IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ARTIFICIAL INTELLIGENCE TECHNOLOGIES FOR ELECTRIC POWER SYSTEMS

Received February 28, 2020, accepted March 23, 2020, date of publication April 3, 2020, date of current version April 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2985381

Wind Power Prediction of Kernel Extreme Learning Machine Based on Differential Evolution Algorithm and Cross Validation Algorithm

NING LI[1,2], (Member, IEEE), FUXING HE[1], (Student Member, IEEE), WENTAO MA[1], (Member, IEEE), RUOTONG WANG[3], AND XIAOPING ZHANG[4], (Fellow, IEEE)

[1]School of Electrical Engineering, Xi'an University of Technology, Xi'an 710048, China
[2]Key Laboratory of Control of Power Transmission and Conversion (SJTU), Ministry of Education, Shanghai 200240, China
[3]Department of Electrical Engineering and Electronics, University of Liverpool, Liverpool L69 3GJ, U.K.
[4]Department of Electronic, Electrical, and Systems Engineering, School of Engineering, University of Birmingham, Birmingham B15 2TT, U.K.

Corresponding authors: Ning Li (lining83@xaut.edu.cn) and Wentao Ma (mawt@xaut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61976175 and Grant 51507140, in part by the Key Project of Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2019JZ-05, in part by the China Scholarship Council (CSC) State Scholarship Fund International Clean Energy Talent Project under Grant [2018]5046,[2019]157, in part by the Key Laboratory of Control of Power Transmission and Conversion (SJTU), Ministry of Education, under Grant 2018AC02, in part by the State Key Laboratory of Electrical Insulation and Power Equipment (EIPE17209), in part by the Natural Science Basic Research Plan of Shaanxi province under Grant 2018JM5041, in part by the Science and Technology Plan Project of Xi'an under Grant GXYD14.23 and Grant GXYD14.10, and in part by the Xi'an Beilin District Science and Technology Project (GX1918).

ABSTRACT As fossil fuel is being depleted, the percentage of wind power capacity in total electricity generation is increasing. In order to improve the absorption capacity of wind power, wind power prediction has been introduced. Aiming at the disadvantage of low prediction accuracy and unstable model of traditional extreme learning machine (ELM), a kernel extreme learning machine based on differential evolution (DE) and cross validation optimization method is proposed to predict short-term wind power generation. Firstly, the average mean square error (MSE) verified by k folding and cross validation is adopted as the error function of the model to improve the stability and generalization performance of the model. Secondly, differential evolution algorithm is used to optimize the regularization coefficient and kernel width of the kernel extreme learning machine with cross validation and improve the precision of model is 8.34%. Finally, compared with the application of extreme learning machine with genetic algorithm and cross validation to a wind farm prediction case in northwest China, the experimental results show that the convergence rate of this method is twice that of genetic algorithm (GA) optimization algorithm, and the accuracy is higher.

INDEX TERMS Differential evolutionary algorithm, Kernel extreme learning machine, k fold cross validation, wind power prediction.

I. INTRODUCTION

At present, the basic trend of global energy transformation is to realize the transition from fossil energy system to low-carbon energy system, and the ultimate goal is to enter the era of sustainable energy dominated by renewable energy. As early as the end of the 19th century, Denmark began to make use of wind power. Until 1973, when the world oil crisis occurred, the concern about the oil shortage and the environmental pollution caused by the use of fossil fuels brought wind power back to the attention. Since then, the United States, Denmark, the Netherlands, the United Kingdom, Germany, Sweden, Canada and other countries have invested a lot of manpower and money in the research and application of wind power. Nowadays, wind power could provide 20% of Denmark's electricity consumption. Wind power brings more uncertainty than conventional power generation to grid. Accurate and reliable wind power prediction is very important to optimize the operating cost of wind power and improve the reliability of wind power system [1].

The associate editor coordinating the review of this manuscript and approving it for publication was Canbing Li.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

To solve the wind power prediction problem, an increasing number of prediction methods have been proposed by researchers. The traditional methods include continuous prediction method [2], kalman filter method [3], [4], and time series method [5], [6]. While the support vector machine and neural network method as the modern intelligent prediction methods have been received extensive attention in the last few years. [7]. In recent years, a novel single-layer neural network model, called extreme learning machine, has been proposed by professor Huang [8], [9]. It only needs to set the number of hidden nodes in the network, does not need to set the input weight matrix of the network, and does not need to set the deviation of hidden elements. Therefore, the method has fast learning speed and good generalization performance [10].

At present, there are many methods to optimize ELM algorithm, which have been widely used in various fields, such as wind power prediction. It can be divided into different categories, such as data preprocessing, error function optimization and parameter optimization. Data preprocessing can filter data, eliminate redundant data, simplify algorithm complexity and improve accuracy [11]–[16]. Error function optimization can be divided into two types, namely, regularization and kernel function. Regularization is to prevent overfitting, while kernel function is constructed on the basis of regularization [17]–[24]. To achieve parameter optimization, a combination of methods is adopted, including genetic algorithm, particle swarm optimization, firefly algorithm [25]–[30].

Kernel extreme learning machine (KELM), as an error function optimization method, has been rarely studied in applications of wind power prediction. On the one hand, as wind power data are nonlinear, and the kernel method performs good performance in the processing of nonlinear data, the KELM shows better generalization performance and accuracy than the traditional ELM. Heterogeneous Ensemble of ELMs (HE2LM) proposed by Literature [24] for classification has different ELM algorithms including the Regularized ELM (RELM), the KELM, and the L2-norm-optimized ELM (ELML2). The method has high prediction accuracy, good generalization ability and low sensitivity to outliers. On the other hand, Literature [31] adopts the method of leaving one out cross validation to improve the stability and generalization performance of the model, which cannot be realized in the case of processing wind power big data. Therefore, this paper proposes the method of K folding and cross validation for wind power prediction. Literature [26] proposes an approximate search algorithm, named weighted ELM with differential evolution, to improve network accuracy. In KELM, the parameters (including regularization coefficient and kernel width) selection is a crucial problem for improving the performance of the KELM. Therefore, this method is applied to the case of wind power prediction to improve the generalization and stability of wind power prediction.

The contribution of this paper is as follows. A combined prediction algorithm of kernel extreme learning machine based on differential evolution algorithm and cross validation algorithm (DECVKELM) is proposed and applied to wind power prediction.

1) Firstly, KELM is used to predict the wind power. Compared with various classification prediction methods, KELM is more effective in nonlinear wind power prediction data.

2) Secondly, the average MSE value of cross-validation is used as the loss function of the model to improve the stability and generalization performance of the model.

3) Finally, the kernel width and regularization coefficient of the model optimized by DE algorithm are used for parameter optimization to improve the prediction accuracy of wind power, and the accuracy of the proposed model is verified by comparing and analyzing the GA optimization algorithm.

The remainder of this paper is organized as follows. Section II introduces ELM algorithm and KELM algorithm. Section III exploits the novel KELM with cross validation and DE methodology to optimize the KELM objective function and the free parameters for improving the stability of the model and the accuracy of model prediction. The experimental results in real wind power prediction scenarios are given in Section IV to show the advantages of the proposed method. Finally, Section V summarizes the full text.

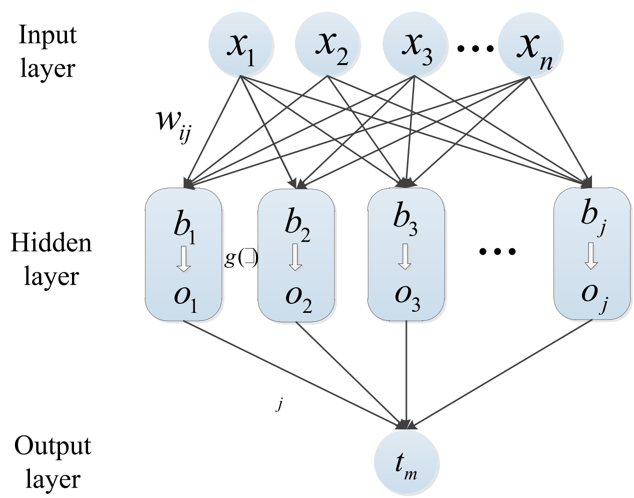


FIGURE 1. ELM structure diagram.

II. REVIEW OF THE ELM AND KELM

A. THE ELM MODEL

Extreme learning machine is a feed forward neuron network (FNN) [8], [9] as shown in **Fig.1** below. Its neural Network structure consists of three layers, including input layer, hidden layer and output layer. The basic principle is: randomly generate weight w_{ij} and threshold b_j of input layer and hidden layer, randomly give weight β_j of hidden layer and output layer, then the expected output t_j can be expressed as the following.

$$\sum_{i=1}^{l} \beta_i g(W_i \cdot X_j + b_i) = t_j \tag{1}$$

where $g(\cdot)$ is the activation function of sigmoid, which makes equation (1) nonlinear and reduces the error.

The (1) can be expressed in matrix form as follows.

$$H \cdot \beta = T \tag{2}$$

where H denotes the output of the hidden layer node; β is the output weight, and T stands for the expected output. Here is an example of an H neural network.

$$\begin{aligned} &H(a_1, \ldots, a_L; b_1, \ldots, b_L; x_1, \ldots x_L) \\ &= \begin{bmatrix} G(a_1, b_1, x_1) & \cdots & G(a_L, b_L, x_1) \\ \vdots & \ddots & \vdots \\ G(a_1, b_1, x_N) & \cdots & G(a_L, b_L, x_N) \end{bmatrix}_{N \times L} \end{aligned} \tag{3}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times d} \qquad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_1^T \end{bmatrix}_{N \times m} \tag{4}$$

ELM is the least squares error function; its objective function is as follows.

$$\begin{aligned} E &= \sum_{j=1}^{m} (t_j - y_j)^2 \\ &= \sum_{j=1}^{m} \left(\sum_{i=1}^{l} \beta_i g(W_i \cdot X_j + b_i) - y_j \right)^2 \end{aligned} \tag{5}$$

where y_j is the real output value. In order to reduce the error to 0, the ultimate learning machine updates the weight of hidden layer and output layer, as shown in equation (6).

$$\hat{\beta} = H^+ \cdot Y \tag{6}$$

where $\hat{\beta}$ denotes the updated weight, and H^+ is the generalized inverse matrix of H.

In summary, the extreme learning machine algorithm described above, extreme learning machine is a very simple machine learning methods. There is no need for the process of network parameters in iteration steps, which greatly reduces the network parameters adjustment time. Therefore, the optimization of its parameters optimization and algorithm has become a research hotspot.

B. STRUCTURE OF THE KELM MODEL

The kernel extreme learning machine (KELM) is constructed on the basis of regularization in order to improve the stability and generalization ability of the network. For linear models, regularization is usually achieved by weights of constraint models. General regularization methods mainly include Ridge Regress, Lasso Regression and Elastic Net. Especially Ridge Regression is used for KELM.

Ridge Regress is the regularized version of linear regression, which adds a regular term equal to $\alpha \sum_{i=1}^{n} \theta_i^2$ to the cost function. This makes the learning algorithm not only need to fit data, but also keep the model weight to a minimum. Equation (7) gives the cost function of Ridge Regress model.

$$J(\theta) = MSE(\theta) + \alpha \frac{1}{2} \sum_{I=1}^{n} \theta_i^2 \tag{7}$$

where α is called hyperparameter, which is the degree of regularization of the model. If $\alpha = 0$, the Ridge Regress is a linear model, and if α is very large, then all the weights are close to zero.

Combined with equation (6), the output weight is as follows.

$$\hat{\beta} = H^T (HH^T + \alpha I)^{-1} Y \tag{8}$$

Define kernel matrix

$$\begin{cases} \Omega_{ELM} = HH^T = h(x_i) \cdot h(x_j) = K(x_i, x_j) \\ K(x_i, x_j) = \exp\left(-\dfrac{\|x_i - x_j\|}{\gamma^2} \right) \end{cases} \tag{9}$$

where γ is called the kernel width.

Meanwhile, the output of the network becomes.

$$\begin{aligned} t(x) &= h(x)\hat{\beta} = h(x)H^T(\alpha I + HH^T)^{-1} Y \\ &= \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_m) \end{bmatrix}^T (\alpha I + \Omega_{ELM})^{-1} Y \end{aligned} \tag{10}$$

From equations (9) and (10), it can be seen that the kernel extreme learning machine has two parameters, kernel width γ and regularization coefficient α, which can also be called penalty factor. The different parameters lead to different results. In order to obtain the optimal parameters, evolutionary algorithm can be generally adopted to optimize them.

III. PROPOSED ALORITHM

A. KELM MODEL BASED ON K FOLD CROSS VALIDATION

In the actual training, the training results are usually good for the fitting degree of the training set (the initial condition is sensitive). However, the fitting degree of the data outside the training set is usually not so satisfactory. Therefore, rather than using all the data sets for training, we divide them into parts (which do not participate in the training) to test the parameters generated by the training set, and relatively objectively judge the degree of conformity of these parameters to the data outside the training set. This method is called cross validation.

K folding cross validation, the initial sample is divided into K subsamples, a single subsample is retained as the data of the verification model, and other K-1 samples are used for training. The cross validation was repeated K times, with each subsample verified once. The results were averaged K times or other combinations were used to obtain a single estimate. The advantage of this method is that the randomly generated subsamples are repeatedly used for training and verification, and the results are verified once each time. 10-fold cross validation is the most commonly used method.

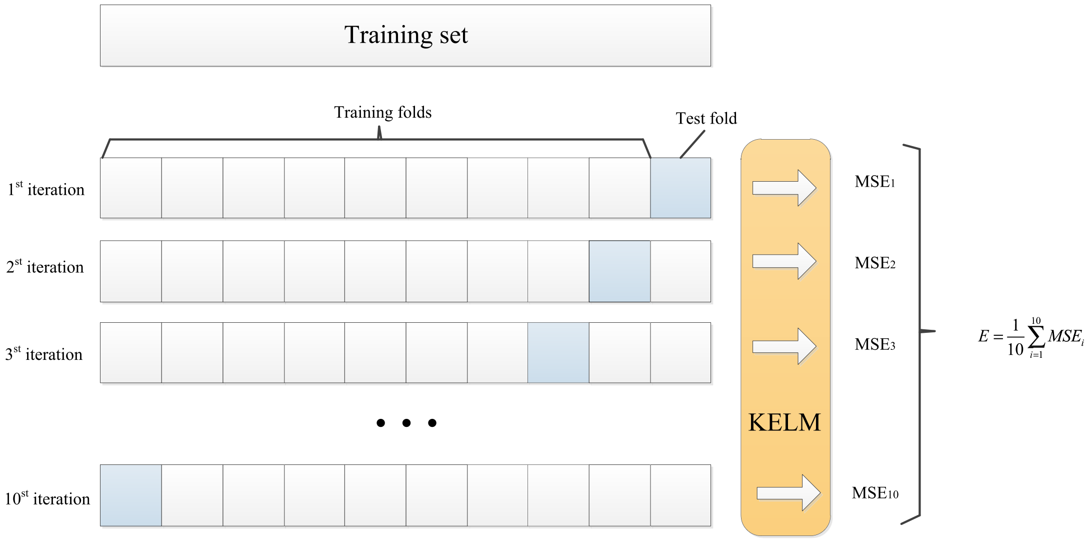


FIGURE 2. KELM structure diagram based on 10 fold cross validation.

In this paper, KELM model based on k-folding and cross validation (CVKELM) is proposed, and its model is shown in **Fig. 2**.

As can be seen from **Fig.2**, the training samples are divided into 10 parts, with 9 parts for each training and 1 part for the remaining test. A total of 10 times of model solving are conducted. With the average MSE error of the 10-degree model as the objective function, the regularization coefficient α and kernel width γ were selected. The model has excellent stability and generalization performance.

B. CVKELM MODEL BASED ON DIFFERENTIAL EVOLUTION ALGORITHM

Firstly, the DE algorithm randomly generates the initial population $\boldsymbol{X}^0 = [\boldsymbol{x}_1^0, \cdots, \boldsymbol{x}_p^0]$ in n-dimensional feasible solution space, where $\boldsymbol{x}_i^0 = [x_{i1}^0, \cdots, x_{in}^0]^T$. p is the population size of DE. The core idea of DE algorithm is to generate test population by mutation and crossover operation, and then to evaluate the fitness of test population. In addition, by applying the selection mechanism of greedy thoughtto conduct a one-to-one comparison between the original population and test population, the best population can be selected in the next generation.

The basic DE algorithm mainly includes three operations: mutation, crossover and selection. First, three individuals were randomly selected from the population for mutation operation.

$$\boldsymbol{v}_i^{t+1} = \boldsymbol{x}_{r1}^t + F(\boldsymbol{x}_{r2}^t - \boldsymbol{x}_{r3}^t) \tag{11}$$

where $\boldsymbol{v}_i^{t+1}$ represents the population obtained after mutation, t represents the population algebra, and F is the scaling factor. Generally, it is (0, 2], whose size can determine the population distribution and make the population search in the global range. $\boldsymbol{x}_{r1}^t$, $\boldsymbol{x}_{r2}^t$ and $\boldsymbol{x}_{r3}^t$ are three different individuals randomly selected from the population.

Then, crossover operation is conducted between the mutant population and the original population.

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1} & rand(j) \le C_R \text{ or } j = randn(i) \\ x_{i,j}^t & rand(j) > C_R \text{ and } j \ne randn(i) \end{cases} \tag{12}$$

where, $u_{i,j}^{t+1}$ represents the population obtained after crossover, $rand(j)$ is the random number between [0,1], j represents the j component of an individual, C_R is the crossover probability, $randn(i)$ is the random quantity between [1,…,n], which is used to ensure that the new individual has at least one dimensional component contributed by the mutant individual.

Finally, DE algorithm selects individuals with higher fitness from the original population and experimental population into the next generation through greedy selection mode.

$$\boldsymbol{x}_i^{t+1} = \begin{cases} \boldsymbol{u}_i^{t+1} & f(\boldsymbol{u}_i^{t+1}) < f(\boldsymbol{x}_i^t) \\ \boldsymbol{x}_i^t & f(\boldsymbol{u}_i^{t+1}) \ge f(\boldsymbol{x}_i^t) \end{cases} \tag{13}$$

$f(\boldsymbol{u}_i^{t+1})$ and $f(\boldsymbol{x}_i^t)$ are the fitness of $\boldsymbol{u}_i^{t+1}$ and $\boldsymbol{x}_i^t$ respectively. When the fitness of the individual $\boldsymbol{u}_i^{t+1}$ is better than that of $\boldsymbol{x}_i^t$, the individual of the test replaces the original individual. On the contrary, the individual of the test is abandoned and the original individual is retained. **Fig. 3** shows the flow chart of the differential evolutionary algorithm.

In **Fig. 3**, N is the total number of iterations, which can be used to satisfy the termination condition and stop the iteration to get the optimal result, or break out of the loop when the training number t is greater than the iteration number N, and get a relatively ideal result. In this paper, the average MSE after cross validation is taken as the objective function, and the differential evolution algorithm is applied to the CVKELM algorithm to select the regularization coefficient α and kernel width γ, namely the DECVKELM algorithm, which improves the accuracy and generalization performance

TABLE 1. Comparison of KELM results with other algorithms.

	MAE	MSE	RMSE	*R-square*	TIME(s)
BP	8.1979	138.8226	11.7823	0.4950	2.3800
SVM	8.1514	135.9627	11.6603	0.4977	1.6116
ELM	8.0149	134.9564	11.6171	0.5135	0.0481
KELM	7.9350	123.5358	11.1147	0.5216	0.1652

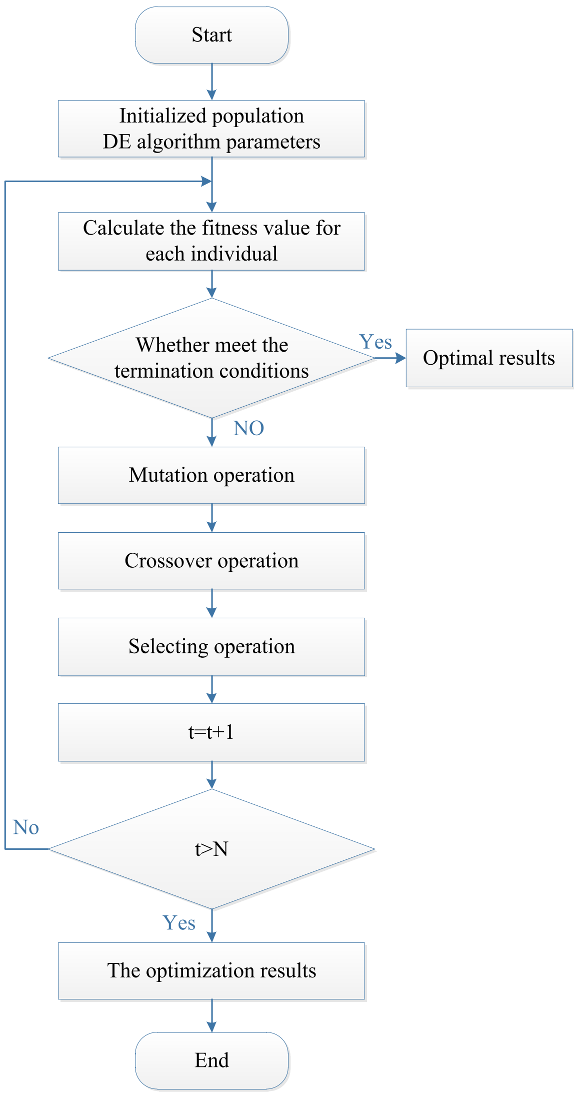


FIGURE 3. Flow chart of differential evolutionary algorithm.

of the model. The algorithm block diagram is shown in **Fig. 4** below, and the algorithm flow is as follows.

1) Initializing the population, the population number is p
2) Secondly, the fitness is calculated, and the average error of K folding and cross validation is used as the objective function
3) DE algorithm parameter mutation, crossover, selection
4) New regularization parameters α and kernel width γ are obtained
5) Then optimize until the termination condition is satisfied or greater than the number of iterations N
6) End

IV. EXPERIMENT RESULTS

A. DATA SOURCES

The test data provided by the northwest China power grid dispatching center, has forecast wind speed and the actual output data type, a total of 6000 samples, sampling time 15 min, this experiment with 5000 groups of model parameters of training samples, 1000 samples as new data to verify this model.

B. COMPARISON OF KELM RESULTS WITH OTHER ALGORITHMS

Data of 4500 groups were randomly trained, and data of 500 groups were tested. Given the regularization coefficient $\alpha = 1000$ and kernel width $\gamma = 1$, the following **Table 1** shows the results of KELM compared with other algorithms.

To evaluate the power wind prediction performance of the proposed method, we use the mean absolute error(MAE), mean square error(MSE), root MSE (RMSE) and R-square as the evaluating criterion defined as.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{14}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{15}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2} \tag{16}$$

$$R-square = \frac{SSR}{SST} = \frac{\sum_{i=1}^{n}\left(\hat{Y}_i - \bar{Y}_i\right)^2}{\sum_{i=1}^{n}\left(Y_i - \bar{Y}_i\right)^2} \tag{17}$$

where Y_i is the ith actual load value; $\hat{Y}_i$ is the ith predicted load value; $\bar{Y}_i$ is the average of the actual load value and n is the total number of predicted points.

For convenience of comparison, the number of input nodes in the BP neural network model is 1, the number of hidden layer is 1, the number of hidden layer nodes is 150 and ELM are the same, and the number of output nodes is 1. From **Table 1**, Compared with the common BP neural network and SVM algorithm, ELM has the advantages of short running time and high precision, as well as the advantages of fewer

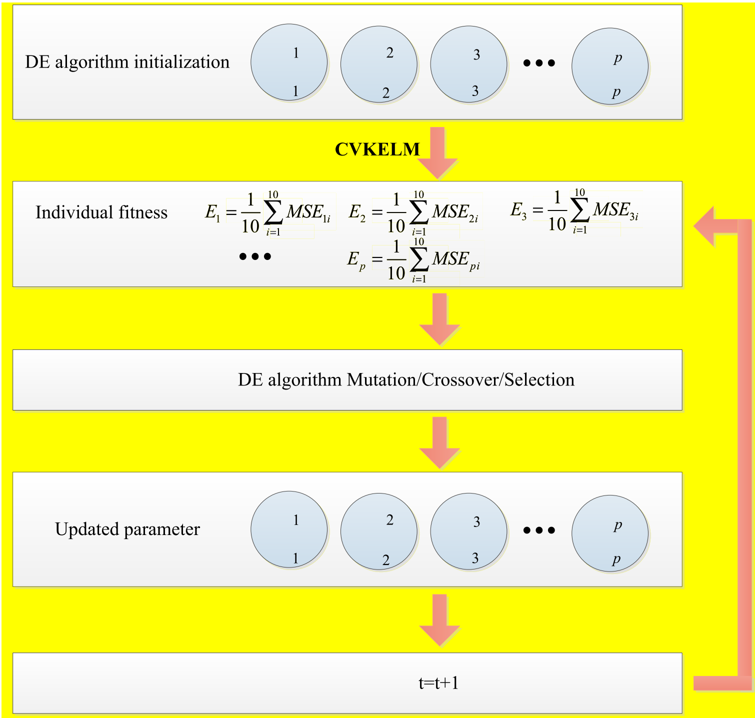


FIGURE 4. CVKELM structure diagram based on differential evolutionary algorithm.

TABLE 2. Wind power error analysis of KELM algorithm based on 10-fold cross validation.

TIMES	MAE	MSE	RMSE	*R-square*
1	7.9350	123.5358	11.1147	0.5216
2	7.4050	108.6529	10.4237	0.5489
3	7.1349	96.1553	9.8059	0.5201
4	7.3427	105.8845	10.2900	0.5237
5	7.5807	120.6932	10.9860	0.5460
6	7.4423	108.1804	10.4010	0.5361
7	7.6049	106.4661	10.3182	0.5142
8	7.9393	120.4047	10.9729	0.5213
9	7.7133	118.6069	10.8907	0.5366
10	7.6341	115.5231	10.7482	0.5122

parameters, simple design and simple parameter optimization. As the ELM algorithm is simple and easy to optimize, KELM is also improved in accuracy. one can see that the KELM algorithm compared to ELM algorithm in every error indicators improved. This data is a single wind turbine and the error value is small. If it is multiple wind motors, or even a region, it will have a great impact on its power dispatching, power generation planning, downtime, maintenance and other services, which improve economic benefits.

C. KELM SIMULATION RESULATION RESULTS BASED ON K FOLDING AND CROSS VALIDATION

In order to improve the stability and generalization performance of the model, 10 fold cross validation, regularization coefficient $\alpha = 1000$ and kernel width $\gamma = 1$ were adopted. The first result is the same as that in the previous section, and the results of the other groups varied due to changes in the training set and test set. The average MAE is 7.5732, the average MSE is 112.4103 and the average *R-square* is 0.5281 as shown in **Table 2**.

D. SIMULATION RESULTS OF CVKELM BASED ON DIFFERENTIAL EVOLUTION ALGORITHM

Taking the 10-fold cross validation average MSE as the objective function, differential evolution was adopted to optimize the regularization coefficient and kernel width. The population size was 10, and the number of iterations was 10. The results were shown in **Fig 5** below, and finally α =14657 and γ =0.5163 were obtained, It can be seen that the error decreases gradually until it reaches a stable state, which is

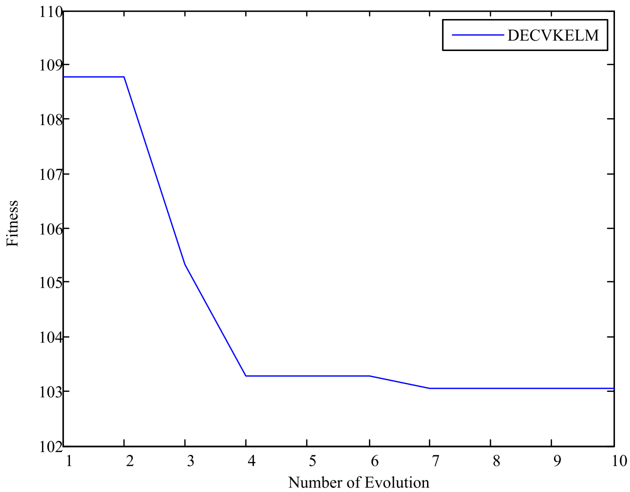


FIGURE 5. Optimization process of CVKELM algorithm based on differential evolution algorithm.

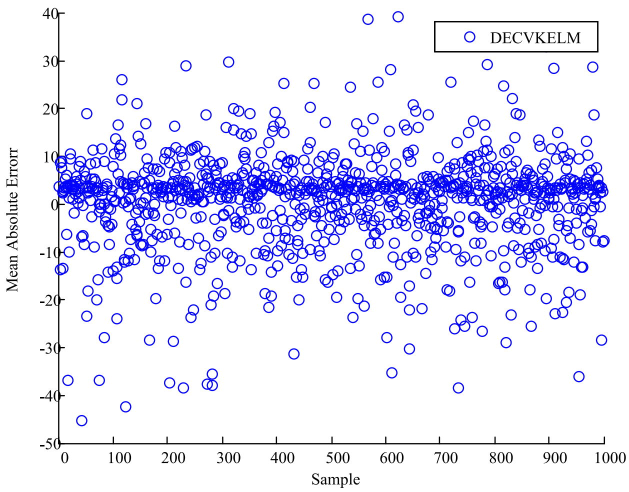


FIGURE 6. Prediction error of CVKELM wind power based on differential evolution algorithm.

consistent with the theory of differential evolution algorithm, and the final fitness was 103.0324, compared with the prior precision improved by 8.34%.

The parameters obtained from optimization were assigned to the model, and the new 1000 sets of data were used for prediction. The final prediction error was obtained which can be seen in **Fig 6**, wherein MAE = 7.3046, MSE = 104.5982 and RMSE = 10.2273. The rated power of a single wind turbine is 50MW, and the predicted power is mostly concentrated within 10MW, providing a good reference for power dispatching operation. However, there are still some large errors, which may be caused by the large gap between the predicted wind speed and the actual wind speed.

E. COMPARSION OF WIND POWER PREDICTION RESULTS BETWEEN DECVKELM AND GACVKELM

DECVKELM algorithm converges faster than GACVKELM, but it may fall into the local optimal. Since there are only two optimization parameters, the two results are very similar in the optimization process. The basic parameters are: population size 10, iteration number 10. **Table 3** gives the comparison results of parameter optimization and error analysis, It can be seen that the parameters and results obtained by the two methods are very similar, and the DECVKELM algorithm is more accurate. One can see that the DECVKELM algorithm gets the optimal result. in addition, the fitness curves of optimization process are shown in **Fig.7**. One can observe that the initial fitness of DECVKELM algorithm is lower than that of GACVKELM, but with the increase of the number of iterations, the final result is better than that of GACVKELM. **Fig.8** is the comparison chart of prediction error. Since both algorithms are based on CVKELM algorithm, the predicted results are very similar, and the error of DECVKELM is small. In addition, the convergence rate of DECVKELM is faster than that of GACVKELM.

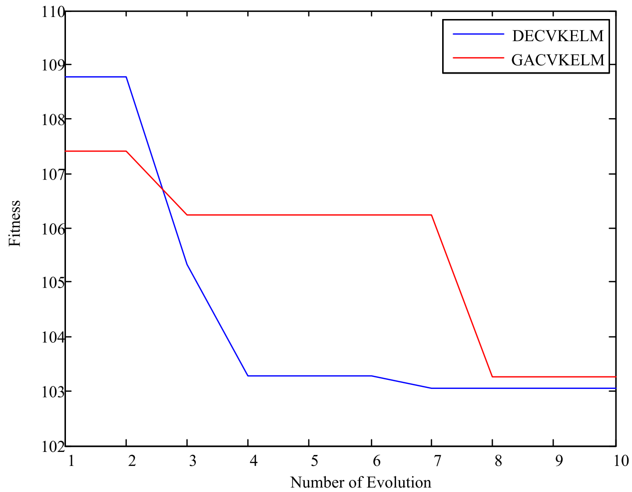


FIGURE 7. Comparison diagram of wind power prediction optimization process between DECVKELM and GACVKELM.

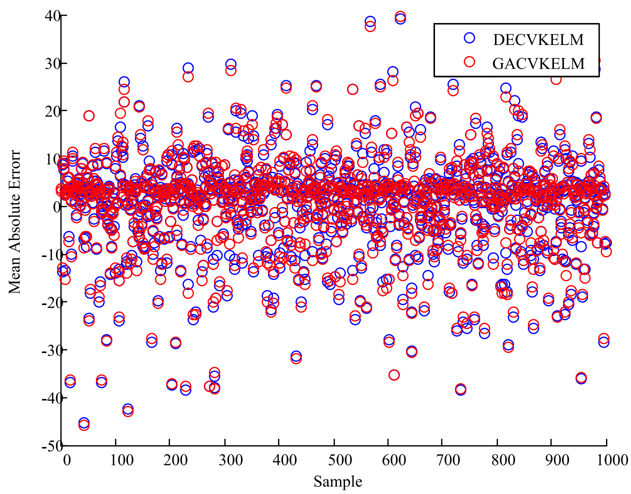


FIGURE 8. Comparison diagram of wind power prediction errors between DECVKELM and GACVKELM.

TABLE 3. Comparison table of parameter optimization and error analysis for DECVKELM and GACVKELM.

	□α	□γ	Fitness	MAE	MSE	RMSE	*R-square*
DECVKELM	14657	0.5163	103.0324	7.3046	104.5982	10.2273	0.5375
GACVKELM	14302	0.5214	103.2452	7.3592	105.6850	10.2803	0.5352
CVKELM	10000	1	112.4103	7.6278	114.0938	10.6815	0.5281

In this experiment, the convergence rate is twice that of GACVKELM.

V. CONCLUSION

In the past five years, with the increasing proportion of wind power generation, the research on the prediction accuracy of wind power has become extremely important. This paper focuses on developing a wind power prediction approach based on a novel KELM algorithm. Due to its simplicity and flexibility, the traditional KELM algorithm has been optimized by many scholars, including data preprocessing, error function optimization, parameter optimization, network model improvement, etc.

This paper mainly studies the following aspects. Firstly, KELM is used for wind power prediction due to the nonlinearity of wind power data. Secondly, to improve the stability and generalization performance of the model, K folding cross validation is used, and its average MSE error is used as the objective function. Finally, DE algorithm is proposed to optimize the parameters of KELM algorithm, compared with the prior precision improved by 8.34%, moreover, compared with GA algorithm, the convergence rate is twice that of GA algorithm when the accuracy is similar. In addition, the wind power prediction based on the above optimization algorithm can improve the prediction accuracy and stability, improve the wind power absorption capacity, and provide a reference for power grid dispatching.

REFERENCES

[1] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1033–1044, May 2014, doi: 10.1109/TPWRS.2013.2287871.

[2] Q. Yao, Y. Liu, W. Zhang, and Y. Hu, "Study on the comprehensive evaluation method of regional wind power prediction," in *Proc. 36th Chin. Control Conf. (CCC)*, 2017, pp. 10259–10264, doi: 10.23919/ChiCC.2017.8028987.

[3] K. Chen and J. Yu, "Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach," *Appl. Energy*, vol. 113, pp. 690–705, Jan. 2014, doi: 10.1016/j.apenergy.2013.08.025.

[4] C. D. Zuluaga, M. A. Álvarez, and E. Giraldo, "Short-term wind speed prediction based on robust Kalman filtering: An experimental comparison," *Appl. Energy*, vol. 156, pp. 321–330, Oct. 2015, doi: 10.1016/j.apenergy.2015.07.043.

[5] Y. Cao, Y. Liu, D. Zhang, W. Wang, and Z. Chen, "Wind power ultra-short-term forecasting method combined with pattern-matching and ARMA-model," in *Proc. IEEE Grenoble Conf.*, Jun. 2013, pp. 1–4, doi: 10.1109/PTC.2013.6652257.

[6] J.Y. Zhang and C. Wang, "Application of ARMA model in ultra-short term prediction of wind power," in *Proc. Int. Conf. Comput. Sci. Appl.*, Dec. 2013, pp. 361–364, doi: 10.1109/CSA.2013.91.

[7] H. Quan, D. Srinivasan, and A. Khosravi, "Short-term load and wind power forecasting using neural network-based prediction intervals," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 2, pp. 303–315, Feb. 2014, doi: 10.1109/TNNLS.2013.2276053.

[8] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, Dec. 2006, doi: 10.1016/j.neucom.2005.12.126.

[9] G. Huang, G.-B. Huang, S. Song, and K. You, "Trends in extreme learning machines: A review," *Neural Netw.*, vol. 61, pp. 32–48, Jan. 2015, doi: 10.1016/j.neunet.2014.10.001.

[10] N. Li, F. He, and W. Ma, "Wind power prediction based on extreme learning machine with kernel mean p-Power error loss," *Energies*, vol. 12, no. 4, p. 673, Feb. 2019, doi: 10.3390/en12040673.

[11] Y. Yang, Q. M. J. Wu, Y. Wang, K. M. Zeeshan, X. Lin, and X. Yuan, "Data partition learning with multiple extreme learning machines," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1463–1475, Aug. 2015, doi: 10.1109/Tcyb.2014.2352594.

[12] H. Zhou, G.-B. Huang, Z. Lin, H. Wang, and Y. Chai Soh, "Stacked extreme learning machines," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 2013–2025, Sep. 2015, doi: 10.1109/Tcyb.2014.2363492.

[13] K. Ning, M. Liu, M. Dong, C. Wu, and Z. Wu, "Two efficient twin ELM methods with prediction interval," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2058–2071, Sep. 2015, doi: 10.1109/Tnnls.2014.2362555.

[14] R. Ak, O. Fink, and E. Zio, "Two machine learning approaches for short-term wind speed time-series prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1734–1747, Aug. 2016, doi: 10.1109/Tnnls.2015.2418739.

[15] Y. Yang and Q. M. J. Wu, "Extreme learning machine with subnetwork hidden nodes for regression and classification," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2885–2898, Dec. 2016, doi: 10.1109/Tcyb.2015.2492468.

[16] L. L. C. Kasun, Y. Yang, G.-B. Huang, and Z. Zhang, "Dimension reduction with extreme learning machine," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3906–3918, Aug. 2016, doi: 10.1109/Tip.2016.2570569.

[17] X. Lu, L. Ming, W. Liu, and H.-X. Li, "Probabilistic regularized extreme learning machine for robust modeling of noise data," *IEEE Trans. Cybern.*, vol. 48, no. 8, pp. 2368–2377, Aug. 2018, doi: 10.1109/Tcyb.2017.2738060.

[18] S. Zhang, Z. Liu, X. Huang, and W. Xiao, "A modified residual extreme learning machine algorithm and its application," *IEEE Access*, vol. 6, pp. 62215–62223, 2018, doi: 10.1109/Access.2018.2876360.

[19] Q. Li, Y. Liu, S. Wang, Q. Gao, and X. Gao, "Image classification using low-rank regularized extreme learning machine," *IEEE Access*, vol. 7, pp. 877–883, 2019, doi: 10.1109/Access.2018.2885982.

[20] R. Li, X. D. Wang, L. Lei, and Y. F. Song, "L_{21}-norm based loss function and regularization extreme learning machine," *IEEE Access*, vol. 7, pp. 6575–6586, 2019, doi: 10.1109/Access.2018.2887260.

[21] J. Liu, X. Liu, C. Liu, B. T. Le, and D. Xiao, "Random search enhancement of incremental regularized multiple hidden layers ELM," *IEEE Access*, vol. 7, pp. 36866–36878, 2019, doi: 10.1109/Access.2019.2905077.

[22] S. Scardapane, D. Comminiello, M. Scarpiniti, and A. Uncini, "Online sequential extreme learning machine with kernels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 9, pp. 2214–2220, Sep. 2015, doi: 10.1109/Tnnls.2014.2382094.

[23] H. Wen, H. Fan, W. Xie, and J. Pei, "Hybrid structure-adaptive RBF-ELM network classifier," *IEEE Access*, vol. 5, pp. 16539–16554, 2017, doi: 10.1109/Access.2017.2740420.

[24] A. O. M. Abuassba, Y. Zhang, X. Luo, D. Z. Zhang, and W. Aziguli, "A heterogeneous ensemble of extreme learning machines with correntropy and negative correlation," *Tsinghua Sci. Technol.*, vol. 22, no. 6, pp. 691–701, Dec. 2017, doi: 10.23919/TST.2017.8195351.

[25] S. Q. Wang, Q. Liu, E. Zhu, J. P. Yin, and W. T. Zhao, "MST-GEN: An efficient parameter selection method for one-class extreme learning machine," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3266–3279, Oct. 2017, doi: 10.1109/Tcyb.2017.2707463.

[26] M. Rafiei, T. Niknam, and M.-H. Khooban, "Probabilistic forecasting of hourly electricity price by generalization of ELM for usage in improved wavelet neural network," *IEEE Trans. Ind. Informat.*, vol. 13, no. 1, pp. 71–79, Feb. 2017, doi: 10.1109/Tii.2016.2585378.

[27] H. R. Oliveira Rocha, L. J. Silvestre, W. Cardoso Celeste, D. J. Custodio Coura, and L. O. Rigo Junior, "Forecast of distributed electrical generation system capacity based on seasonal micro generators using ELM and PSO," *IEEE Latin Amer. Trans.*, vol. 16, no. 4, pp. 1136–1141, Apr. 2018, doi: 10.1109/TLA.2018.8362148.

[28] X. Yang, K. Xu, S. Xu, and P. X. Liu, "Image noise level estimation for rice noise based on extended ELM neural network training algorithm," *IEEE Access*, vol. 7, pp. 1943–1951, 2019, doi: 10.1109/Access.2018.2886294.

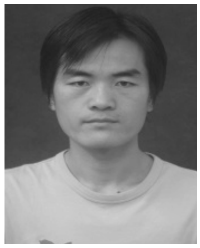
[29] Y. Sui, W. Yu, and Q. Luo, "Jointly optimized extreme learning machine for short-term prediction of fading channel," *IEEE Access*, vol. 6, pp. 49029–49039, 2018, doi: 10.1109/Access.2018.2868480.

[30] J. Ri, L. Liu, Y. Liu, H. Wu, W. Huang, and H. Kim, "Optimal weighted extreme learning machine for imbalanced learning with differential evolution [Research Frontier]," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 32–47, Aug. 2018, doi: 10.1109/Mci.2018.2840707.

[31] Z. Shao, M. J. Er, and N. Wang, "An efficient Leave-One-Out Cross-Validation-Based extreme learning machine (ELOO-ELM) with minimal user intervention," *IEEE Trans. Cybern.*, vol. 46, no. 8, pp. 1939–1951, Aug. 2016, doi: 10.1109/Tcyb.2015.2458177.

NING LI (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Xi'an Jiaotong University (XJTU), Xi'an, China, in 2006, 2009, and 2013, respectively, all in electrical engineering. He was a Faculty Member of electrical engineering with the Xi'an University of Technology, where he is currently an Associate Professor. In 2018, he was with Energy Engineering, Mälardalen University, Västerås, Sweden, as a Visiting Scholar. From 2019 to 2020, he was with the Department of Electronic, Electrical, and Systems Engineering, School of Engineering, University of Birmingham, Birmingham, U.K., as a Visiting Scholar. His research interests include topology, modulation, and control technology of high power multilevel converter, analysis, and control of power quality, adaptive signal processing, and artificial intelligence algorithm in power systems.

FUXING HE (Student Member, IEEE) is currently pursuing the M.S. degree in electrical engineering with the Xi'an University of Technology, Xi 'an, China. His research interests include wind power forecasting and EV battery management systems.



WENTAO MA (Member, IEEE) received the B.S. degree in mathematics and applied mathematics from the Shannxi University of Technology, in 2003, the M.S. degree in computing mathematics from the Huazhong University of Science and Technology, in 2008, and the Ph.D. degree in information and communication engineering from Xi'an Jiaotong University, in 2015. He is currently an Associate Professor with the School of Electrical Engineering, Xi'an University of Technology, Xi'an, China. His research interests include machine learning, information theory, adaptive signal processing, and their applications in electrical and computer engineering.

RUOTONG WANG received the B.Eng. degree in electronics communication and engineering from the University of Liverpool, U.K., in 2014, and the M.Sc. degree in communication and signal processing from Imperial College London, U.K., in 2016. He is currently pursuing the Ph.D. degree with the University of Liverpool, U.K. His current research interests include the control of voice coil motor and its applications in tribotest and motion control.

XIAOPING ZHANG (Fellow, IEEE) is currently a Professor of electrical power systems with the University of Birmingham, U.K. He is also the Director of Smart Grid, Birmingham Energy Institute, and the Co-Director of the Birmingham Energy Storage Center. He has coauthored the first and second edition of the monograph *Flexible AC Transmission Systems: Modeling and Control* (Springer in 2006 and 2012, respectively). He has also coauthored the book *Restructured Electric Power Systems: Analysis of Electricity Markets With Equilibrium Models* (IEEE Press/Wiley, 2010). He pioneered the concept of "Energy Quality", "Global Power & Energy Internet," "Energy Union," and "UK's Energy Valley." He is an Advisor to the IEEE PES UK and Ireland Chapter.