Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 18(3): 2496–2507.
DOI: 10.3934/mbe.2021127
Received: 04 December 2020
Accepted: 02 March 2021
Published: 15 March 2021

Research article

Prediction of cooling moisture content after cut tobacco drying process based on a particle swarm optimization-extreme learning machine algorithm

Ming Zhu[1,†], Kai Wu[2,†], Yuanzhen Zhou[1], Zeyu Wang[1], Junfeng Qiao[2], Yong Wang[2], Xing Fan[2], Yonghong Nong[2] and Wenhua Zi[2,*]

[1] Honghe Cigarette Factory, Hongyunhonghe Tobacco Group Co., Ltd., Honghe 652300, China
[2] College of Energy and Environment Science, Yunnan Normal University, Kunming 650500, China

† These authors contributed to this work equally.

* **Correspondence:** Email: kgclxwh@163.com.

Abstract: The stability of the moisture content of the cigarette is an important index to evaluate the quality of the cigarette. The cooling moisture content after cut tobacco drying process is a key factor affecting the stability of the moisture content of the cigarette. In order to realize its accurate prediction and ensure the stability, in Honghe cigarette factory, a cooling moisture content prediction model is built based on a particle swarm optimization-extreme learning machine (PSO-ELM) algorithm via the historical production data. Besides, the proposed PSO-ELM algorithm is also compared with multiple linear regression (MLR), support vector machine (SVM) and the traditional extreme learning machine (ELM) algorithms in the same data set on the prediction. The prediction accuracy of PSO-ELM method is the highest and the average error of the prediction standard is the lowest. The results indicated the proposed method can achieve a better prediction performance over compared methods and it provides a new method to realize the prediction of the cooling moisture content after cut tobacco drying process.

Keywords: PSO-ELM; cooling moisture content; export moisture content; environment temperature; environment humidity

1. Introduction

Cut tobacco drying and cooling are two important part of cigarette producing process [1]. During the tobacco primary processing, moisture control plays an important role on cigarette quality. The cooling moisture content of the cut tobacco after drying will directly affect final cigarette quality. The stability of the moisture content of the cigarette is an important index to evaluate the quality of the cigarette. Some foreign scholars have studied how different moisture contents affect the smoking quality of cigarettes and summarized a reasonable moisture content of finished product of tobacco. In actual production, the moisture content of finished product of cut tobacco is controlled in a reasonable range of moisture content. Whether the moisture content of finished product of cut tobacco can be controlled stably or not, the control of the cooling moisture content of the cut tobacco is crucial. Usually, the cooling moisture content is depended on the export moisture content after drying. Therefore, how to set a previous export moisture content is of great importance.

Nowadays the export moisture content of the cut tobacco is adjusted by artificial experience, which is subjective, fatigable and unstable. In Honghe cigarette factory, the time that cut tobacco transfers from drying process to cooling process is 270 seconds and the final cooling moisture content is also affected by the environment temperature and humidity. However, there is still no related research on building the mentioned model so far. As the result, how to establish the relationship between the export moisture content, environment temperature, environment humidity and the cooling moisture content to improve the stability of the cooling moisture content has the great practical value.

Huang has proposed a machine learning method-the extreme learning machine (ELM) [2], which has been widely used in classification [3], regression [4], clustering [5], and feature learning [6]. Comparing with the traditional neural networks and support vector machine (SVM) algorithm [7], ELM is composed of single hidden layer feed-forward neural networks (SLFNs) and has faster learning speed and better generalization [8]. One significant achievement made in those years is to successfully prove the universal approximation and regression capabilities of ELM in theory [9–13]. In ELM, the input weights and hidden layer biases are randomly assigned. However, ELM tends to require more hidden neurons than traditional tuning-based algorithms in many cases, which may lead ELM to respond slowly to unknown data [14,15].

Particle Swarm Optimization (PSO) algorithm is a species of the intelligent optimization algorithm [16–18]. It has not only the merits of the traditional optimization algorithm, which adopts the global search strategy, but also avoids the complex genetic operation and enhances the learning and competition between the particles. At the same time, it possesses excellent global search ability. Therefore, PSO has no complicated operators as evolutionary algorithms and it has less parameters which need to be adjusted. It is very suitable for engineering practice. In this work, the PSO algorithm will be used in combination with ELM to improve the generalization capacity of the SLFNs. We will show that this strategy promises excellent performance and produces compact and well-conditioned SLFN than other ELM approaches.

Considering the above challenges, in this paper, we propose a PSO-ELM export moisture content prediction method. The PSO-ELM method optimizes the input weights and hidden biases according to the root mean square error (RMSE) on the validation set instead of random weight generation. The structure of this paper is as follows. Section 2 introduces the description of production process and data pre-processing. The proposed PSO-ELM is presented in section 3. Section 4 shows the experimental results and confirms the effectiveness of the proposed algorithm, and the article is concluded in section 5.

2. Description of production process and data pre-processing

2.1. Description of production process

The description of production process is shown in Figure 1. In Honghe cigarette factory, the time that cut tobacco transfers from drying process to cooling process is 270 seconds. Usually, the cooling moisture content is not only affected by export moisture content after drying but also by the environment temperature and humidity. Therefore, two moisture meters are installed to record the moisture content of the drying export and cooling export respectively. Besides, the environment temperature and humidity are measured in real time.

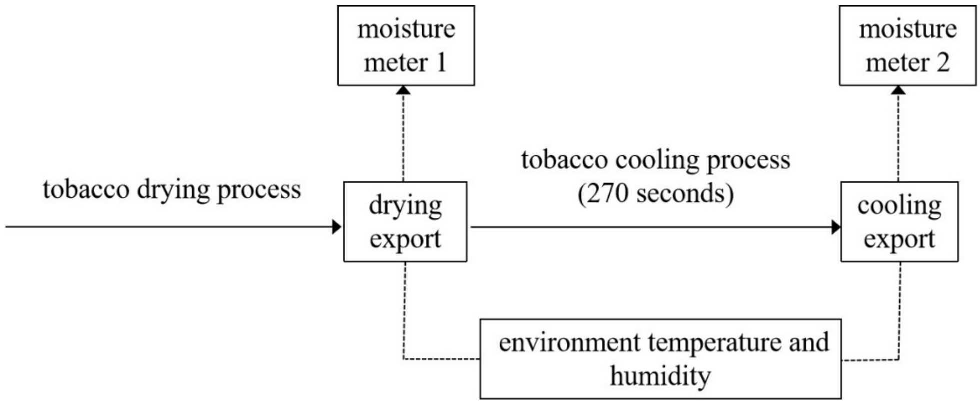


Figure 1. Description of production process.

2.2. Data pre-processing

The experimental data is provided by Honghe cigarette factory. All the samples are collected by manufacturing execution system of the factory in order to ensure they are on the same situation, the same method of cutting and also the same size. It contains HCACS, HBSRY and HBSYH three production brands. The total samples of the three brands are 11,862, 3605 and 6247 respectively. The detail of the experimental data is shown in Table 1.

Table 1. Details of date set 1 and date set 2.

Production Brand	Time	Samples	Training samples	Testing samples
HCACS	2019.5–2019.6	11,862	9877	1985
HBSRY	2019.5–2019.6	3605	3005	600
HBSYH	2019.5–2019.6	6247	5000	1247

2.2.1. Excluding data outliers

Firstly, the data should be cleaned before building the models. It mainly contains excluding null values and data outliers. 3δ rule and Grubbs Criterion are often used to exclude the data outliers. Usually, Grubbs Criterion is chosen when the size of the samples is small and rule is chosen when the size of the samples is big. As the result, 3δ rule is adopted to exclude the data outliers considering the size of the samples of the paper [19]. The rule can be described as Eq (1).

$$\delta = \sqrt{(\sum_{i=1}^{N} V_i^2) / (N-1)} \tag{1}$$

where, $V_i = x_i - (\sum_{i=1}^{N} x_i^2) / N$ is the residual of the variable x_i. If the residual of the data is greater than 3δ, it will be regarded as the outlier and excluded. The operation should be implemented multiple times until there is no outliers in the data.

2.2.2. Data standardization

As the magnitude features and the dimension of the data are inconsistent, the data normalization operation should also be implemented before building the models. The normalization formula is shown as Eq (2).

$$\tilde{x} = \frac{x - \max(x)}{\max(x) - \min(x)} \tag{2}$$

All the data will be scaled in the 0–1 interval after normalization. Then, the data are used to build the prediction models.

2.3. Evaluation indexes

2.3.1. Model building evaluation indexes

Coefficient of determination (R^2), root mean square error (RMSE), mean squared error (MSE) and mean absolute error (MAE) are four common evaluation indexes that are used for evaluate the performance of a regression model. R^2 is a key output of regression analysis. It is interpreted as the proportion of the variance in the dependent variable that is predictable from the independent variable. It ranges from 0 to 1. The value of R^2 indicates that the model is a good or bad fit for the data. If the value is close to 1, it means the model is a good fit for the data. The use of RMSE is very common and it makes an excellent general purpose error metric for numerical predictions. Compared to the similar MSE and MAE, RMSE amplifies and severely punishes large errors. As in Honghe cigarette factory production, the error of cooling moisture content is controlled in 0.5%. It is necessary to punish large error that the cooling moisture content is larger than 0.5%. Therefore, the quality of different building models is evaluated based on the coefficient of determination (R^2) and root mean square error (RMSE) with Eqs (3) and (4), respectively.

$$R^2 = (1 - RSS / SS) \tag{3}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(f(x_i)-t_i\right)^2} \tag{4}$$

RSS and SS represent the residual sums of squares of the calibration data prediction of the final model and the variance of the response variable respectively. $f(x_i)$ is the prediction value and t_i is the actual value.

2.3.2. Prediction evaluation indexes

Mean absolute error (MAE) and mean absolute percentage error (MAPE) are often used to evaluate the performance of the machine learning algorithms. However, MAE can only achieve an evaluation value but it cannot reflect the model is good or bad. MAPE can not only consider the error of the prediction value and true value, but also reflect the proportion of the error and the true value. As the result, MPAE is a very important evaluation index of representing the prediction accuracy. Besides, as the cooling moisture content prediction is a true value prediction issue, mean absolute percentage error (MAPE) and accurate proportion (AP) are chosen as the prediction evaluation indexes in order to guarantee a small enough error between the actual value and prediction value [20]. Here, MAPE is the mean or average of the absolute percentage errors of forecasts. In actual factory production, the error of cooling moisture content is controlled in 0.5%. As the result, AP defines the accurate proportion that the MAPE of prediction value and actual value is smaller than 0.5%. MAPE of AP are defined as follows Eqs (5) and (6).

$$MAPE = \frac{1}{n}\sum_{i=0}^{n}\left|f(x_i)-t_i\right| \tag{5}$$

$$AP = \frac{N(\left|f(x_i)-t_i\right| < 0.5\%)}{n}\times 100\% \tag{6}$$

In Eqs (5) and (6), $f(x_i)$ is the prediction value and t_i is the actual value. $N(\left|f(x_i)-t_i\right| < 0.5\%)$ is the number that the error between the prediction value and actual value is smaller than 0.5%.

3. Methods

3.1. ELM algorithm

ELM was given to such models by its principal inventor Huang [2]. ELM is formulated as a linear-in-the-parameter model that boils down to solve a linear system. Compared with traditional feed-forward neural network (FNN) learning methods, ELM is remarkably efficient and tends to reach a global optimum. Theoretical studies show that even with randomly generated hidden nodes, ELM maintains the universal approximation capability of SLFNs [3,4].

Given a single hidden layer of ELM, suppose that the output function of the *i*-th hidden node is $h_i(\mathbf{x}) = G(a_i, b_i, \mathbf{x})$, where a_i and b_i are the parameters of the *i*-th hidden node. The output function of the ELM for SLFNs with *L* hidden nodes is Eq (7).

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \beta_i h_i(\mathbf{x}) \tag{7}$$

In Eq (7), β_i is the output weight of the *i*-th hidden node. $\mathbf{h}(\mathbf{x}) = [G(h_i(\mathbf{x}), ..., h_L(\mathbf{x}))]$ is the hidden layer output mapping of ELM. Given *N* training samples, the hidden layer output matrix **H** of ELM is given as Eq (8).

$$\mathbf{T} = \begin{bmatrix} \mathrm{t}_1 \\ . \\ . \\ . \\ \mathrm{t}_\mathrm{N} \end{bmatrix} \tag{8}$$

The training data-target matrix *T* is given as Eq (9):

$$\mathbf{H} = \begin{bmatrix} \mathrm{h}(\mathrm{x}_1) \\ . \\ . \\ . \\ \mathrm{h}(\mathrm{x}_\mathrm{N}) \end{bmatrix} = \begin{bmatrix} G(a_1, b_1, x_1)...G(a_L, b_L, x_1) \\ ... \\ ... \\ ... \\ G(a_1, b_1, x_N)...G(a_L, b_L, x_N) \end{bmatrix} \tag{9}$$

Generally speaking, ELM is a kind of regularization neural networks with non-tuned hidden layer mappings (formed by either random hidden nodes, kernels or other implementations), its objective function is:

$$\text{Minimize:} \left\|\beta\right\|_p^{\sigma_1} + C\left\|\mathbf{H}\beta - \mathbf{T}\right\|_q^{\sigma_2} \tag{10}$$

where $\sigma_1 > 0$, $\sigma_2 > 0$ and $p, q = 0, \frac{1}{2}, 1, 2, ..., +\infty$.

Different combinations of σ_1, σ_2, *p* and *q* can be used and result in different learning algorithms for regression, classification, sparse coding, compression, feature learning and clustering.

As a particular case, a most straightforward training algorithm learns a model of the form (for single hidden layer sigmoid neural networks):

$$\hat{\mathbf{Y}} = \mathbf{W}_2 \sigma(\mathbf{W}_1 x) \tag{11}$$

where $\mathbf{W}_1$ is the matrix of input-to-hidden-layer weights, σ is an activation function, and $\mathbf{W}_2$ is the matrix of hidden-to-output-layer weights. The algorithm proceeds as follows:

(i) Fill $\mathbf{W}_1$ with random values, such as Gaussian random noise;

(ⅱ) Estimate $\mathbf{W}_2$ by least-squares fit to an array of response variables Y, computed using the pseudoinverse, given a design matrix X: $\mathbf{W}_2 = \sigma(\mathbf{W}_1\mathbf{X})^{+}\mathbf{Y}$.

3.2. Theory of particle swarm optimization algorithm

In computer science, PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality [18]. Here PSO algorithm uses particles moving in an m-dimensional space to search solutions of an optimization problem with *m* variables. In our approach, PSO is used to search for the optimal particle iteratively. Each particle represents a candidate solution. ELM classifier is built for each candidate solution to evaluate its performance. The velocity and position of particles can be updated by the Eq (12).

$$v_{ij}^{(t+1)} = \omega v_{ij}^{\ t} + c_1 rand_1 (pbest^t_{\ ij} - x_{ij}^{\ t}) + c_2 rand_2 (gbest^t_{\ ij} - x_{ij}^{\ t}) x_{ij}^{\ t+1} = x_{ij}^t + v_{ij}^{t+1} \tag{12}$$

Where t is an evolutionary generation, v_{ij} and x_{ij} stand for the velocity and position of particle i on dimension j, respectively. ω is the inertia weight and it is used to balance the global exploration and local exploitation. Rand represents the random function, c_1 is the personal learning factor and c_2 is the social learning factor.

PSO algorithm and Genetic algorithm (GA) are two common algorithms to optimize the parameters of the machine learning algorithms. Comparing with GA algorithm, PSO algorithm has the advantages of fast searching speed, less parameters adjusting, simple structure and it is more suitable for engineering practice. Therefore, PSO is adopted for parameter optimization of ELM and SVM in the following part.

3.3. Theory of particle swarm optimization-extreme learning machine algorithm

Figure 2 is the flow chart of building a cooling moisture content prediction model by using PSO-ELM algorithm, followed by data pre-processing, parameter optimization of ELM by PSO, PSO-ELM prediction.

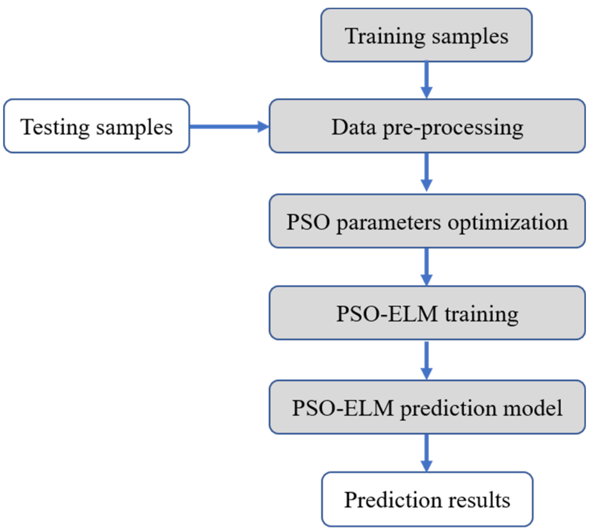


Figure 2. Flow chart of building prediction model by using PSO-ELM algorithm.

4. Experimental results and discussions

4.1. Prediction modeling built

In the following research, all the experiments are performed on an Intel machine (Core TM i5-4590s, 3.00 GHz, CPU with 8 GB RAM, with 64-bit Windows 7 Professional operation system). As one kind of powerful and simple engineering calculation software, Matlab is widely used in automatic control, mechanical design, fluid mechanics, mathematical statistics and other engineering fields. Engineering and technical personnel can solve complex engineering problems efficiently, simulate the system dynamically and achieve the numerical calculation results with powerful graphic functions by means of using the toolbox provided by Matlab. Matlab is more suitable for engineering practice comparing with C++, python, R. Therefore, all methods are implemented in the language MATLAB, 64-bit version 2010b.

Firstly, the parameters used in the PSO are defined as follows. The coefficients c_1 and c_2 were both set to 2.0 and the adaptive inertia is used where the initial inertia is 0.9 and the end inertia is 0.4. The PSO is executed for 100 iterations. All components are limited within the range [−1,1]. Figure 3 shows the evolution of the RMSE along the iterations and we can observe the behavior of the PSO in the figure. It can be seen from Figure 3 that there is an optimal value for the number of neurons in the hidden layer of ELM. In order to calculate the optimal number of neurons automatically in the actual production process, PSO is implemented according to the section 3.2. The optimization interval is [0,100] and the PSO reaches the best RMSE values for three different testing brands after 60 iterations.

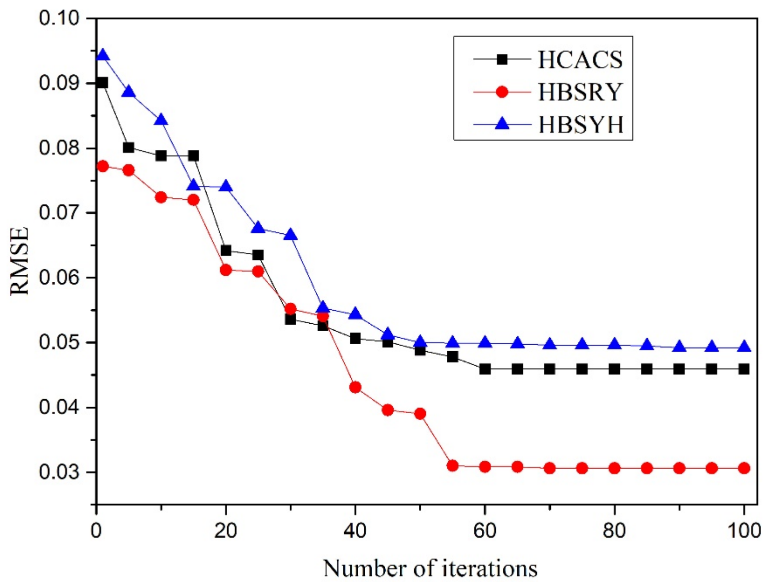


Figure 3. PSO-ELM fitness function change process.

All the training data of the three different brands (HCACS, HBSRY, HBSYH) are cleaned and normalized by using Eqs (1) and (2). MLR and SVM algorithms are two traditional regression algorithms which are widely used in all kinds of areas. As the result, it is very typical to compare the results of PSO-ELM algorithm with MLR and SVM algorithms. Besides, the paper use PSO algorithm to optimize the parameters of ELM algorithm, so it is also necessary to compare the results of PSO-

ELM algorithm with traditional ELM algorithm. Then, four different regression algorithms were built and compared: MLR [21], PSO-SVR [22] and ELM and PSO-ELM. The SVM algorithm has two very important parameters punishment coefficient and Gamma. If the punishment coefficient is too large, the model is easy to occur overfit. If the punishment coefficient is too small, the model is easy to occur underfit. The value of Gamma determines the number of support vectors. The number of support vectors affects the speed of training and prediction. In order to make a fair comparison, the parameters of SVM algorithm are also optimized by using PSO algorithm. Here, the kernel function of SVM algorithm is radial basis function (RBF), and the punishment coefficient is 42.14 and Gamma is 0.02 after PSO optimization. Here the parameters of PSO algorithm are the same with the above section. For ELM algorithm, the number of hidden neurons is 20, and the transfer function is Sigmoidal function. The statistics of established models in calibration set is shown in Table 2. For all the models, the calibration results of the PSO-ELM model were compared with that of the MLR, SVR and ELM models. PSO-ELM models had much higher R^2 values and lower RMSE values. The results show PSO algorithm can optimize the parameters effectively, in order to improve the accurate of the ELM model.

Table 2. Statistics of established models in calibration set.

Brand	Evaluation index	Method			
		MLR	SVR	ELM	PSO-ELM
HCACS	R^2	0.8617	0.8619	0.9173	0.9517
	RMSE	0.0801	0.0788	0.0619	0.0459
HBSRY	R^2	0.6356	0.7019	0.8342	0.8846
	RMSE	0.0459	0.0416	0.0328	0.0306
HBSYH	R^2	0.7574	0.8416	0.8813	0.9372
	RMSE	0.0844	0.0674	0.0595	0.0492

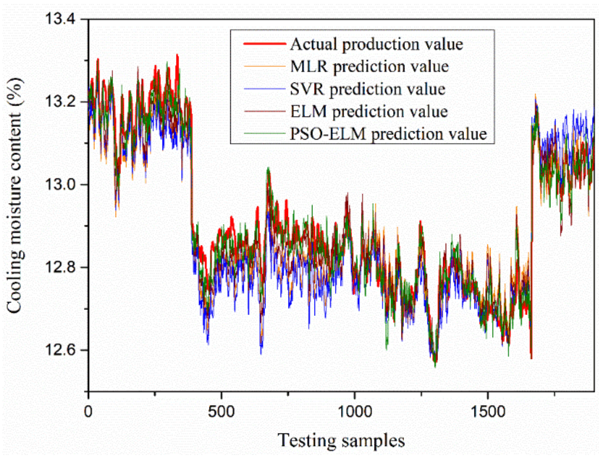
4.2. Prediction results and discussion

The prediction performance comparison results of PSO-ELM and the traditional methods could be obtained and listed in Table 3. It can be clearly seen that: 1) the MAPE of PSO-ELM method is the lowest among all the four methods for each testing brand; 2) the AP of PSO-ELM method is the highest among all the four methods for each testing brand. The results indicate that the proposed PSO-ELM method outperformed the traditional methods. It should be noted that the optimal results used for comparison are all obtained on the same data set for all the comparison methods and all the results are obtained through fine parameters tuning.

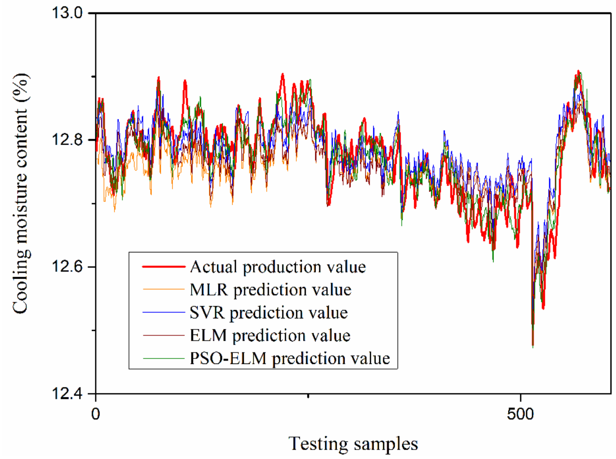
In Figure 4, it can be easily seen that the prediction results of PSO-ELM are better than those of the other MLR, SVR and traditional ELM methods. The prediction values are the closest to the actual production values for all the three different brands (HCACS, HBSRY and HBSYH). Besides, the PSO-ELM algorithm has the better prediction accuracy than the traditional ELM algorithm after parameters optimization. ELM can find the optimal number of neurons after PSO algorithm optimizes the number of extreme learning neurons. As the result, PSO-ELM algorithm reaches the highest prediction accuracy and the prediction error is reduced to the minimum. Due to the computational rapidity and strong generalization ability of extreme learning, ELM method can realize the real-time prediction of production and ensure the optimal model all the time. Its performance and feasibility are much higher than those of MLR and SVM methods and it is very suitable for engineering practice.

Table 3. Prediction accuracy and error of different methods.

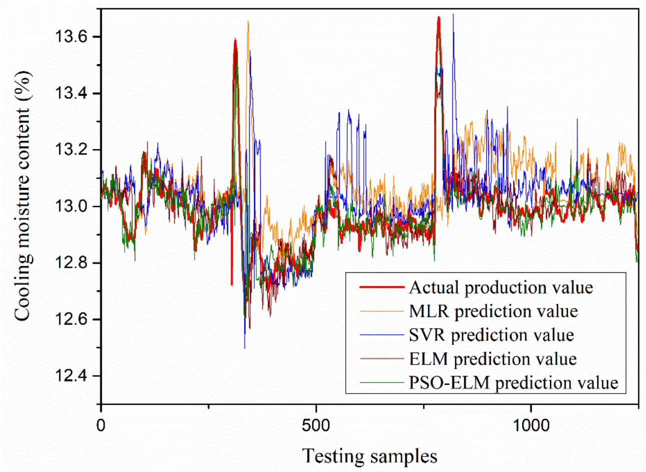
Brand	Evaluation index	Method			
		MLR	SVR	ELM	PSO-ELM
HCACS	MAPE	0.0571	0.0651	0.0401	0.0259
	AP	62.06%	53.9%	79.3%	92.99%
HBSRY	MAPE	0.0417	0.0327	0.0282	0.0126
	AP	78.02%	83.47%	86.16%	94.72%
HBSYH	MAPE	0.1219	0.0919	0.0491	0.0348
	AP	34.72%	50.84	72.89%	87.49%



(a)



(b)



(c)

Figure 4. Comparison results of different prediction methods of three different testing brands. (a) is the result of the brand of HCACS, (b) is the result of the brand of HBSRY, (c) is the result of the brand of HBSYH.

5. Conclusions

The paper uses PSO algorithm to optimize the parameters of ELM algorithm and builds a novel

PSO-ELM model to predict the cooling moisture content after cut tobacco drying process. The experimental results show that PSO-ELM algorithm works better than MLR, SVR and traditional ELM algorithms on the prediction accuracy and generalization ability. In addition, the fast calculation speed of the ELM algorithm makes the factory easy to build and update the online model. In the following research, the online update learning method of ELM will also be studied.

All the calculated results in the paper are all from the actual production data of the Honghe cigarette factory. It is of great importance to improve the quality of the cigarette, enhance the economic benefit of enterprises and reduce the unnecessary investment in physical test. Besides, the method proposed by the paper can also be used in the other cigarette factories. However, the new model should be built if the method is adopted in the other cigarette factories according to the collected data of themselves. That is the work that is planned in the future.

Acknowledgments

Thanks Dr. Zhang Jianqiang (Yunnan Police College) for the help and advice of data analysis. This work is financially supported by Hongyun Honghe Tobacco (Group) Co., Ltd (HYHH2018GY04). The authors would like to thank the Honghe cigarette factory for providing the experimental samples.

Conflict of interest

The authors declare that they have no conflicts of interest.

References

1. C. L. Yuan, W. Yi, Y. Bin, Relationship between Quality of dried cut tobacco and cooling temperature, *Tob. Sci. Technol.*, **36** (2003), 9–12.
2. G. B. Huang, Q. Y. Zhu, C. K. Siew, Extreme learning machine: theory and applications, *Neurocomputing*, **70** (2006), 489–501.
3. G. B. Huang, H. M. Zhou, X. J. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, *IEEE Trans. Syst. Man. Cybern. B*, **42** (2010), 513–529.
4. M. V. Heeswijk, Y. Miche, E. Oja, A. Lendasse, GPU-accelerated and parallelized ELM ensembles for large-scale regression, *Neurocomputing*, **74** (2011), 2430–2437.
5. Y. Miche, A. Akusok, D. Veganzones, K. M. Björk, E. Séverin, P. Du Jardin, et al., SOM-ELM-Self-Organized Clustering using ELM, *Neurocomputing*, **165** (2015), 238–254.
6. Y. Jin, J. Li, C.Y. Lang, Q. Ruan, Multi-task clustering ELM for VIS-NIR cross-modal feature learning, *Multidimens. Syst. Signal Process.*, **28** (2017), 905–920.
7. H. Zhong, C. Miao, Z. Shen, Y. Feng, Comparing the learning effectiveness of BP, ELM, I-ELM, and SVM for corporate credit ratings, *Neurocomputing*, **128** (2014), 285–295.
8. G. B. Huang, L. Chen, C. K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes, *IEEE Trans. Neural Network*s, **174** (2006), 879–892.
9. G. B. Huang, C. Lei, Convex incremental extreme learning machine, *Neurocomputing*, **70** (2007), 3056–3062.

10. G. B. Huang, C. Lei, Enhanced random search based incremental extreme learning machine, *Neurocomputing*, **71** (2008), 3460–3468.
11. X. M. Wang, X. H. Wan, Y. Y. Zhu, Z. L. Jiang, J. X. Liu, Prediction for Building Vibration Velocity Caused by Blasting Based on PSO-ELM, *Sci. Technol. Rev.*, **32** (2014), 15–20.
12. A. Banan, A. Nasiri, A. Taheri-Garavand, Deep learning-based appearance features extraction for automated carp species identification, *Aquacult. Eng.*, **89** (2020), 102053.
13. R. Taormina, K.W. Chau, ANN-based interval forecasting of streamflow discharges using the LUBE method and MOFIPS, *Eng. Appl. Artif. Intel.*, **45** (2015), 429–440.
14. S. F. Ardabili, B. Najafi, S. Shamshirband, B. M. Bidgoli, R. C. Deo, K. W. Chau, Computational intelligence approach for modeling hydrogen production: a review, *Eng. Appl. Comput. Fluid*, **12** (2018), 438–458.
15. C. L. Wu, K. W. Chau, Prediction of rainfall time series using modular soft computing methods, *Eng. Appl. Artif. Intel.*, **26** (2013), 997–1007.
16. C. Cheng, W. Niu, Z. Feng, J. Shen, K. Chau, Daily reservoir runoff forecasting method using artificial neural network based on quantum-behaved particle swarm optimization, *Water*, **7** (2015), 4232–4246.
17. R. Taormina, K. W. Chau, Data-driven input variable selection for rainfall-runoff modeling using binary-coded particle swarm optimization and Extreme Learning Machines, *J. Hydrol*., **529** (2015), 1617–1632.
18. H. Liang, J. Zou, Z. Li, M. J. Khan, Y. Lu, Dynamic evaluation of drilling leakage risk based on fuzzy theory and PSO-SVR algorithm, *Future Gener. Comput. Syst.*, **95** (2019), 454–466.
19. C. Shang, X. Huang, F. You, Data-driven robust optimization based on kernel learning, *Comput. Chem. Eng*., **106** (2017), 464–479.
20. P. Goodwin, R. Lawton, On the asymmetry of the symmetric MAPE, *Int. J. Forecast.*, **15** (1999), 405–408.
21. J. J. Da Costa, F. Chainet, B. Celse, M. Lacoue-Nègre, C. Ruckebusch, D. Espinat, Comparing kriging, spline, and MLR in product properties modelization: application to cloud point prediction, *Energ Fuels*, **32** (2018), 5623–5634.
22. A. Kavousi-Fard, Modeling uncertainty in tidal current forecast using prediction interval-based SVR, *IEEE Trans. Sustainable Energy*, **99** (2016), 1–3.