A WSN-Based Prediction Model of Microclimate in a Greenhouse Using Extreme Learning Approaches

Qi Liu*, Dandan Jin*, Jian Shen*, Zhangjie Fu**, Nigel Linge***

* College of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, Jiangsu, China

** Jiangsu Engineering Centre of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing, Jiangsu, China

*** The University of Salford, Salford, Greater Manchester, UK

qi.liu@nuist.edu.cn, 1005949332@qq.com, s_shenjian@126.com, wwwfzj@126.com, n.linge@salford.ac.uk

***Abstract*—Monitoring and controlling microclimate in a greenhouse becomes one of the research hotspots in the field of agrometeorology, where the application of Wireless Sensor Networks (WSN) recently attracts more attentions due to its features of self-adaption, resilience and cost-effectiveness. Present microclimate monitoring and control systems achieve their prediction by manipulating captured environmental factors and traditional neural network algorithms; however, these systems suffer the challenges of quick prediction (e.g. hourly and even minutely) when a WSN network is deployed. In this paper, a novel prediction method based on an Extreme Learning Machine (ELM) algorithm and KELM (Kernel based ELM) is proposed to predict the temperature and humidity in a practical greenhouse environment in Nanjing, China. Indoor temperature and humidity are measured as data samples via WSN nodes. According to the results, our approach (0.0222s) has shown significant improvement on the training speed than Back Propagation (BP) (0.7469s), Elman (11.3307s) and Support Vector Machine (SVM) (19.2232s) models, the accuracy rate of our model is higher than those models. In the future, research on faster learning speed of the ELM and KELM based neural network model will be conducted.**

***Keyword*—Wireless Sensor Networks; Kernel based Extreme Learning Machine; Greenhouse Microclimate; Prediction Model**

Manuscript received October 1, 2015. This work is supported by the NSFC (61300238, 61300237, 61232016, U1405254, 61373133), Basic Research Programs (Natural Science Foundation) of Jiangsu Province (BK20131004), Scientific Support Program of Jiangsu Province (BE2012473) and the PAPD fund.

Q. Liu is with the College of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China (e-mail: qi.liu@nuist.edu.cn).

D. Jin is with the College of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China (e-mail: 1005949332@qq.com).

S. Shen is with the College of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China (e-mail: s_shenjian@126.com).

Z. Fu is with the Jiangsu Engineering Centre of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing, 210044, China (corresponding author; e-mail: wwwfzj@126.com).

N. Linge is with the School of Computing, Science and Engineering, University of Salford, Salford, M5 4WT, UK (E-mail: n.linge@salford.ac.uk).

I. INTRODUCTION

MODERN greenhouses provide a suitable indoor microclimate meeting the requirements of plant growth. A prediction model of the microclimate in a greenhouse therefore becomes critical for the establishment of control strategies and consequent evaluation [1]. The design of such a model becomes challenging due to the features of the greenhouse microclimate, i.e. nonlinear, multiple input multiple output, and its strong coupling between relevant factors. It is also affected by the indoor and outdoor climate environment, crops grown inside and movements of control facilities [2]. All reasons above make it difficult to establish a precise mathematical model to achieve fast and accurate prediction [3]. With rapid development of short-range wireless communication, e.g. Wireless sensor networks (WSN), real-time (or nearly real-time) collection of relevant environmental data in a greenhouse turns out to be convenient, but also raises new challenges on microclimate prediction [4].

Mechanism modelling method based on energy balance because of its multiple parameters and low accuracy, it is difficult to meet the need of practical application [1]. Another system identification method based on the data input and output because it needs less parameters, good adaptive ability and higher simulation accuracy [1-3]. And it has been widely applied. The traditional neural network algorithm, Back Propagation (BP) was applied to build the prediction model of microclimate in a greenhouse. Support Vector Learning was also used for the prediction [5-6]. However, the existence of slow training speed, easy to fall into local minima and the choice of learning rate sensitive etc. inherent shortcomings in the neural network makes its application in the greenhouse prediction model is not ideal.

Extreme Learning Machine (ELM) algorithm [7] is a new algorithm for these mentioned shortcomings of the neural networks, and it has the advantages of faster training speed, the global optimal solution and good generalization. In addition, Kernel Extreme Learning Machine (KELM) has applied kernel function algorithm to the ELM [8]. In this paper, the ELM and KELM algorithms are used to establish prediction models of microclimate in a greenhouse.

This paper presents the WSN-based prediction model of microclimate in a greenhouse using both ELM and KELM algorithms is structured as follows: Section II introduces the related work. Section III presents the principles of ELM algorithm and KELM algorithms and the principles of modelling the greenhouse environment using the ELM and KELM algorithms. Section IV presents the experimental results from real-world data and discussion, and Section V presents conclusions.

II. Related Work

The system identification method based on input-output data needs less parameter and obtains high simulation precision. Patril et al. [9] used the auto-regression and neural network to build a temperature model of tropical greenhouse. A greenhouse microclimate model was built based on neural network in [10], where it was found that outdoor wind and temperature is not vital input factors for the greenhouse microclimate model in summer. Indoor temperature was gathered in a greenhouse in [11], where relative humidity, the intensity of solar radiation and wind speed was collected as input items. In [12], an environment factor model was established in a greenhouse to predict its microclimate based on a fuzzy neural network. Wang et al. [13] used BP algorithm to establish the rainy season greenhouse microclimate model in Jianghuai area, and results show that the model has higher precision and is a beneficial supplement to the physical model. Ferreira [14] used Radical Basis Function (RBF) neural networks to establish fitting model of a hydroponic greenhouse, and obtained the very good fitting results. Fourati [15] used an Elman neural network to emulate the direct dynamics of a greenhouse, the Elman model was used to train control model. Another identification method of nonlinear systems, support vector machine regression (SVMR) was applied to the modelling of greenhouse microclimate system areas, such as online modelling method of weighted least squares support vector machine based on [16]. In [17], indoor data, such as inner temperature, humidity, wind speed, solar radiation intensity, etc. were gathered, so that a greenhouse microclimate humidity model was designed for the prediction in the north of china in winter by using a BP neural network improved by a genetic algorithm.

III. Prediction Model

A. Principles of ELM

For a given set of random samples, $S_Q = \{(\boldsymbol{x}_i, t_i)\}_{i=1}^{Q}$, where $\boldsymbol{x}_i = [x_i, x_{i+1}, \dots, x_{i+n-1}], t_i = x_{i+n}$, x_i is the input vector, $\boldsymbol{t}_i$ is the output corresponding to x_i, n is the embedded dimension. An ELM regression model containing the L hidden layer neurons can be expressed as in [18]:

$$\sum_{i=1}^{L} \beta_i f(\boldsymbol{\omega}_i \boldsymbol{x}_j + b_i) = t_j, \qquad j = 1,2, \dots, Q \tag{1}$$

where Q is the number of samples in the training set, ω_i is the input weights between the i^th neuron and the input layer, and $\omega_i = [\omega_{i1}, \omega_{i2}, \dots, \omega_{in}]$, β_i is the output weights between the i^th neuron and the output layer, b_i is the threshold values of the i^th neuron. Eq. (1) can be written in a matrix form as:

$$\boldsymbol{H\beta} = \boldsymbol{T}' \tag{2}$$

where $\boldsymbol{H}$ is an output matrix of the hidden layer, can be written specifically as:

$$\mathrm{H} = \begin{bmatrix} f(\boldsymbol{\omega}_1 \boldsymbol{x}_1 + b_1) & \dots & f(\boldsymbol{\omega}_L \boldsymbol{x}_1 + b_L) \\ \vdots & \vdots & \vdots \\ f(\boldsymbol{\omega}_1 \boldsymbol{x}_Q + b_1) & \dots & f(\boldsymbol{\omega}_L \boldsymbol{x}_Q + b_L) \end{bmatrix}_{Q \times L} \tag{3}$$

$\boldsymbol{\beta}$ is an output weight, which can be written specifically as: $\beta = [\beta_1, \beta_2, \cdots, \beta_L]^T$.

$\boldsymbol{T}$ is an output weight, which can be written specifically as: $\mathrm{T} = [t_1, t_2, \cdots, t_L]^T$.

In most cases of Eq. (3), $\boldsymbol{Q}$ is much greater than $\boldsymbol{L}$. By solving Eq. (2), the output weights $\boldsymbol{\beta}$ can be calculated as in [19]:

$$\hat{\boldsymbol{\beta}} = \boldsymbol{H}^{+}\boldsymbol{T} \tag{4}$$

where H^{+} is the Moore-Penrose generalized inverse of the hidden layer output matrix $\boldsymbol{H}$, and it can be calculated as: $\mathbf{H}^{+} = (\mathbf{H}^{T}\mathbf{H})^{-1}\mathbf{H}^{T}$.

Therefore, after training the final prediction model using ELM can be written as:

$$\mathrm{t} = \sum_{i=1}^{L} \beta_i f(\boldsymbol{\omega}_i \boldsymbol{x} + b_i) \tag{5}$$

where $\boldsymbol{x}$ is the input vector of prediction model using ELM, $\boldsymbol{t}$ is the output vector of prediction model using ELM.

B. Principles of KELM

Due to the fact that two thresholds, w and b in the ELM model are generated randomly, the performance of an ELM based prediction model suffers from its poor stability. KELM is therefore designed in order to introduce a stable kernel for the ELM algorithm for constant fitness [8].

The KELM algorithm can be summarized as follows:

Input: the training set $(\boldsymbol{x}_i, \boldsymbol{t}_i)$, the kernel function $\boldsymbol{K}$, the ridge regression parameter $\boldsymbol{C}$.

Output: prediction of test set $\boldsymbol{f(x)}$.

Step 1. Calculate kernel matrix $\boldsymbol{K}$:

$$\boldsymbol{\Omega}_{ELM} = \boldsymbol{HH}^T : \ \Omega_{ELM_{ij}} = h(\boldsymbol{x}_i) \cdot h(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Step 2. Calculate matrix inverse:

$$(\frac{\boldsymbol{I}}{\boldsymbol{C}} + \Omega_{ELM})^{-1}$$

Step 3. Calculate the mapping of test data set:

$$\boldsymbol{K}_x = \begin{bmatrix} K(\boldsymbol{x}, \boldsymbol{x}_1) \\ \vdots \\ K(\boldsymbol{x}, \boldsymbol{x}_N) \end{bmatrix}$$

Step 4. Calculate the predicted results:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{K}_x (\frac{\boldsymbol{I}}{\boldsymbol{C}} + \Omega_{ELM})^{-1} \boldsymbol{T}$$

C. Prediction Model of Microclimate in a Greenhouse Using ELM and KELM

Greenhouse microclimate is an extremely complicated system, influenced by outdoor environmental factors, the structures of greenhouse and the operation status of environmental control equipment, etc. So the historical sample of indoor temperature and humidity imply the above information. Hence in order to predict temperature and humidity in greenhouse that can only use the historical sample of indoor temperature and humidity to establish prediction models. While establishing prediction model To predict the next sample prediction model uses the three past samples: $s_{t-3}, s_{t-2}, s_{t-1}$, and the current sample s_t. Hence, the t[th] input-output instance is:

$$\boldsymbol{x}_t = [s_{t-3}, s_{t-2}, s_{t-1}, s_t],\ \mathbf{t}_t = s_{t+1}$$

where **s** can be temperature or humidity value.

The ELM algorithm can be represented as follows:

Step 1. Initialization. Set the number of the neurons and activation function in the hidden layer, randomly generate input weights $\boldsymbol{\omega}$ and the threshold value **b**;

Step 2. Use the parameters obtained in Step 1 and the input matrix of the training set to calculate the output matrix H of the hidden layer;

Step 3. Use H and T to calculate the output weights $\hat{\beta}$: $\hat{\beta} = H^{+}T$;

Step 4. Use Eq. (5) to calculate the predicted results.

IV. Results and Discussion

A. WSN

A weather observation system is implemented based on a ZigBee enabled wireless sensor network. Customised sensor nodes are designed providing four open communication interfaces for wide compatibility of sensors. The nodes also support multiple wireless data communication methods including free short-range ISM radio bands at 2.4GHz and 915MHz, as well as enhanced mobile telecommunication technologies, e.g. EDGE and HSDPA. The prototype design, hardware block diagram and PCBs of the sensor node are shown in Fig. 1.

A closed polycarbonate casing design makes the dust particle and liquid ingress protection of our nodes reach IP6x and IPx5 respectively. Four generic communication interfaces are offered to accept third-party analogue and digital sensors. Adapters/converters, shown in the perception part of Fig. 1(c) have been organised and fitted into the corresponding pins of the interfaces, as depicted in Fig. 2.

B. Results of the predicting using the ELM model

To establish the temperature model, the training parameters of ELM were as follows: the number of neurons in the hidden layer is 26 and the activation function of neurons in the hidden layer is sin. In order to meet the requirements of the prediction model, all data have been normalized to range [0, 1] according to Eq. (6):

$$s_t = (s_t - s_{min})/(s_{max} - s_{min}) \tag{6}$$

where s_{min} is the minimum number of sample series, s_{max} is the maximum number of sample series.

Establishing humidity model used the same method with temperature model, the difference is the sample set, so establishing humidity model was not described in detail.

The fitting results of using the ELM algorithm to predict the temperature and humidity have been done through simulation tests using Matlab. The results are shown in Fig. 3 and Fig. 4, where the red line indicates the actual value, and the blue line indicates the predicted value. As can be seen from the figures, using ELM algorithm to predict the greenhouse environmental factors basically reached the expected results. The models have higher accuracy, with a good fit between the predictive values and the actual values. This shows that the simulation of greenhouse environment factors using ELM algorithm is effective and can meet the needs of agricultural production. And, the effect in the temperature simulation is more superior.

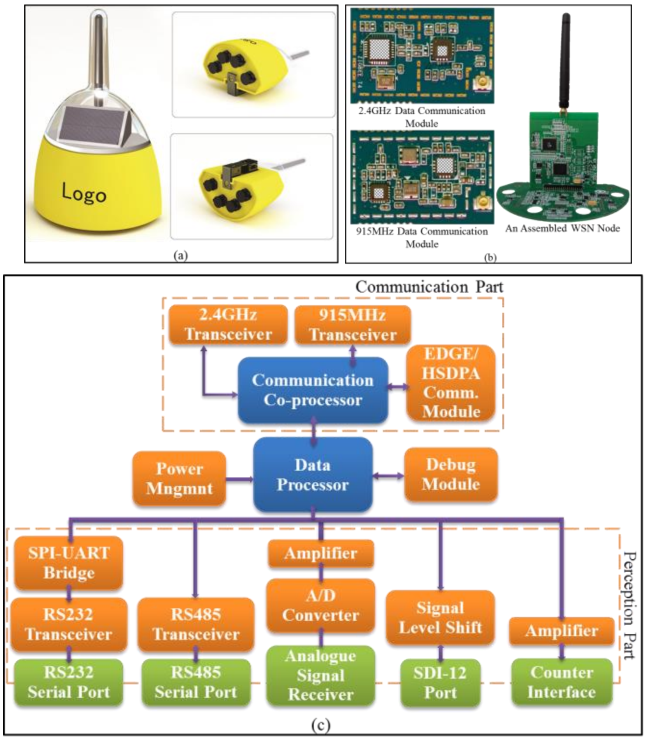
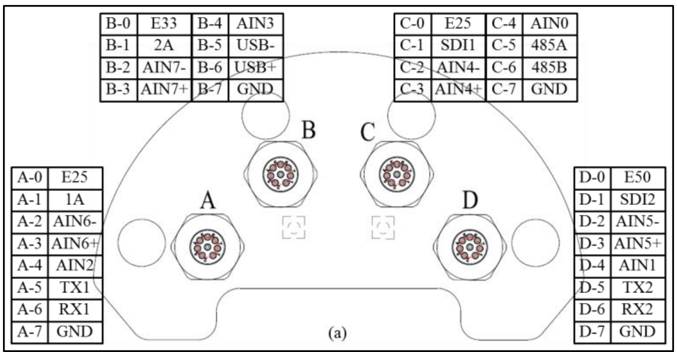


Fig. 1. The hardware implementation of a customised WSN node. (a) The prototype design; (b) the PCB design of communication module and the entire assembled design of a node; (c) the block diagram.



Pin Name	Description	Pin Name	Description
Exx	Power Output*	TXx	UART Tx**
xA	Counter Input**	RXx	UART Rx **
AINx-	Negative Differential Analog Input**	USB-	USB Negative Signal
AINx+	Positive Differential Analog Input**	USB+	USB Positive Signal
AINx	Analog Input**	SDIx	SDI-12 data signal **
GND	Ground	485	RS485 signal

*: "xx" specifies the output voltage level
**: 'x' specifies the sequence number

(b)

Fig. 2. The pin arrangement of four interfaces. (a) Pin numbers and their names specified in four pin tables; (b) the pin names and their corresponding description.

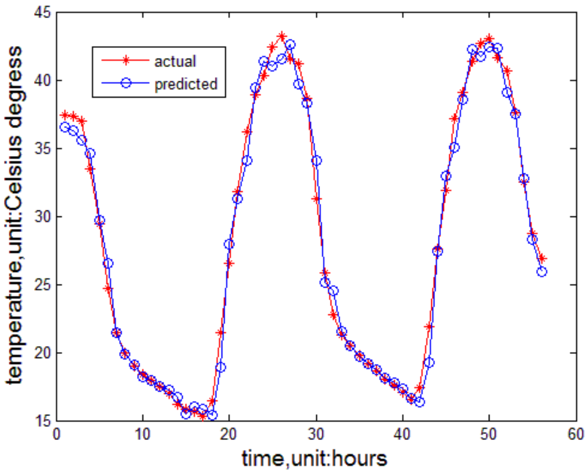


Fig. 3. The curve fitting of predicted and actual data for the greenhouse temperature.

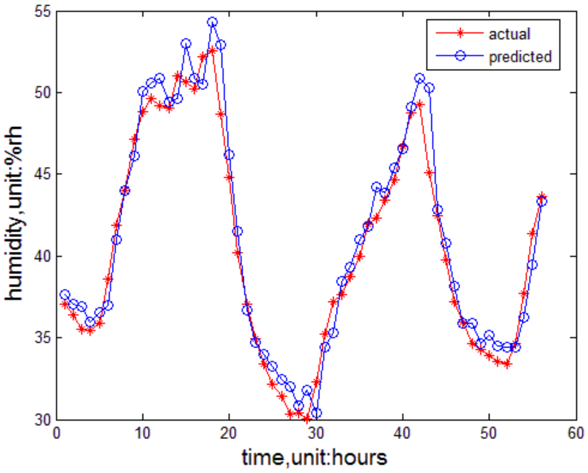


Fig. 4. The curve fitting of predicted and actual data for the greenhouse humidity.

C. *Results of the predicting using the KELM model*

Same history samples as from the ELM were used for the KELM model. Related parameters for Genetic Algorithms (GA) configuration to Optimize KELM learning parameters were set as well, as shown in Table I.

TABLE I
GA SETTINGS

GA Parameters	Value
Max. Population No.	20
Max. Evolution Generation	200
Gap Rate	0.9
Crossover Probability	0.7
Mutation Probability	0.07

Fig. 5 and 6 depict that when using genetic algorithm to optimize KELM learning parameters, the optimal fitness value of GA can quickly reached extremes. It can also be used to find the optimal learning combination (C, σ), which makes the fitting performance of KELM prediction model optimal.

D. *Discussion*

In order to better evaluate the performance of the ELM models, three algorithms are used to compare with ELM. Three algorithms were selected, i.e. BP, Elman, and SVM. These models have same training and testing sets with ELM models, the input and output data also take the same way with ELM models.

BP neural network has 5 hidden neurons of temperature and humidity model, the maximum number of iterations is 1000, the learning rate is 0.2, and the mean squared error goal is 0.0001, other parameters default. Elman neural network has 13 hidden neurons of temperature model and 18 hidden neurons of humidity model, the hidden layer transfer function is the tansig, the output layer transfer function is the purelin, and the mean squared error goal is 0.0001, other parameters default. Kernel function of SVM model is RBF. In temperature Model penalty factor parameter c = 16, variance g= 0.125, in humidity model c = 11.3137, g = 0.353553, other parameters are set as default values.

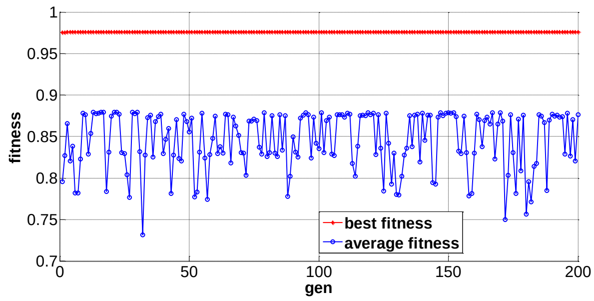


Fig. 5. The Fitness of temperature using GA to optimize KELM parameters.

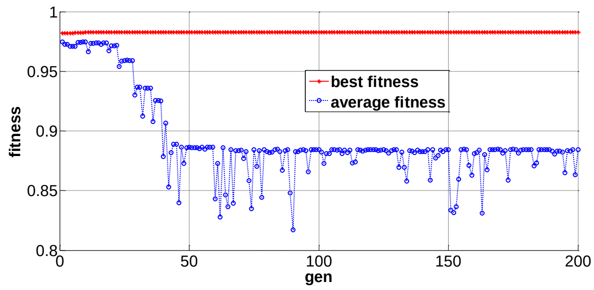


Fig. 6. The Fitness of Humidity using GA to optimize KELM parameters.

Table II shows the results of predicting the environment factors using three algorithms just mentioned. Since the input weights and thresholds are generated randomly, for without loss of generality, performance evaluation parameters for all algorithms were taken average of 50 times as the results. The performance evaluation parameters are training time, root mean square error (RMSE) and the coefficient of determination (R^2). And the smaller training time, the smaller RMSE and the higher R2 indicated a better performance of the model. Specific calculation formulas are shown as Eq. (7) and (8):

$$\mathrm{RMSE} = \sqrt{\frac{1}{l}\sum_{i=1}^{l}(\hat{t}_i - t_i)^2} \tag{7}$$

$$R^2 = \frac{\left(l\sum_{i=1}^{l}\hat{t}_i t_i - \sum_{i=1}^{l}\hat{t}_i\sum_{i=1}^{l}t_i\right)^2}{\left(l\sum_{i=1}^{l}\hat{t}_i^{\,2} - (\sum_{i=1}^{l}\hat{t}_i)^2\right)\left(l\sum_{i=1}^{l}t_i^{\,2} - (\sum_{i=1}^{l}t_i)^2\right)} \tag{8}$$

where, l is the number of testing samples, $t_i (i = 1,2, \ldots, l)$ is the actual value for the i[th] sample, $\hat{t}_i (i = 1,2, \ldots, l)$ is the predicted value for the i[th] sample.

Seen from Table II, the ELM algorithm ran around 33 times faster than BP, 510 times faster than Elman, and 865 times faster than SVM for predicting the temperature in the greenhouse. So the ELM model showed great superiority in training speed. At the same time, the RMSE of ELM model are lower than and R^2 was higher than BP, Elman and SVM model, the ELM model showed the high accuracy and fitting ability. The ELM model of humidity also exhibited the same advantages. It can also been that ELM models can quickly predict the greenhouse environmental factors in the context of maintaining the accuracy. Therefore, ELM model is more suitable to predict the greenhouse environmental factors.

From Table III, it can be seen the KELM model outperforms the training speed and generalization ability. First, the train speed of the KELM model is 2.9 times faster than the ELM model. In addition, the standard deviation of the KELM model is 0, showing its stability. The results on humidity have shown similar trends. Compared to BP, Elman and SVR, the KELM has also depicted better performance, as shown in Table IV.

TABLE II
PERFORMANCE COMPARISON OF THE ELM, BP, ELMAN AND SVM ALGORITHMS

Environment factors	Algorithms	Training Time(seconds)	RMSE	R2	Nodes of Neurons
temperature	ELM	0.0222	1.0586	0.9883	26
	BP	0.7469	1.1841	0.9858	5
	Elman	11.3307	1.1840	0.9853	13
	SVM	19.2232	1.1537	0.9865	-
humidity	ELM	0.0187	1.4177	0.9648	18
	BP	0.6833	1.6398	0.9586	5
	Elman	11.5784	1.4191	0.9641	18
	SVM	20.0633	1.4635	0.9605	-

TABLE III
PERFORMANCE COMPARISON OF THE ELM AND KELM ALGORTHAMS

Environment Factors	Algorithms	Parameter	Value	Training Time (seconds)	R^2 ±SD
temperature	KELM	(C，σ)	($2^{16.1170}$，$2^{-6.1797}$)	0.0023	0.9762±0
	ELM	*L*	13	0.0068	0.9749±0.0019
humidity	KELM	(C，σ)	($2^{999.9870}$，$2^{1.2519}$)	0.0026	0.9829±0
	ELM	*L*	27	0.0069	0.9828±0.0003

TABLE IV
PERFORMANCE COMPARISON OF THE ELM, BP, ELMAN AND SVR ALGORITHMS

Environment Factors	Algorithms	Parameter	Value	Training Time (seconds)	R^2 ±SD
temperature	KELM	(C，σ)	($2^{16.1170}$，$2^{-6.1797}$)	0.0023	0.9762±0
	BP	*L*	5	0.9126	0.9727±0.0033
	Elman	*L*	10	15.8528	0.9748±0.0009
	SVR	(C，σ)	($2^{1.4142}$，$2^{1.4142}$)	0.0318	0.9731±0
humidity	KELM	(C，σ)	($2^{999.9870}$，$2^{1.2519}$)	0.0026	0.9829±0
	BP	*L*	5	0.9999	0.9809±0.0026
	Elman	*L*	12	15.0532	0.9807±0.0006
	SVR	(C，σ)	(2^{16}，$2^{1.4142}$)	0.0330	0.9827±0

V. CONCLUSION

This paper applied ELM algorithm and KELM algorithm to predict the greenhouse environmental factors. Different from the traditional learning algorithms, ELM algorithm randomly generated input weights and thresholds, and simply set the number of hidden layer neurons, we can obtain the unique global optimal solution. The algorithm is simple, fast and high simulation precision. Comparison of BP, Elman and SVM algorithms in environmental factors prediction, ELM showed better performance. It proved that it is feasible to use ELM algorithm to predict environmental factors, and which can provide support for the intelligent control of greenhouse. Compared with the prediction model based on the ELM, the refined model based on the KELM depicts better results on computing speed and accuracy, with the stability of the model being maintained. Compared with the prediction models based on BP, Elman and SVR, the KELM model requires less training time, but shows stronger fitness and more stable performance.

Because of the influence of environment factors in greenhouse by different types of greenhouse structure and material, the types of crops and planting mode, the weather changes, human disturbance and run state of control equipment and many other factors, the steady model to predict the environmental factor is inappropriate. In future research, the model will be improved further. Building an online model of greenhouse environment factors is very necessary. Moreover, if shortening time interval (e.g. 5min, 10min), whether or not to make model more perfect, there is need for further study.

REFERENCES

[1] L. Qin and G. Wu, "The present situation and prospect of modeling and control of the greenhouse microclimate", *Automation Panorama*, vol. 2010, no. 2, pp. 58 - 64, 2010. (Chinese)

[2] N. Bennis, J. Duplaixb, G. Enéab, M. Halouac and H. Youlal, "Greenhouse climate modelling and robust control", *Computers and electronics in agriculture*, vol. 61, no. 2, pp. 96-107, 2008.

[3] J. P. Coelho, P. B. de Moura Oliveira and J. B. Cunha, "Greenhouse air temperature predictive control using the particle swarm optimisation algorithm", *Computers and Electronics in Agriculture*, vol. 49, no. 3, pp. 330-344, 2005.

[4] R. Pahuja, H. K. Verma and M. Uddin, "A Wireless Sensor Network for Greenhouse Climate Control", *IEEE Pervasive Computing*, vol. 12, no. 2, pp. 49-58, 2013.

[5] B. Gu, V. S. Sheng, K. Y. Tay, W. Romano and S. Li, "Incremental Support Vector Learning for Ordinal Regression", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no.7, pp. 1403-1416, 2015.

[6] B. Gu, V. S. Sheng, Z. Wang, D. Ho, S. Osman and S. Li, "Incremental learning for v-Support Vector Regression", *Neural Networks*, vol. 67, pp. 140-150, 2015.

[7] G.-B. Huang, Q.-Y. Zhu and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks", *IEEE Proceedings of International Joint Conference on Neural Networks*, pp.985-990, 2004.

[8] G.-B. Huang and C.-K. Siew, "Extreme learning machine with randomly assigned RBF kernels", *International Journal of Information Technology*, vol. 11, no. 1, pp. 16-24, 2005.

[9] S. L. Patil, H. J. Tantau and V. M. Salokhe, "Modelling of tropical greenhouse temperature by auto regressive and neural network models", *Biosystems Engineering*, vol. 99, no. 3, pp. 423-431, 2008.

[10] I. Seginer, T. Boulard and B. J.Bailey, "Neural network models of the greenhouse climate", *Journal of Agricultural Engineering Research*, vol. 59, no. 3, pp. 203-216, 1994.

[11] J. B. Cunha, C. Couto and A. E. Ruano, "Real-time parameter estimation of dynamic temperature models for greenhouse environmental control", *Control Engineering Practice*, vol. 5, no. 10, pp. 1473-1481, 1997.

[12] B. T. Tien and G. Van Straten, "A Neuro-Fuzzy approach to identify lettuce growth and greenhouse climate", *Artificial Intelligence Review*, vol. 12, no. 1, pp. 71-93, 1998.

[13] X. Wang, W. Ding, W. Luo and J. Dai, "Simulation and analysis of micro-climate of gutter connected Venlo greenhouse during rainy season in Jianghuai region of China using BP neural network", *Transactions of the Chinese Society of Agricultural Engineering*, vol. 20, no. 2, pp. 235-238, 2004. (Chinese)

[14] P. M. Ferreira, E. A. Faria and A. E. Ruano, "Neural network models in greenhouse air temperature prediction", *Neurocomputing*, vol. 43, no. 1, pp. 51-75, 2002.

[15] F. Fourati, "Multiple neural control of a greenhouse", *Neurocomputing*, vol. 139, pp. 138-144, 2014.

[16] D. Wang, M. Wang and X Qiao, "Support vector machines regression and modeling of greenhouse environment", *Computers and electronics in agriculture*, vol. 66, no. 1, pp. 46-52, 2009.

[17] F. He and C. Ma, "Genetic algorithm to optimize neural network model in the application of solar greenhouse humidity forecast", *China agriculture bulletin*, vol. 24, no. 1, pp. 492-495, 2008. (Chinese)

[18] G.-B. Huang, H. Zhou, X. Ding and R. Zhang, "Extreme learning machine for regression and multiclass classification", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513-529, 2012.

[19] G.-B. Huang, D.-H. Wang and Y. Lan, "Extreme learning machines: a survey", *International Journal of Machine Learning and Cybernetics*, vol. 2, no. 2, pp. 107-122, 2011.

Qi Liu (M'11) received his BSc degree in Computer Science and Technology from Zhuzhou Institute of Technology, China in 2003, and his MSc and PhD in Data Telecommunications and Networks from the University of Salford, UK in 2006 and 2010. His research interests include context awareness, data communication in MANET and WSN, and smart grid. His recent research work focuses on intelligent agriculture and meteorological observation systems based on WSN.

Dandan Jin received her bachelor's degree in Software Engineering from Nanjing University of Information, Science and Technology in 2015, and she will continue to pursue a master's degree in computer science and technology at the Nanjing University of Information Science and Technology. Her research interests include smart grid and greenhouse microclimate modeling.

Jian Shen received his bachelor's degree in Electronic Science and Technology Specialty from Nanjing University of Information, Science and Technology in 2007, and he received his masters and PhD in Information and communication from CHOSUN University, South Korean in 2009 and 2012. His research interests includes Computer network security, information security, mobile computing and network, wireless ad hoc network.

Zhangjie Fu received his BS in education technology from Xinyang Normal University, China, in 2006; received his MS in education technology from the College of Physics and Microelectronics Science, Hunan University, China, in 2008; obtained his PhD in computer science from the College of Computer, Hunan University, China, in 2012. Currently, he works as an assistant professor in College of Computer and Software, Nanjing University of Information Science and Technology, China. His research interests include cloud computing, digital forensics, network and information security.

Nigel Linge received his BSc degree in Electronics from the University of Salford, UK in 1983, and his PhD in Computer Networks from the University of Salford, UK, in 1987. He was promoted to Professor of Telecommunications at the University of Salford, UK in 1997. His research interests include location based and context aware information systems, protocols, mobile systems and applications of networking technology in areas such as energy and building monitoring.