Hindawi
Science and Technology of Nuclear Installations
Volume 2023, Article ID 2267376, 14 pages
https://doi.org/10.1155/2023/2267376

Research Article

Prediction of Automatic Scram during Abnormal Conditions of Nuclear Power Plants Based on Long Short-Term Memory (LSTM) and Dropout

Hanying Chen,[1] **Puzhen Gao**,[2] **Sichao Tan**,[2] **Hongsheng Yuan**,[3] **and Mingxiang Guan**[1]

[1]*Shenzhen Institute of Information Technology, Shenzhen 518100, China*
[2]*Heilongjiang Provincial Key Laboratory of Nuclear Power System and Equipment, Harbin Engineering University, Harbin 150001, China*
[3]*China Nuclear Power Technology Research Institute Co., Ltd., Shenzhen 518100, China*

Correspondence should be addressed to Hanying Chen; chenhanying@hrbeu.edu.cn

Received 17 October 2022; Revised 15 January 2023; Accepted 4 February 2023; Published 3 March 2023

Academic Editor: Alexander Zulauf

A deep-learning model was proposed for predicting the remaining time to automatic scram during abnormal conditions of nuclear power plants (NPPs) based on long short-term memory (LSTM) and dropout. The proposed model was trained by simulated condition data of abnormal conditions; the input of the model was the deviation of the monitoring parameters from the normal operating state, and the output was the remaining time from the current moment to the upcoming reactor trip. The predicted remaining time to the reactor trip decreases with the development of abnormal conditions; thus, the output of the proposed model generates a predicted countdown to the reactor trip. The proposed prediction model showed better prediction performance than the Elman neural network model in the experiments but encountered an overfitting problem for testing data containing noise. Therefore, dropout was applied to further improve the generalization ability of the prediction model based on LSTM. The proposed automatic scram prediction model can provide NPP operators with an alert to the automatic scram during abnormal conditions.

1. Introduction

Various faults and failures cannot be completely avoided throughout the lifetime of nuclear power plants (NPPs). During abnormal conditions in NPPs that do not immediately lead to automatic scram, control room operators need to gather information through human-machine interfaces (HMIs) to follow the operating procedures [1]. Conventional monitoring techniques cannot provide control room operators with information regarding when the abnormal condition will lead to automatic scram without further operator intervention. Thus, control room operators usually cannot be alerted about the reactor trip ahead of it. Intelligent prediction for automatic scram can help operators to know when will the reactor trip be triggered during abnormal conditions, so the operators can better prepare or take countermeasures for the upcoming automatic scram to avoid further damage to the reactor. Therefore, intelligent prediction for automatic scram is a promising technique to improve the operating safety of NPPs.

Intelligent prediction for automatic scram belongs to the research field of NPP prognostics, which is an important aspect of nuclear safety. Ayo-Imoru and Cilliers [2] reviewed the studies on prognostics in the nuclear industry, and they pointed out that prognostics is a very important aspect of condition-based maintenance (CBM) in the nuclear power industry as it will help the operator and maintenance personnel to better understand how to schedule maintenance. Coble et al. [3] reviewed the studies on prognostics and health management (PHM) for nuclear power systems.

Many relevant studies focused on long-term remaining useful life (RUL) prediction for NPP systems. Liu et al. [4] proposed an exponential degradation model based on the Bayesian process and prior information to predict the RUL of equipment after the failure warning system alarm. Zio and di Maio [5] proposed a method to predict the RUL of lead-bismuth reactor subsystems based on fuzzy similarity analysis. Coble and Hines [6] proposed a prediction method of RUL based on a general path model and Bayesian updating, which can predict the RUL of the system by predicting failure-related parameters. Furthermore, among the studies on prognostics and health management of NPP, the application of machine learning algorithms, especially deep-learning methods showed many benefits and potentials [7, 8]. In these existing studies on NPP prognostics, the prediction target is the RUL of NPP subsystems or components, but the prediction of short-term behavior of NPPs during abnormal conditions such as automatic scram is not involved.

On the other hand, some other studies on the prognostics of NPPs involve the short-term trend prediction of NPP operating conditions. An online condition forecasting method for a natural circulation system with flow instability was developed by Chen et al. [9] based on the ensemble of the online sequential extreme learning machine (EOS-ELM). Marseguerra et al. [10, 11] presented a predictive model based on neuro-fuzzy techniques to predict the water level in a steam generator of a pressurized water reactor (PWR). Liu et al. [12] developed a condition prediction model for NPP operating parameters based on backpropagation neural networks (BPNN). An approach for predicting NPP components condition was proposed by Liu et al. [13] based on a modified probabilistic support vector machine (PSVM). Koo et al. [14] developed a model to provide the internal containment states information during LOCA accidents based on the rule dropout deep fuzzy neural networks (DFNN). Zhang and Hu [15] predicted radionuclides in the receiving water of an inland nuclear power plant based on a difference gated neural network. The aforementioned studies focus on the short-term prediction of NPP operating parameters such as coolant flow rate, coolant temperature, steam generator water level, and radionuclides, but these prediction methods are not applied on the NPP events like automatic scram.

There are very limited studies investigating the prediction methods for automatic scram during fault conditions of NPPs. Park et al. [16] pointed out that the prognosis usually means the remaining useful lifetime, so from the point of view of an operator, the remaining time to reactor trip is regarded as the remaining useful lifetime of NPPs during abnormal conditions. As such, Park et al. [16] proposed a method to predict the expected remaining time to the reactor trip by calculating the similarity between the monitored data and the simulated data in the transient database with the operating data and used principal component analysis (PCA) to compress the dimension of data used in the similarity comparison. However, the prediction method for automatic scram proposed by Park et al. [16] is based on simple transient similarity measures and relies on calling the transient database during abnormal conditions, so the prediction results cannot be updated with the change of operating conditions. The advanced data-driven methods developed in recent years should provide better solutions to NPP automatic scram prediction problems.

In recent years, advanced data-driven methods such as long short-term memory (LSTM), random forest, and attention mechanism have been widely used in the studies of condition prognostics and diagnostics of NPPs. Bae et al. [17] conducted real-time prediction of nuclear power plant parameter trends based on LSTM. Zhang et al. [18] developed a cost-sensitive long short-term memory (CSLSTM) model to predict abnormal or false pressurizer water levels. A method for multistep ahead prediction of leakage flow from the first reactor coolant pump seals was proposed by Nguyen et al. [19] based on LSTM combined with ensemble empirical mode decomposition. Radaideh et al. [20] developed a DNN/LSTM expert system for predicting NPP condition parameters during LOCA accidents. Wang et al. [21] applied several LSTM-based networks for abnormal event detection, identification, and isolation to help maintain the safe operations of NPPs. Yu et al. [22] proposed a continuous learning monitoring strategy for NPPs based on random forest and PCA. Qian and Liu [23] applied a new gated recurrent unit (GRU) network combined with attention mechanism and transfer learning (TL) in the fault diagnosis of NPPs. Furthermore, the mutation of the aforementioned advanced data-driven methods, such as bidirectional long short-term memory (BiLSTM), interpretable machine learning, knowledge-data-driven attention (CFKDA), and self-attention, was also successfully applied in many prognostics fields, such as renewable energy generation forecast [24], battery capacity prediction [25], battery health prognostics [26], charging demand forecast [27], etc. These existing studies have proved the capability of advanced data-driven methods in the prediction of time series and showed the potential of these methods in solving the problem of automatic scram prediction during fault conditions of NPPs.

In the present paper, a deep-learning model based on LSTM and dropout was proposed for predicting the remaining time to automatic scram during abnormal conditions of NPPs. The proposed prediction model was trained and tested using abnormal condition data generated by a full-scale simulator for a 300 MW PWR. The input of the proposed model is set to be the deviation from the normal operating state, and the output is the remaining time from the current moment to the upcoming reactor trip. The influence of data noise on automatic scram prediction was studied, and dropout was applied to improve the generalization ability of the LSTM prediction model. Different from the existing method [16], the proposed models based on LSTM do not depend on the transient database when conducting automatic scram prediction, so the proposed method has better deployment flexibility and response speed. The proposed automatic scram prediction models can assist NPP operators to prepare for automatic scram during abnormal conditions.

The present paper is structured as follows: Section 2 introduces the specific process of the automatic scram prediction of NPPs. Section 3 illustrates the prediction algorithm used in this study. The experimental results of the proposed prediction model are shown and analyzed in Section 4. Section 5 shows the improvements in the proposed prediction model based on dropout for avoiding overfitting. Conclusions are drawn in Section 6.

2. Process of Automatic Scram Prediction

When an abnormal condition occurs in a NPP because of component fault or failure, the operating parameters of the NPP will deviate from the normal operating state. An automatic scram will be triggered when the deviation of NPP operating parameters exceeds a certain limit. Therefore, for a specific mode of NPP abnormal condition, there is a close association between the time series of operating parameters and the remaining time to automatic scram. A machine learning model can be trained to learn this association and predict when will the automatic scram be triggered if there is no operator intervention.

In the proposed prediction model, the input of the network is the deviation of monitoring parameters from the operating state before an abnormal condition occurred, and the output is the remaining time from the current moment to the upcoming reactor trip. The deviation of the operating state is selected to be the input of the network instead of the absolute value of the monitoring parameters because, under the same type of fault mode, the operating parameters of NPP will have the similar deviation mode, but the operating conditions before the abnormal condition are not necessarily the same, so the absolute values of the operating parameters may not reflect the development process of the abnormal condition. In such a way, by using the deviation of the operating state as the input of the prediction model, even if the fault event occurs during nonstandard operating conditions (for example, operating below full power), the prediction model still has a high probability of being effective. The output of the network will be provided to the operators as an alert for the reactor trip.

Every prediction model only corresponds to one fault mode (for example, steam generator secondary side leakage) in this work. Since different fault modes of NPPs have different causes and mechanisms, it is almost impossible to train one prediction model corresponding to all fault modes of NPPs, and it is reasonable to train plural prediction models for different fault modes. During the NPP operation monitoring using the proposed method, only when a certain fault mode is detected and identified by operators or by an intelligent fault diagnosis system, the corresponding automatic scram prediction model will be activated to predict the remaining time to automatic scram. In this way, one prediction model does not need to be applicable to different abnormal modes, and the difficulty of model training is reduced.

During the training process of the prediction model, the entire dataset of the abnormal condition is available at once, so the complete time series of operating parameters from a fault condition occurrence to a reactor trip can be input into the prediction model. While in the testing of the models or automatic scram prediction in the real world, the time series of the monitored parameters are acquired sequentially and input into the prediction model in each time step. Every time the prediction model obtains the input of operating parameters, it outputs the remaining time to automatic scram. In this way, the prediction of the reactor trip can be updated continuously during abnormal conditions. Without external intervention, the output of the prediction model should form a countdown for the upcoming automatic scram, which can be provided to NPP operators as a reference. The proposed process of automatic scram prediction is shown in Figure 1.

The fault mode of the steam generator's secondary side leakage was used to build the proposed automatic scram prediction model. After the steam generator's secondary side leak occurs, the steam generator pressure will decrease gradually, and finally, trigger turbine tripping and automatic reactor scram. Without the intervention of the operators, the steam generator's secondary side leakage event usually causes the reactor scram within a few minutes.

The monitoring data used in this study was generated by a full-scale simulator of the nuclear power plant. The reference plant for the simulator is a 300 MW PWR plant. The reference plant has two primary coolant loops with one steam generator and one coolant pump in each loop. The schematic diagram of the primary coolant system (PCS) of the reference plant is shown in Figure 2.

The fault conditions with different severity were used in the training and testing of the prediction model. The fault condition with the worst severity was given a severity of 1.00, and the normal operating condition has a severity of 0. The prediction model was trained by 10 groups of fault conditions with the severity of 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, and 1.00, so the prediction model can be robust to abnormal conditions with different severity. The fault conditions with the severity of 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, and 0.95 were used to test the performance of the trained prediction system.

The NPP operating parameters are input into the automatic scram prediction model by the step size of one second. The input vector of the prediction model is composed of normalized deviation values of 22 operating parameters. The input operating parameters and their units are listed in Table 1. The operating parameters listed in Table 1 are all measurable in NPPs and are closely associated with the operation condition of the reactor, and the deviation of these operating parameters can show the transient process of the abnormal conditions; thus, these operating parameters are selected to build the prediction model for predicting the remaining time to automatic scram.

3. Prediction Algorithm

To predict the remaining time to automatic scram during NPP abnormal conditions, the prediction model must be able to obtain information from the time series of operating parameters. If conventional feedforward neural network is applied to build the prediction model, then at each time step, the prediction output will only be determined by the network input of the current time step. Since the fault

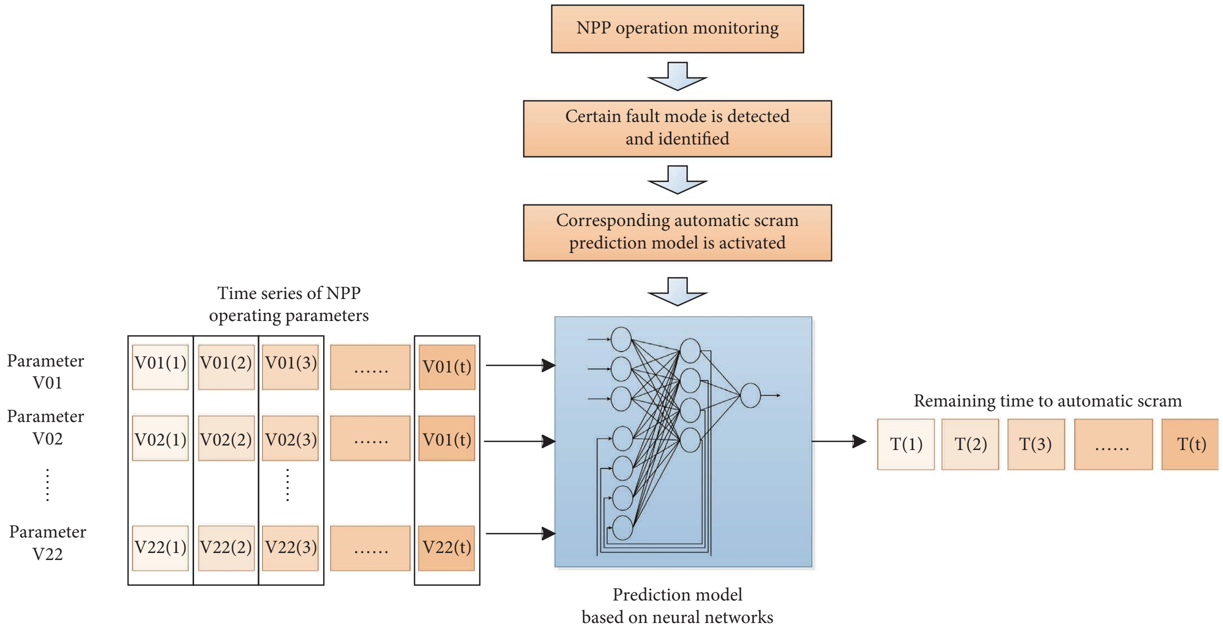

NPP operation monitoring
Certain fault mode is detected and identified
Corresponding automatic scram prediction model is activated
Time series of NPP operating parameters
Parameter V01
Parameter V02
Parameter V22
V01(1)
V01(2)
V01(3)
......
V01(t)
V02(1)
V02(2)
V02(3)
......
V01(t)
V22(1)
V22(2)
V22(3)
......
V22(t)
Prediction model based on neural networks
Remaining time to automatic scram
T(1)
T(2)
T(3)
......
T(t)

Figure 1: The process of automatic scram prediction.

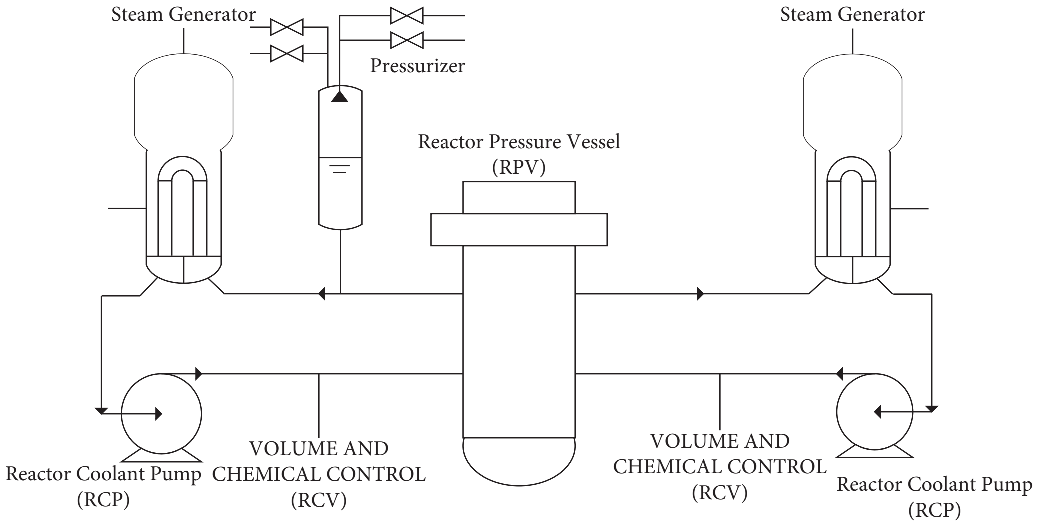

Steam Generator
Pressurizer
Steam Generator
Reactor Pressure Vessel (RPV)
VOLUME AND CHEMICAL CONTROL (RCV)
VOLUME AND CHEMICAL CONTROL (RCV)
Reactor Coolant Pump (RCP)
Reactor Coolant Pump (RCP)

Figure 2: The primary coolant system (PCS) of the reference nuclear power plant.

conditions of NPPs are transient processes, the prediction model should know not only the instantaneous operating condition of NPP but also the changing tendency of various parameters, so as to predict the automatic scram. Therefore, the feedforward neural network prediction model has to take the complete sequence or at least a time window of operating parameter time series as the network input, which will increase the scale of the network and make it difficult to train.

Different from the feedforward neural network, recurrent neural networks (RNNs) can obtain information from the historical network input. If RNNs are applied to build the prediction model, only the current operating parameters need to be input into the network at each time step, which improves the performance of the prediction model. Therefore, a simple RNN and its variant LSTM were applied to predict the remaining time to automatic scram in this study.

3.1. Recurrent Neural Networks (RNNs). RNNs are a class of deep-learning algorithms first proposed by Rumelhart et al. [28]. For conventional feedforward neural networks, the outputs of neural nodes in each layer are connected to the inputs of neural nodes in the next layer, and the nodes in the same layer are not connected to each other. The history node outputs of feedforward neural networks have no effect on subsequent node outputs. While the layer outputs of RNNs can be stored and fed back to the input of neural nodes in subsequent time steps so that the history node outputs will affect the later ones.

Table 1: The monitoring parameters used in the training and testing of the proposed prediction model.

No.	Parameters	Unit
V01	Coolant flow rate of loop 1	kg/s
V02	Coolant flow rate of loop 2	kg/s
V03	Hot leg coolant temperature of loop 1	°C
V04	Hot leg coolant temperature of loop 2	°C
V05	Cold leg coolant temperature of loop 1	°C
V06	Cold leg coolant temperature of loop 2	°C
V07	Steam generator pressure of loop 1	MPa
V08	Steam generator pressure of loop 2	MPa
V09	Steam generator narrow range water level of loop 1	m
V10	Steam generator narrow range water level of loop 2	m
V11	Steam generator feed water flow rate of loop 1	kg/s
V12	Steam generator feed water flow rate of loop 2	kg/s
V13	Main feed water temperature of the steam generator 1	°C
V14	Main feed water temperature of the steam generator 2	°C
V15	Steam generator outlet steam flow rate of loop 1	kg/s
V16	Steam generator outlet steam flow rate of loop 2	kg/s
V17	Steam generator outlet steam temperature of loop 1	°C
V18	Steam generator outlet steam temperature of loop 2	°C
V19	Pressurizer pressure	MPa
V20	Pressurizer water level	m
V21	Pressurizer steam temperature	°C
V22	Pressurizer water temperature	°C

One of the most known simple RNNs is Elman neural network [29], which can be regarded as a special feed-forward network including an additional context layer. The architecture of the Elman neural network is shown in Figure 3.

The Elman neural network uses the outputs of the hidden layer at the previous time step as part of the inputs to the hidden layer at the present time step. The network output is as follows:

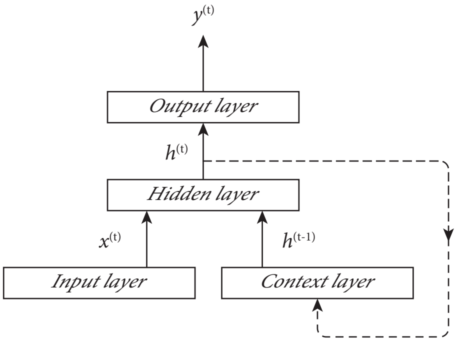


Figure 3: Architecture of Elman neural network.

$$\mathbf{y}^{(t)} = \varphi_O\left(\mathbf{w}_{\mathrm{HO}}\mathbf{h}^{(t)} + \mathbf{b}_O\right), \tag{1}$$

$$\mathbf{h}^{(t)} = \varphi_H\left(\mathbf{w}_{\mathrm{IH}}\mathbf{x}^{(t)} + \mathbf{w}_{\mathrm{CH}}\mathbf{h}^{(t-1)} + \mathbf{b}_H\right), \tag{2}$$

where $\mathbf{y}$ is the output vector of the network, $\mathbf{h}$ is the output vector of the hidden layer, $\mathbf{x}$ is the input vector of the network, $\mathbf{w}_{\mathrm{HO}}$ is the weight matrix from the hidden layer to the output layer, $\mathbf{w}_{\mathrm{IH}}$ is the weight matrix from the input layer to the hidden layer, $\mathbf{w}_{\mathrm{CH}}$ is the weight matrix from the context layer to the hidden layer, φ_O and φ_H are activation functions of the output layer and hidden layer, and $\mathbf{b}_O$ and $\mathbf{b}_H$ are biases of the output layer and hidden layer. In Elman neural network, φ_O is generally the linear function and φ_H is generally the sigmoid or tanh function. The superscript (t) represents the time step.

By bringing the hidden layer outputs of previous time steps into equations (1), (3–5), and so on can be obtained in turn, where the layer biases are omitted. The hidden layer output $\mathbf{h}^{(t)}$ can be decomposed sequentially until the start of the input data sequence.

$$\mathbf{y}^{(t)} = \varphi_o\left(\mathbf{v}\varphi_h\left(\mathbf{u}\mathbf{x}^{(t)} + \mathbf{w}\mathbf{h}^{(t-1)}\right)\right), \tag{3}$$

$$\mathbf{y}^{(t)} = \varphi_o\left(\mathbf{v}\varphi_h\left(\mathbf{u}\mathbf{x}^{(t)} + \mathbf{w}\varphi_h\left(\mathbf{u}\mathbf{x}^{(t-1)} + \mathbf{w}\mathbf{h}^{(t-2)}\right)\right)\right), \tag{4}$$

$$\mathbf{y}^{(t)} = \varphi_o\left(\mathbf{v}\varphi_h\left(\mathbf{u}\mathbf{x}^{(t)} + \mathbf{w}\varphi_h\left(\mathbf{u}\mathbf{x}^{(t-1)} + \mathbf{w}\varphi_h\left(\mathbf{u}\mathbf{x}^{(t-1)} + \mathbf{w}\mathbf{h}^{(t-3)}\right)\right)\right)\right). \tag{5}$$

The time-dependent learning characteristics of RNNs can also be illustrated by unfolding the neural network architecture through time, as shown in Figure 4. It can be found that all the data inputs back in time contribute to the current output of the Elman neural network model $\mathbf{y}^{(t)}$. Through adding the context layer, RNNs can store and learn information from the whole time-dependent input data sequence. Therefore, the prediction models based on RNNs seem to have the ability to learn the changing tendency of operating parameters to predict the automatic scram.

However, in the backpropagation through time (BPTT) training algorithm commonly used by RNNs, the error gradient of the current time step propagates backward to the previous time steps and usually decreases exponentially during the backpropagation. Therefore, the outputs of RNNs are only affected by inputs within several time steps before the current time, while the inputs far back from the current time will not affect the network outputs. This is the so-called vanishing gradient problem [30]. The prediction model of NPP automatic scram needs to learn the long-term changing tendency of operating parameters, so it is necessary to find a more suitable algorithm to improve the prediction performance.

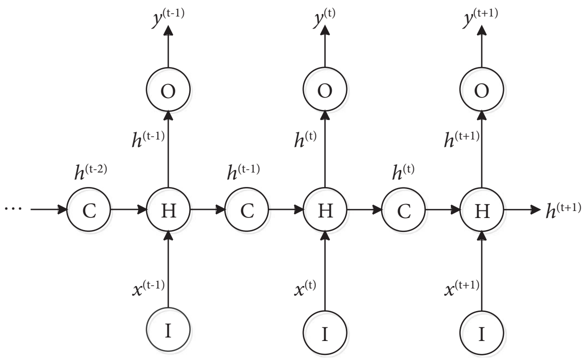


Figure 4: Unfolding the Elman neural network architecture through time.

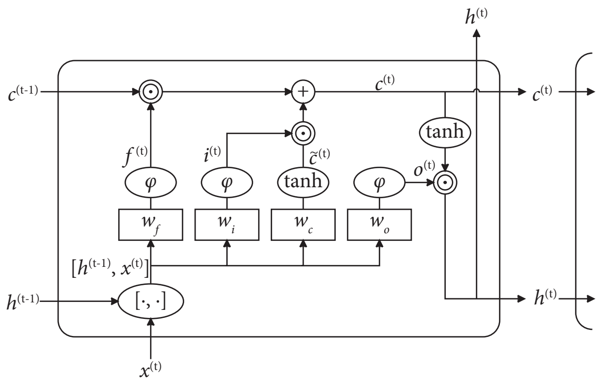


Figure 5: Architecture of LSTM node.

3.2. Long Short-Term Memory (LSTM). Long short-term memory (LSTM) was proposed by Hochreiter and Schmidhuber [31] as a solution to the gradient vanishing problem. LSTM is designed to adaptively control the memory length of the learned features and shows good performance in Seq2Seq problems. To solve the problem that the RNNs cannot learn information from long data sequences, the LSTM nodes add a new state value c, called cell state, and its function is to save the information of data input far away from the current moment, that is, long-term memory. Structures called gates are set in LSTM nodes to remove or add information to the cell state c. The gates are essentially fully connected layers with outputs between 0 and 1.

Generally, a LSTM node has the following three types of gates that update and control the cell state: the forget gate, the input gate, and the output gate. The forget gate controls what information in the cell state to forget, whose output is given by the following equation:

$$\mathbf{f}^{(t)} = \varphi\left(\mathbf{w}_f\left[\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}\right] + \mathbf{b}_f\right), \tag{6}$$

where $\mathbf{w}_f$ is the weight matrix of the forget gate, $\mathbf{b}_f$ is the bias vector of the forget gate, $\mathbf{h}^{(t-1)}$ denotes the hidden layer outputs of the previous time step and $\mathbf{x}^{(t)}$ denotes the input of the current time step, $[\cdot, \cdot]$ represents the cascading two vectors, and φ is the activation function.

The input gate controls what new information will be encoded into the cell state. The input gate output has the form:

$$\mathbf{i}^{(t)} = \varphi\left(\mathbf{w}_i\left[\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}\right] + \mathbf{b}_i\right), \tag{7}$$

where $\mathbf{w}_i$ is the weight matrix and $\mathbf{b}_i$ is the bias vector of the input gate. The output of the input gate $\mathbf{i}$ determines whether the candidate value $\tilde{\mathbf{c}}$ generated by the new input will be added to the cell state or not. The candidate value $\tilde{\mathbf{c}}$ is given by

$$\tilde{\mathbf{c}}^{(t)} = \tan h\left(\mathbf{w}_c\left[\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}\right] + \mathbf{b}_c\right), \tag{8}$$

where $\mathbf{w}_c$ and $\mathbf{b}_c$ are the corresponding weight matrix and bias vector.

The cell state of the current moment step $\mathbf{c}^{(t)}$ is calculated based on the cell state of the previous time step and the candidate value of the current time step.

$$\mathbf{c}^{(t)} = \mathbf{f}^{(t)} \odot \mathbf{c}^{(t-1)} + \mathbf{i}^{(t)} \odot \tilde{\mathbf{c}}^{(t)}, \tag{9}$$

where $\odot$ represents multiplying corresponding elements of two vectors.

The output gate controls what information encoded in the cell state is sent to the network as hidden layer output. The output gate's output is given by

$$\mathbf{o}^{(t)} = \varphi\left(\mathbf{w}_o\left[\mathbf{h}^{(t-1)}, \mathbf{x}^{(t)}\right] + \mathbf{b}_o\right), \tag{10}$$

where $\mathbf{w}_o$ is the weight matrix, $\mathbf{b}_o$ is the bias vector, and $\mathbf{o}$ is the output of the output gate.

The output vector of the LSTM layer is given by

$$\mathbf{h}^{(t)} = \mathbf{o}^{(t)} \odot \tan h\left(\mathbf{c}^{(t)}\right), \tag{11}$$

where $\mathbf{h}^{(t)}$ will be transmitted to the next network layer as well as the LSTM layer of the next time step.

The LSTM network can be built by the above calculation flow, which is shown in Figure 5. By introducing state value and three types of gates, LSTM networks can effectively learn the long-term tendency by efficiently resisting the gradient vanishing [32]. It can be expected that the automatic scram prediction model based on LSTM will have better performance.

Furthermore, some other advanced data-driven methods will be applied in future work to improve the proposed prediction model for automatic scram in NPPs. BiLSTM [33] is an improvement of LSTM, which enable additional training by traversing the input data twice in forward and backward directions. BiLSTM provides better predictions compared to LSTM in many cases [24, 34], and can be used to benefit the automatic scram prediction accuracy. The interpretable machine learning framework proposed by Liu et al. [25] can also be combined with the proposed models to analyze how the variations of NPP operating parameters during abnormal conditions affect the process leading to automatic scram.

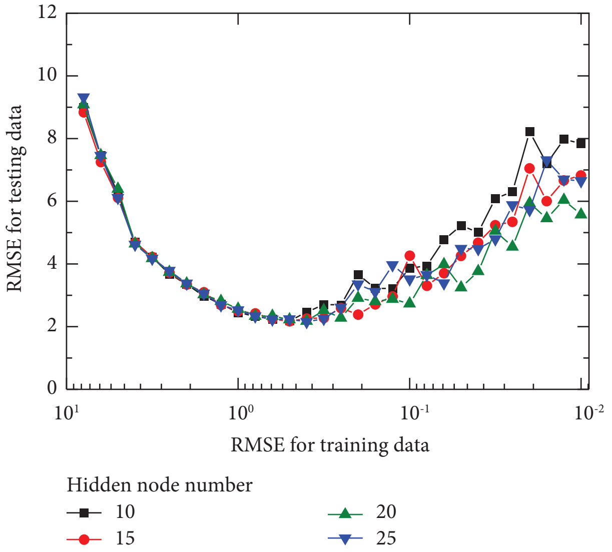


Figure 6: Average testing RMSE vs. training RMSE by Elman neural network models.

4. Results and Discussion

4.1. Prediction Experiments for Original Testing Data. The prediction results for automatic scram during the steam generator secondary side leak condition are shown in this section. In the training of prediction models, the training error or training loss decreases gradually with the increase in training epochs. The prediction models with different training error usually have different prediction performance, and deep-learning models such as RNN and LSTM networks often encounter overfitting problems; that is, the models are too closely fit to the training datasets, but has poor fit with the testing datasets. Therefore, in the present study, the training of prediction models was stopped when the preset goal of the training error was reached, and the performance of prediction models with different training errors was tested, so as to analyze the influence of fitting degree on the prediction performance. The training error and testing error of prediction models in this study were measured by root mean square error (RMSE), which is defined as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} \left(y(i) - \hat{y}(i)\right)^2}{N}}, \tag{12}$$

where $y(i)$ is the actual value and $\hat{y}(i)$ is the predicted value, and N is the number of data inputs used in the tests.

Elman neural networks and LSTM networks were applied in the modeling of automatic scram prediction, respectively. Firstly, the automatic scram prediction model based on the Elman neural network was tested. The Elman neural network models were trained using Bayesian regularization backpropagation, and the learning rate was set to 0.01. During the training process, it was found that the training RMSE of the prediction models can be reduced from dozens to 0.01 or even lower within 1000 epochs. Therefore, logarithmic coordinates were used to represent the training RMSE of the prediction model in this study. The goal of training RMSE was set to be 10^1, $10^{0.9}$, $10^{0.8}$, . . ., $10^{-1.9}$, and 10^{-2}, and the training was stopped when the goal of the training error was reached, then the testing dataset was applied to the trained prediction models with different training RMSE. In addition, Elman neural network models with 10, 15, 20, and 25 hidden layer nodes were trained and tested to evaluate the influence of the hidden layer node number on prediction performance.

The Elman neural network prediction model was trained to achieve every set training RMSE goal for 20 times. The average testing RMSE of Elman neural network prediction models with different training errors is shown in Figure 6, and the testing RMSE results with error bars are shown in Figure 7. The x-axis of training RMSE in Figures 6 and 7 is in logarithmic coordinate.

As shown in Figures 6 and 7, when the training RMSE of the Elman neural network prediction model is greater than $10^{-0.3}$ (about 0.5), the testing error decreases with the increase in training accuracy. However, when the training RMSE of the Elman neural network model is too low (less than 0.5), the prediction performance deteriorates, that is, overfitting occurs. When the Elman neural network prediction model is overfitted, its average testing RMSE increases (as shown in Figure 6), and the stability of the prediction model becomes worse, which may lead to prediction results with a very large deviation (as shown in Figure 7). In addition, the results show that when the hidden layer node number is in a reasonable range (10–25), it has no significant influence on the performance of the prediction model based on Elman neural networks.

The examples of predicted remaining time to automatic scram generated by the Elman neural network models with the training RMSE of 5.00, 0.50, and 0.01 are shown in Figure 8, where the severity of the abnormal transient used in the testing is 0.55. It should be noted that, as shown in Figure 7, the outputs of the prediction models with very small training error are especially unstable, so the prediction results generated by the prediction models with training RMSE of 0.01 varies largely at each time of training.

It can be observed that the model with training RMSE of 0.50 generated prediction results fitting well with the actual values. The predicted remaining time to reactor trip decreases with the development of abnormal conditions; thus, the output of the prediction model can generate a countdown to the reactor trip for NPP operators. However, the prediction results generated by the models with training RMSE of 5.00 and 0.01 were not fitting so well with the actual values. The output of the model with training RMSE of 5.00 increases with time in the initial stage of an abnormal condition, so the changing tendency of the predicted results was inconsistent with the actual value. Meanwhile, there is a significant deviation between the predicted results and the actual value of the model with a training RMSE of 0.01.

The results shown in Figures 6–8 jointly prove that the training error does have an impact on the prediction performance of the automatic scram prediction models based on the Elman neural network, and whether underfitting (RMSE of 5.00) or overfitting (RMSE of 0.01) occurs, the prediction model will not achieve the best prediction performance.

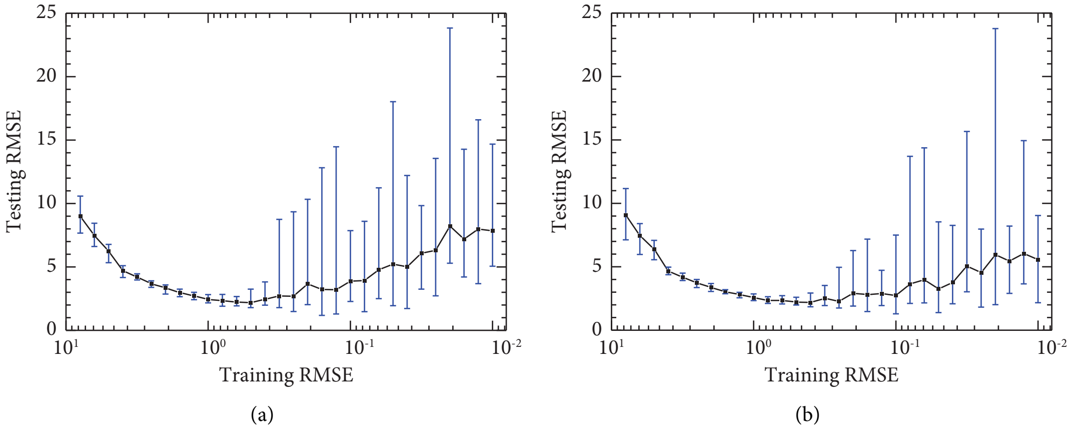


Figure 7: Testing RMSE (with error bar) vs. training RMSE by Elman neural network models. (a) Hidden node number = 10. (b) Hidden node number = 20.

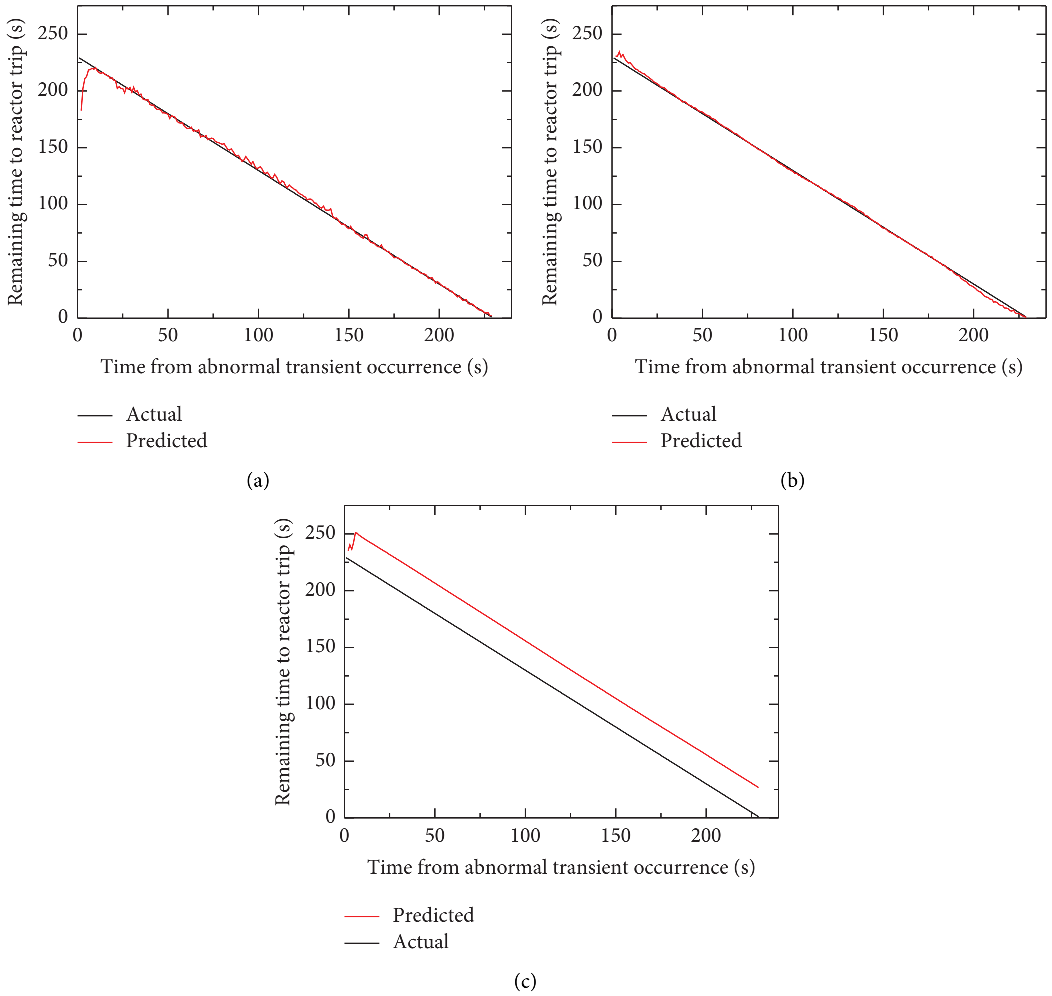


Figure 8: Prediction results for automatic scram generated by Elman neural network models: (a) training RMSE = 5.00; (b) training RMSE = 0.50; (c) training RMSE = 0.01.

Generally, the optimal training error of the Elman neural network prediction model cannot be easily determined for different datasets, so the overfitting problem will bring some difficulties to the practical application of the automatic scram prediction model based on the Elman neural network.

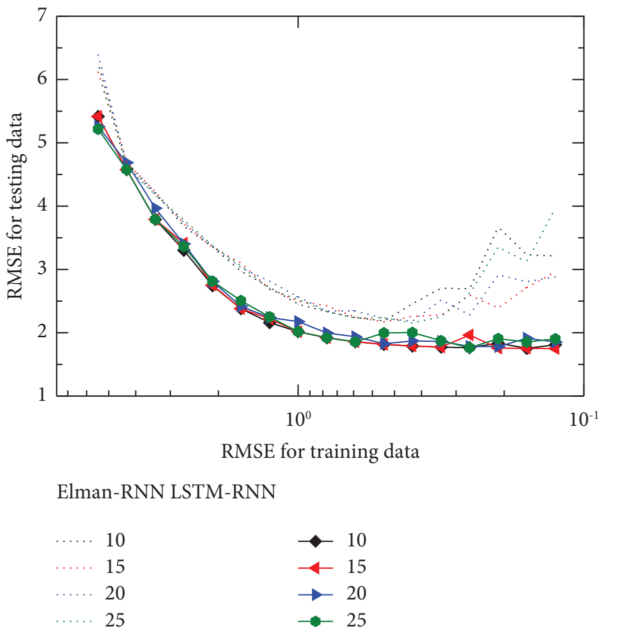


Figure 9: Average testing RMSE vs. training RMSE by prediction models based on LSTM models (compared with the Elman neural network models).

To improve the performance of the automatic scram prediction model, LSTM, as an improved algorithm for RNN, was applied in this study. The same training dataset and testing dataset were used in the experiments of automatic scram prediction models based on LSTM. As the performance of LSTM prediction models is also significantly affected by training errors, LSTM prediction models with different training errors were also trained and tested to analyze the impact of training errors on the performance of LSTM prediction models. Figure 9 shows the testing RMSE of the LSTM prediction model with 10, 15, 20, and 25 hidden layer nodes, while the testing errors of the Elman neural network prediction models are also shown in Figure 9 for comparison. During the training process, it was found that even if the training epoch number of the LSTM prediction model was increased to more than 100,000, it was still difficult to reduce the training RMSE to less than 0.1, so only the results with training RMSE greater than 0.1 are shown in Figure 9. In addition, the testing RMSE results of LSTM models with error bars are shown in Figure 10.

It can be seen from Figure 9 that the average testing RMSE of LSTM models is lower than the Elman neural network models, but the difference is not significant. When the training RMSE of the LSTM prediction model is greater than 1.0, the testing RMSE decreases with the decrease in training error. After its training RMSE drops below 1.0, the testing error of the LSTM model does not continue to decrease, but remains basically unchanged. Furthermore, it can be seen from Figure 10 that the prediction performance of the LSTM model is stable in the range of training RMSE from 0.1 to 1.0, and the prediction model will not generate prediction results that deviate from the actual value largely like Elman neural network models.

Therefore, the LSTM prediction model can achieve good prediction performance in a large range of training RMSE from 0.1 to 1.0, which indicates that the LSTM prediction model can avoid the overfitting problem to a great extent in the training process. This feature is an important advantage in the practical application of automatic scram prediction of NPPs.

4.2. Prediction Experiments for Testing Data Containing Noise.

The training and testing data set used in the above experiments are based on simulated fault condition data without measurement errors and noise. In the actual monitoring process of NPP fault conditions, the monitored data obtained by the prediction model will inevitably contain measurement noise. This requires the prediction model to be robust to the data noise.

In order to analyze the influence of data noise on the prediction performance of the automatic scram prediction model proposed in this paper, random white noises with different amplitudes were added to the normalized testing data, and the prediction models based on the Elman neural network and LSTM were tested by these noisy testing data, respectively, while keeping the training data set unchanged. Figure 11 shows the average prediction RMSE for noisy testing sets by prediction models based on the Elman neural network and LSTM with different training errors.

It can be seen from the prediction results of Figure 11(a) that the prediction accuracy of the Elman neural network model with RMSE greater than 0.5 is relatively less affected by data noise, but its testing RMSE is still significantly higher than that of LSTM prediction model shown in Figure 11(b). After the training RMSE drops below 0.5, the testing RMSE of the Elman neural network models for noisy testing data increases sharply. Especially for the testing data containing the noise with the maximum amplitude of 0.05 and 0.07, the testing RMSE of the Elman neural network models exceeds 30, which means that the prediction results at this time have deviated greatly from the actual values and are no longer of the reference value.

The above results show that overfitting has a serious influence on the generalization ability of Elman neural network prediction models. In practical applications, once the condition monitoring data contains measurement noise, the prediction accuracy of Elman neural network models, especially the overtrained Elman neural network models, will be obviously affected.

In contrast, the influence of testing data noise on the performance of LSTM prediction models is much smaller. This is mainly because the learning process of LSTM models is based on a longer data sequence, so the training error cannot decrease so low as Elman neural networks. The generalization ability of LSTM prediction models is relatively high, so they are more suitable for actual automatic scram time prediction of the NPPs.

However, when the added noise is relatively large (the maximum amplitude is greater than or equal to 0.03), after the training RMSE error decreases below 1.0, the testing error increases with the decrease of the training error, which

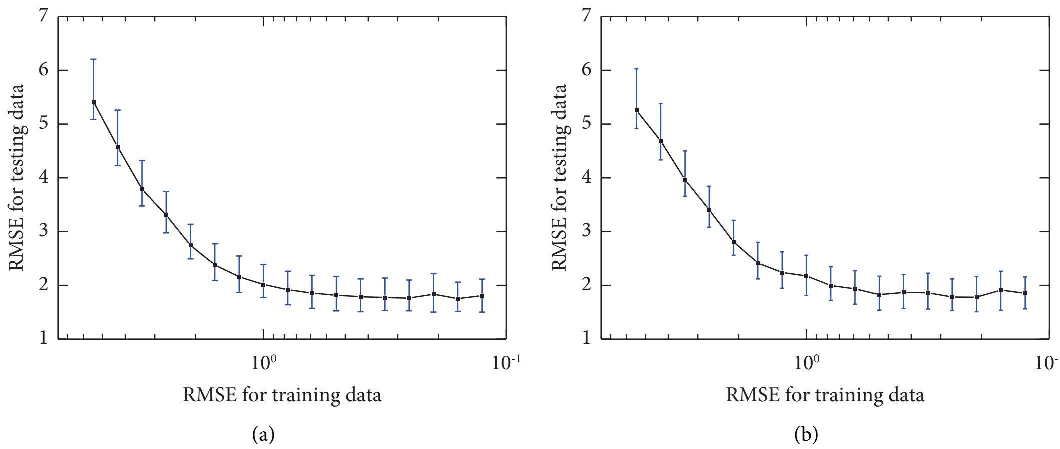


Figure 10: Testing RMSE versus training RMSE with error bar by prediction models based on LSTM. (a) Hidden node number = 10. (b) Hidden node number = 20.

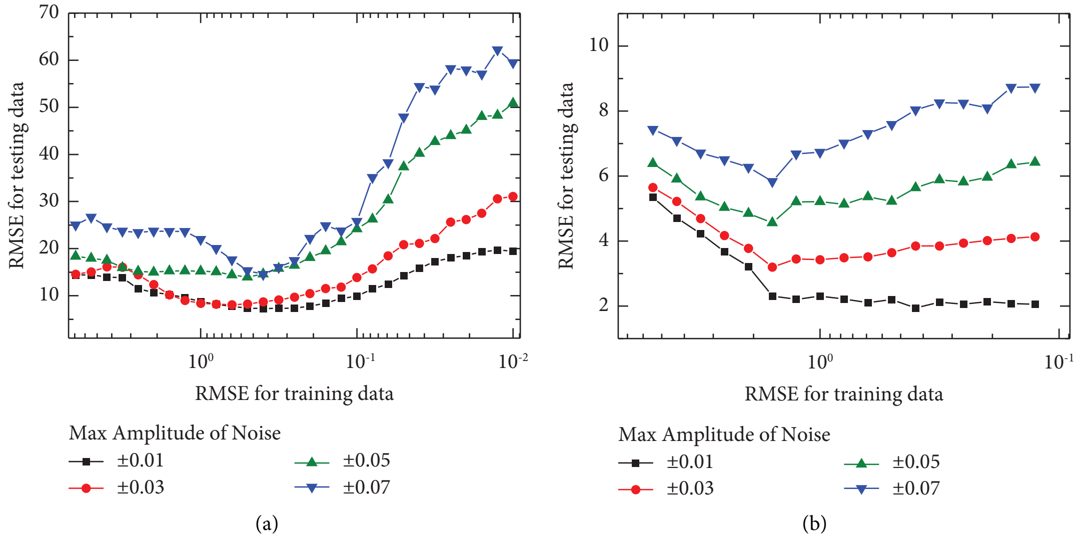


Figure 11: Testing RMSE vs. training RMSE by automatic scram prediction models with testing data containing artificial noise: (a) a model based on the Elman neural network and (b) a model based on LSTM.

indicates that the LSTM prediction models also encountered overfitting for the testing data with noise.

To further demonstrate the overfitting of the LSTM models, the prediction results of LSTM models with RMSE of 1.47 and 0.12 are shown in Figure 12, where the artificial noise with the maximum amplitude of 0.07 is added to the testing data, and the severity of the abnormal condition used in the test is 0.55. It can be seen from the prediction results in Figure 12 that the LSTM prediction model with a larger training error (training RMSE = 1.47) has a relatively high prediction accuracy, while the outputs of the model with a smaller training error (training RMSE = 0.12) shows a larger deviation from the actual value in the initial stage of the abnormal condition. Figure 11(b) and Figure 12 jointly show that when there is noise in the testing data, the LSTM-based automatic scram prediction model will also have the overfitting problem, and the prediction model should be improved to solve this problem.

5. Solution to Overfitting Based on Dropout

To solve the overfitting problem in the test of input data with noise, the dropout technique was used to improve the LSTM prediction model. Dropout is a technique used for addressing the overfitting problem in deep neural networks, first proposed by Hinton et al. [35] and further discussed by Srivastava et al. [36].

The key idea of dropout is that in each epoch of training, some neural nodes (along with their connections) are randomly dropped with a certain probability (called the dropout rate) to prevent neural nodes from coadapting with each other too much, thus improving the generalization ability of the network.

At test time, the neural network without dropout and composed of all nodes are used, but the outputs of each node need to be multiplied by the probability that the node is not discarded in the dropout training (that is, 1-dropout rate), so

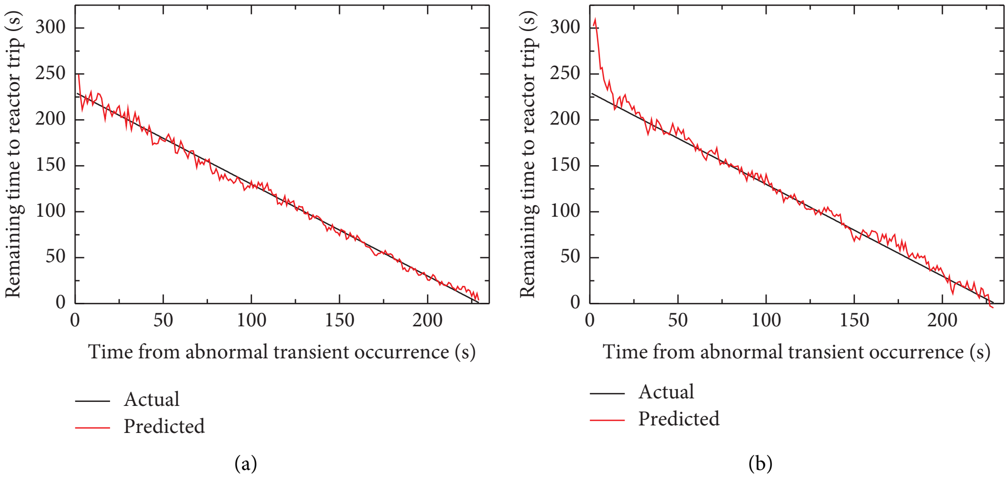


Figure 12: Prediction results for testing data containing artificial noise (max ±0.07) by LSTM model: (a) training RMSE = 1.47 and (b) training RMSE = 0.12.

as to ensure the correct network output. In this study, dropout was applied to the LSTM layer of the prediction model.

The following is the detailed calculation flow. The output layer of the LSTM prediction model uses a linear activation function, so the network output without dropout is as follows:

$$\hat{y} = \sum_{j=1}^{J} v_j h_j + b, \tag{13}$$

where $\hat{y}$ is the output of the network, which is a real number indicating the remaining time to reactor trip. J denotes the number of LSTM layer nodes in the network, v_j is the weight from the j-th LSTM layer node to the output node, h_j is the output of the j-th LSTM layer node, and b is the output layer bias. Let the dropout rate of the hidden layer node be p, then the prediction model output becomes

$$\begin{aligned} \tilde{h}_j &= r_j \cdot h_j, \\ \hat{y} &= \sum_{j=1}^{J} v_j \tilde{h}_j + b, \end{aligned} \tag{14}$$

where r_j is a Bernoulli random variable (with a value of 0 or 1) with a probability of $1-p$, and when r_j is 0, the corresponding node is discarded. At test time, the prediction model uses all hidden layer nodes, but the output of each node should be scaled down to ensure the expected network output; that is,

$$\hat{y}' = \sum_{j=1}^{J} (1-p) v_j h_j + b, \tag{15}$$

where $\hat{y}^{'}$ denotes the testing network output of the prediction model trained with dropout.

The automatic scram prediction model based on LSTM and dropout was tested by a dataset with and without noise. The prediction model with 10 LSTM layer nodes was used for testing, and dropout was applied to its LSTM layer in the training. The dropout rate was set to be 0.05, 0.10, 0.20, 0.30, 0.40, and 0.50 to show its effect on prediction performance. Every model was trained by 100,000 epochs to ensure sufficient training. Table 2 lists the training RMSE of prediction models trained with different dropout rates, as well as the testing RMSE for the testing dataset with and without noise.

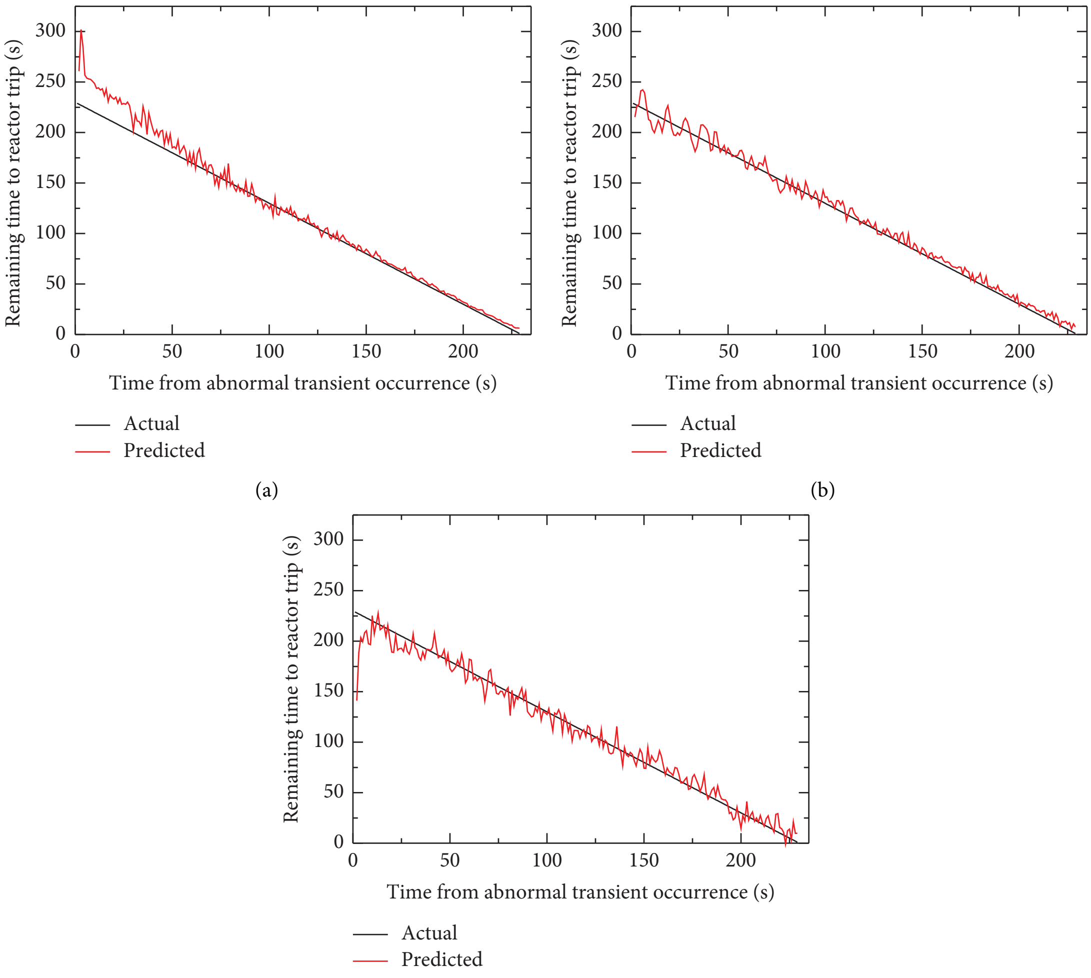
It can be found that, when applying dropout to train the prediction model, the training RMSE of the sufficiently trained model increased. The prediction models with greater dropout rates achieved larger training errors, which indicates that the prediction models are more likely to avoid overfitting, but an excessive dropout rate may lead to underfitting and deteriorate prediction performance. On the contrary, the prediction models with a smaller dropout rate generated smaller training errors and a more similar prediction performance with the models without dropout.

As shown in Table 2, for the testing data with no noise or small noise amplitude (±0.01), adopting dropout did not improve the prediction performance, while increasing the dropout rate may raise the prediction RMSE to some extent. This is because the original LSTM prediction model does not show obvious overfitting for testing data with no noise or small noise amplitude, so increasing training error would not benefit the prediction performance. However, for the testing data with the maximum noise amplitude above ±0.03, if the applied dropout rate is appropriate, applying dropout can reduce overfitting, thus reducing the testing RMSE of the prediction model. When the dropout rate is 0.2, the prediction performance for testing data with larger maximum noise amplitude (±0.03, ±0.05, ±0.07) is significantly improved and the overfitting problem is largely avoided, while the prediction accuracy for the testing data with no noise and the maximum noise amplitude of ±0.01 has only deteriorated very slightly.

Figure 13 gives some examples of prediction results for the aforementioned sufficiently trained LSTM models with different dropout rates, where the testing data contains

Table 2: Training and testing RMSE of LSTM prediction model trained by 100,000 epochs with different dropout rates.

Dropout rate	Training RMSE	Testing RMSE				
		No noise	Noise ±0.01	Noise ±0.03	Noise ±0.05	Noise ±0.07
0	0.1258	1.8098	2.0543	4.1357	6.4225	8.7408
0.05	0.7406	1.9228	2.2484	3.5289	5.2413	7.2842
0.10	0.8566	1.9896	2.2555	3.5443	5.1793	6.5903
0.20	1.3063	2.1803	2.3142	3.2418	4.5737	5.9841
0.30	1.7377	2.6401	2.8563	3.6682	4.9724	6.3992
0.40	2.7155	3.4187	3.6943	3.9189	5.1976	6.7855
0.50	3.6668	4.2122	4.4845	4.9057	5.7148	7.0513



(a)

(b)

(c)

Figure 13: Prediction results for testing data containing artificial noise (max ±0.07) by sufficiently trained LSTM model with a different dropout rate: (a) no dropout; (b) dropout rate = 0.2; (c) dropout rate = 0.5.

artificial noise with the maximum amplitude of ±0.07, and the severity of the abnormal condition used in the test is 0.55. The illustrated prediction results show that the predicted results generated by the LSTM prediction model with an appropriate dropout rate (0.2) are in relatively good agreement with the actual remaining time to reactor trip, while the outputs of the model with no dropout and large dropout rate (0.5) both have a larger deviation from the actual value in the initial stage of the abnormal condition. The experimental results of Table 2 and Figure 13 jointly proved that applying an appropriate dropout rate to the proposed LSTM prediction model can address the overfitting problem for noisy testing data.

However, it should be emphasized that all the above comparisons in this section were made on the premise of a large training epoch number (100,000). Although the LSTM prediction models with dropout can achieve better performance when the training epoch number is sufficiently

Table 3: Comparison of LSTM prediction models with the best performance.

Models with the best performance	Training RMSE	Testing RMSE				
		No noise	Noise ±0.01	Noise ±0.03	Noise ±0.05	Noise ±0.07
No dropout	1.5849	2.2960	2.3094	3.1993	4.5615	5.8320
Dropout rate of 0.20	1.3063	2.1803	2.3142	3.2418	4.5737	5.9841

large, applying dropout only avoids overfitting, but cannot improve the best prediction performance that can be achieved by the prediction models based on LSTM. The best prediction performance of the models with no dropout (i.e., the best prediction result shown in Figure 11(b)) and the prediction performance of a sufficiently trained model with a dropout rate of 0.2 are compared in Table 3. The comparison results show that the best testing RMSE of a prediction model with a dropout rate of 0.2 is not better than that of the prediction model without dropout. Therefore, in fact, the main advantage of applying dropout is to simplify the training process, that is, the training epoch number can be set sufficiently large without the risk of overfitting, and it is not necessary to search for the best training error, which can be difficult for the new training dataset.

6. Conclusions

In the present paper, a deep-learning model based on LSTM and dropout was proposed for predicting the remaining time to automatic scram during abnormal conditions of NPPs. The prediction result of the proposed model can provide NPP operators with a predicted countdown for the upcoming reactor trip, as such the operators can prepare or take countermeasures for it to avoid further damage to the reactor.

The proposed prediction method was trained and tested by monitoring data of abnormal conditions generated by a full-scale PWR simulator. The input of the prediction model was the monitoring data during abnormal conditions, and the output was the remaining time from the current moment to the upcoming reactor trip. The monitoring data of abnormal conditions with different severity was used to train and test the proposed prediction method separately.

The experiment results show that the prediction model based on LSTM generated a more accurate automatic scram prediction than the prediction model based on simple RNNs such as the Elman neural network. The LSTM prediction model did not show an obvious overfitting problem when tested by the input data without noise. Furthermore, the performance of the LSTM prediction model is not sensitive to the number of hidden layer nodes.

The influence of data noise on the automatic scram prediction performance was also studied. When the prediction model was tested by input data containing artificial noise, the prediction model based on LSTM achieved significantly better performance than the Elman neural network model, but showed some degree of overfitting.

The overfitting problem of the LSTM prediction model can be addressed by adding a dropout layer. When the dropout rate is 0.2, the prediction model can largely avoid overfitting for noisy testing datasets even if the training epoch number is sufficiently large. However, applying dropout cannot improve the best prediction performance achieved by the proposed prediction model, and the main advantage of applying dropout is to simplify the training process; that is, the training epoch number can be set sufficiently large without the risk of overfitting.

The proposed automatic scram prediction model based on LSTM and dropout can achieve good prediction performance within a wide range of model parameters, so the proposed prediction model is suitable to be applied in real NPP systems.

In the present work, the proposed prediction method for automatic scram was only applied in the fault mode of the steam generator's secondary side leakage. In the future work, more fault modes will be tested to verify the proposed model. Furthermore, advanced machine learning algorithms such as random forest, BiLSTM, and attention mechanism will be applied in the future work to improve the performance of the prediction model.

Data Availability

The original data of abnormal condtions used in the present work are stored in the simulator system of Harbin Engineering University and can be obtained after approval.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper was supported by the University Scientific Research Project of Guangdong Provincial Department of Education (grant no. KJ2021C016), University Stability Support Program of Shenzhen City (grant no. 20220817214021001), and PhD Research Startup Project of Shenzhen Institute of Information Technology (grant no. SZIIT2021KJ048).

References

[1] J. S. Kang and S. J. Lee, "Concept of an intelligent operator support system for initial emergency responses in nuclear power plants," *Nuclear Engineering and Technology*, vol. 54, no. 7, pp. 2453–2466, 2022.

[2] R. M. Ayo-Imoru and A. C. Cilliers, "A survey of the state of condition-based maintenance (CBM) in the nuclear power industry," *Annals of Nuclear Energy*, vol. 112, pp. 177–188, 2018.

[3] J. Coble, P. Ramuhalli, L. Bond, J. W. Hines, and B. Upadhyaya, "A review of prognostics and health management applications in nuclear power plants," *International*

Journal of Prognostics and Health Management, vol. 6, no. 3, pp. 1–22, 2020.

[4] G. Liu, W. Fan, F. Li, G. Wang, and D. You, "Remaining useful life prediction of nuclear power machinery based on an exponential degradation model," *Science and Technology of Nuclear Installations*, vol. 2022, Article ID 9895907, 9 pages, 2022.

[5] E. Zio and F. di Maio, "A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system," *Reliability Engineering & System Safety*, vol. 95, no. 1, pp. 49–57, 2010.

[6] J. Coble and J. W. Hines, "Applying the general path model to estimation of remaining useful life," *International Journal of Prognostics and Health Management*, vol. 2, pp. 71–82, 2011.

[7] B. Rezaeianjouybari and Y. Shang, "Deep learning for prognostics and health management: state of the art, challenges, and opportunities," *Measurement*, vol. 163, Article ID 107929, 2020.

[8] A. Oluwasegun and J. C. Jung, "The application of machine learning for the prognostics and health management of control element drive system," *Nuclear Engineering and Technology*, vol. 52, no. 10, pp. 2262–2273, 2020.

[9] H. Chen, P. Gao, S. Tan, J. Tang, and H. Yuan, "Online sequential condition prediction method of natural circulation systems based on EOS-ELM and phase space reconstruction," *Annals of Nuclear Energy*, vol. 110, pp. 1107–1120, 2017.

[10] M. Marseguerra, E. Zio, P. Baraldi, and A. Oldrini, "Fuzzy logic for signal prediction in nuclear systems," *Progress in Nuclear Energy*, vol. 43, no. 1-4, pp. 373–380, 2003.

[11] M. Marseguerra, E. Zio, and P. Avogadri, "Model identification by neuro-fuzzy techniques: predicting the water level in a steam generator of a PWR," *Progress in Nuclear Energy*, vol. 44, no. 3, pp. 237–252, 2004.

[12] Y. k. Liu, F. Xie, C. l. Xie, M. j. Peng, G. h. Wu, and H. Xia, "Prediction of time series of NPP operating parameters using dynamic model based on BP neural network," *Annals of Nuclear Energy*, vol. 85, pp. 566–575, 2015.

[13] J. Liu, R. Seraoui, V. Vitelli, and E. Zio, "Nuclear power plant components condition monitoring by probabilistic support vector machine," *Annals of Nuclear Energy*, vol. 56, pp. 23–33, 2013.

[14] Y. D. Koo, H. Seon Jo, M. G. Na, K. H. Yoo, and C. H. Kim, "Prediction of the internal states of a nuclear power plant containment in LOCAs using rule-dropout deep fuzzy neural networks," *Annals of Nuclear Energy*, vol. 156, Article ID 108180, 2021.

[15] Y. J. Zhang and L. S. Hu, "Real time estimation of radionuclides in the receiving water of an inland nuclear power plant based on difference gated neural network," *Radiation Physics and Chemistry*, vol. 176, Article ID 109019, 2020.

[16] S. Park, J. Park, and G. Heo, "Transient diagnosis and prognosis for secondary system in nuclear power plants," *Nuclear Engineering and Technology*, vol. 48, no. 5, pp. 1184–1191, 2016.

[17] J. Bae, G. Kim, and S. J. Lee, "Real-time prediction of nuclear power plant parameter trends following operator actions," *Expert Systems with Applications*, vol. 186, Article ID 115848, 2021.

[18] J. Zhang, X. Wang, C. Zhao et al., "Application of cost-sensitive LSTM in water level prediction for nuclear reactor pressurizer," *Nuclear Engineering and Technology*, vol. 52, no. 7, pp. 1429–1435, 2020.

[19] H. P. Nguyen, P. Baraldi, and E. Zio, "Ensemble empirical mode decomposition and long short-term memory neural network for multi-step predictions of time series signals in nuclear power plants," *Applied Energy*, vol. 283, Article ID 116346, 2021.

[20] M. I. Radaideh, C. Pigg, T. Kozlowski, Y. Deng, and A. Qu, "Neural-based time series forecasting of loss of coolant accidents in nuclear power plants," *Expert Systems with Applications*, vol. 160, Article ID 113699, 2020.

[21] M. D. Wang, T. H. Lin, K. C. Jhan, and S. C. Wu, "Abnormal event detection, identification and isolation in nuclear power plants using LSTM networks," *Progress in Nuclear Energy*, vol. 140, Article ID 103928, 2021.

[22] Y. Yu, M. j. Peng, H. Wang, Z. g. Ma, S. y. Cheng, and R. y. Xu, "A continuous learning monitoring strategy for multi-condition of nuclear power plant," *Annals of Nuclear Energy*, vol. 164, Article ID 108544, 2021.

[23] G. Qian and J. Liu, "Fault diagnosis based on gated recurrent unit network with attention mechanism and transfer learning under few samples in nuclear power plants," *Progress in Nuclear Energy*, vol. 155, Article ID 104502, 2023.

[24] T. Hu, K. Li, H. Ma, H. Sun, and K. Liu, "Quantile forecast of renewable energy generation based on Indicator Gradient Descent and deep residual BiLSTM," *Control Engineering Practice*, vol. 114, Article ID 104863, 2021.

[25] K. Liu, M. F. Niri, G. Apachitei, M. Lain, D. Greenwood, and J. Marco, "Interpretable machine learning for battery capacities prediction and coating parameters analysis," *Control Engineering Practice*, vol. 124, Article ID 105202, 2022.

[26] T. Hu, H. Ma, K. Liu, and H. Sun, "Lithium-ion battery calendar health prognostics based on knowledge-data-driven attention," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 1, pp. 407–417, 2023.

[27] T. Hu, H. Ma, H. Liu, H. Sun, and K. Liu, "Self-attention-based machine theory of mind for electric vehicle charging demand forecast," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 8191–8202, 2022.

[28] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[29] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[30] 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.

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

[32] X. Li, L. Zhang, Z. Wang, and P. Dong, "Remaining useful life prediction for lithium-ion batteries based on a hybrid model combining the long short-term memory and Elman neural networks," *Journal of Energy Storage*, vol. 21, pp. 510–518, 2019.

[33] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[34] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series," in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, pp. 3285–3292, Los Angeles, CA, USA, December 2019.

[35] G. E. Hinton, N. Srivastava, and A. Krizhevsky, "Improving neural networks by preventing co-adaptation of feature detectors," *Computer Science*, vol. 3, no. 4, pp. 212–223, 2012.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from over-fitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.