Application of Time Series Model in Relative Humidity Prediction

Hongxiang Li\textsuperscript{1,2}, Yang Yang\textsuperscript{1,2}, Yifei Cheng\textsuperscript{1,2}, Yan Jin\textsuperscript{1,2}, Huan Luo\textsuperscript{1,2} and Liang Zhang\textsuperscript{1,2,*}

\textsuperscript{1} College of Information Engineering Sichuan Agricultural University Yaan, Sichuan
\textsuperscript{2} Key Laboratory of Agricultural Information Engineering of Sichuan Province, Sichuan Agriculture University

\textsuperscript{*}liang840416@126.com

Abstract. Relative humidity is the percentage of the vapor pressure in the air and the saturated vapor pressure at the same temperature. In the context of global climate change, accurate and reliable relative humidity prediction is of great significance in all fields. In this paper, taking the relative humidity data of Ya’an city, Sichuan province as an example, we used three methods, Holt-Winters, SARIMA and XGBoost to establish a time series model to predict relative humidity. We use the grid search and other methods for selecting the optimal parameters, the mean absolute error, mean square error and mean absolute percentage error as evaluation standard, the experimental results show that although the three model error is within an acceptable range, but XGBoost model predicted results are more accurate, better performance and stronger ability to resist a fitting, obviously better than the other two models, which can provide a reference for practical work.

1. Introduction
Relative humidity affects the resistance of the respiratory system; It is one of the most important factors affecting the efficiency and life span of perovskite solar cells [1], and also affects the adaptive thermal comfort and ozone concentration [2]. In addition, relative humidity plays an important role in the prevention and control of air pollution, industrial and technological production, agricultural production and plant growth. In recent years, people have tried many methods to predict relative humidity. XIN SHI and others will be based on cloud database of the improved BP neural network is applied in predicting the indoor relative humidity [3], Fen He put forward a kind of combined with genetic algorithm, IBP neural network model [4], Suryono Suryono et transform relative humidity of physical quantity into electric signal and fuzzy time series - markov model for relative humidity forecasts [5], G. Mustafaraj people use NNARX model such as relative humidity forecasts, and achieved good results. ZhiQiang Li et al. respectively established ARIMA and LSTM models to predict relative humidity and found that ARIMA was better, indicating that compared with the current popular neural network, the traditional time series model can also achieve better results.Holt-Winters and SARIMA time series model have been widely used in the fields of finance [6], health care [7], transportation [8–10]and so on. Xgboost is an integrated learning algorithm. Compared with the traditional GBDT, Xgboost makes improvements in the aspects of target function, regularization,
missing value processing and so on, thus significantly improving the effect and performance. In this paper, the three models are applied to relative humidity prediction to prove the applicability of the models.

2. Methodology

2.1. Holt-Winters Model
Holt-winters is developed by P.R. Winters on the basis of Charles Holt's exponentially weighted moving average method which is suitable for time series with tendency and seasonality [6]. It takes into account the three smoothing parameters, the, the, the, the, the, three smoothing parameters are all constrained between 0 and 1. There is an additive model and a multiplication model. The equation of the multiplication model is as follows:

\[ l_t = \alpha \frac{y_t}{s_{t-m}} + (1 - \alpha)(l_{t-1} + b_{t-1}) \]  
\[ b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \]  
\[ s_t = \gamma \frac{y_t}{l_{t-1} + b_{t-1}} + (1 - \gamma)s_{t-m} \]  
\[ \hat{y}_{t+h} = (l_t + b_t h) s_{t-m + h m} \]  

The addition model equation is as follows:

\[ l_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}) \]  
\[ b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} \]  
\[ s_t = \gamma(y_t - l_{t-1} + b_{t-1}) + (1 - \gamma)s_{t-m} \]  
\[ \hat{y}_{t+h} = l_t + b_t h + s_{t-m + h m} \]  

Where \( l_t \) is the long-term trend, \( b_t \) denotes the trend of increment, \( s_t \) represents the seasons change, \( \hat{y}_{t+h} \) is the future \( h \) period in advance, \( y_t \) is observation data for time \( t, m \) is the length of the seasonal cycle.

2.2. SARIMA Model
SARIMA is a seasonal ARIMA model, and its generalized form is \( SARIMA(p, d, q)(P, D, Q)s \). Where \( p \) is the long-term autoregressive order, \( q \) is the order of the long-term moving average, \( P \) is the seasonal autoregression, \( Q \) is the order of seasonal moving average, \( d \) is the regular difference, \( D \) is the seasonal difference and \( s \) is the period length. The establishment of SARIMA model can be divided into three steps. First, the stationarity test of time series is carried out. In this paper, augmented dickey-fuller (ADF) will be used for stationarity test (both parameters d and d are determined). The second step is to analyze autocorrelation function (ACF) and partial autocorrelation function (PACF) and determine the approximate range of \( P, Q \). Then, to fit each set of parameters through grid search, and to determine the optimal parameters according to akaike information criterion (AIC). Finally, the residual is verified, and Ljung-box is used to verify whether the residual is white noise. If the model residual is white noise, the available information has been extracted and the optimal parameters of the model are obtained.

2.3. XGBoost Model
XGBoost [11] is a kind of boosting algorithm, which is a gradient promotion algorithm based on the decision tree. Through multiple iterations, each iteration produces a weak classifier, and each classifier is trained on the basis of the residual of the previous classifier.

\[ \hat{y}_t = \varphi(x_t) = \sum_{k=1}^{K} f_k(x_t) \]  

Where \( f_k \) is the regression tree. \( K \) is the number of regression trees, and \( f_k(x_t) \) is the score of the i-th observation given by the k-th tree. In order to establish the optimal model, the following regularization objective functions need to be minimized:

\[ \text{minimize} \quad \sum_{k=1}^{K} f_k(x_t) \]  

Here $y_i$ is the actual value, $\hat{y}_i$ is the predicted value of round $T$, $l$ is a differentiable convex loss function, which represents the difference between the predicted value and the target value. $\Omega$ is punishment to prevent model too complex. $\gamma$ is the parameter that controls the number of leaf nodes $T$, $\lambda$ is the parameter that controls the weight of the leaf.

Whereas the tree integration model in equation (10) is difficult to optimize in Euclidean Spaces using traditional methods, XGBoost iterates using an additive approach, adding $f_t$ to minimize the following target functions.

$$\text{Obj}(\omega) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k)$$  \hspace{1cm} (10)$$

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \| \omega \|^2$$  \hspace{1cm} (11)$$

Where $\hat{y}_i^{(t)}$ is the prediction of the $i$-th instance in the $t$-th iteration. In general, the gradient of objective function is not easy to obtain. Because second-order approximation can be rapidly optimized [12], we use the second-order Taylor expansion to simplify the equation.

$$\text{Obj}^{(t)} = \sum_{i=1}^{n} \left[ l(y_i, \hat{y}_i^{(t-1)}) + g_t f_t(x_i) + \frac{1}{2} h_t f_t^2(x_i) \right] + \Omega(f_t)$$  \hspace{1cm} (12)$$

Where $g_t = \partial \text{Obj}^{(t-1)}(y_i, \hat{y}_i^{(t-1)})$ and $h_t = \partial^2 \text{Obj}^{(t-1)}(y_i, \hat{y}_i^{(t-1)})$ are the first and second order values of the function respectively. Assumptions of the leaf node $f$ collection $I_f = \{ [q(x_i) = f] \}$ by extending the $\omega$ and remove the constant term at $t$-th iteration, we can rewrite the equation (13)

$$\text{Obj}^{(t)} = \sum_{i=1}^{n} \left[ g_t f_t(x_i) + \frac{1}{2} h_t f_t^2(x_i) \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^{T} \omega_j^2$$  \hspace{1cm} (13)$$

$$\text{Obj}^{(t)} = \sum_{j=1}^{T} \left[ \left( \sum_{i \in I_f} g_t \omega_j \right) + \frac{1}{2} \left( \sum_{i \in I_f} h_t \omega_j + \lambda \right) \omega_j^2 \right] + \gamma T$$  \hspace{1cm} (14)$$

In a tree structure $q(x)$, the weight of leaf node $f$ omega is expressed as $\omega_j = -\frac{\sum_{i \in I_f} g_t \omega_j}{\sum_{i \in I_f} h_t + \lambda}$ then we can calculate the corresponding optimal value:

$$\text{Obj}^{(t)}(q) = \frac{1}{2} \sum_{j=1}^{T} \left( \frac{\sum_{i \in I_f} g_t}{\sum_{i \in I_f} h_t + \lambda} \right)^2 + \gamma T$$  \hspace{1cm} (15)$$

Equation (15) is the evaluation equation of tree structure $q$. However, it is not practical to enumerate all the tree structure $q$ in general. Instead, greedy algorithm is used to continuously split leaf nodes and repeatedly add subtrees. Equation (16) is used as the evaluation equation of evaluation split node.

$$\text{Obj}_{\text{split}} = \frac{1}{2} \sum_{i \in I_f} \left( \frac{\sum_{i \in I_f} g_t}{\sum_{i \in I_f} h_t + \lambda} \right)^2 + \frac{1}{2} \sum_{i \in I_f} h_t \omega_j^2 + \gamma$$  \hspace{1cm} (16)$$

Where $I_L$ and $I_R$ are the left and right nodes of the instance set after leaf node $I$ is split, respectively.

3. Model establishment and parameter selection

3.1. The experimental data

In this study, the experimental data from China's greenhouse data sharing platform (http://data.sheshiyuanyi.com/), select yaan of sichuan province weather stations from January 1984 to December 2016 as experiment data, Python is used for model building and research experiments.

3.2. Evaluation metrics

When evaluating the performance of the model, we mainly consider the Error between the predicted value and the real value, therefore the Mean Absolute Error(MAE),Mean Square Error(MSE) and
Mean Absolute Percentage Error (MAPE) are used for model evaluation. The calculation equation is as follows:

\[
MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (17)
\]

\[
MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (18)
\]

\[
MAPE(y, \hat{y}) = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \times 100 \quad (19)
\]

### 3.3. Holt-Winters model establishment

In order to find the optimal parameters of the model more accurately and not destroy the time structure of the data, we use the scroll cross validation to perform the scroll cross validation of the data by 4 folds. The three parameters of the Holt-Winters model \((\alpha, \beta, \gamma)\) are all constrained from 0 to 1, so we must choose the optimization algorithm that supports the model parameter constraints. Here we use the optimizer in the scipy library to minimize the target function using truncated Newton (TNC) algorithm. The values of the values of \(\alpha, \beta, \) and \(\gamma\) are shown in table 1.

| Parameter selection of addition model and multiplication model. | \(\alpha\)          | \(\beta\)          | \(\gamma\)          |
|---------------------------------------------------------------|-------------------|-------------------|-------------------|
| Multiplicative                                               | 0.131439129       | 0.012047853       | 0.089363757       |
| Additive                                                     | 0.135142669       | 0.014924424       | 0.094415762       |

According to the parameters in table 1, we established the multiplication model and the addition model. After comparing the fitting effect of the model, the addition model is more accurate. Therefore, we will keep the addition model for further comparison with the other two models.

### 3.4. SARIMA model establishment

We used python's statsmodels library for the ADF test to determine time series stationarity as shown in table 2.

| Table 2. The ADF test information of the original data and the data after the difference operation. |
|---------------------------------------------------------------|-------------------|-------------------|
| ADF                                                           | Original          | After the difference |
| P-value                                                       | 2.479991195       | 9.676373607       |
| 1%                                                            | 0.120418893       | 1.24E-16          |
| 5%                                                            | 3.447494667       | 3.448393521       |
| 10%                                                           | 2.869096244       | 2.869491234       |
| 2.5707953                                                     | 2.571005879       |

It was found that the ADF value of the original data was more than 1%, and the P value was large, indicating that the null hypothesis could not be rejected, and the time series showed obvious seasonality as shown in the figure 1.
After regular difference and seasonal difference, the ADF value of time series is less than 1%, and the P value is close to 0, indicating that the sequence is stable and the seasonality is eliminated as shown in the figure 2. Through the above difference operation, we have determined that the values of d, D and s are 1,1,12 respectively. According to the ACF and PACF diagrams of the sequence after difference, the approximate range of p, q, P and Q can be obtained as shown in the table 3 and table 4.

| Table 3 | grid search scope of some parameters. |
|---------|---------------------------------------|
| P       | range(5,8)                            |
| q       | range(2,5)                            |
| P       | range(0,3)                            |
| Q       | range(1,3)                            |

| Table 4 | Partial parameter combination AIC value (p, q, P, Q) respectively. |
|---------|---------------------------------------------------------------|
| Number  | Parameters         | AIC                  |
| 1       | (6,2,0,1)          | 2142.965602          |
| 2       | (6,2,2,1)          | 2144.719665          |
| 3       | (7,2,0,1)          | 2144.098407          |

Finally, SARIMA(6,1,2)(0,1,1)12 is selected as the optimal parameter through grid search of these 54 groups of parameters. Ljung-box test shows that the model residuals are white noise. As shown in the figure 3 and figure 4, p values are all greater than 0.05, indicating that the available information of the time series is fully extracted.
3.5. XGBoost model establishment

We use grid search and early stopping (early_stopping_rounds=10) to optimize the parameter selection of XGBoost model. The parameter search interval is shown in the figure 5:

- **max_depth**: The depth of the tree. The search range is [1, 2, 3, 4, 5, 6, 7, 8, 9, 10].
- **min_child_weight**: The minimum sum of instance weight (hessian) needed in a child. The search range is [2, 4, 6, 8, 9].
- **gamma**: Specifies the minimum loss reduction required for a split to happen. The search range is [0.0005, 0.01, 0.1, 0.2, 0.3].
- **reg_alpha**: L1 regularization. The search range is [0, 0.01, 0.05, 0.1, 0.2, 0.4, 0.5, 0.6, 0.8, 0.9, 1].
- **reg_lambda**: L2 regularization. The search range is [0, 0.0005, 0.01, 0.1, 0.4, 0.5, 0.6, 0.8, 0.9, 1].
- **subsample**: Controls the random proportion of each tree. The search range is [0.6, 0.7, 0.8, 0.9, 0.95, 1].
- **colsample_bytree**: Controls every tree of the characteristics of random sampling. The search range is [0.6, 0.7, 0.8, 0.9, 0.95, 1].

**Figure 5.** Parameter interval and selection.

Where **max_depth** is the depth of the tree, **reg_alpha** is the weight of L1 regularization, **reg_lambda** is the weight for L2 regularization, **subsample** controls the random proportion of each tree, **colsample** controls every tree of the characteristics of random sampling, **gamma** specified node
split the required minimum loss function decline in value and **min_child_weight** defines the minimum weight sum for all observations of a subtree.

4. **Comparison and Result**

This paper proves the applicability of Holt-Winters, SARIMA and XGBoost in relative humidity prediction. The prediction results of the model are shown in figure 6 and table 5. By calculating the errors, the prediction results of the XGBoost model are more accurate, and the model is more interpretive. However, in this experiment, the only data used is univariate data, and the prediction is only based on the data of relative humidity. In the future research, other data with strong correlation with relative humidity and influence can be considered to carry out multivariate prediction, so as to enhance the reliability of the prediction.

![Figure 6. Model prediction effect comparison](image)

**Table 5. Model error comparison.**

| Model       | MAE  | MSE  | MAPE   |
|-------------|------|------|--------|
| XGBoost     | 2.29 | 9.17 | 2.99%  |
| SARIMA      | 2.97 | 14.76| 3.89%  |
| HW(additive)| 2.74 | 13.4 | 3.55%  |

**References**

[1] Gangishetty M K, Scott R W J and Kelly T L 2015 Effect of Relative Humidity on Crystal Growth, Device Performance and Hysteresis in Planar Heterojunction Perovskite Solar Cells *Nanoscale* 8

[2] Quansah E, Preko K and Amekudzi L K 2012 The influence of temperature and relative humidity on indoor ozone concentrations during the Harmattan *J. Emerg. Trends Eng. Appl. Sci.*

[3] X. Shi, W. Lu, Y. Zhao and P. Qin 2018 Prediction of Indoor Temperature and Relative Humidity Based on Cloud Database by Using an Improved BP Neural Network in Chongqing *IEEE Access* 6 30559–66

[4] Fen He, Chengwei Ma and Junxiong Zhang 2007 Greenhouse Air Temperature and Humidity Prediction Based on Improved BP Neural Network and Genetic Algorithm *Adv. Neural Netw. - ISNN 2007 4th Int. Symp. Neural Netw. ISNN 2007 Nanjing China June 3-7* 2007 Proc. Part III
[5] S. Suryono, R. Saputra, B. Surarso and H. Sukri 2017 Web-based fuzzy time series for environmental temperature and relative humidity prediction 2017 IEEE International Conference on Communication, Networks and Satellite (Comnetsat) 2017 IEEE International Conference on Communication, Networks and Satellite (Comnetsat) pp 36–41

[6] CLAUDIMAR, PEREIRA, DA, VEIGA, CASSIA, RITA, PEREIRA, DA, VEIGA and and W Demand forecasting in food retail: a comparison between the Holt-Winters and ARIMA models

[7] Zhou L, Ping Z, Dongdong W, Cheng C and Hao H Time series model for forecasting the number of new admission inpatients Bmc Med. Inform. Decis. Mak. 18 39-

[8] Yang Y M, Yu H and Sun Z 2017 Aircraft failure rate forecasting method based on Holt-Winters seasonal model 2017 IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)

[9] Dantas T M, Cyrino Oliveira F L and Varela Repolho H M Air transportation demand forecast through Bagging Holt Winters methods J. Air Transp. Manag. 59 116–23

[10] Miloš Milenkovиć, Libor Švadlenka, Vlastimil Melichar, Nebojša Bojović and Zoran Avramović SARIMA modelling approach for railway passenger flow forecasting Transport 1–8

[11] Chen T, Tong H and Benesty M 2016 xgboost: Extreme Gradient Boosting

[12] Jerome, Friedman, Trevor, Hastie, Robert and Tibshirani Special Invited Paper. Additive Logistic Regression: A Statistical View of Boosting