Nuclear Engineering and Technology 52 (2020) 1429–1435

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

Application of cost-sensitive LSTM in water level prediction for nuclear reactor pressurizer

Jin Zhang [a, b, 1], Xiaolong Wang [c, 1], Cheng Zhao [a, d], Wei Bai [a], Jun Shen [b], Yang Li [a], Zhisong Pan [a, *], Yexin Duan [b, **]

[a] Command and Control Engineering College, Army Engineering University of PLA, Nanjing, 210000, China
[b] Zhenjiang Campus, Army Military Transportation University of PLA, Zhenjiang, 212003, China
[c] College of Nuclear Science and Technology, Naval University of Engineering, Wuhan, 430033, China
[d] Anhui Provincial Key Laboratory of Network and Information Security, Anhui Normal University, Wuhu, 241000, China

ARTICLE INFO

Article history:
Received 21 May 2019
Received in revised form
20 December 2019
Accepted 24 December 2019
Available online 27 December 2019

Keywords:
LSTM
Parameter prediction
Cost sensitive
Pressurizer
Pressurized water reactor
Time series

ABSTRACT

Applying an accurate parametric prediction model to identify abnormal or false pressurizer water levels (PWLs) is critical to the safe operation of marine pressurized water reactors (PWRs). Recently, deep-learning-based models have proved to be a powerful feature extractor to perform high-accuracy prediction. However, the effectiveness of models still suffers from two issues in PWL prediction: the correlations shifting over time between PWL and other feature parameters, and the example imbalance between fluctuation examples (minority) and stable examples (majority). To address these problems, we propose a cost-sensitive mechanism to facilitate the model to learn the feature representation of later examples and fluctuation examples. By weighting the standard mean square error loss with a cost-sensitive factor, we develop a Cost-Sensitive Long Short-Term Memory (CSLSTM) model to predict the PWL of PWRs. The overall performance of the CSLSTM is assessed by a variety of evaluation metrics with the experimental data collected from a marine PWR simulator. The comparisons with the Long Short-Term Memory (LSTM) model and the Support Vector Regression (SVR) model demonstrate the effectiveness of the CSLSTM.

1. Introduction

Pressurizer water level (PWL) is a crucial parameter for marine pressurized water reactors (PWRs). Operators use it as key evidence to comprehend reactors' operating conditions and identify their transients [1]. However, affected by load fluctuations in marine nuclear power equipment as well as the harsh working conditions caused by high temperature and humidity, pressurizers are prone to problems such as measurement faults and steam-and-water mixture [2]. These unfavorable conditions could lead to occurrences of abnormal indications or false water levels, making it challenging for operators to identify pressurizers' real water levels. This could consequently add to the difficulty of operation and increase the probability of human errors, resulting in nuclear disasters such as the Three Mile Island accident [3]. Fortunately, relevant studies [4] shows that there is a strong internal relation among the main parameters of PWRs. Therefore, when the water level signal is lost, the real water level value can be reconstructed and predicted via other easily obtained and normally displayed feature parameters. Moreover, the predicted PWL can help with the accurate evaluation of meter reading or the recalibration of the lost signal.

In general, the reconstruction and prediction methods of PWL can be classified into the following two categories [5]:

1. Mathematical-physical models, which rely on a large security analysis program to realize parametric prediction. However, the security analysis program needs to be modified for different PWRs, and it is not easy to interact with actual running data [4]. As a result, such methods have demonstrated poor generality and transfer adaptability. In addition, the more complicated the model is, the more rounding and transfer errors tend to occur in the calculation process, which consequently enlarges the deviation between the predicted value and the actual value [5]. This deficiency is not likely to be compensated by a simplified model, because it cannot achieve effective forecasting accuracy.

* Corresponding author.
** Corresponding author.
E-mail addresses: hotpzs@hotmail.com (Z. Pan), duanyexin0713@163.com (Y. Duan).
[1] Contributed equally to this work.

https://doi.org/10.1016/j.net.2019.12.025

2. Data-driven approaches, which use machine learning models to learn the mapping between the PWL and normal feature parameters. Such methods are able to achieve high accuracy and strong applicability. In particular, the deep neural networks (DNNs), as a new focus across the research community, have proved to outperform traditional machine learning methods (such as SVR models and BP networks) by large margins [6,7].

We are thus motivated to adopt LSTM, a DNN with temporal characteristic, to build a PWL prediction model to track the temporal changes of PWL. However, as LSTM models usually treat all examples with equal importance during training, it is necessary to make some modifications to our proposed LSTM model. Otherwise, the model is very likely to be overwhelmed by the majority data (stable condition examples) [8] and suffer from the correlations shifting over time among the PWL and other feature parameters. This might seriously undermine the model's performance and updating [9].

Considering the temporal and transient working condition features of nuclear equipment's operating parameters, the examples should vary in importance for training. First, training examples appearing later in a time series are more important than previous ones due to the temporal shifting of the correlations between parameters. Second, volatile examples are more important than those with stable changes as they usually contain more information than stable ones. Therefore, we propose a cost-sensitive mechanism for training examples, which should demonstrate four properties: time series sensitivity, fluctuation sensitivity, eternal positivity, and convergence. Based on this mechanism, a Cost-Sensitive LSTM (CSLSTM) model is established, which takes six parameters highly coupled with the PWL as the feature parameters to predict and reconstruct the PWL.

With experimental data collected from a marine PWR simulator, the comparative experiments are performed under both *global learning* and *local learning* modes. The results show that the CSLSTM outperforms the LSTM model and the SVR model on multiple evaluation metrics.

The main contributions of this paper are as follows:

1. For the first time, a deep learning framework is applied to predict a nuclear power equipment parameter. We propose a novel parameter prediction model for the PWL in an end-to-end way based on the LSTM deep learning network.
2. Considering both temporal and fluctuation characteristics of PWL, we introduce a cost-sensitive mechanism and establish a CSLSTM model, which could further improve the performance by letting the model pay more attention to the later examples and fluctuation examples during training.

The rest of this paper is arranged as follows: Section 2 describes the preliminary knowledge of this paper, including feature selection, data preparation and non-deep learning methods for the prediction of nuclear power equipment parameters; Section 3 introduces in detail the LSTM model, the CSLSTM model, the training and optimization; In Section 4, we verify the effectiveness of the CSLSTM model by comparing it with the LSTM and SVR models, and analyze the experimental results of the time series prediction of PWL. Finally, the conclusion of this paper is drawn in Section 5.

2. Related works

2.1. Feature selection and data preparation

Time series that affect the PWL are high-dimensional features, which not only contain rich and useful information but also include irrelevant or redundant features. These irrelevant and redundant features can undermine the prediction accuracy and efficiency of the model. Thus, selecting valuable features based on their correlations with the PWL is crucial for prediction. After theoretical analysis and simulation test, we follow Wang's practice [10] and select six parameters highly coupled with the PWL as the model's input features, including reactor inlet and outlet average temperatures, the pressure and temperature of the pressurizer, the main pump flow of the primary circuit, and the reactor's nuclear power.

In this study, our time series data is collected from a RELAP5 marine PWR simulator in terms of seven parameters including the six input feature parameters and output parameter, i.e. the PWL. Since these parameters usually have different dimensions and magnitude orders, we normalize the raw data (Equation (1)) to generate our experimental data set. According to the transient process of increasing reactor power from 30% to 90%, our experiments collect 3667 groups of original data sets with the time step of 1 s. After normalization, the input feature parameters are expressed as $\mathrm{X} = [x_1, x_2, \cdots, x_{3667}]$, where vector x_i denotes the normalized value of the six input feature parameters at moment *i*. The output variable, namely the PWL, is expressed as $\mathrm{Y} = [y_1, y_2, \cdots, y_{3667}]$, where y_i denotes the normalized PWL at moment *i*.

$$\mathrm{X} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

2.2. PWRs parametric prediction methods

In essence, time series parametric prediction for PWRs is a kind of regression analysis. Machine learning methods applied by recent studies to regression analysis on PWR parameters can be mainly classified into the following two categories, i.e., shallow neural network approaches and non-neural-network approaches.

Before the rise of deep learning, SVR is the most successful non-neural-network method in regression analysis owing to its efficiency [5]. For example, Wang et al. studied the prediction and reconstruction of marine nuclear power parameters via an SVR method [10]; Liu et al. [11] applied an SVR model to predict marine nuclear power equipment's faults and failures. Although these SVR models have demonstrated quick learning ability and high efficiency, their weaknesses are still obvious. First, except learning support vectors, they are unable to learn the feature representation from other examples sufficiently, which limits their generalization and capacity. More importantly, in SVR models, data is assumed to be independent and identically distributed, which poorly reflects the strong temporal features of PWR parameters.

Another prevalent strategy uses shallow neural network models such as the Auto-Associative Neural Network (AANN) and the Back-propagation (BP) Neural Network for regression analysis [13–22]. For instance, Sameer et al. [17] and Maio et al. [18] proposed an Auto-Associative Kernel Regression (AAKR) model to predict signals in nuclear power plants. Huang et al. [15] adopted a BP Neural Network to predict PWR's departure from nucleate boiling ratio. Such models have powerful nonlinear mapping ability. In theory, and the deeper the network goes, the stronger the ability gets. However, in practice, they tend to be easily stuck with local minimum due to gradient vanishing [23].

Recently, to avoid gradient vanishing and ensure models' strong nonlinear mapping ability, deep learning methods have emerged in the spotlight [24]. By abstractly simulating human neurons and their internal links, deep learning models do not have to rely on high-quality features, so they are able to extract features from input signals layer by layer and find deeper underlying rules [25]. Therefore, an increasing number of deep learning-based approaches are employed in nuclear engineering tasks such as nuclear

reactor perturbation analysis [12] and anomaly detection [34]. In particular, the Recurrent Neural Network (RNN) introduce the concept of time series into network structure design, which makes models more adaptative to time series data analysis. As an improvement of RNN, the Long Short-Term Memory (LSTM) model has shown impressive performance in regression analysis for tasks such as traffic flow forecasting [26], power demand prediction [27], machine translation [28], and PWR accident diagnosis [29]. Because of its weight sharing mechanism and cyclic structure, it is able to effectively compensate for problems such as gradient vanishing, gradient exploding, and insufficient long-term memory capacity [25]. Based on previous research, we extend the application of the LSTM to nuclear power equipment parametric prediction and provide a novel method, the CSLSTM, to predict the PWL.

3. Method and implementation

3.1. LSTM model

Different from standard neurons, RNN neurons have a cyclic structure and can transfer the information from the previous state to the current state. As shown in Fig. 1, when the input information is a time series, it can be expanded into a series of mutual-connected standard neurons.

By using a standard RNN model for an input sequence $X = [x_1, x_2, \cdots, x_n]$ with a given length of time n, a hidden layer sequence $H = [h_1, h_2, \cdots, h_n]$ and an output sequence $\widehat{Y} = (\widehat{y}_1, \widehat{y}_2, \cdots, \widehat{y}_n)$ can be obtained through iterative formulas (2) and (3). At a given time t, the hidden layer sequence, the output sequence, and the input eigenvector are defined as:

$$h_t = \sigma(W_{xh} \otimes x_t + W_{hh} \otimes h_{t-1} + b_h),\ t = 1, 2, \cdots, n \tag{2}$$

$$\widehat{y}_t = W_{hy} \otimes h_t + b_y,\ t = 1, 2, \cdots, n \tag{3}$$

$$x_t = \left[x_t^1, x_t^2, \cdots, x_t^p\right],\ t = 1, 2, \cdots, n \tag{4}$$

Where x_t^p denotes the value of the p^{th} input feature at moment t; W represents a matrix of weight coefficients (for example, W_{xh} represents a matrix of weight coefficients from the input layer x to the hidden layer h), b refers to the bias vector (for example, b_h and b_y respectively denotes the bias vector in the hidden layer and the output layer, respectively); $\otimes$ means matrix multiplication; and σ represents an activation function such as sigmoid, tanh or ReLU [30]. Note that the RNN has shared variables including W, h, and b at different moments.

Although the RNN can effectively deal with nonlinear time series, the following two problems still exist. First, it cannot deal with long delay time series due to gradient vanishing and exploding.

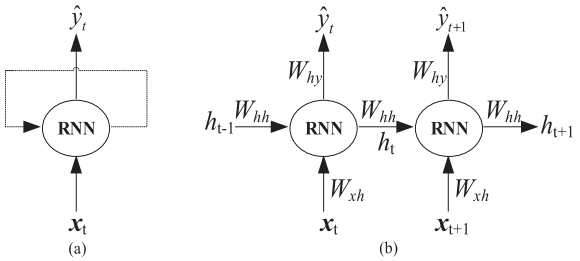

Fig. 1. The Structure of RNN Network: (a) The RNN unit; (b) RNN sequences. x_t and $\widehat{y}_t$ represent the input features and output variable at moment t. h and W refer to the hidden layer unit and the weight to be learned.

Second, to train RNN models, it is necessary to pre-determine the length of the delay window. However, in reality, it is difficult to automatically obtain this parameter's optimal value. This is where the LSTM model can be applied. The LSTM model can replace the RNN cells in the hidden layer with LSTM cells, so as to provide the hidden layer with long-term memory. Currently, the cell structure of the most widely used LSTM model has been the one exhibited in Fig. 2. Its forward calculation method is expressed as follows:

$$i_t = \sigma(W_{xi} \otimes x_t + W_{hi} \otimes h_{t-1} + W_{ci} \otimes c_{t-1} + b_i) \tag{5}$$

$$f_t = \sigma\left(W_{xf} \otimes x_t + W_{hf} \otimes h_{t-1} + W_{cf} \otimes c_{t-1} + b_f\right) \tag{6}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tanh(W_{xc} \otimes x_t + W_{hc} \otimes h_{t-1} + b_c) \tag{7}$$

$$o_t = \sigma(W_{xo} \otimes x_t + W_{ho} \otimes h_{t-1} + W_{co} \otimes c_t + b_o) \tag{8}$$

$$h_t = o_t \otimes \tanh(c_t) \tag{9}$$

Wherein, i_t, f_t, o_t, and c_t respectively represents the values of the input gate, the forgotten gate, the output gate and the cell state at moment t; σ and tanh refers to the sigmoid and the hyperbolic tangent activation function.

Combining formulas (2) - (9), an LSTM prediction model and training framework for the PWL is therefore established (Fig. 3). The model takes parameters strongly correlated with PWL as input feature parameters $x_t \in R^p$, in which, p represents the number of parameters. We follow the convention of regression analysis and adopt mean square error (MSE) as the loss function for training.

$$\text{Min}\ \ Loss = \frac{1}{N}\sum_{t=1}^{N}(y_t - \widehat{y}_t)^2 \tag{10}$$

In Formula (10), y_t and $\widehat{y}_t$ refer to the real value and the predicted value of the PWL at a given moment t, respectively. N is the

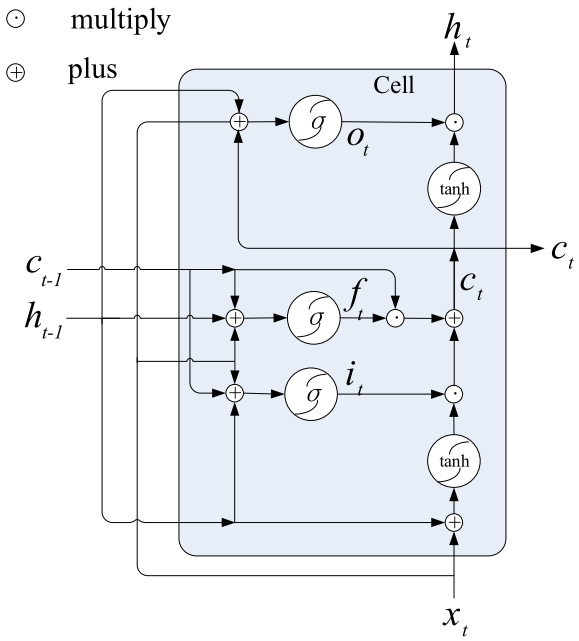

Fig. 2. The structure of LSTM cells in the hidden layer. x_t, h_t, i_t, f_t, c_t and o_t respectively represents the values of the input eigen vector, the hidden layer sequence, the input gate, the forgotten gate, the cell state, and the output gate at moment t. σ and tanh respectively refers to the sigmoid and the hyperbolic tangent activation function.

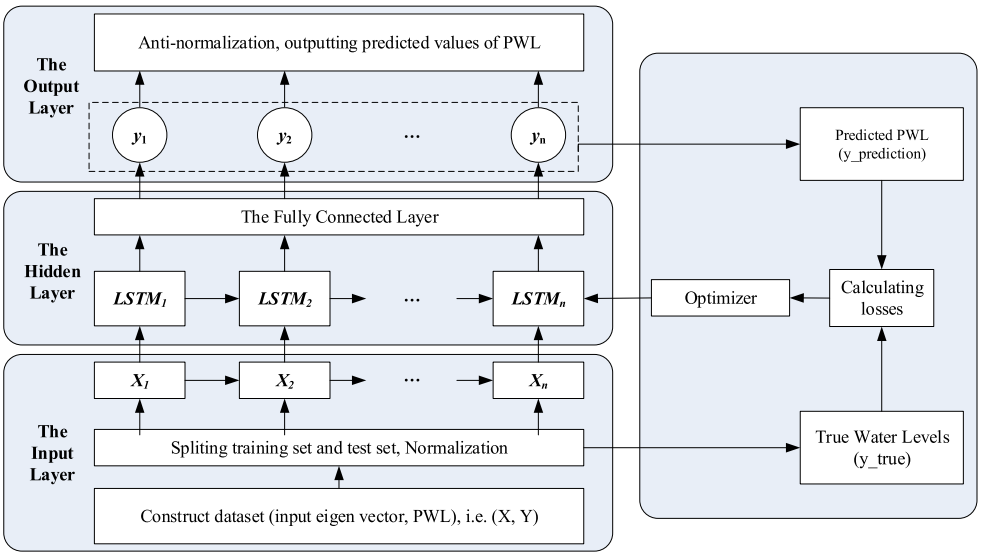

Fig. 3. LSTM-Based prediction model for the PWL and its training framework.

total number of training examples.

3.2. CSLSTM model

Since the LSTM regression model typically regards Formula (10) as a loss function, it in fact implies an assumption that all training examples have the same importance. However, this is not reasonable in PWL prediction. Unlike standard datasets where examples are independent identically distributed and class balance, our training data suffer from two drawbacks that reduce the generalization of the models: 1) the correlations between the PWL and input feature parameters might shift over time. Thus, the model should pay more attention to recent examples during training. However, ignoring long-ago historical data is not a good strategy because we don't know when and how the shifting takes place; 2) as fluctuation condition examples (the minority data) usually contain more complicated feature representation than stable condition ones (the majority data), the latter might be easily overwhelmed by the former. We accordingly propose to introduce a cost sensitive factor r_t to help the model pay more attention to more important and recent training data, so that we can obtain correct mappings between the PWL and input features. To be specific, this factor r_t should meet with the following properties:

1. Fluctuation sensitivity, which shares positive correlation with the fluctuation factor a_t.
2. Time series sensitivity, which monotonically increases with time t, i.e. $r_t < r_{t+1}$.
3. Eternally positive, $r_t > 0$.
4. Convergence, $\lim_{t\to\infty} r_t < \infty$.

Since the second derivative can better represent the speed of parametric change, we use it to describe the fluctuation of PWL and introduce the fluctuation factor a_t.

$$a_t = \frac{||y_t - y_{t-1}| - |y_{t-1} - y_{t-2}||}{\sum_{t=1}^{N} ||y_t - y_{t-1}| - |y_{t-1} - y_{t-2}||}, \quad t = 1, 2, \cdots, N \tag{11}$$

In the above equation, we set $y_{-1} = y_0 = y_1$, then $0 \le a_t \le 1$ and $\sum_{t=1}^{m} a_t = 1$. Based on a_t, we define r_t as:

$$r_t = \prod_{i=1}^{t}\left(a_i + 1 + \frac{1}{N}\right), \ t = 1, 2, \cdots, N \tag{12}$$

Where r_t obviously meets with property 1 and 3, and the existence of $1/N$ precisely ensures that r_t should satisfy property 2. Based on the rule that the geometric mean of positive numbers is less than or equal to the arithmetic mean, r_t can also be proved to meet with property 4:

$$r_N = \prod_{i=1}^{N}\left(a_i + 1 + \frac{1}{N}\right) \le \left(\frac{\sum_{i=1}^{N}\left(a_i + 1 + \frac{1}{N}\right)}{N}\right)^N = \left(1 + \frac{2}{N}\right)^N \tag{13}$$

Since $\lim_{N\to\infty}\left(1 + \frac{2}{N}\right)^N = e^2$, r_t is definitely convergent.

3.3. Training and optimization

Due to the CSLSTM model's cyclic and weight sharing mechanism, the back-propagation algorithm cannot be directly applied to the model's training. We therefore need to adopt the Back-propagation Through Time (BPTT) algorithm, the training process of CSLSTM is illustrated in Table 1.

There are many typical gradient-based optimization algorithms, such as Stochastic Gradient Descent (SGD), AdaGrad [32] and RMSProp [33]. For our proposed model, we adopt the Adam algorithm [31], an effective gradient-based stochastic optimization algorithm that combines the advantages of AdaGrad and RMSProp algorithms. In addition, it also has strong robustness in choosing hyper-parameters.

In the experiments, for a fair comparison, both the CSLSTM and LSTM model adopt the same configurations.

i.e., a single hidden layer with 128 LSTM cell followed with a fully-connected layer. Based on Tensorflow framework, we train both models using Adam optimizer with a batch size of 8, setting learning rate to 0.001 and training step to 1e5, 1e6 and 2e6 respectively.

Table 1
Algorithm for the training process of the CSLSTM.

Input: Train data (x_t, y_t), where $t = 1, 2, \cdots, N$. x_t is highly coupled with y_t.
Output: variables including W and b
1: initialize W and b
2: calculate the cost-sensitive factor r_t
3: **for** each batch of $(x_t, y_t) \in$ training data **do**
4: forward calculation $\hat{y}_t$
5: calculate the batch prediction *loss*
6: back calculate the gradient of each weight W and offset b according to corresponding BPTT
7: update W and b
8: **end for**

4. Results and discussion

4.1. Data description

In this study, we follow the conventional practice [12] and adopt four metrics to evaluate the performance of PWL time series prediction models, namely root mean square error (RMSE), normalized root mean square error (NRMSE), symmetric mean absolute percentage error (SMAPE), and mean absolute error (MAE).

$$\mathrm{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}[\hat{y}_i - y_i]^2} \tag{14}$$

$$\mathrm{NRMSE} = \frac{1}{\hat{y}_{\max} - \hat{y}_{\min}}\sqrt{\frac{1}{M}\sum_{i=1}^{M}[\hat{y}_i - y_i]^2} \tag{15}$$

$$\mathrm{SMAPE} = \frac{1}{M}\sum_{i=1}^{M}\frac{|\hat{y}_i - y_i|}{(|\hat{y}_i| + |y_i|)/2} \tag{16}$$

$$\mathrm{MAE} = \frac{1}{M}\sum_{i=1}^{M}|\hat{y}_i - y_i| \tag{17}$$

In the above evaluation metrics, the smaller the values, the better the prediction results get. M is the total number of testing examples. y_i and $\hat{y}_i$ refer to the PWL and its prediction value, respectively.

Moreover, we evaluate the performance of models under two modes, i.e. the global learning mode and the local learning mode. In global learning experiments, training examples are sampled from the entire operating condition, i.e. global feature space (Fig. 4 (a)). In this case, the whole data set includes 3667 groups of data. We split 80% of it for training and 20% for testing. Specifically, if a data group's sequence number can be divided by 5, it is thus cataloged as a test example (Fig. 4 (a)), i.e. $(x_5, y_5), (x_{10}, y_{10}), \cdots, (x_{3665}, y_{3665})$. We thus obtain 733 groups of testing examples in total. The rest 2994 groups of examples are used for training. In local learning experiments, training examples are sampled from part of the operating condition, i.e. part of the feature space (Fig. 4 (b)), while testing examples belong to another part of the feature space. To be concrete, we set the first 1600 groups data (the transient process of increasing reactor power from 30% to 78%) as the training set, i.e. $(x_1, y_1), (x_2, y_2), \cdots, (x_{1600}, y_{1600})$. The next 400 data groups (the transient change of increasing reactor power from 78% to 90%) form the testing set, including $(x_{1601}, y_{1601}), (x_{1602}, y_{1602}), \cdots, (x_{2000}, y_{2000})$.

4.2. Experimental results and analysis

On both the global learning and local learning setting, we perform comparative experiments between our models and the SVR models that have produced inspiring results in recent years on time series prediction. The SVR experiments are implemented on Scikit-learn framework using multiple kernel functions. In practice, we adopt default configurations for SVR models, but set *epsilon* to 10^{-4} for competitive performance.

In global learning experiments, our deep learning methods surpass the SVR models. In particular, the CSLSTM model with fluctuation sensitivity produces the lowest error of [0.001195, 0.006083, 0.001023, 0.000966] (Table 2) for RMSE, NRMSE, SMAPE, and MAE, respectively. Although the SVR models have high computational efficiency, they show poor robustness which can be affected by the selection of kernel functions. For example, the error produced by the radial basis kernel function in the global learning experiments is three times the error of the linear kernel function; however, in the local learning experiments, it works better than the latter (Table 2). On the contrary, the LSTM and CSLSTM models are more robust, which gradually converge as the number of iterations goes from 1e5, 1e6 to 2e6 (Fig. 5). In general, the LSTM and CSLSTM models can easily learn the feature representation between the PWL and other input feature parameters in the global learning mode, since the training data is sampled from the global feature space through the entire operating condition. Nonetheless, to further verify models' generalization ability, we need to perform experiments on the local learning setting.

In the local learning experiments, the performance of all models inevitably decreases by some margins as they need to make predictions outside the feature space of the training data. Nonetheless, our deep learning models still outperform the SVR models by large margins. To be specific, the LSTM model could decrease the RMSE, NRMSE, SMAPE, and MAE of SVR models with the polynomial kernel function from [0.007837, 0.521378, 0.007515, 0.007361] to [0.005250, 0.319511, 0.004867, 0.004776] (Table 2), respectively. Moreover, by integrating the cost-sensitive factor, the errors could be further reduced by almost 30%. This indicates that by focusing on the later examples and fluctuation examples, we can improve the learning of feature representation over time and prevent the rapid increase of errors in later time.

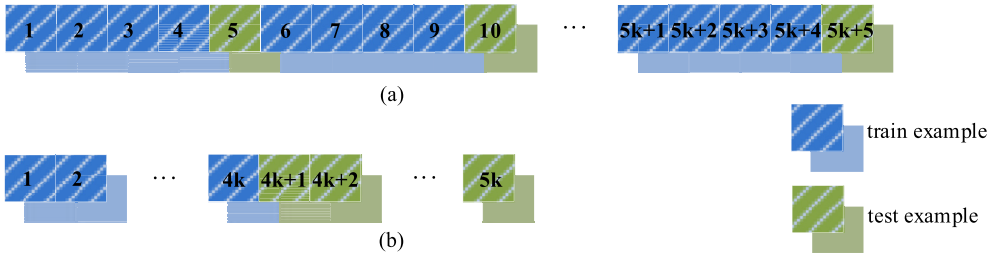

Fig. 4. Sampling of two learning settings: (a) The global learning; (b) The local learning. Examples are arranged in time series.

Table 2
Comparisons among the CSLSTM, LSTM and SVR for PWL prediction on the global learning setting and the local learning setting. Abbreviations: root mean square error (RMSE), normalized root mean square error (NRMSE), mean absolute error (MAE), symmetric mean absolute percentage error (SMAPE). The SVR models use three kernel functions, i.e. radial basis kernel function (rbf), polynomial kernel function (poly), and the linear kernel function (linear). The results of the LSTM and CSLSTM models are obtained by training 1e5, 1e6 and 2e6 step iterations respectively.

	Metrics	SVR			LSTM			CSLSTM		
		rbf	poly	linear	1e5 steps	1e6 steps	2e6 steps	1e5 steps	1e6 steps	2e6 steps
Global learning	RMSE	0.004826	0.002538	0.001558	0.006827	0.001688	0.001234	0.006430	0.001399	**0.001195**
	NRMSE	0.025083	0.012971	0.007977	0.035783	0.008668	0.006268	0.033826	0.007200	**0.006083**
	SMAPE	0.003498	0.001858	0.001226	0.006333	0.001444	0.001052	0.005948	0.001211	**0.001023**
	MAE	0.003335	0.001787	0.001153	0.005909	0.001319	0.000991	0.005543	0.001133	**0.000966**
Local learning	RMSE	0.012393	0.007837	0.014252	0.009081	0.005614	0.005250	0.009388	0.004872	**0.003848**
	NRMSE	0.903358	0.521378	0.760411	0.627127	0.332609	0.319511	0.627269	0.294474	**0.212058**
	SMAPE	0.012379	0.007515	0.014517	0.008893	0.005343	0.004867	0.009234	0.004439	**0.003442**
	MAE	0.012088	0.007361	0.014152	0.008703	0.005239	0.004776	0.009034	0.004358	**0.003381**
	NRMSE	0.903358	0.521378	0.760411	0.627127	0.332609	0.319511	0.627269	0.294474	**0.212058**
	SMAPE	0.012379	0.007515	0.014517	0.008893	0.005343	0.004867	0.009234	0.004439	**0.003442**
	MAE	0.012088	0.007361	0.014152	0.008703	0.005239	0.004776	0.009034	0.004358	**0.003381**

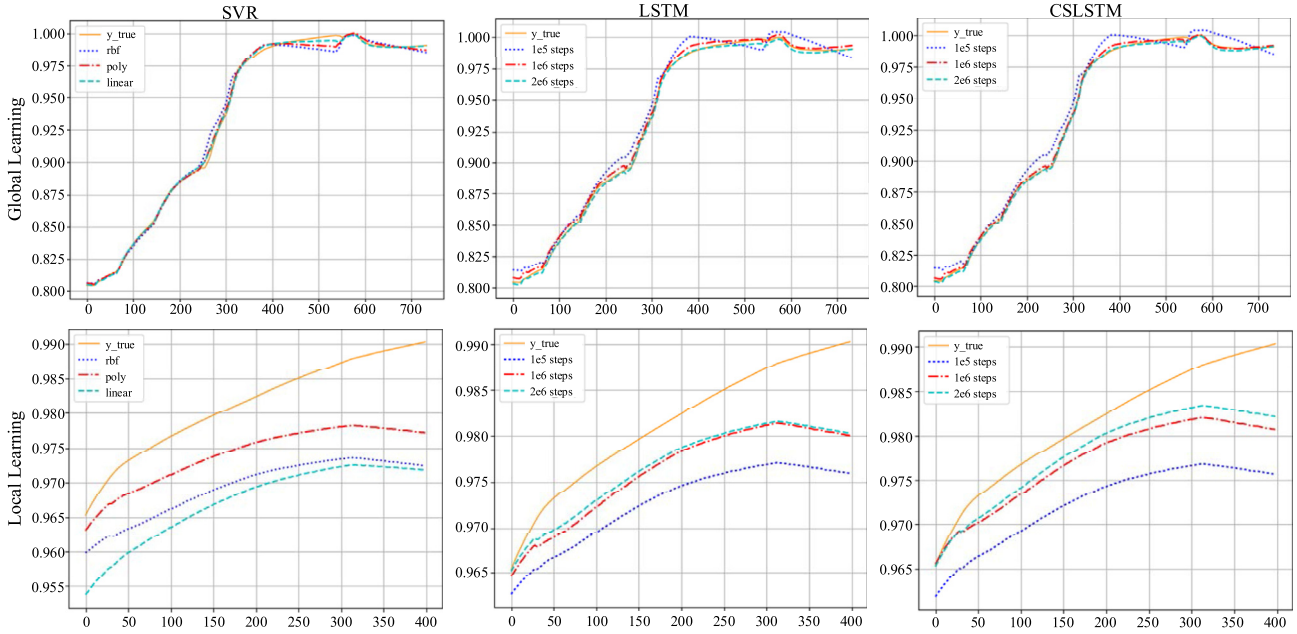

Fig. 5. The PWL prediction curves. The vertical ordinates represent the normalized PWL. The orange solid lines represent the real value of the PWL, and the dotted lines represent the values predicted by models. In global learning mode, the horizontal ordinates 1–733 respectively correspond to time sequences $[5, 10, \cdots, 3665]$. In local learning mode, the horizontal ordinates 1–400 respectively correspond to time sequences $[1601, 1602, \cdots, 2000]$. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

5. Conclusions

To ensure the safe operation of marine nuclear power equipment, we apply a deep learning method the LSTM to predict equipment parameter for the first time. Considering the time series and fluctuation characteristics of PWRs, we propose a cost-sensitive mechanism which should demonstrate four properties including time series sensitivity, fluctuation sensitivity, eternal positivity, and convergence. Based on these properties, we construct the cost-sensitive factor and create a novel CSLSTM model for PWL prediction. With experimental data collected from a marine PWR simulator, the overall performance of the CSLSTM is evaluated in the global learning experiments and the local learning experiments. The results show that the CSLSTM model outperforms both the LSTM model and SVR model on a variety of evaluation metrics. In the future, we will further improve the training efficiency and shorten the training time.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research was funded by the National Natural Science Foundation of China (61806220) and the National Key Research and Development Program of China (2017YFB0802800).

References

[1] J.H. Ye, J.M. Yi, H.Y. Ji, Research on pressurizer water level control of nuclear reactor based on RBF neural network and PID controller, Int. Conf. Mach. Learn. Cybern. (2010) 1486–1489.
[2] M. Marseguerra, M.E. Ricotti, E. Zio, Neural network-based fault detections in a pressurized water reactor pressurizer, Nucl. Sci. Eng. 124 (2) (1996) 339–348.

[3] P.L. Bot, Human reliability data, human error and accident models—illustration through the Three Mile Island accident analysis, Reliab. Eng. Syst. Saf. 83 (2) (2014) 153–167.
[4] X.L. Wang, Q. Cai, Y.Q. Chen, Abnormal parameter reconstruction for marine nuclear power plant based on multi-signal coupling, J. Shanghai Jiaot. Univ. 48 (2) (2014) 210–213.
[5] W. Xiao-long, C. Qi, Z. Zi-min, C. Yu-qing, L. Shou-xiang, Prediction for water level of steam generator based on multi-signal fusion by support vector regression, Atomic Energy Sci. Technol. 47 (2013) 1806–1811.
[6] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[7] H. Shen, X. Liang, A time series forecasting model based on deep learning integrated algorithm with stacked autoencoders and SVR for FX prediction, Int. Conf. Artif. Neural Netw. (2016) 326–335.
[8] N.V. Chawla, N. Japkowicz, A. Kolcz, Editorial: special issue on learning from imbalanced data sets, Acm Sigkdd Explor. Newslett. 6 (1) (2014) 1–6.
[9] Z. Wu, Y. Guo, W. Lin, A weighted deep representation learning model for imbalanced fault diagnosis in cyber-physical systems, Sensors 18 (4) (2018) 1096.
[10] W. Xiao-long, C. Qi, C. Yu-qing, Study on pressurizer water level signal reconstruction based on support vector regression, Atomic Energy Sci. Technol. 47 (6) (2013) 1003–1007.
[11] L. Jie, V. Valeria, Z. Enrico, R. Seraoui, A novel dynamic-weighted probabilistic support vector regression-based ensemble for prognostics of time series data, IEEE Trans. Reliab. 64 (4) (2015) 1203–1213.
[12] F.D. Ribeiro, et al., Towards a deep unified framework for nuclear reactor perturbation analysis, Symp. Ser. Comput. Intell. (2018) 120–127.
[13] C. Lombardi, A. Mazzola, Prediction of two-phase mixture density using artificial neural networks, Ann. Nucl. Energy 24 (17) (1997) 1373–1387.
[14] S. Zaferanlouei, D. Rostamifard, S. Setayeshi, Prediction of critical heat flux using ANFIS, Ann. Nucl. Energy 37 (6) (2010) 813–821.
[15] Y. Huang, J. Liu, L. Liu, Application of BP neural network in DNBR prediction, Nucl. Tech. 38 (7) (2015) 90–94.
[16] A. Shaheryar, X. Yin, H. Hao, et al., A Denoising based Auto-Associative model for robust sensor monitoring in Nuclear Power, Plant. Sci. Tech. Nucl. Inst. (2016) (2016) 1–17.
[17] A. Sameer, P. Baraldi, F.D. Maio, E. Zio, A novel fault detection system taking into account uncertainties in the reconstructed signals, Ann. Nucl. Energy 73 (2014) 131–144.
[18] F.D. Maio, P. Baraldi, E. Zio, R. Seraoui, Fault detection in nuclear power plants components by a combination of statistical methods, IEEE Trans. Reliab. 62 (4) (2013) 833–845.
[19] P. Baraldi, G. Gola, E. Zio, D. Roverso, M. Hoffmann, A randomized model ensemble approach for reconstructing signals from faulty sensors, Expert Syst. Appl. 38 (8) (2011) 9211–9224.
[20] P. Baraldi, E. Zio, G. Gola, D. Roverso, M. Hoffmann, Two novel procedures for aggregating randomized model ensemble outcomes for robust signal reconstruction in nuclear power plants monitoring systems, Ann. Nucl. Energy 38 (2–3) (2011) 212–220.
[21] P. Baraldi, A. Cammi, F. Mangili, E. Zio, An ensemble approach to sensor fault detection and signal reconstruction for nuclear system control, Ann. Nucl. Energy 37 (6) (2010) 778–790.
[22] P. Baraldi, A. Cammi, F. Mangili, E. Zio, Local fusion of an ensemble of models for the reconstruction of faulty signals, IEEE Trans. Nucl. Sci. 57 (2) (2010) 793–806.
[23] W. Wu, J. Wang, M. Cheng, Z. Li, Convergence analysis of online gradient method for BP neural networks, Neural Netw. 24 (1) (2011) 91–98.
[24] D. Guo, W. Zhou, H. Li, M. Wang, Hierarchical LSTM for sign language translation, Conf. Artificial Intell. (AAAI) (2018) 6845–6852.
[25] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[26] Y. Jia, J. Wu, M. Xu, Traffic flow prediction with rainfall impact using a deep learning method, J. Adv. Transp. (2017) (2017) 1–10.
[27] H. Shao, H. Jiang, H. Zhang, T. Liang, Electric locomotive bearing fault diagnosis using a novel convolutional deep belief network, IEEE Trans. Ind. Electron. 65 (3) (2018) 2727–2736.
[28] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: Int. Conf. Learn Represent. (ICLR), 2014.
[29] J. Yang, J. Kim, An accident diagnosis algorithm using long short-term memory, Nucl. Eng. Technol. 50 (4) (2018) 582–588.
[30] X. Jin, C. Xu, J. Feng, Y. Wei, J. Xiong, S. Yan, Deep learning with S-shaped rectified linear activation units, Natl. Int. Conf. Artif. Intell. (AAAI) (2016) 1737–1743.
[31] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, Int. Conf. Learn Represent, ICLR), 2014.
[32] J.C. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, Conf. Learn. Theory (COLT) (2010) 257–269.
[33] G. Hinton, L. Deng, D. Yu, Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
[34] F. Calivà, et al., A deep learning approach to anomaly detection in nuclear reactors, Int. Joint Conf. Neural Netw. (IJCNN) (2018) 1–8.