Hydro-Turbine Coordination Power Predictive Method of Improved Multi-Layer Neural Network Considered Adaptive Anti-Normalisation Strategy

Fannie Kong*, Jiahui Xia, Daliang Yang, Tianshun Lan
Department of Electrical Engineering, University of Guangxi, China
kongfannie@163.com, 604081439@qq.com, angdl@gxu.edu.cn, 891597340@qq.com

***Abstract*—Due to the limitation of economics and time cost, the data obtained from hydro-turbine coordination field test are insufficient to fully guide the setting of unit operating parameters. To enlarge the amount of data, realise power point tracking, and avoid the problems of high non-linearity with hydro-turbine physical model which is difficult to simulate in actual field, a mathematical prediction model is proposed based on an improved multi-layer neural network. Using the rule activation function, L2 regularisation, Adam optimiser and its gradient parameters are optimised by PSO algorithm in the prediction model. It is found that lacking true value in the process of anti-normalisation leads to difficulty for actual forecast of neural network. Therefore, an adaptive anti-normalisation strategy is proposed to improve the actual prediction accuracy, which can judge the value of the interval. According to the analysis of examples with hydro-turbine coordination and non-coordination test, the results show that the proposed prediction model and interval strategy can effectively forecast the coordination operating conditions of the turbine with high accuracy under small samples.**

***Index Terms*—Hydro-turbine power predictive; Improved multi-layer neural network; Anti-normalisation strategy; Blade angle; Guide vane openings.**

I. INTRODUCTION

To ensure the efficient operation of hydropower units, in the actual field, through the coordination and non-coordination tests of turbine units, the blade angle and guide vane openings are adjusted under a certain head to obtain data such as output power and efficiency of the units, which provides guidance for setting turbine parameters. Coordination test refers to the operating condition of hydro-turbine in which the blade angle and the guide vane openings are in a non-linear corresponding relationship. But both coordination and non-coordination experiments require a lot of economic and time costs, and usually only a limited amount of data can be obtained. The shortage of test data can be effectively compensated by establishing a prediction model to realise data enhancement.

Due to the complexity of the hydraulic turbine system, the built physical prediction model is often in a highly non-linear form [1], [2]. Generally, considering that there are fluctuations of penstock, spiral case, draft tube [3]–[5] at the actual coordination test site, it is almost impossible that the physical model fully simulated the actual environmental factors of field tests, so it is difficult to satisfy the accuracy and universality of the physical model. Therefore, to avoid the shortcomings of the physical model, this paper conducts research from the perspective of establishing a mathematical prediction model.

The neural network is a mathematical model with the ability to predict data. Guo, Wu, Lu, and Wang [6] utilise the Back Propagation (BP) neural network combined with historical data for hybrid wind speed prediction. Complex non-linear mapping can be realised by BP neural network, but there exist weak points of slow convergence speed and fall into local optimum easily with gradient descent method. To increase the speed of convergence, Montes-Atenas *et al.* [7] construct a deep neural network (DNN) model to realise the size and rate of bubble forecasting by increasing the number of multiple hidden layers. However, when the activation function is sigmoid for weight threshold correction, it needs a chain derivative from the input layer to the output layer one by one. Once the number of layers is too large, it is possible that the parameters stop updating, since the continuous multiplication of multi-layer derivatives will produce a vanishing gradient. Due to the limitation of the gradient descent algorithm, Chang, Zhang, and Chen [8] use Adam gradient optimiser to improve it. But only a brief introduction of Adam algorithm is reported; mathematical process that combine Adam with the parameters updating the neural of the network is not proposed in detail. In terms of data handling, the authors in [9] and [10] divide historical data into train set and test set, which are normalised for the prediction model. Nevertheless, the effect of the anti-normalisation process on the actual forecast is neglected, and the anti-normalisation formula is usually inverted simply according to the original format without specifying the physical meaning of the corresponding parameters.

Manuscript received 5 March, 2022; accepted 19 December, 2022.
This work was supported by the National Natural Science Foundation of China under Grant No. 51167003 and Guangxi Natural Science Foundation under Grant No. 2014GXNSFAA118320.

In view of the above points that need to be improved, this paper proposes an improved multi-layer neural network (IMNN) prediction model based on the BP network framework where the number of hidden layers is extended to a limited number to accelerate the convergence speed of the network and prevent the gradient from vanishing. The rule function is used as the activation function, and Adam optimiser is adopted as the gradient algorithm whose parameter settings are optimised by Particle Swarm Optimisation (PSO). Furthermore, L2 regularization is used to stop overfitting [11]–[14]. In this paper, an explicit mathematical derivation of the IMNN model is derived from the operating principle and optimisation aspects. An adaptive anti-normalisation interval strategy is proposed for the practical forecasting application of neural network. The effectiveness of the mathematical model and the adaptive strategy is proved with the coordination and non-coordination tests.

II. BP NEURAL NETWORK

The BP neural network consists of three layers: input layer, hidden layer, and output layer, whose structure is shown in Fig. 1. It includes two calculation processes: forward propagation and reverse learning [15]. Data samples enter the network from the input layer and reach the hidden layer after the weights and threshold calculation. Then the data are activated by functions in hidden layer and after that will arrive at the output layer through a new round of weights and thresholds calculation. In the end, the predicted values are obtained at the output layer. This process is called "forward propagation". During reverse learning, the weights and thresholds are corrected based on the loss function for the next forward propagation, which keeps the network error within the accuracy range under several iterations of calculation. The calculation process of BP neural network is as follows (take output neuron threshold $c = 0$):

$$U_i = \sum_{j=1}^{n} W_{ij} x_j + b_i, \tag{1}$$

$$s(x) = \frac{1}{1+e^{-x}}, \tag{2}$$

$$T_i = s(U_i), \tag{3}$$

$$y = \sum_{i=1}^{m} V_i T_i. \tag{4}$$

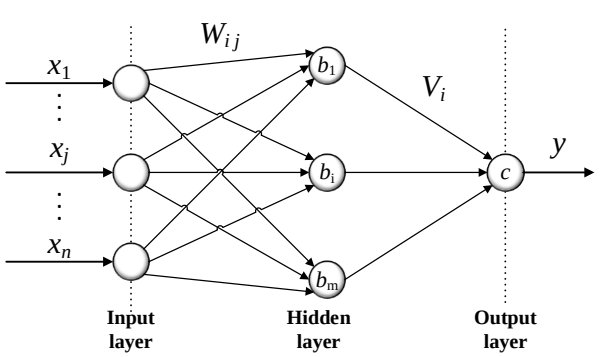

Fig. 1. Structure of the BP neural network.

In (1)–(4), x_j is the value of the sample to the input layer, U_i represents the input signals of the hidden layer, W_{ij} and b_i are the weight threshold between the input layer and the hidden layer. $s(x)$ denotes the activation function sigmoid of the hidden layer [16]. T_i represents the output signals of the hidden layer, y is the result of the output layer, and V_i is the weight between the hidden layers.

Reverse learning regards the loss function as the objective function of gradient descent and achieves the effect that the loss function of the network gradually decreases with the number of iterations by modifying weights and thresholds. p is the number of samples group, $\boldsymbol{x}_p = [x_1^p, x_2^p, \cdots x_j^p, \cdots x_n^p]^T$ is the vector of input data of the network model, and define its loss function as

$$err_p = \frac{1}{2}(t_p - y_p)^2. \tag{5}$$

In (5), t_p represents the value of expectation, in other words, the actual historical data.

Taking the weight W_{ij} as an example, if the negative gradient operation is carried out, the modified part is as in the following

$$\Delta W_{ij}^p = -c_1 \frac{\partial err_p}{\partial W_{ij}^p}. \tag{6}$$

In (6), c_1 is the learning rate, $0<c_1<1$, substitute formula (5) into (6), and $\delta_p = t_p - y_p$

$$\begin{aligned} \Delta W_{ij}^p &= c_1 (t_p - y_p) \frac{\partial y_p}{\partial W_{ij}^p} \\ &= c_1 \delta_p \frac{\partial y_p}{\partial W_{ij}^p} \\ &= c_1 \delta_p \frac{\partial y_p}{\partial T_i^p} \frac{\partial T_i^p}{\partial U_i^p} \frac{\partial U_i^p}{\partial W_{ij}^p}. \end{aligned} \tag{7}$$

Reference equations (1), (3), and (4) and the differential characteristics of the sigmoid function:

$$\begin{cases} \dfrac{\partial U_i^p}{\partial W_{ij}^p} = x_j^p, \\ \dfrac{\partial T_i^p}{\partial U_i^p} = T_i^p (1 - T_i^p), \\ \dfrac{\partial y_p}{\partial T_i^p} = V_i^p. \end{cases} \tag{8}$$

Replace (8) into (7) to get the update formula of weight W_{ij}, the derivation process of b_i and V_i is similar to the weight, and the results are as in the following:

$$\begin{cases} \Delta W_{ij}^{p} = c_1 \delta_p V_i^{p} T_i^{p} (1 - T_i^{p}) x_j^{p}, \\ \Delta b_i^{p} = c_1 \delta_p V_i^{p} T_i^{p} (1 - T_i^{p}), \\ \Delta V_i^{p} = c_1 \delta_p T_i^{p}, \end{cases} \tag{9}$$

$$\begin{cases} W_{ij}^{p+1} = W_{ij}^{p} + \Delta W_{ij}^{p}, \\ b_i^{p+1} = b_i^{p} + \Delta b_i^{p}, \\ V_i^{p+1} = V_i^{p} + \Delta V_i^{p}. \end{cases} \tag{10}$$

III. IMNN PREDICTION MODEL

With the number of hidden layers which are expanded to 3 based on BP structure, the convergence speed of the model can be accelerated, while the problem of vanishing gradient caused by deep layers can be avoided. Its structure is shown in Fig. 2.

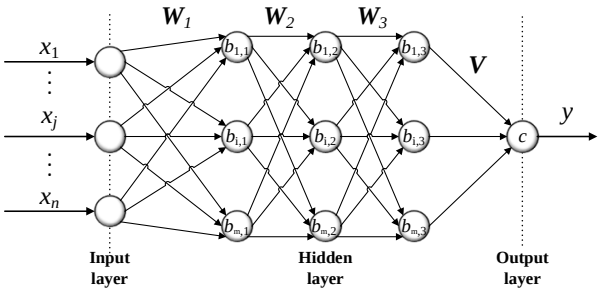

Fig. 2. Structure of IMNN.

The rule function is used as the activation function as follows

$$R(x) = \max(0, x). \tag{11}$$

According to (3),

$$T_i = R(U_i). \tag{12}$$

The matrix expression for the forward calculation of IMNN (Fig. 2) is as follows

$$\begin{cases} \boldsymbol{U}_1 = \boldsymbol{W}_1 \boldsymbol{x}_p + \boldsymbol{b}_1, \\ \boldsymbol{U}_2 = \boldsymbol{W}_2 R(\boldsymbol{U}_1) + \boldsymbol{b}_2, \\ \boldsymbol{U}_3 = \boldsymbol{W}_3 R(\boldsymbol{U}_2) + \boldsymbol{b}_3, \\ \boldsymbol{y} = \boldsymbol{V} R(\boldsymbol{U}_3). \end{cases} \tag{13}$$

In (13), W_1, W_2, and W_3 are the weight matrix, $\boldsymbol{b}_\Omega = [b_{\Omega 1}, b_{\Omega 2}, b_{\Omega 3}]^T$ $(\Omega = 1,2,3)$ is the threshold vector, $\boldsymbol{V} = [V_1, V_2, V_3]$ is the weight vector between the last hidden layer and the output layers, and $\boldsymbol{y}$ is the output vector of the prediction model.

Compared with sigmoid whose calculation is exponential function, the only need of the rule function is to judge whether the input is greater than 0, which has obvious advantages in calculation speed and convergence speed. In the case of avoiding too many negative inputs, the rule function solves the problem of vanishing gradient from the mathematical theory [17], [18], and its differential characteristics are

$$R'(x) = \begin{cases} 1, & x \geq 0, \\ 0, & x < 0. \end{cases} \tag{14}$$

Owing to the fact that the activation function has been replaced, according to (12) and (14), (8) is changed as

$$\begin{cases} \dfrac{\partial U_i^{p}}{\partial W_{ij}^{p}} = x_j^{p}, \\ \dfrac{\partial T_i^{p}}{\partial U_i^{p}} = R_i'^{p}, \\ \dfrac{\partial y_p}{\partial T_i^{p}} = V_i^{p}. \end{cases} \tag{15}$$

Putting (14) in (7), the gradient update formula of the rule function can be obtained

$$\begin{cases} \Delta W_{ij}^{p} = c_1 \delta_p V_i^{p} R_i'^{p} x_j^{p}, \\ \Delta b_i^{p} = c_1 \delta_p V_i^{p} R_i'^{p}, \\ \Delta V_i^{p} = c_1 \delta_p R_i^{p}. \end{cases} \tag{16}$$

Combined (16) with (15), when the input sample number is P, the weighted gradient and threshold can be obtained as

$$\begin{cases} \mathrm{g}_{Wij}^{p} = \nabla err_p = \dfrac{\partial err_p}{\partial W_{ij}^{p}} = -\dfrac{\Delta W_{ij}^{p}}{c_1}, \\ \mathrm{g}_{bi}^{p} = -\dfrac{\Delta b_i^{p}}{c_1}, \\ \mathrm{g}_{vi}^{p} = -\dfrac{\Delta V_i^{p}}{c_1}. \end{cases} \tag{17}$$

The overfitting phenomenon is that the model fits the current data too deeply and it is difficult to make correct judgment on new input data which have not been learnt. Since the model lacks generalisation, L2-norm is used to prevent overfitting of the numerical model, and regularity term is added to the loss function to represent complexity of model

$$\mathrm{err}_p = \frac{1}{2}(t_p - y_p)^2 + \frac{\lambda}{2} \times \sum_{\xi} \|\xi\|^2. \tag{18}$$

Define the loss function for the total sample as

$$\mathrm{err} = \sum_{p=1}^{n} \frac{1}{2}(t_p - y_p)^2 + \frac{\lambda}{2} \times \sum_{\xi^p} \left\|\xi^p\right\|^2. \tag{19}$$

In (18) and (19), λ is the regularisation coefficient, $\boldsymbol{\xi} = [\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3, \boldsymbol{V}]^{\mathrm{T}}$ is the total weight matrix. The change of the loss function causes the gradient equation (17) to become

$$\begin{cases} \mathrm{g}_{Wij}^{p} = \nabla err_p = \dfrac{\partial err_p}{\partial W_{ij}^{p}} + \dfrac{\lambda}{2} \dfrac{\partial \sum_{\xi} \xi^2}{\partial W_{ij}^{p}} = -\dfrac{\Delta W_{ij}^{p}}{c_1} + \lambda W_{ij}^{p}, \\ \mathrm{g}_{bij}^{p} = \nabla err_p = \dfrac{\partial err_p}{\partial W_{ij}^{p}} = -\dfrac{\Delta b_{ij}^{p}}{c_1}, \\ \mathrm{g}_{vij}^{p} = \nabla err_p = \dfrac{\partial err_p}{\partial W_{ij}^{p}} + \dfrac{\lambda}{2} \dfrac{\partial \sum_{\xi} \xi^2}{\partial V_i^{p}} = -\dfrac{\Delta W_{ij}^{p}}{c_1} + \lambda V_i^{p}. \end{cases} \tag{20}$$

Adam optimiser is used as the gradient algorithm, taking the weight W_{ij} as an example (g_{Wij}^{p} is abbreviated as g_{ij}^{p}), the calculation process of Adam optimiser is as in the following:

$$m_{ij}^{p}=\begin{cases} g_{ij}^{p}, & p=1, \\ \beta_1 m_{ij}^{p-1}+(1-\beta_1)g_{ij}^{p}, & p\geq 2, \end{cases} \tag{21}$$

$$I_{ij}^{p}=\begin{cases} 1, & p=1 \\ \beta_2 I_{ij}^{p\text{-}1}+(1-\beta_2),(g_{ij}^{p})^2, & p\geq 2 \end{cases} \tag{22}$$

$$\Delta m_{ij}^{p}=\frac{m_{ij}^{p}}{1-\beta_1^{p}}, \tag{23}$$

$$\Delta I_{ij}^{p}=\frac{I_{ij}^{p}}{1-\beta_2^{p}}, \tag{24}$$

$$W_{ij}^{p+1}=W_{ij}^{p}-c_1\frac{\Delta m_{ij}^{p}}{\sqrt{\Delta I_{ij}^{p}}}. \tag{25}$$

In (21)–(25), m_{ij}^{p} is the first momentum related to the first power of the gradient, I_{ij}^{p} is the second momentum associated with the square of the gradient, and β_1, β_2 are the coefficients. Δm_{ij}^{p} is bias-correct for the first momentum and ΔI_{ij}^{p} is bias-correct for the second momentum. Considering that the first and second momentum coefficients of each sample are consistent, the updating formula of the Adam gradient optimisation algorithm is simplified as follows

$$\begin{aligned} W_{ij}^{p+1}&=W_{ij}^{p}-c_1\frac{m_{ij}^{p}}{1-\beta_1^{p}}\Big/\sqrt{\frac{I_{ij}^{p}}{1-\beta_2^{p}}} \\ &=W_{ij}^{p}-c_1\frac{m_{ij}^{p}}{1-\beta_1}\Big/\sqrt{\frac{I_{ij}^{p}}{1-\beta_2}}. \end{aligned} \tag{26}$$

The Adam gradient algorithm of b_i and V_i deduces the same principle and gives an updating formula directly:

$$b_i^{p+1}=b_i^{p}-c_1\frac{m_i^{p}}{1-\beta_1}\Big/\sqrt{\frac{I_i^{p}}{1-\beta_2}}, \tag{27}$$

$$V_i^{p+1}=V_i^{p}-c_1\frac{m_i^{p}}{1-\beta_1}\Big/\sqrt{\frac{I_i^{p}}{1-\beta_2}}. \tag{28}$$

The IMNN flow chart is shown in Fig. 3.

Particle Swarm Optimisation is an evolutionary algorithm for intelligent population searching, which can optimise network parameters. Taking the total loss function after the IMNN iteration as the PSO fitness function, considerlearning rate c_1, the momentum coefficients β_1 and β_2 as the parameters to be optimised. The flow chart is shown in Fig. 4.

IV. Adaptive Anti-Normalisation Interval Strategy

The symbolic unit of each feature x_j is not uniform due to the different features in the sample data $\boldsymbol{x} = [x_1, \ldots x_j, \ldots x_n]^T$, so it is necessary to adopt normalisation that controls the input data of each feature in the range of [0, 1]. In addition, normalisation can accelerate the convergence of the model with higher accuracy. In this paper, the normalised formula [19] is used as

$$x^{*}=\alpha\frac{x-x_{\min}}{x_{\max}-x_{\min}}+\beta. \tag{29}$$

In (29), α = 0.95, β = 0.05, $x_{\max}$, and $x_{\min}$ are divided into the maximum and minimum values of the input eigenvectors.

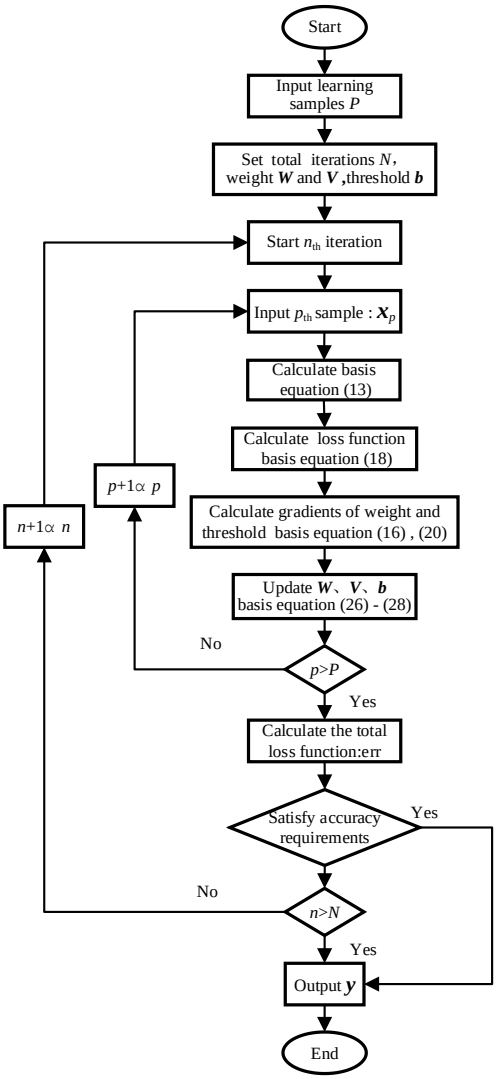

Fig. 3. Flow chart of the IMNN calculation.

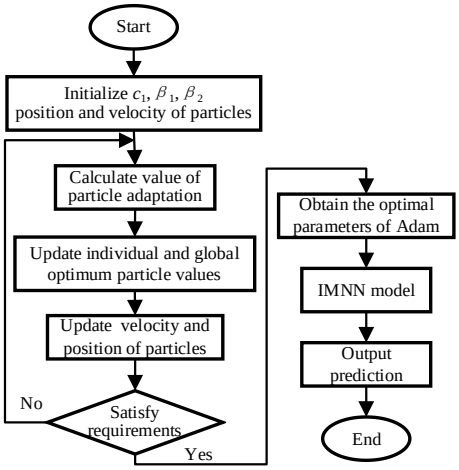

Fig. 4. Flow chart of the optimised Adam parameters based on PSO.

The non-linear description ability of IMNN is realised by learning the sample data that input into the model one by one and modifying the weights and thresholds $\boldsymbol{W}_p$, $\boldsymbol{V}_p$, and $\boldsymbol{b}_p$. Each time the weight threshold is modified on the basis of the samples that have been learnt in the previous iteration, which makes the update of the weight threshold have "memory". The weight threshold learns and remembers the non-linear relationship between the current sample and the learnt sample. Through one iteration of all sample data, the weights and thresholds obtained have the ability of non-linear description of the complete sample, and a round of memory of the data is realised. Iteration of the network is a process of strengthening the memory by the learnt rounds. As the number of network iterations increases, the deeper the "memory degree" of the weights and thresholds, the stronger the non-linear description ability of the model. After several iterations of learning, the sample data obtain the final weights, thresholds $\boldsymbol{W}_{last}$, $\boldsymbol{V}_{last}$, and $\boldsymbol{b}_{last}$ to form a mature IMNN model, which has the ability of prediction.

In this paper, samples are divided into two types, learning sample $\boldsymbol{x}_{\mathrm{I}}$ and samples to be predicted $\boldsymbol{x}_{\mathrm{II}}$. According to (13), for the normalised input data $\boldsymbol{x}_{\mathrm{II}}^*$ of the sample that to be predicted, the mature IMNN calculation procedure is as follows:

$$\begin{cases} \boldsymbol{U}_1 = \boldsymbol{W}_{last1}\boldsymbol{x}_{\mathrm{II}}^* + \boldsymbol{b}_{last1}, \\ \boldsymbol{U}_2 = \boldsymbol{W}_{last2}R(\boldsymbol{U}_1) + \boldsymbol{b}_{last2}, \\ \boldsymbol{U}_3 = \boldsymbol{W}_{last3}R(\boldsymbol{U}_2) + \boldsymbol{b}_{last3}, \\ \boldsymbol{y}_{\mathrm{II}}^* = \boldsymbol{V}_{last}R(\boldsymbol{U}_3). \end{cases} \tag{30}$$

In (30), $\boldsymbol{y}_{\mathrm{II}}^*$ is the normalised value of network output.

Anti-normalisation is the key step to convert the output value of neural network into actual value. The calculation form is obtained from the inverse operation of (29)

$$\boldsymbol{Y} = \frac{1}{\alpha}(\boldsymbol{y}^* - \beta)(Y_{\max} - Y_{\min}) + Y_{\min}. \tag{31}$$

In (31), $Y_{\max}$, $Y_{\min}$ are the maximum and minimum values of the actual output values.

For the learning samples $\boldsymbol{x}_{\mathrm{I}}$ and the samples to be predicted $\boldsymbol{x}_{\mathrm{II}}$, the anti-normalisation formulas are as in following:

$$\boldsymbol{Y}_{\mathrm{I}} = \frac{1}{\alpha}(\boldsymbol{y}_{\mathrm{I}}^* - \beta)(Y_{\mathrm{I}\max} - Y_{\mathrm{I}\min}) + Y_{\mathrm{I}\min}, \tag{32}$$

$$\boldsymbol{Y}_{\mathrm{II}} = \frac{1}{\alpha}(\boldsymbol{y}_{\mathrm{II}}^* - \beta)(Y_{\mathrm{II}\max} - Y_{\mathrm{II}\min}) + Y_{\mathrm{II}\min}. \tag{33}$$

In (32), $Y_{\mathrm{I}\max} = t_{\max}$, $Y_{\mathrm{I}\min} = t_{\min}$. However, in (33), the values of $Y_{\mathrm{II}\max}$ and $Y_{\mathrm{II}\min}$ are undetermined. Because the actual real value of $\boldsymbol{Y}_{\mathrm{II}}$ is unknown, in other words, $\boldsymbol{Y}_{\mathrm{II}}$ is the predictive output of samples to be predicted $\boldsymbol{x}_{\mathrm{II}}$. Since it is significant to find the substitutes for $Y_{\mathrm{II}\max}$ and $Y_{\mathrm{II}\min}$, the physical process of learning and forecasting is shown in Fig. 5.

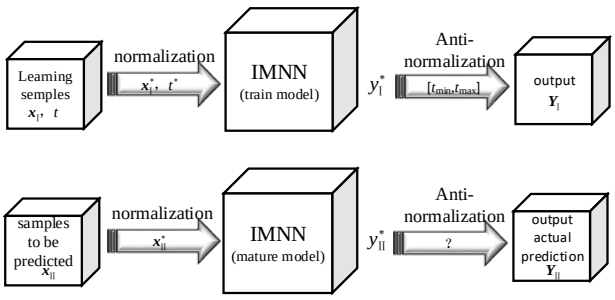

Fig. 5. Processes of data normalisation.

The IMNN train model is to realise the non-linear capability of the network output through continuous iterative learning of the neural network under the condition of the known input $\boldsymbol{x}_{\mathrm{I}}$ and expected output value $\boldsymbol{t}$. The predicted output $\boldsymbol{Y}_{\mathrm{I}}$ is obtained after anti-normalisation of the expected values $t_{\max}$ and $t_{\min}$, which approximately fits the actual expected value $\boldsymbol{t}$.

The known input of the IMNN mature model is only $\boldsymbol{x}_{\mathrm{II}}$, while the expected output value is unknown. According to (33), only the values of $Y_{\mathrm{II}\max}$ and $Y_{\mathrm{II}\min}$ determine the actual prediction $\boldsymbol{Y}_{\mathrm{II}}$ output. In reverse thinking, if the maximum and minimum values of $\boldsymbol{Y}_{\mathrm{II}}$ can be approximated, find a set of $N_{\max}$ and $N_{\min}$ that can be used to approximately replace $Y_{\mathrm{II}\max}$ and $Y_{\mathrm{II}\min}$ ($Y_{\mathrm{II}\max}$, $Y_{\mathrm{II}\min} = N_{\min}$), the predicted actual value $\boldsymbol{Y}_{\mathrm{II}}$ can be obtained. The closer the substitute approximate values are, the closer the forecast results are to the true level.

The values of $N_{\max}$ and $N_{\min}$ have the following schemes:

– Case 1: Direct adoption of learning samples with maximum and minimum expectations, $N_{\max} = t_{\max}$, $N_{\min} = t_{\min}$;

– Case 2: An adaptive anti-normalisation interval strategy is proposed according to the characteristics of hydro-turbine test data.

The flow of adaptive anti-normalisation interval strategy is shown in Fig. 6; note that the corresponding expression of the converted power of the hydro-turbine coordination test is as $N = f(\varphi, a)$. Take the k^{th} sample of input data ($x_{\mathrm{II}}^k = [\varphi_{\mathrm{II}}^k, a_{\mathrm{II}}^k]$) for example:

Step 1: Judge whether φ_{II}^k belongs to the learning sample vector $\boldsymbol{\varphi}_{\mathrm{I}}$. If yes, turn to Step 2; if not, move to Step 4.

Step 2: Find a group of a_{I}^{k-}, a_{I}^{k+} that belongs to the learning sample vector $\boldsymbol{a}_{\mathrm{I}}^k$ and satisfy

$$a_{\mathrm{I}}^{k-} < a_{\mathrm{I}}^k < a_{\mathrm{I}}^{k+}. \tag{34}$$

In (34), a_{I}^{k-} is the maximum in an array of values smaller than a_{I}^k in the learning sample, a_{I}^{k+} is the minimum in an array of values greater than a_{I}^k in the learning sample. Move to Step 3.

Step 3: Find the corresponding power value as follows:

$$\begin{cases} N_{\max}^{k} = f(\varphi_{\mathrm{I}}^{k}, a_{\mathrm{I}}^{k+}), \\ N_{\min}^{k} = f(\varphi_{\mathrm{I}}^{k}, a_{\mathrm{I}}^{k-}). \end{cases} \tag{35}$$

Step 4: Find a group of $\varphi_{\mathrm{I}}^{k-}$, $\varphi_{\mathrm{I}}^{k+}$ that belongs to the learning sample vector $\boldsymbol{\varphi}_{\mathrm{I}}^{k}$ and satisfy

$$\varphi_{\mathrm{I}}^{k-} < \varphi_{\mathrm{I}}^{k} < \varphi_{\mathrm{I}}^{k+}. \tag{36}$$

In (36), $\varphi_{\mathrm{I}}^{k-}$ is the maximum in an array of values smaller than φ_{I}^{k} in the learning sample and $\varphi_{\mathrm{I}}^{k+}$ is the minimum in an array of values greater than φ_{I}^{k} in the learning sample. Move to Step 5.

Step 5: Judge whether a_{II}^{k} belongs to the learning sample vector $\boldsymbol{a}_{\mathrm{I}}$. If yes, move to Step 6; if not, turn to Step 7.

Step 6: Find the corresponding power value as follows

$$\begin{cases} N_{\max}^{k} = f(\varphi_{\mathrm{I}}^{k+}, a_{\mathrm{I}}^{k}), \\ N_{\min}^{k} = f(\varphi_{\mathrm{I}}^{k-}, a_{\mathrm{I}}^{k}). \end{cases} \tag{37}$$

Step 7: Find the corresponding power value as follows:

$$\begin{cases} N_{\max}^{k} = [f(\varphi_{\mathrm{I}}^{k+}, a_{\mathrm{I}}^{k-}) + f(\varphi_{\mathrm{I}}^{k+}, a_{\mathrm{I}}^{k+})]/2, \\ N_{\min}^{k} = [f(\varphi_{\mathrm{I}}^{k-}, a_{\mathrm{I}}^{k+}) + f(\varphi_{\mathrm{I}}^{k-}, a_{\mathrm{I}}^{k-})]/2. \end{cases} \tag{38}$$

To sum up, there exist three relationships between samples to be predicted ($\varphi_{\mathrm{II}}^{k}$, a_{II}^{k}) and learning samples (φ_{I}^{k}, a_{I}^{k}), and the corresponding $N_{\max}^{k}$ and $N_{\min}^{k}$ formulas are:

1. $\varphi_{\mathrm{II}}^{k} \in \boldsymbol{\varphi}_{\mathrm{I}}$, $a_{\mathrm{II}}^{k} \notin \boldsymbol{a}_{\mathrm{I}}$ corresponding equation (35);
2. $\varphi_{\mathrm{II}}^{k} \notin \boldsymbol{\varphi}_{\mathrm{I}}$, $a_{\mathrm{II}}^{k} \in \boldsymbol{a}_{\mathrm{I}}$ corresponding equation (37);
3. $\varphi_{\mathrm{II}}^{k} \notin \boldsymbol{\varphi}_{\mathrm{I}}$, $a_{\mathrm{II}}^{k} \notin \boldsymbol{a}_{\mathrm{I}}$ corresponding equation (38).

Take the third as an example to demonstrate the physical flow of the adaptive interval, as shown in Fig .7.

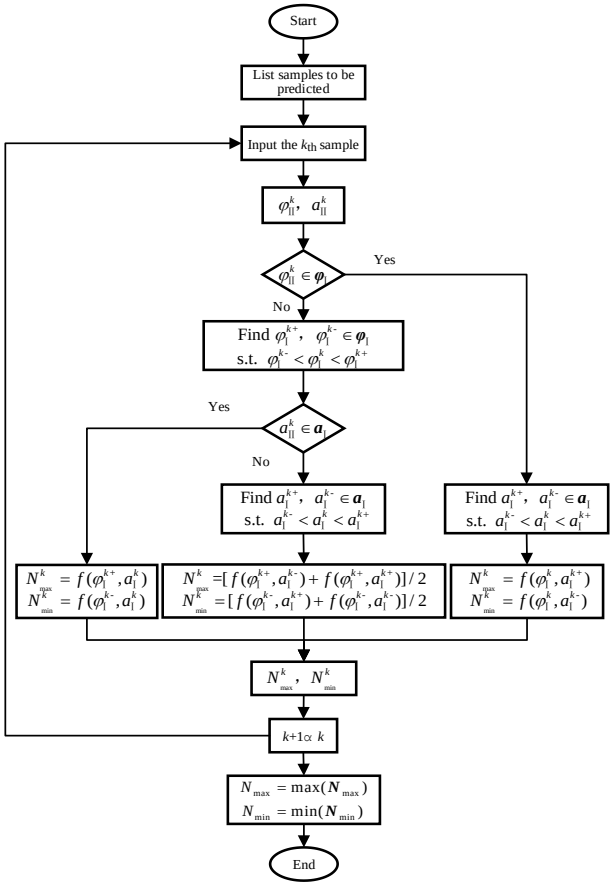

Fig. 6. Flow chart of the adaptive interval strategy.

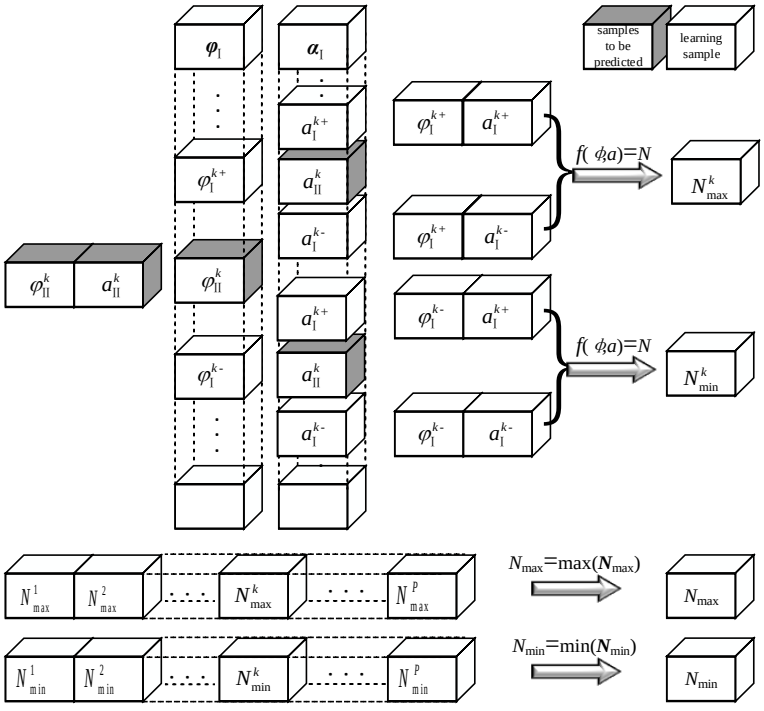

Fig. 7. Physical process of adaptive interval.

V. GRADIENT OPTIMISATION ALGORITHM SIMULATION

To verify the effectiveness of the Adam optimiser proposed in this paper, it is compared with the Stochastic gradient decent (SGD), SGD with momentum (SGDM), Adagrad gradient algorithm [20], [21], and BP network under the condition of unchanged IMNN network parameters. Figure 8 shows the convergence curve of the loss function after 300 iterations of the network.

The parameters of each gradient algorithm are shown in Table I, where the learning rate of Adam gradient algorithm is much lower than that of SGD, SGDM and BP neural network (6 times of SGD and 4.5 times of BP network).

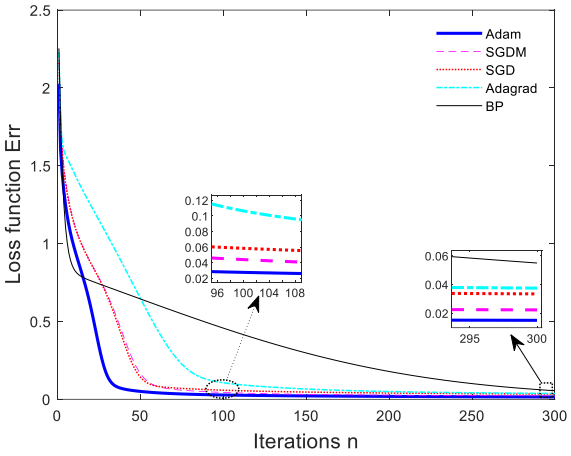

Fig. 8. Convergence rates of IMNN from different gradient algorithms.

Adam gradient algorithm can converge faster and keep the error accuracy minimum in the last stage of iteration. Under the condition of the same learning rate, the loss function of Adagrad gradient algorithm is close to Adam after IMNN iterates about 104 times, and the convergence accuracy is not as good as Adam. Adam has obvious advantages over Adagrad in convergence speed.

PSO is used to optimise the gradient parameters of Adam, set the number of particles to 20, and iterate 50 times. The effect is shown in Fig. 9, where the convergence speed of the optimised Adam gradient is significantly higher than the unoptimised one. The parameter settings after optimisation are shown in Table II.

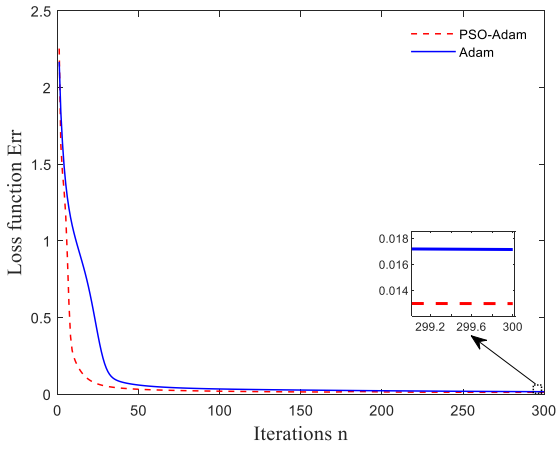

Fig. 9. Comparison between Adam & optimised parameters based on PSO.

TABLE I. PARAMETERS OF DIFFERENT GRADIENT ALGORITHM.

Gradient algorithm	Learning rate c_1	Momentum coefficients
Adam	0.002	$\beta_1 = 0.11$ $\beta_2 = 0.99$
SGDM	0.012	$\beta = 0.1$
SGD	0.012	
Adagrad	0.002	
BP	0.009	

TABLE II. DATA OF OPTIMISED PARAMETERS BASED ON PSO.

Algorithm \ Parameter	c_1	β_1	β_2
Adam	0.0020	0.110	0.990
PSO-Adam	0.0039	0.037	0.998

VI. SIMULATION PREDICTION

Example 1: Taking the non-coordination test data from a hydropower station in Guangxi, China, as an example, the hydro-turbine head is 9.3 m ($H = 9.3\ m$). Due to the huge mechanical inertia of the hydraulic turbine, it is difficult to flexibly adjust the blade angle φ in a wide range. The value of the blade angle φ in the non-coordination test site changes gradually along the same direction. First, fix the angle of the blade φ and adjust the value of guide vane opening a; then change φ and adjust a. Therefore, the historical data are randomly arranged to improve the randomness of the learning samples and ensure the generalisation of the model. Randomised historical data are divided into learning samples and samples to be predicted. As shown in Table III, unlike learning samples that have the expected power value, the samples to be predicted only contain the input $\boldsymbol{\varphi}$ and $\boldsymbol{a}$, which satisfy the actual forecast conditions.

TABLE III. DATA OF LEARNING & PREDICTION WITH NON-COORDINATION CONDITION.

Learning samples				Samples to be predicted		
No.	φ_I	a_I	P_I	No.	φ_{II}	a_{II}
1	85.2	79.0	28269.94	26	85.2	86.0
2	13.0	42.8	8382.45	27	65.0	74.8
3	53.3	78	20699.97	28	26	50
⋮	⋮	⋮	⋮	⋮	⋮	⋮
24	65.0	83	23791.20	33	26	62
25	26.0	45.8	11245.66	34	53.3	74.0

First, the learning samples are imported into the IMNN model, and the mature prediction model is obtained after the iterative training is completed. The samples to be predicted are then entered into the IMNN mature model to obtain the prediction. As shown in Fig. 10, the blue line is the actual value and the red dotted line is the training value of the learning samples, and almost fit of two lines reflects that IMNN already possess mature predictive capabilities. After learning samples, it is time to turn to the samples to be predicted. The pink line represents the prediction values that adopt the Case 1 anti-normalisation strategy, and the cyan line represents the prediction values that use the Case 2 adaptive anti-normalisation interval strategy. When the model prediction is finished, the field test is carried out to obtain the black line. When comparing two forecast lines with black lines, the results show that the predicted values which utilise the adaptive anti-normalisation interval strategy

are closer to the actual value of the field test.

Finally, to reflect the advantages of the proposed strategy, as shown in Fig. 11, the green dotted line is drawn, which applies Case 3 anti-normalisation strategy. The data are shown in Table IV.

– Case 3: Because the actual value of field test has been obtained (black line), that means that the vector $\boldsymbol{Y}_{\mathrm{II}}$ is known, so the values of $Y_{\mathrm{II}\max}$ and $Y_{\mathrm{II}\min}$ are available. Put $N_{\max}=Y_{\mathrm{II}\max}$, $N_{\min}=Y_{\mathrm{II}\min}$ as an anti-normalisation strategy.

The predicted value of IMNN through the Case 3 strategy approximates the field test actual value, which is based on the result of anti-normalisation of the test true value interval. Ignoring the systematic error of the model itself, in theory, the best forecast effect is the anti-normalisation realised by the actual value interval. The gap between the predicted value and the actual value of the field test is measured by relative error, root mean square error (RMSE), and mean absolute error (MAE) indicators, as shown in Table V.

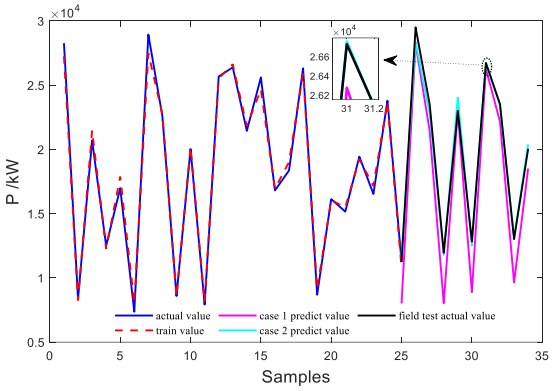

Fig. 10. Predicting effect of IMNN with non-coordination condition.

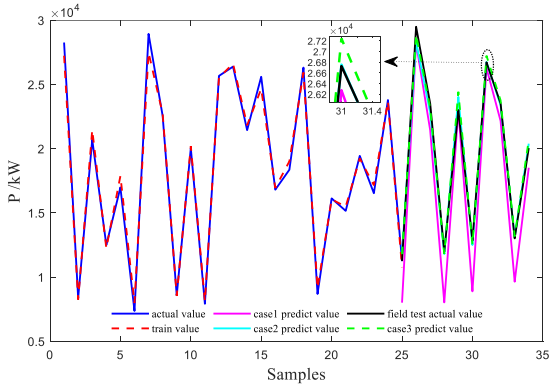

Fig. 11. Diagram of prediction at work with non-coordination condition.

TABLE IV. DATA OF PREDICTION & EXPERIMENT WITH NON-COORDINATION CONDITION.

No.	Case 1 predict value	Case 2 predict value	Field test actual value	Case 3 predict value
26	28344.03	28212.86	29497.40	28880.58
27	22789.34	21439.86	23515.07	23151.55
28	11792.56	8031.26	11961.48	11809.72
⋮	⋮	⋮	⋮	⋮
33	13114.07	9642.61	13016.12	13172.70
34	20392.87	18517.83	20048.96	20679.91

TABLE V. EVALUATIVE INDICATORS OF PREDICTIVE VALUE WITH NON-COORDINATION CONDITION.

Indicator / Strategy	Relative error	RMS	MAE
Case 1	10.88 %	2514.56	2124.82
Case 2	2.33 %	589.07	455.03
Case 3	2.53 %	607.14	493.47

Example 2: Taking as an example the coordination test data of a hydropower station in Guangxi, China, the head of the hydro-turbine is 6.0 m (H = 6.0 m). The data for the learning samples and the samples to be predicted are shown in Table VI.

TABLE VI. DATA OF LEARNING & PREDICTION AT WORK WITH COORDINATION CONDITION.

Learning samples				Samples to be predicted		
No.	φ_{I}	a_{I}	P_{I}	No.	φ_{II}	a_{II}
1	77.0	88.4	14368.31	12	26.8	61.6
2	12.4	49.6	4205.19	13	36.3	69.3
⋮	⋮	⋮	⋮	14	23.8	60.0
10	80.5	89.5	15382.32	15	68.6	85.1
11	55.3	79.7	11770.61	16	6.62	44.2

Since the blade angle φ and guide vane opening a need to be changed simultaneously in the coordination test, the adjustment time of a single sample is much longer than the non-coordination test, so the data obtained by the coordination test are much less than that of the non-coordination test in limited test time. The results of the IMNN forecast are shown in Table VII and Fig. 12.

TABLE VII. DATA OF PREDICTION & EXPERIMENT WITH COORDINATION CONDITION.

No.	Case 1 predict value	Case 2 predict value	Field test actual value	Case 3 predict value
12	6412.27	7410.68	6878.26	6813.08
13	8314.66	9787.38	8768.37	8516.95
14	5910.57	6783.89	6467.69	6363.74
15	13546.35	16323.44	13131.04	13202.67
16	3169.60	3359.53	3226.52	3908.88

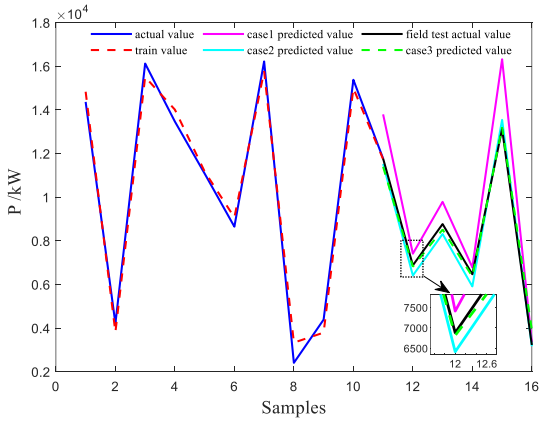

Fig. 12. Predicting the effect of IMNN with the coordination condition.

The case3 line is closest to the actual value of the field test (black line), and the effect of case2 line is slightly lower than that of case3 line. Obviously, this result is due to the fact that the anti-normalisation interval is the actual true value and that the adaptive anti-normalisation value can approach the field

test value as close as possible to ensure the accuracy of the prediction. Taking into account the actual prediction, it is impossible to directly obtain the true value of the experiment and the case3 line. Therefore, the adaptive anti-normalisation interval strategy possesses significance for practical forecasting application.

The results of evaluation indices are shown in Table VIII, under the small sample learning data, predictive values based on anti-normalisation interval strategy can still approximate the field actual test values with low error accuracy. It further proves the effectiveness of the proposed IMNN prediction model and adaptive anti-normalisation strategy.

TABLE VIII. EVALUATIVE INDICATORS OF PREDICTIVE VALUE WITH COORDINATION CONDITION.

Indicator / Strategy	Relative error	RMS	MAE
Case 1	14.37 %	1619.46	1203.19
Case 2	4.37 %	402.07	365.96
Case 3	3.09 %	339.81	258.97

VII. CONCLUSIONS

To solve the problem that the hydro-turbine coordination test data are not enough to guide the parameter setting of units comprehensively, this paper starts from the perspective of building a mathematical prediction model. For BP neural network prediction model, there exist problems such as slow convergence rate of gradient descent method, vanishing gradient probably caused by sigmoid activation function in multi-layer network structure, and data overfitting, etc. An improved multi-layer neural network (IMNN) prediction model is proposed and the optimisation process is derived mathematically. It explains the role of non-linear fitting and prediction significance of the weight threshold to the sample data in the iterative process and gives out the predictive calculation formula of mature model after training. It is pointed out that the problem of practical application of neural networks is that the interval size of the actual value to be predicted cannot be obtained. Based on this, an adaptive anti-normalisation interval strategy is put forward; besides, the accuracy and effectiveness of the proposed prediction model are verified by case study of turbine coordination and non-coordination operating conditions.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] Y. Tian, B. Wang, D. Zhu, and F. Wu, "Takagi–Sugeno fuzzy generalised predictive control of a time-delay non-linear hydro-turbine governing system", *IET Renewable Power Generation*, vol. 13, no. 13, pp. 2338–2345, 2019. DOI: 10.1049/iet-rpg.2019.0329.

[2] Y. Wu, S. Liu, H.-S. Dou, S. Wu, and T. Chen, "Numerical prediction and similarity study of pressure fluctuation in a prototype Kaplan turbine and the model turbine", *Computers & Fluids*, vol. 56, pp. 128–142, 2012. DOI: 10.1016/j.compfluid.2011.12.005.

[3] O. H. Souza, N. Barbieri, and A. H. M. Santos, "Study of hydraulic transients in hydropower plants through simulation of nonlinear model of penstock and hydraulic turbine model", *IEEE Transactions on Power Systems*, vol. 14, no. 4, pp. 1269–1272, 1999. DOI: 10.1109/59.801883.

[4] R. Xiao, Z. Wang, and Y. Luo, "Dynamic stresses in a francis turbine runner based on fluid-structure interaction analysis", *Tsinghua Science and Technology*, vol. 13, no. 5, pp. 587–592, 2008. DOI: 10.1016/S1007-0214(08)70096-8.

[5] P. Guo, H. Zhang, and D. Gou, "Dynamic characteristics of a hydro-turbine governing system considering draft tube pressure pulsation", *IET Renewable Power Generation*, vol. 14, no. 7, pp. 1210–1218, 2020. DOI: 10.1049/iet-rpg.2019.1061.

[6] Z.-h. Guo, J. Wu, H.-y. Lu, and J.-z. Wang, "A case study on a hybrid wind speed forecasting method using BP neural network", *Knowledge-Based Systems*, vol. 24, no. 7, pp. 1048–1056, 2011. DOI: 10.1016/j.knosys.2011.04.019.

[7] G. Montes-Atenas *et al.*, "Predicting bubble size and bubble rate data in water and in froth flotation-like slurry from computational fluid dynamics (CFD) by applying deep neural networks (DNN)", *International Communications in Heat and Mass Transfer*, vol. 76, pp. 197–201, 2016. DOI: 10.1016/j.icheatmasstransfer.2016.05.031.

[8] Z. Chang, Y. Zhang, and W. Chen, "Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform", *Energy*, vol. 187, art. no. 115804, 2019. DOI: 10.1016/j.energy.2019.07.134.

[9] J. Wang, S. Li, Z. An, X. Jiang, W. Qian, and S. Ji, "Batch-normalized deep neural networks for achieving fast intelligent fault diagnosis of machines", *Neurocomputing*, vol. 329, pp. 53–65, 2019. DOI: 10.1016/j.neucom.2018.10.049.

[10] Z. Shi, H. Liang, and V. Dinavahi, "Direct interval forecast of uncertain wind power based on recurrent neural networks", *IEEE Transactions on Sustainable Energy*, vol. 9, no. 3, pp. 1177–1187, 2018. DOI: 10.1109/TSTE.2017.2774195.

[11] S. Grubisic, W. P. Carpes, J. P. A. Bastos, and G. Santos, "Association of a PSO optimizer with a Quasi-3D ray-tracing propagation model for mono and multi-criterion antenna positioning in indoor environments", *IEEE Transactions on Magnetics*, vol. 49, no. 5, pp. 1645–1648, 2013. DOI: 10.1109/TMAG.2013.2243831.

[12] H. Bendu, B. B. V. L. Deepak, and S. Murugan, "Multi-objective optimization of ethanol fuelled HCCI engine performance using hybrid GRNN–PSO", *Applied Energy*, vol. 187, pp. 601–611, 2017. DOI: 10.1016/j.apenergy.2016.11.072.

[13] Z. Zhang, F. Li, M. Zhao, L. Zhang, and S. Yan, "Robust neighborhood preserving projection by nuclear/L2,1-norm regularization for image feature extraction", *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1607–1622, 2017. DOI: 10.1109/TIP.2017.2654163.

[14] S. S. Liew, M. Khalil-Hani, R. Bakhteri, "Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems", *Neurocomputing*, vol. 216, pp. 718–734, 2016. DOI: 10.1016/j.neucom.2016.08.037.

[15] L. Zhang and P. B. Luh, "Neural network-based market clearing price prediction and confidence interval estimation with an improved extended Kalman filter method", *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 59–66, 2005. DOI: 10.1109/TPWRS.2004.840416.

[16] Y. Qin, X. Wang, and J. Zou, "The optimized deep belief networks with improved logistic igmoid units and their application in fault diagnosis for planetary gearboxes of wind turbines", *IEEE Transactions on Industrial Electronics*, vol. 66, no. 5, pp. 3814–3824, 2019. DOI: 10.1109/TIE.2018.2856205.

[17] X. Wang, Y. Qin, Y. Wang, S. Xiang, and H. Chen, "ReLTanh: An activation function with vanishing gradient resistance for SAE-based DNNs and its application to rotating machinery fault diagnosis", *Neurocomputing*, vol. 363, pp. 88–98, 2019. DOI: 10.1016/j.neucom.2019.07.017.

[18] H. Guo, S. Li, B. Li, Y. Ma, and X. Ren, "A new learning automata-based pruning method to train deep neural networks", *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3263–3269, 2018. DOI: 10.1109/JIOT.2017.2711426.

[19] F. J. Martínez López, S. Martínez Puertas, and J. A. Torres Arriaza, "Training of support vector machine with the use of multivariate normalization", *Applied Soft Computing*, vol. 24, pp. 1105–1111, 2014. DOI: 10.1016/j.asoc.2014.08.020.

[20] S. R. Dubey, S. Chakraborty, S. K. Roy, S. Mukherjee, S. K. Singh, and B. B. Chaudhuri, "diffGrad: An optimization method for convolutional neural networks", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4500–4511, 2020. DOI: 10.1109/TNNLS.2019.2955777.

[21] C. Yan and Q. Zhang, "Merging visual features and temporal dynamics in sequential recommendation", *Neurocomputing*, vol. 362, pp. 11–18, 2019. DOI: 10.1016/j.neucom.2019.07.015.