A Method for Greenhouse Temperature Prediction Based on XGBoost Algorithm and Linear Residual Model

Huijin Han, Chengjie Tong, Yanpeng Ma[*], Meiling Zhao
North China Electric Power University(Baoding), Baoding, CO 071003,China
**Corresponding Author.*

Abstract

Temperature prediction is significant for precise control of the greenhouse environment. Traditional machine learning methods usually rely on a large amount of data. Therefore, it is difficult to make a stable and accurate prediction based on a small amount of data. This paper proposes a temperature prediction method for greenhouses. With the prediction target transformed to the logarithmic difference of temperature inside and outside the greenhouse,the method first uses XGBoost algorithm to make a preliminary prediction. Second, a linear model is used to predict the residuals of the predicted target. The predicted temperature is obtained combining the preliminary prediction and the residuals. Based on the 20-day greenhouse data, the results show that the target transformation applied in our method is better than the others presented in the paper. The MSE (Mean Squared Error) of our method is 0.0844, which is respectively 20.7%, 76.0%, 10.2%, and 95.3% of the MSE of LR (Logistic Regression), SGD (Stochastic Gradient Descent), SVM (Support Vector Machines), and XGBoost algorithm. The results indicate that our method significantly improves the accuracy of the prediction based on the small-scale data.

Keywords:*greenhouse, temperature prediction, machine learning, XGBoost, linear residual model*

I. Introduction

Greenhouse is a typical scene of modern agriculture. By creating a small closed ecosystem beneficial to plant growth, the greenhouse improves the quantity and quality of agricultural products. As the most important environmental parameters of the greenhouse, the indoor temperature is highly coupled with humidity,air pressure, outdoor temperature, and other environmental parameters [1]. Because of the interference of external signals and the sensor itself, the collected time series data are often mixed with noise. As a result, the collected time series data is nonlinear and unstable [2], which is difficult to achieve precise control of the greenhouse environment.

Existing studies mainly use numerical algorithms to fit the numerical characteristics between the greenhouse temperature and the predictable variables of the environment. According to the numerical characteristics, it is able to predict the indoor temperature. Traditional time series analysis, such as principal component regression forecasting method [3], grey forecasting method [4], and autoregressive integrated moving average model (ARIMA) [5], are widely applied in greenhouse temperature prediction. However, these methods can be easily affected by external environmental noise. Besides, the accuracy of these methods is related to the selection of the numerical algorithms and independent variables.

With the development of machine learning, SVM (support vector machines) [6], extreme learning machines [7] artificial neural networks [8-10], and other methods have been applied to greenhouse temperature prediction. Some hybrid methods based on the above methods [11, 12] have also been used for greenhouse temperature prediction. These methods emphasize on unearthing the intrinsic autocorrelation characteristics and capturing different pattern features within the data to reduce the dependence on external parameters. However, the amount of collected training data deeply influences their accuracy, which makes it difficult to maintain the prediction stable and accurate on a small amount of training data.

XGBoost (Extreme Gradient Boosting) is an extended and optimized version of the gradient boosting machine learning algorithm [13] proposed by T. Chen and C. Guestrin [14] in 2016. It improves the convergence speed of the training and prevents the model from overfitting. Additionally, the XGBoost algorithm spends less time adjusting the hyper-parameters. Because of its high computing speed and high accuracy, XGBoost has been widely used in data mining, recommendation systems and other fields.

As a new machine learning algorithm, the XGBoost algorithm has not yet been used to predict indoor temperature in greenhouses. This paper attempts to combine the XGBoost algorithm with residual prediction to propose a greenhouse temperature prediction method, which can maintain accurate on small-scale data. In section II, XGBoost algorithm and linear model are introduced. In the section III, the process of our method is presented. In the section IV, model training and numerical results are given. The conclusion is in the last section.

II. Model

2.1 XGBoostalgorithm

XGBoost is an algorithm based on the GBDT (Gradient Boosting Decision Tree), which consists of multiple decision tree iterations. Its principle is to first establish multiple CART (Classification and Regression Trees) models to predict target value, and then to integrate these trees into a new tree model. This model is improved through continuous iterations. The new tree model generated in each iteration will fit the residuals of the previous tree. As the number of trees increases, the complexity of the ensemble model will gradually increase until it approaches the complexity of the data itself and the results obtained by training are accurate enough. The detailed derivation of the XGBoost model has been given by T. Chen [14] et al.

The XGBoost algorithm model is as follows:

$$\hat{y}_i = \phi(x_i) = \sum_{t=1}^{T} f_t(x_i) \#(1)$$

where$\{\mathrm{f_t(x_i) = w_q(x)}\}$ is the space of CART, $w_q(x)$ is the scoring of sample x, and the predicted value $\hat{y}_i$ of the model is obtained by accumulation, and q denotes the structure of each tree, T is the number of trees. Each f_t corresponds to an independent tree structure q and leaf weight w_q.The internal decision tree of XGBoost uses a regression tree. The iterative process of residual fitting is as follows:

$$\hat{y}_i^{(0)} = 0\#(2)$$

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)\#(3)$$

wherex_i is the input ith sample, $\hat{y}_i^{(t)}$ is the predicted value of the ith sample after t iterations, $\hat{y}_i^{(0)}$is the initial value of the ith sample value.

According to the iterative process of residual error, the objective optimization function of the algorithm$Obj^{(t)}$, namely the loss function, can be obtained.

$$Obj^{(t)} = \sum_{i=1}^{n} l\left(y_i,\ \hat{y}_i^{(t)}\right) + \sum_{i=1}^{t} \Omega(f_t) \#(4)$$

where$l\left(y_i,\ \hat{y}^{(t)}\right)$ is the loss function that measures the degree of similarity between $\hat{y}$ and y, and $\Omega(f)$is the regularization term. $\Omega(f)$represents the complexity of the tree. The smaller the function value, the stronger the generalization ability of the tree.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \#(5)$$

where w_i is the weight on the ^{j}th leaf node in the tree f, T is the total number of leaf nodes in the tree, γ is the penalty item of the L_1 regularity, and λ is the penalty item of the L_2 regularity, which is a custom parameter of the algorithm. $\Omega(f)$is the complexity of the tree. The generalization ability of the tree will be better with the function value decreasing.

In order to train the gradient descent method better, we need to expand the XGBoost's second-order Taylor and remove the constant term of the XGBoost. The loss function at step t is

$$Obj^{(t)} = \sum_{i=1}^{T} \left[l\left(y_i,\ \hat{y}_i^{(t)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_i)\#(6)$$

$$g_i = \partial_{\hat{y}^{(t-1)}} L\left(y_i, \hat{y}^{(t-1)}\right)\#(7)$$

$$h_i = \partial_{\hat{y}^{(t-1)}}^2 L\left(y_i, \hat{y}^{(t-1)}\right)\#(8)$$

whereg_i and h_i are the first and second derivatives, respectively. Therefore, the objective function is

$$Obj = -\frac{1}{2}\sum_{j=1}^{T} \frac{G_j^2}{H_j^2 + \lambda} + \gamma T\#(9)$$

$$G_j = \sum_{i \in I_j} g_i \#(10)$$

$$H_j = \sum_{i \in I_j} h_i \#(11)$$

where$I_j = \{i | q(x) = j\}$ denotes the sample set of the jth leaf node. The detailed derivation of the XGBoost model has been given by T. Chen [14] et al.

It is necessary to split the nodes in the process of building a decision tree. The greedy algorithm enumerates the partitioning schemes, starting from a leaf, and iteratively adding branches to the tree. The difference between the scores before and after splitting is the gain value of the decision tree. The optimal split is the split with the largest gain value. $I_L$andI_R are supposed to create an instance set of left and right nodes after splitting. Assuming $I = I_L \cap I_R$, the divided gain value is given by the following equation. The segmentation with the largest gain value is the optimal segmentation,

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma\#(12)$$

2.2 Linear residual model

The linear model is

$$p(x) = w_1 x_1 + w_2 x_2 + \cdots + w_d x_d + b\#(13)$$

where $p(x)$ is the predicted value of our linear model, $x_1, x_2, ..., x_d$ are our feature data, w_1 , w_2 ,..., w_d are the weights corresponding to the feature data, and b is a constant term. The vector form of the linear model is

$$p(x) = w^T x + b\#(14)$$

where$w$ is the vector format of w_1 , w_2 , ...,w_d . $w$and$b$ are determined by learning the residual of the XGB algorithm.

III. Method

3.1 Forecasting process

The principle of the method is to use a linear model to learn the residuals produced by the XGBoost algorithm. For convenience, we mark the XGBoost algorithm and our method as XGB and XGB-R, respectively. The XGB-R can be divided into the following steps:

(a) Data acquisition and data preprocessing.
(b) Extract, select and construct the data features required for training the XGBoost algorithm.
(c) Train the XGBoost algorithm and predict the indoor temperature $t_{indoor\ \_predict\ \_XGB}$ of the greenhouse.
(d) Train a linear model to predict the residuals $t_{Residual}$ produced by the XGBoost.
(e) Add $t_{Residual}$ to $t_{indoor\ \_predict\ \_XGB}$ to get the predicted temperature $t_{indoor\ \_predict}$ of the XGB-R.
(f) Accuracy evaluation. This article will use the MSE (Mean Square Error) as the index to evaluate the accuracy of the method,

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(t_{indoo\ r_{predict}} - t_{actual}\right)^2 \#(15)$$

where$N$ is the amount of data of indoor temperature in the test set. The smaller the MSE, the better the model fitting effect and the higher the accuracy

3.2 Target transformation

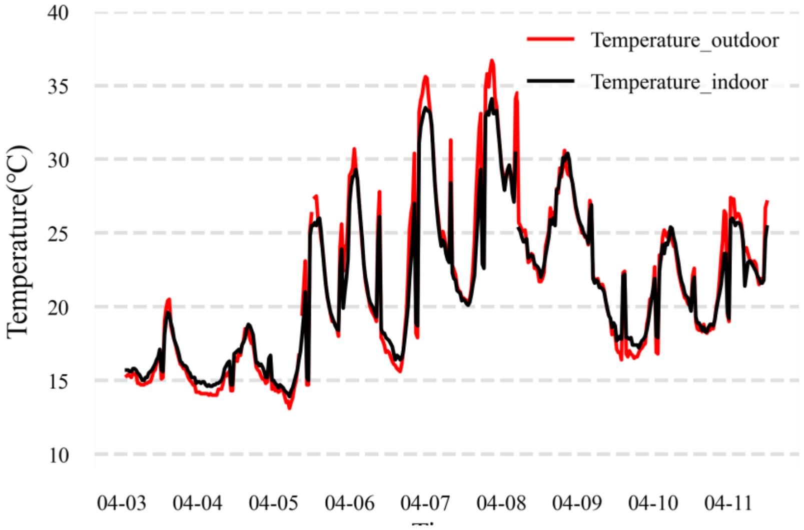

Fig 1: Indoor and outdoor temperature curve of the greenhouse.

Considering the distribution of indoor temperature, it is prone to a situation where the predicted value at certain moments differ greatly from the actual value, which may result in a larger overall MSE. From Fig.1, we observe that the indoor and outdoor temperature changes have similar trends. We try to use the difference between the inside and outside temperatures of the greenhouse to indirectly predict the indoor temperature.

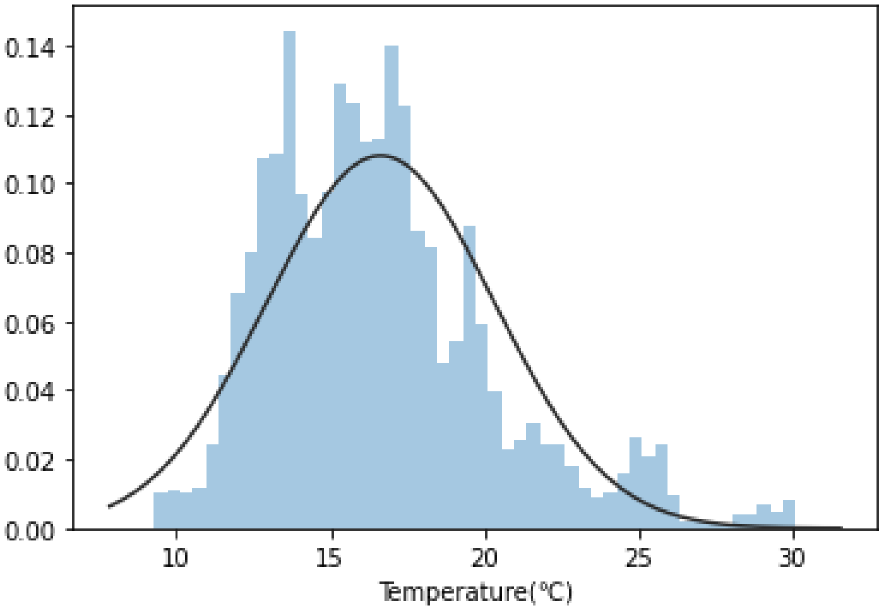

Fig 2: Distribution of indoor temperature.

ISSN: 0010-8189

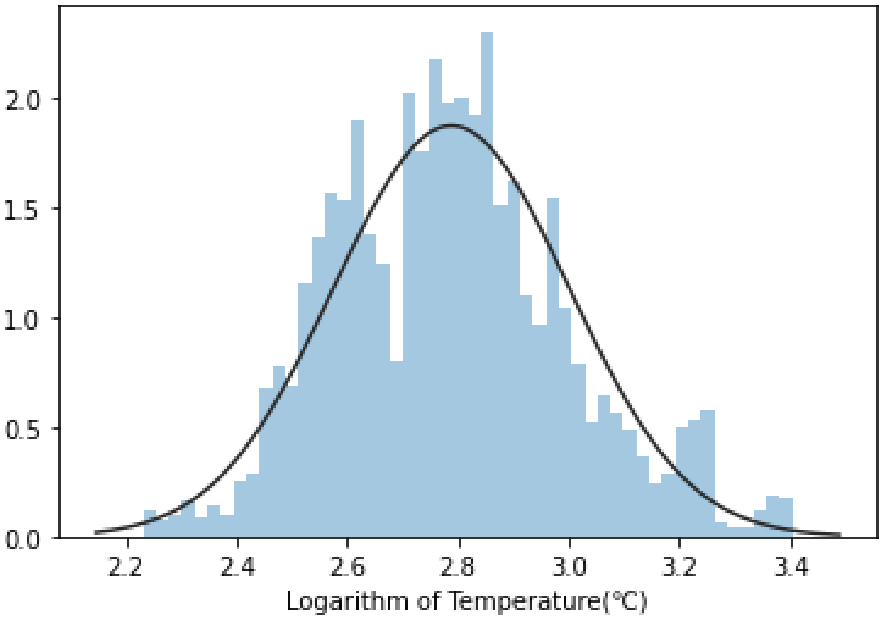

Fig 3: Distribution of the logarithm of the indoor temperature.

Figs. 2 and 3 show that after log transformation, the indoor temperature is closer to the normal distribution.Because MSE has the poor robustness to the skew distribution of indoor temperature, the prediction target of the XGBoost algorithm is transformed to the difference between the logarithmic temperature inside and outside the greenhouse to avoid overfitting, In the following formula, t is the prediction target of our model. After t is obtained, the predicted value $t_{indoo\ r_{predic\ t_{XGB}}}$ of the indoor temperature we need can be obtained by exponential transformation.

$$t = log\, t_{indoor}\ - log\, t_{outdoor}\ \#(16)$$
$$t_{indoo\ r_{predic\ t_{XGB}}}\ = e^{(t+log\,(t_{outdoor}\ ))}\#(17)$$

Therefore, the prediction flow chart of this method is following.

The processed data first uses xgboost to predict the difference between the logarithmic temperature inside and outside the greenhouse. Then restore the predicted result to obtain the predicted indoor temperature. Then a linear model is used to predict the residual difference between the predicted indoor temperature and the real indoor temperature, and finally the two are added to get the prediction result of our model. (Fig.4)

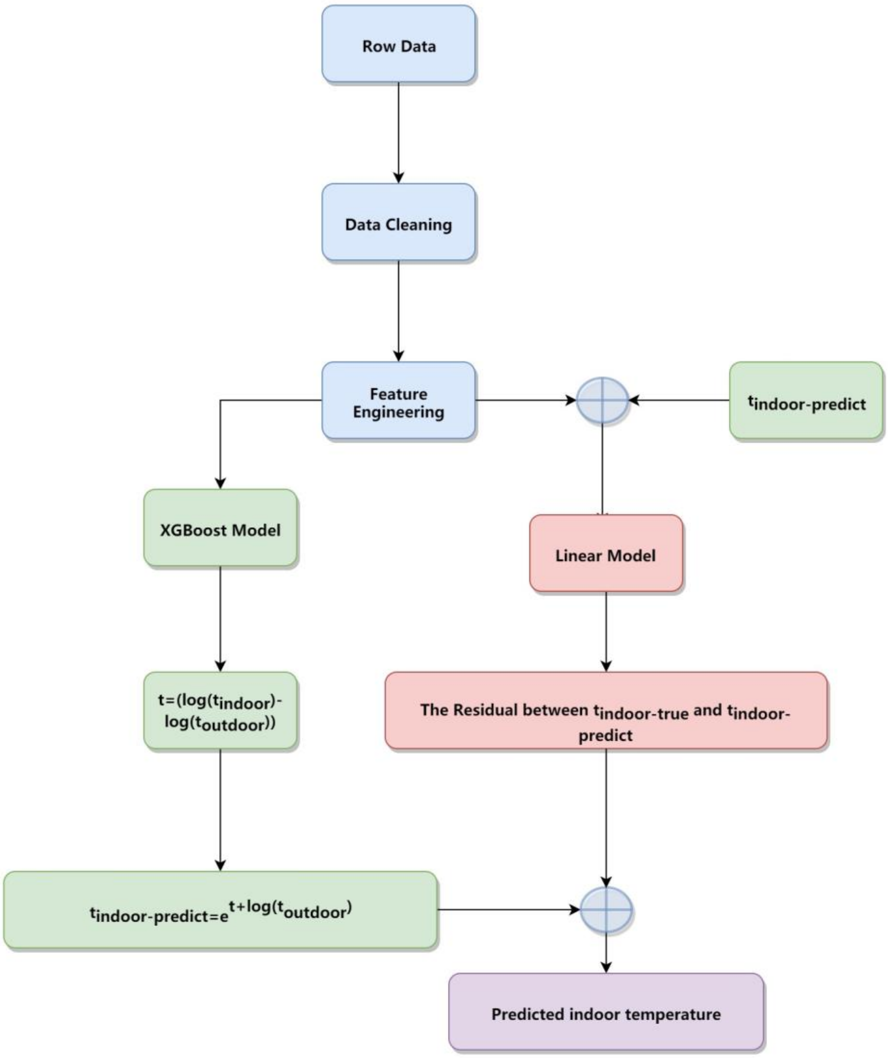
Row Data
Data Cleaning
Feature Engineering
$t_{indoor\text{-}predict}$
XGBoost Model
Linear Model
$t=(\log(t_{indoor})-\log(t_{outdoor}))$
The Residual between $t_{indoor\text{-}true}$ and $t_{indoor\text{-}predict}$
$t_{indoor\text{-}predict}=e^{t+\log(t_{outdoor})}$
Predicted indoor temperature

Fig 4: Process of the XGB-R.

IV.Results

We collect about thirty-day data of a smart greenhouse project, including six types of environmental parameters, including temperature, humidity and air pressure respectively inside and outside the greenhouse. The amount of the data is 12843 rows.

4.1 Data preprocessing

In order to eliminate the influence of duplicate data, abnormal data, and missing data in the original data, it is necessary to preprocess the time series data. The steps are as follows:

(a) Eliminate duplicate data. The repeated time series data is averaged to reduce the data collection error caused by the sensor itself.
(b) Correct the abnormal data. Identify outliers in the data through box plots. By Figs.5 It can be seen that indoor air pressure and outdoor air pressure have many abnormal values. With reference to the opinions of relevant experts, the abnormal value of indoor air pressure is corrected by interpolation. The abnormal value of outdoor air pressure is corrected by exponential average.

(c) Fill in missing values. Fill in the missing values of indoor and outdoor air pressure by filling in backwards.
(d) Fill the data obtained by cleaning to the faults of the test set.

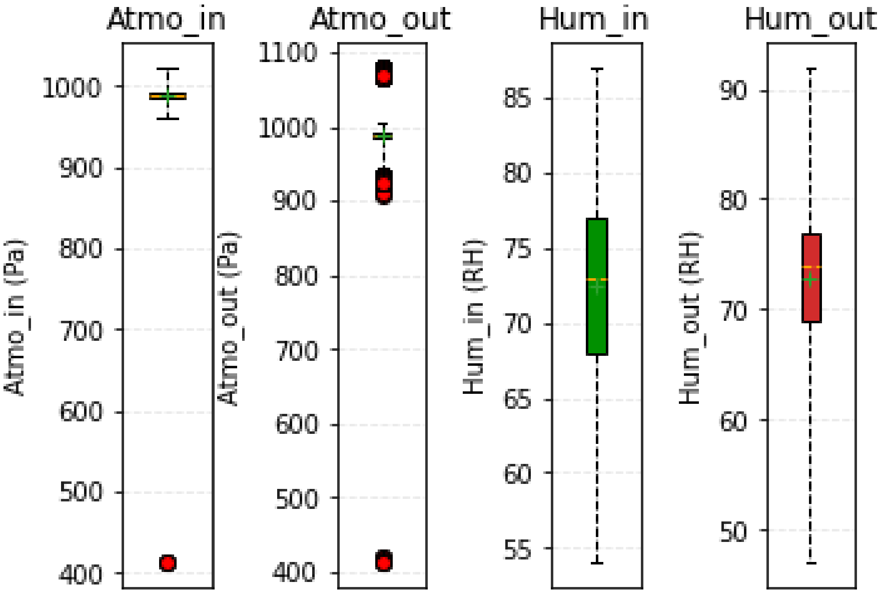

Fig 5: Outlier analysis of the air pressure and the humidity.

4.2 Feature engineering

Since there are only six kinds of time series data in the greenhouse data, the features of the preprocessed data need to be extracted, selected and structured to improve the reliability and generalization of prediction.

First, we extract third-order polynomial features, statistical features, aggregate features, crossover features, and historical information features of temperature, humidity, and air pressure. The historical information feature is the average of the data of the day before the current day. Additionally, the hourly difference over the same period is calculated.Statistical features include:

(a) the average, maximum, minimum, median, standard deviation for each hour and each day;
(b) the ratio between statistical features;
(c) the upper 0.25 quantile value, lower 0.75 quantile value, skewness, kurtosis, and average absolute distance difference of the data from 48, 96, and 144 hours ago.

Second, discretize continuous data. The data is divided into buckets to enhance the generalization of the model. Then, the statistical characteristics of the humidity and the air pressure are constructed.

Third, On the basis of humidity and temperature features, we add discrete features based on them. The greenhouse data is divided into buckets according to hour and day. Statistical features are constructed in the bucket.

Fourth, to reduce the influence of temperature abnormal values on the overall data, we use the 0.05 quantile and 0.95 quantile to replace the min and max of the statistical value, respectively.

Fifth, splice part of the training set data to the test set in order to fill in the complete statistical features. Sample the spliced data of the test set and calculate the median.

4.3 Algorithm parameter optimization

This article uses the XGBoost algorithm package in python, and takes the constructed features as the input. The data samples of the first 20 days are used as the train set, and the data samples of the next 10 days are used as the test set. To optimize the parameters of the XGBoost algorithm, we use the first 16 days of training data as the training set, and the last 4 days of data as the validation set for selecting model parameters. The training set, validation set, and test set are divided as shown in the figure below.

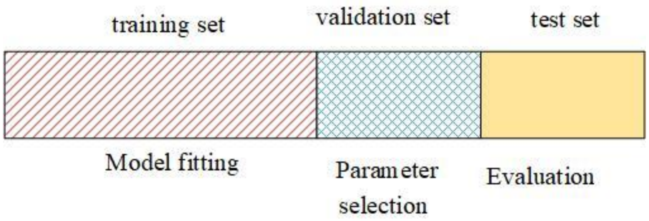

Fig 4: Data division.

Table 1: XGBoost Algorithm Parameters

parameter	Parameter value
booster	gbtree
tree_method	gpu_hist
eval_metric	rmse
objective	reg:squarederror
Silent	TRUE
Seed	2020
subsample	0.5
colsample_bytree	0.5
min_child_weight	5
max_depth	8
eta	0.001

The relevant parameters of the optimized model are shown in the table. Refer to the training rounds when optimizing parameters (22199 rounds in total), use the training set and the validation set as the new training set to train the XGBoost algorithm again.

4.4 Numerical experiment

In order to test the influence of the prediction target transformation on the prediction accuracy, the prediction target is recorded as follows. (a) the indoor temperature, (b) the difference between indoor and outdoor temperature, (c) the difference between logarithm of the indoor and outdoor temperature. We use the XGBoost algorithm to train under the same conditions except for the different prediction targets. The results are as follows (Actual refers to the real indoor temperature, XGB-R(a), XGB-R(b), XGB-R(c) respectively represent the indoor temperature obtained by using the corresponding prediction target to obtain the prediction result landscape transformation Predicted value).

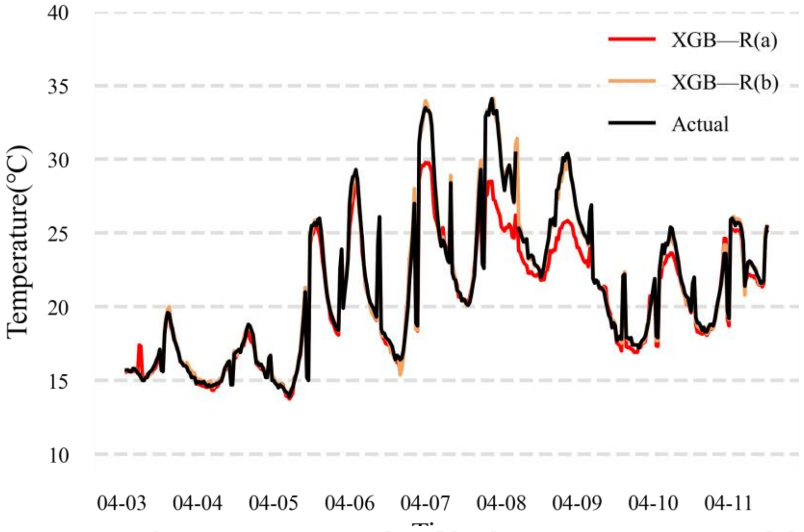

Fig 7: Indoor temperature predicted by the XGB-R(Target (a) and (b)).

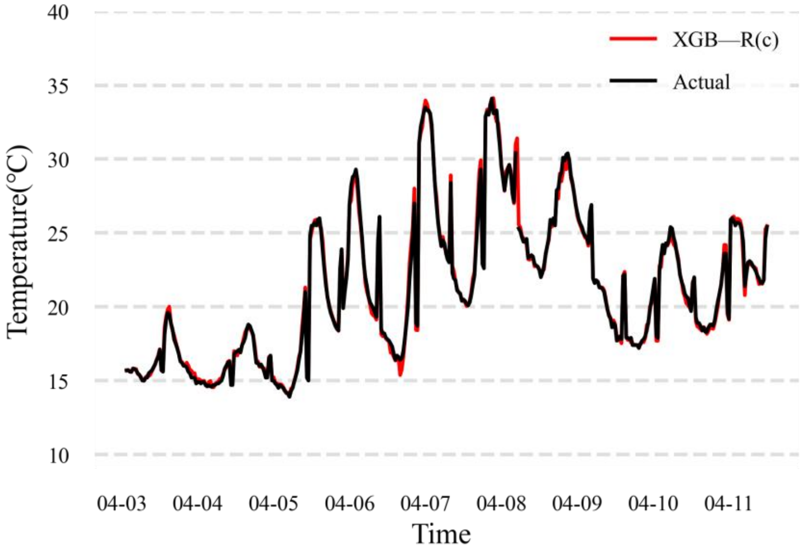

Fig 8: Indoor temperature predicted by the XGB-R(Target (c)).

Figs.7 and 8 show the indoor temperature predicted by the XGB-R with the predicted Targets (a), (b) and (c). It is found that the effect of prediction with Target (b) and (c) are much better than that with Target (a), while the differences between the prediction curves of Targets (a) and (b) are not obvious. All three Targets do not fit as well as expected in the peaks and valleys of the curves.

Fig.9 visually shows the MSE of the temperature prediction results under different prediction targets. The MSEs of the prediction of the XGBoost algorithm with the Targets (a), (b), and (c) are 0.9201, 0.0992, and 0.0886, respectively. The MSEs of the prediction of the XGB-R are 0.7797, 0.0980, and 0.0844, which is 84.7%, 98.8%, and 95.3% before processing residuals, respectively.

In order to improve the persuasiveness, we use LR (logistic regression), SGD (stochastic gradient descent) and SVM (support vector machine), which are usually used to solve such problems.Target (c) is chosen to be the predicted target. The results are as follows. (Actual refers to the real indoor temperature, LR, SGD, and SVM respectively refer to the predicted results of the indoor temperature obtained in the corresponding model)

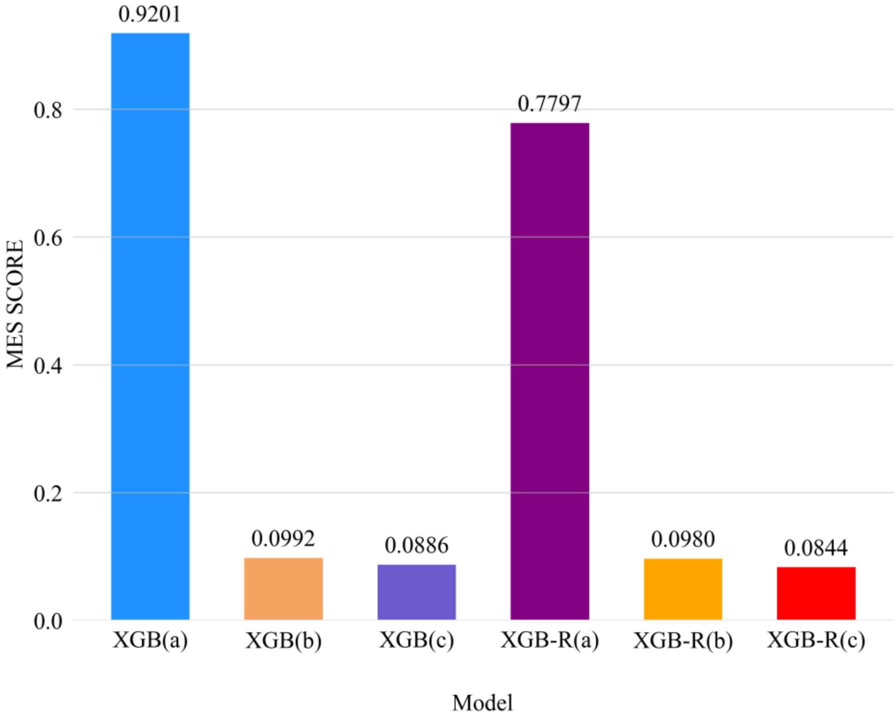

Fig 9:MSE of the prediction with different prediction Targets.

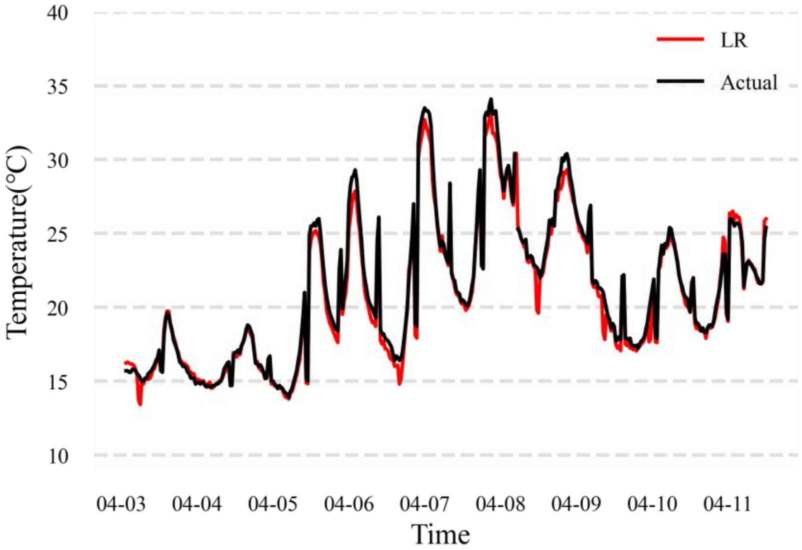

Fig 10: Indoor temperature predicted by the LR.

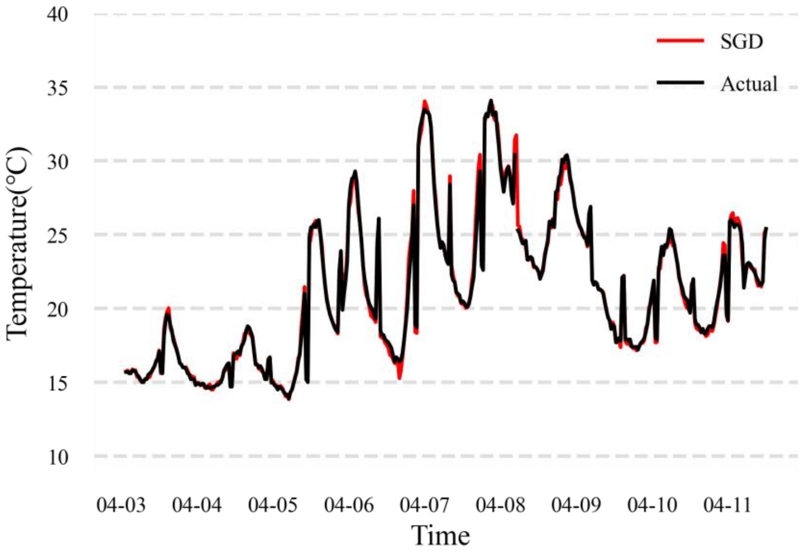

Fig 11: Indoor temperature predicted by the SGD.

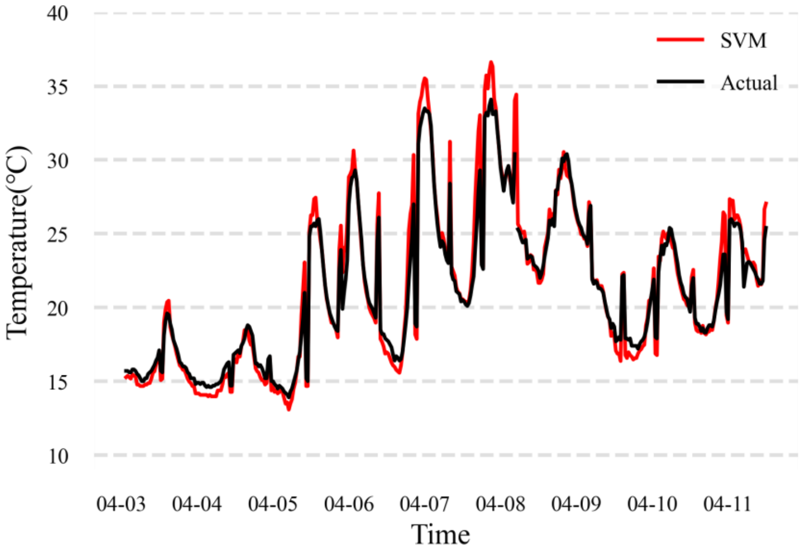

Fig 12: Indoor temperature predicted by the SVM.

Figs. 10, 11, and 12 show the temperature prediction curves of the three algorithms of the LR, SGD, and SVM. From these figures, we can observe that the predictions obtained using the SGD model are the most accurate, while the prediction results obtained using the SVM model are the least accurate. The temperature prediction curve of the three methods has large differences between the predicted value and the actual value at the peak and valley of the temperature curve.

ISSN: 0010-8189

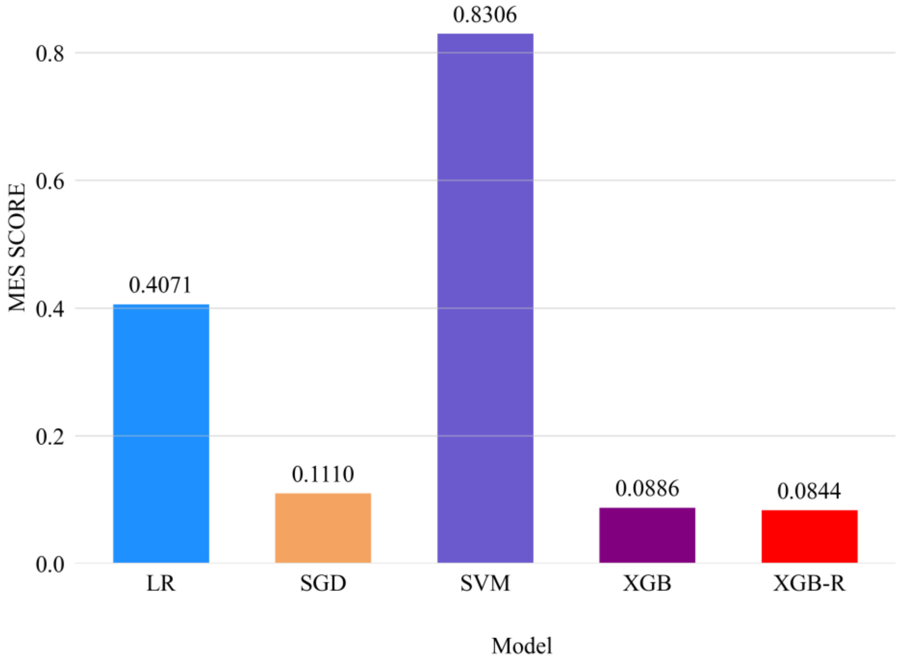

Fig 13: MSE of the prediction using LR, SGD, SVM, XGB, and XGB-R(target (c)).

Fig.13 shows the MSEs of the prediction of various methods. With the same prediction target(Target (c)), compared with LR, SGD, SVM,XGBoost, the method of XGB-R has the lowest prediction error (MSE = 0.0844), the prediction error of LR, SGD, SVM is 0.4071, 0.1110, 0.8306, 0.0886 respectively, decrease by 0.3227, 0.0266, 0.7462, 0.0042.

V. CONCLUSION

This paper proposes a temperature prediction method called XGB-R. This method is based on the XGBoost algorithm and linear residuals model. Three transformations of prediction target are also applied in the method.

Based on the 20-day greenhouse data, the results indicate the following points:

(a) Using linear model to process residuals produced by XGBoost can reduce about 10% MSE of the prediction. both before and after processing the residuals of the XGBoost (Fig.9).
(b) Among the three transformations of predicted target, it is the difference between the logarithm of the temperature inside and outside the greenhouse that has the highest accuracy(Targets (c)).
(c) The impacts of transformation on the accuracy of the prediction are more significant than the impact of residual processing.
(d) With the same predicted target(Target (c)), the error of the XGB-R is lower compared with LR, SGD, SVM and XGBoost. The MSE predicted by XGB-R is 0.0844, while the MSE predicted by LR, SGD, SVM and XGBoost are 0.4071, 0.1110, 0.8306 and 0.0886 respectively. (Fig.13).

Although the XGB-R method with thetransformation of the predicted target improves the accuracy of the prediction, there are several limitations in the method and the numerical experiments. Firstly,the method of predicting target conversion may not have the same effect on other evaluation indicators. Secondly, XGB-R is only applied in one small-scale data set. More data sets are needed to prove the superiority of XGB-R.

According to the application scenario, various evaluation indexes and data features can be selected in the subsequent researches. Also, different transformations can be applied. Additionally, models such as polynomial model can be used to predicted the residuals produced by XGBoost.

Acknowledgement

The authors would like to thank JiaHui Liang and Ze Chen for their valuable discussions and feedback.

References

[1] Xu Yu, JiRonghua. Research on temperature prediction of intelligent greenhouse based on complex neural network. Journal of Chinese Agricultural Mechanization, 2019, 40(04),174-178.

[2] ZHOU Xiangyu, CHENG Yong, WANG Jun. Agricultural greenhouse temperature prediction method based on improved deep belief network. Journal of Computer Applications, 2019,39(04), 1053-1058.

[3] Li Ning, Shen Shuang-he, Li Zhen-fa, LI Chun, LIU Shu-mei, XUE Qing-yu. Forecast model of minimum temperature inside greenhouse based on principal component regression. Chinese Journal of Agrometeorology,2013, 34(03),306-311.

[4] Qin Linlin, Ma Guoqi, Chu Zhudong, Wu Gang. Modeling and control of greenhouse temperature-humidity system based on grey prediction model. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE),2016, 32(01), 233-241.

[5] H.Uchida Frausto, J.G.Pieters, J.M.Deltour, Modelling greenhouse temperature by means of auto regressive models. Biosystems Engineering,2003, 84(02), 147-157.

[6] HuihuiYu,Yingyi Chen, Shahbaz Gul Hassan, DaoliangLi.Prediction of the temperature in a Chinese solar greenhouse based on LSSVM optimized by improved PSO. Computers and Electronics in Agriculture, 2016,122(03), 94-102.

[7] Sachin Kumar,Saibal K Pal,Rampal Singh. A novel hybrid model based on particle swarm optimization and extreme learning machine for short-term temperature prediction using ambient sensors. Sustainable Cities and Society, 2019,49(06), 101—601.

[8] Wang Hongkang, Li Li, Wu Yong, MengFanjia, Wang Haihua,N.A.Sigrimis. Recurrent neural network model for prediction of microclimate in solar greenhouse. IFAC-PapersOnLine,2018, 51(17),790-795.

[9] Seginer,T. Boulard, B.J.Bailey. Neural network models of the greenhouse climate. Journal of Agricultural Engineering Research, 1994,59(03),203-216.

[10] FathiFourati. Multiple neural control of a greenhouse. Neurocomputing,2014, 139, 138-144.

[11] Tian Dong, Wei Xinhua, Wang Yue, Zhao Anping, Mu Weisong, Feng Jianying. Prediction of temperature in edible fungi greenhouse based on MA-ARIMA-GASVR. Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE),2020, 36(03), 190−197.

[12] S.L.Patil, H.J.Tantau, V.M.Salokhe. Modelling of tropical greenhouse temperature by auto regressive and neural network models. Biosystems Engineering,2008, 99(03),423-431.

[13] Jerome H.Friedman. Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics, 2001,29(05),1189-1232.

[14] Tianqi Chen, Carlos Guestrin. 2016.XGBoost: A Scalable Tree Boosting System. The 22nd ACM SIGKDD International Conference.