Research Article

Hybrid Time Series Method for Long-Time Temperature Series Analysis

Guangdong Huang¹ and Jiahong Li²

¹*School of Science, China University of Geosciences, Beijing 100083, China*

²*School of Economics and Management, Beihang University, Beijing 100191, China*

Correspondence should be addressed to Jiahong Li; by2008131@buaa.edu.cn

Received 8 March 2021; Revised 11 June 2021; Accepted 15 July 2021; Published 23 July 2021

Academic Editor: Ya Jia

Copyright © 2021 Guangdong Huang and Jiahong Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper combines discrete wavelet transform (DWT), autoregressive moving average (ARMA), and XGBoost algorithm to propose a weighted hybrid algorithm named DWTs-ARMA-XGBoost (DAX) on long-time temperature series analysis. Firstly, this paper chooses the temperature data of February 1 to 20 from 1967 to 2016 of northern mountainous area in North China as the observed data. Then, we use 10 different discrete wavelet functions to decompose and reconstruct the observed data. Next, we build ARMA models on all the reconstructed data. In the end, we regard the calculations of 10 DWT-ARMA (DA) algorithms and the observed data as the labels and target of the XGBoost algorithm, respectively. Through the data training and testing of the XGBoost algorithm, the optimal weights and the corresponding output of the hybrid DAX model can be calculated. Root mean squared error (RMSE) was followed as the criteria for judging the precision. This paper compared DAX with an equal-weighted average (EWA) algorithm and 10 DA algorithms. The result shows that the RMSE of the two hybrid algorithms is much lower than that of the DA algorithms. Moreover, the bigger decrease in RMSE of the DAX model than the EWA model represents that the proposed DAX model has significant superiority in combining models which proves that DAX has significant improvement in prediction as well.

1. Introduction

In the context of global climate change, surface atmospheric temperature prediction has become a hot topic. In recent years, researchers have proposed a variety of methods for atmospheric temperature simulation and prediction. Commonly used methods include time series analysis, empirical mode decomposition (EMD), wavelet transform, Markov model, and gray prediction model (GM). Among these methods, the most popular measure is time series analysis including ARMA and its derivative models. Badu and Reddy [1] imported ARIMA in the forecast of temperature comparing the accuracy of whether ARIMA is based on trend or not. Zhou et al. [2] proposed a seasonal index-based Gray-Markov prediction model according to the interannual periodicity and seasonal variation of historical data and combined longitudinal and horizontal

analysis in temperature prediction. They corrected the lateral seasonal change of temperature through a seasonal index, then used the gray model to predict, and finally used Markov to correct the error. Wang et al. [3] used the C-C method and least error method to construct daily maximum and minimum time series, then applied the method of extracting nearest neighbors by segment to local support vector regression, and established a local prediction model of daily maximum and minimum temperature one day in advance. But these methods can only deal with linear time series and cannot extract nonlinear information or features of time series. So, the research about temperature forecasting needs some idea that comes from other fields which apply nonlinear methods on time series.

Besides being applied in temperature prediction, time series analysis always plays a vital role in practical application covering finance, traffic flow, and so on. Huang et al.

[4] proposed empirical mode decomposition (EMD). It can decompose both stable and unstable time series into IMFs. But it is empirical and approximate, and sometimes mode mixing will occur. Wang et al. [5] used Fourier transform to improve the prediction accuracy of the debris flow disaster. The prediction accuracy which was formerly obtained by the gray model was significantly improved by Fourier transform. Nevertheless, the Fourier transform has its limitations in which the scale of the Fourier basis function is fixed; thus, the scale cannot switch flexibly according to the time series. But wavelet transform offers a great option for improvement because the wavelet basis function's scale can adjust based on the local features of time series (Polikar [6]).

As a signal time-domain analysis method, the significant feature of wavelet transform is self-adaptation, which can automatically adjust relevant parameters according to the research object (Chen [7]). The biggest advantage of wavelet analysis is that the time series is decomposed into different levels according to different scales, which is convenient for analysis and prediction. Moreover, the basic function of wavelet decomposition is orthogonal which will avoid the appearance of mode mixing.

For time series data that has high fluctuation and complexity, another way of simulation and prediction is using the deep learning method. Wu et al. [8] and Chang et al. [9] developed artificial neural networks (ANN) algorithms for short-term river flow forecasting. Sudheer et al. [10] used Particle Swarm Optimization (PSO) algorithm to select Support Vector Machine (SVM) parameters and developed an SVM-PSO algorithm to predict monthly discharge. Jiang et al. [11] utilized the GAN-CNN model for predicting the complex product costs in the case of small samples, aiming at solving the problem that traditional deep learning networks have low fitting performance in the training process.

A growing consensus suggests that combining forecasting has advantages over a single algorithm in terms of not only accuracy but also simplifying algorithm selection and building. Yu et al. [12] proposed an EELM-EEMD-ADD algorithm applied in crude oil price forecasting. Comparing with single ANN, SVM, and FNN, it had less loss cost and much more robust. It shows that the worst combining algorithm is even better than the best single algorithm. Yong et al. [13] used an ensemble neural network method combining BP network, SVM, EEMD, and wavelet transform to forecast the wind speed. The experimental result indicated the ensemble model outperformed individual models. Shi et al. [14] combined the lasso method with XGBoost to optimize the error of the algorithm used for WTI price forecasting. The lasso method works well in feature selection which will be set as the inputs of the XGBoost model.

From the literature above, we conclude that the hybrid algorithm outperforms single models. In the process of choosing single models, we consider the interpretability, complexity, and time costs of a model as the choosing principle. ARMA is the most frequently used linear model which has low time costs and does not sacrifice regression accuracy in time series analysis. DWT overperforms Fourier transform because the wavelet basis has a self-adaptation

scale, while the scale of the Fourier basis is fixed. Moreover, DWT has less complexity than neural networks for signal denoising. For ensemble tasks, XGBoost is the improvement of GBDT, and it has better interpretability than neural networks because XGBoost is constructed based on decision trees. As a result, this paper proposes a novel algorithm named DAX for time series prediction. The DAX takes advantage of discrete wavelet transform and XGBoost in optimizing the error of prediction. This paper chooses the temperature data of February 1st to 20th from 1967 to 2016 of Northern mountainous area in North China as the dataset. Comparing with the DA and EWA model, we found that DAX algorithm increased the accuracy and strengthened the robustness of the prediction algorithm apparently.

2. Methodology

The DAX model is based on three submodels: ARMA, discrete wavelet transform, and XGBoost. ARMA is used to describe the dynamic trend of time series but is only applicable to the stationary and linear time series. What is more, it is easy to be disturbed by noises. The volatility of temperature time series is always caused by noises with different frequencies (Wang and Xiao [15]). Thus, the performance of simulation and prediction is not excellent in these cases. It is necessary to eliminate the noises. Fourier transform is a commonly used method but its basis function is fixed which cannot adjust its scale according to the feature of time series. However, the DWT does well in denoising since the biggest advantage of DWT is its self-adaptation. The DWT basis function adjusts its scale according to the time series feature and thus avoids the interference of noise with different frequencies. Although the DWT-based ARMA model (i.e., DA) solves the problem of noise, the model highly relies on the type of wavelet basis function. It is vital to get a comprehensive model which combines the strength of different wavelet basis function. XGBoost is a mature ensemble learning algorithm that aims to improve the performance of basis prediction methods. What is more, XGBoost is much more interpretability than other ensemble methods such as neural networks. As a result, XGBoost is added to our former method to learn a hybrid model. In the process of model training, XGBoost focuses on those data which have weak performance in the basic method. The DAX first utilizes DA to obtain simulations and predictions of basic methods and then uses XGBoost to optimize the error of simulation and prediction. A detailed description of each method is illustrated in this section.

2.1. ARMA Model. The fundamental part of the DAX method is the ARMA model. ARMA is an abbreviation of autoregression and moving average which suggests it relies on historical data. The basic principle of the ARMA model is to regard time series as a random series arranged in time. The dependence of these random variables is reflected as the continuity in time. ARMA analyzes time series based on the historical data.

In general, $\text{ARMA}(p, q)$ is defined as follows:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q}, \quad (1)$$

where X_t is the value at time t , $X_{t-1}, X_{t-2}, \dots, X_{t-p}$ are the value at time $t-1, t-2, \dots, t-p$, $\varepsilon_{t-1}, \varepsilon_{t-2}, \dots, \varepsilon_{t-q}$ are the observed noises in time series at time $t-1, t-2, \dots, t-q$, and p, q are the order of autoregression and moving average, respectively, that can be determined by minimum information criterion AIC. θ, ϕ are the nonzero undetermined coefficient which can be obtained through least-squares estimation. $\{\varepsilon_t\}$ is the independent error item which is the random item extracted from data in this paper. But ARMA is easy to be interfered by noise with different frequencies in the data. It is often manifested in areas where data fluctuations are large or uneven; thus both the simulation and prediction results deviate from the observed value.

2.2. Wavelet Transform (WT). Wavelet is a mathematical function of amplifying the local part of the signal through scale changing. Moreover, the basis wavelet can be flexibly selected according to the characteristics of the time series. WT contains continuous wavelet transform (CWT) and DWT.

2.2.1. Continuous Wavelet Transform (CWT). The CWT of a time series function $f(t)$ is given as

$$T(a, b) = \int_{-\infty}^{\infty} f(t) \psi_{(a,b)}^*(t) dt, \quad (2)$$

where $*$ denotes complex conjugation and $\psi_{(a,b)}(t)$ is a wavelet basis, which is given as

$$\psi_{(a,b)}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad (a, b \in \mathbb{R}, a \neq 0). \quad (3)$$

The transformed signal $T(a, b)$ is defined on the $a-b$ plane, where a and b are used to adjust the frequency and the time location.

2.2.2. Discrete Wavelet Transform (DWT). The DWT method is derived from the CWT through discretization of the wavelet $\psi_{(a,b)}(t)$. The general formulation of discrete wavelet basis is

$$\psi_{(j,k)}(t) = \frac{1}{\sqrt{2^j}} \psi\left(\frac{t-2^j k}{2^j}\right), \quad (4)$$

where a, b in CWT are replaced by 2^j and $2^j k$, respectively. So, $\psi_{(j,k)}(t)$ is an orthonormal basis according to the suitable condition. Based on the above, the original time function can be written as

$$f(t) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} C_{j,k} \psi_{(j,k)}(t), \quad (5)$$

where we denote the wavelet transform coefficient as

$$C_{j,k} = \int_{-\infty}^{\infty} f(t) \psi_{(j,k)}^*(t) dt. \quad (6)$$

2.2.3. Mallat Decomposition and Reconstruction Theory. Mallat [16] proposed multiresolution analysis in building an orthogonal wavelet basis. For each signal, we could decompose it into several subsignals through scale transformation. The subsignal can be divided into two parts according to their frequency. The lower frequency section is also called the approximate signal which can reflect the overall information. The higher frequency section is also called detail signal which can reflect the detail of the signal.

Based on multiresolution theory, Mallat proposed the construction of an orthogonal wavelet and its fast transformation algorithm. We could decompose a signal into lower frequency space and higher frequency space by constructing a wavelet basis function with different frequencies. Since the decomposition, we could obtain the original signal without noise through the inverse transformation of the wavelet basis function.

2.3. XGBoost Method. XGBoost is a hybrid method proposed by Chen and Guestrin [17], which is constructed on a decision tree. For a given task, we first use the decision rules in the trees to classify it into the leaves, and each leaf has its score. Next, we calculate the final prediction by summing up the score in the corresponding leaves. XGBoost has three cores which are boosting, constructing the objective loss function, and solving the loss function. The three cores will be introduced in this section.

2.3.1. Boosting. For a given dataset $D = \{(x_i, y_i)\} (|D| = n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R})$ with n samples and m attributes, an ensemble tree algorithm uses K additive functions to output the prediction. The formula of ensembled prediction is shown as follows:

$$\hat{y}_i = \Phi(x_i) = \sum_{k=1}^K f_k(x_i), \quad f_k \in F, \quad (7)$$

where $F = \{f(x) = w_{q(x)}\} (q: \mathbb{R}^m \rightarrow T, w \in \mathbb{R}^T)$ is the space of regression trees. Here, q denotes the structure of each tree and maps a sample to the corresponding leaf factor. T is the number of leaves in the tree. Each f_k represents to the output of each tree and corresponds to an independent tree structure q and leaf weight w .

2.3.2. Objective Loss Function. The objective loss function consists of the loss function and regularization term. The regularization term is the sum of regularization of each tree, and it is used for regular structure of each tree to avoid overfitting. In a word, the regularization aims to keep both accuracy and generalization. The regularized objective loss function is written as follows:

$$L(\Phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k). \quad (8)$$

The first term $l(\bullet)$ is a loss function which measures the error between prediction $\hat{y}_i$ and the target y_i . The second term $\Omega(f_k) = \gamma T_k + (1/2)\lambda \|w\|^2$ in (8) is the regularization term which penalizes the complexity of the algorithm. T_k represents the number of leaves in the k -th tree. Since $\|w\|^2 = \sum_{j=1}^{T_k} w_j^2$, w_j is the score of the j -th leaf. γ, λ are the parameters of the algorithm which need to be initialized and calculated through training and testing. The larger the values γ and λ , the greater the penalty for attributes with a high number of leaves. The addable regularization term helps the XGBoost algorithm avoid overfitting. And the objective loss function can be written in detail as follows:

$$L(\Phi) = \sum_i l(\hat{y}_i, y_i) + \sum_{k=1}^K \left(\gamma T_k + \frac{1}{2} \lambda \sum_{j=1}^{T_k} w_j^2 \right). \quad (9)$$

2.3.3. Solve the Objective Loss Function. Since the algorithm is trained by an additive process, the solving of the objective loss function is implemented through additive steps. $\hat{y}_i^{(t)}$ is defined as the prediction of the i th instance at the t th iteration. First, the constant term is initialized at the beginning denoting as $\hat{y}_i^{(0)} = 0$. Then, the function f_t called tree in XGBoost is added to both loss function and regularization term to minimize the objective function at the t th iteration. The detailed solving process is described in this part. To introduce the complex process clearly, we take the prediction term $\hat{y}_i^{(t)}$, for example. The first tree is added into the prediction function at the first iteration, as shown in the following:

$$\hat{y}_i^{(1)} = f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i). \quad (10)$$

The second tree is added into the prediction function at the second iteration, as shown in the following:

$$\hat{y}_i^{(2)} = f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i). \quad (11)$$

Generally, the t th tree is added into the prediction function at the t th iteration, as shown in the following:

$$\hat{y}_i^{(t)} = \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i). \quad (12)$$

Substitute equation (12) into (8) and use f_t to minimize the objective loss function at the t th iteration. The objective loss function is rewritten as follows:

$$L^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \Omega(f_t) = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t). \quad (13)$$

According to the nature of the square error, equation (13) can extend as

$$\begin{aligned} L^{(t)} &= \sum_{i=1}^n \left(y_i - (\hat{y}_i^{(t-1)} + f_t(x_i)) \right)^2 + \Omega(f_t) \\ &= \sum_{i=1}^n \left[2(\hat{y}_i^{(t-1)} - y_i) f_t(x_i) + f_t(x_i)^2 \right] + \Omega(f_t). \end{aligned} \quad (14)$$

The second-order term of the Taylor expansion is kept in the loss function. Equation (14) can be written as

$$L^{(t)} \approx \sum_{i=1}^n \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t), \quad (15)$$

where $g_i = \partial l(y_i, \hat{y}_i^{(t-1)}) / \partial \hat{y}_i^{(t-1)}$ and $h_i = \partial^2 l(y_i, \hat{y}_i^{(t-1)}) / \partial \hat{y}_i^{(t-1)^2}$ are the first- and second-order gradient statistics of the loss function. At the t th iteration, $l(y_i, \hat{y}_i^{(t-1)})$ is a constant since $\hat{y}_i^{(t-1)}$ is known. Equation (15) can be simplified as

$$L^{(t)} = \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + C. \quad (16)$$

Since $f_t(x_i)$ is the prediction of the t th tree according to the definition, it is correlated with the leave nodes and their locations. Based on this theory, the structure of $f_t(x_i)$ can be assumed as

$$f_t(x_i) = w_{q_t(x)}, \quad w \in R^{T_t}, q_t: R^d \longrightarrow \{1, 2, \dots, T_t\}. \quad (17)$$

In equation (17), $q_t(x)$ maps the sample x to specific leave nodes which represent the structure. w is the value of leave nodes, and it is also called weight.

Define $I_j = \{i | q(x_i) = j\}$ as the instance set of leaf j . The objective loss function can be transformed from the sum of instances to the sum of leaves. Equation (16) is rewritten by replacing instance by leaf and expanding Ω , as shown in the following:

$$\begin{aligned} L^{(t)} &= \sum_{i=1}^n \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \\ &= \sum_{j=1}^T \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T. \end{aligned} \quad (18)$$

From equation (18), we notice that only $(\sum_{i \in I_j} g_i) w_j + (1/2)(\sum_{i \in I_j} h_i + \lambda) w_j^2$ is correlated with parameter w_j . Thus, the goal of minimizing the objective loss function $L^{(t)}$ is equal to minimizing $(\sum_{i \in I_j} g_i) w_j + (1/2)(\sum_{i \in I_j} h_i + \lambda) w_j^2$. To obtain the optimal weight w_j^* , we only need to differentiate equation (18) according to w_j . The result of optimal weight w_j^* is shown as follows:

$$w_j^* = - \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda}. \quad (19)$$

Next, we substitute the optimal weight w_j^* into the objective loss function to get the corresponding optimal value of $L^{(t)}(q)$, which is shown as

$$L^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^T \frac{\left(\sum_{i \in I_j} g_i \right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T. \quad (20)$$

According to equation (20), the value of the minimum loss can be used to evaluate the quality of a tree structure q .

2.4. DAX Method. Based on the basic model above, we can build our DAX method. First, we use different discrete wavelet functions to decompose time series. In theory, we can do $\log_2 N$ times decomposition for a data series whose length is N . But as the number of decomposition layers increases, information loss occurs in high-frequency signals (Wang et al. [18]). So, it is necessary to choose a reasonable number. We choose the number of layers is 3. The original data is decomposed into lower frequency signal $cA1$ and higher frequency signal $cD1$. Next, we decompose the $cA1$ into $cA2$ and $cD2$ and then decompose $cA2$ into $cA3$ and $cD3$. So, we get one lower frequency signal $cA3$ and three higher frequency signals $cD1$, $cD2$, and $cD3$. Applying Mallat's theory, we get the reconstructed signals $A3$, $D1$, $D2$, $D3$ respectively.

Build ARMA models on each reconstructed signal, respectively, and get their simulations and predictions. Adding up simulation and prediction on each reconstructed signal together to obtain the final simulation and prediction of each DA model. The formula of this method can be written as $\hat{Y} = \hat{Y}_{A_3} + \sum_{i=1}^3 \hat{Y}_{D_i}$, where $\hat{Y}_{A_3}$ denotes the simulation and prediction of reconstructed lower frequency signal $A3$ and $\hat{Y}_{D_i}$ represents the simulation and prediction of higher frequency signal.

With the existence of big errors in the simulation and prediction of some DA methods, a powerful hybrid machine learning method XGBoost is considered to solve this error. XGBoost focuses on those DA models which have big deviations from observed data. Thus, the performance will be improved through iteration until the error is lower than its acceptable threshold. Consequently, we combine the DA and XGBoost named DAX to lower simulation and prediction errors. In the process of XGBoost modeling, some parameters control the structure need to be initialized at the beginning. For example, learning rate η controls the speed of learning, a max depth controls the number of layer of trees in XGBoost, regularization factor λ is used for avoiding overfitting, and γ reflects the number of leaves in regularization term. Due to DAX, which is a hybrid algorithm, the optimal weights of submodels are essential, and they can be obtained through model training and testing. After these processes, 10 weak DA methods can be combined as a strong simulation and prediction method through the XGBoost algorithm. The summary of the DAX algorithm is written as follows: (Algorithm 1)

3. Long-Time Temperature Variation Analysis in North China

3.1. Data Usage. The northern mountainous climate of northern China is a warm temperate semihumid continental

monsoon climate, due to its location on the eastern edge of the Loess Plateau and the northern end of the North China Plain. The outstanding performance is that its winter temperature is low, and the icing period is long. The average temperature is lower by 6°C and 2.8°C than the nearby city and other mountainous areas. Relying on the unique meteorological conditions, in recent years, the area has vigorously developed tourism which attracts many skiers to come and play, reaching the peak in February every year. Studying the long-term changes of winter temperature in the area can infer the temperature changes in other areas with similar climatic characteristics and help to provide a reference for the local development of winter sports.

This paper first collected daily average temperature data of February 1st to 20th from 1967 to 2020 of a station in the northern mountainous region in North China. Then, we divided the data into 2 parts. The first part consists of data range from February 1st to 10th, named temperature of early February. The second part consists of data range from February 11st to 20th, named temperature of mid-February. According to the segmentation, both parts are panel data where the column is the year and the row is the date. For each part, we took an average of each year's data by date. The panel data was transformed into a time series ranging from 1967 to 2020. The time series is denoted as $\{X_t\}$ ($t = 1967, \dots, 2020$), which has been defined in Section 2.1. The two time series were called average temperature of early February and the average temperature of mid-February, respectively. In this paper, data ranging from 1967 to 2016 were used for modeling, while those ranging from 2017 to 2020 were used for predicting.

To judge whether the sequence can establish an ARMA model, the original time series was subjected to pretreatment including stationarity test and white noise test. According to the result of the stationarity test and white noise test, the observed data was stationary so that can be used in the DAX algorithm.

3.2. Model Comparison and Result Analysis

3.2.1. DA Model and Result. Based on the DAX theory proposed in Section 2 and the data preparation process, we first used 10 discrete wavelet basis functions to decompose each observed data into 3 subsignals. Then, we built an ARMA model on each subsignal, and this process was repeated by 10 times since there were 10 discrete wavelet basis functions. To conclude the La Nina that happened in 2008, we utilized 45 pieces of data to estimate the parameter of each DA model and took the remaining 5 pieces of data to test whether the estimation of the parameter is significant. The results of 10 DA models and the comparison with ARMA are shown in Figure 1.

Figure 1 consists of two parts, the first and second subgraphs describe average temperature simulation and prediction of 10 DA method of early and mid-February, respectively. In the field of atmospheric sciences, the output of a model is called simulation if the observed data is used for fitting the model. Similarly, the output of a model is called

Input: observed data; discrete wavelet functions

Step 1: decompose the observed data utilizing DWT to get lower frequency signal cA_3 and higher frequency signals cD_1 , cD_2 , and cD_3

Step 2: reconstruct lower frequency signal A_3 and higher frequency signals D_1 , D_2 , and D_3 applying Mallat reconstruction theory

Step 3: build ARMA models on A_3 , D_1 , D_2 , and D_3 to get their simulations and predictions

Step 4: add up the simulations and predictions of A_3 , D_1 , D_2 , and D_3 to get final simulations and predictions of each DA model

Step 5: set DA model's predictions as features; set real value as the target

Step 6: split the dataset into the training set and testing set

Step 7: parameter initialization

Step 8: optimize weights and parameters in the process of model training and testing

Step 9: if the times of training and testing equal the iteration set in initialization, get the parameters; else, return to Step 8

Step 10: predict the temperature in later years

Output: a strong hybrid learning model; the simulation and prediction of temperature

ALGORITHM 1: DAX algorithm.

prediction if the observed data is used for predicting. According to this theory, the year of simulation ranges from 1967 to 2016, and the year of prediction ranges from 2017 to 2020. The "Real" in legend means the observed real temperature data, the "ARMA" means the output of the ARMA model, and the other 10 words in legend are the names of 10 discrete wavelet basis functions which correspond to the output of the 10 DA model.

It can be seen from Figure 1 that some of the 10 DA methods outputs are close to the observed data, while some have high deviation which represents low precision. It is reasonable because discrete wavelet transform has advantages in self-adaptation and dealing with the frequency fluctuation in time series. It is worth noting that some of the DA methods such as Haar-ARMA, Db2-ARMA, and Rbio2.4-ARMA outperform other DA models significantly. To make these results more convincing, we evaluate the precision of simulation and prediction in line with the RMSE criterion which is shown in Table 1.

According to Table 1, the RMSE of most DA methods decreases apparently comparing with the single ARMA method, demonstrating the discrete wavelet transform is efficient in denoising. It can be seen from Table 1 that the Haar-ARMA method, Bior2.4-ARMA, Coif2-ARMA, Coif3-ARMA, and Rbio2.4-ARMA have smaller RMSE than other DA methods in simulation and prediction of Early February. But for mid-February, Haar-ARMA, Bior1.3-ARMA, Db2-ARMA, and Rbio1.3 have smaller RMSE than other DA methods. This difference suggests that different discrete wavelet basis functions are suitable for different cases. A stable prediction model must involve high precision and good generalization ability simultaneously. Thus, it is necessary to take advantage of those well-performance DA models to make the simulation and prediction with high precision across the whole time range. Then, we need to utilize the XGBoost algorithm to achieve this goal.

3.2.2. DAX Model and Analysis. Based on Section 3.2.1, we first regarded the outputs of 10 DA methods as the inputs of the XGBoost algorithm and set the observed temperature time series as the target for training. Then, the dataset is divided into a training set and testing set according to the

ratio of 8 : 2. What is more, parameters and hyperparameters must be initialized at the beginning. For example, the learning rate is universally set from 0.01 to 0.35 to control the learning speed. If the learning rate is too small, it will take more time to get the optimal solution or it will not reach the global optimal before the iteration reaches its maximum number. As for max depth, which controls the depth of the decision tree to reflect the fitness, it should not be set too big. According to the set parameters, the DAX method next calculates the optimal parameters through the training and testing process. Among the parameters, the optimal weights are used to combine the DA models and thus get the hybrid DAX model. At last, we obtain the simulation and prediction of early February and mid-February. A comparison between the output of the DAX method and observed data is shown in Figure 2.

Figure 2 consists of two parts; the first subgraph shows the comparison between the observed data and the result of the DAX method of Early February, while the second subgraph represents the average temperature simulation and prediction of mid-February. In Figure 2, the yellow line represents the real observed value, and the blue line represents the simulation and prediction of the DAX method. The year of simulation ranges from 1967 to 2016, while the year of prediction ranges from 2017 to 2020.

Since the blue curves in the two parts both fit closely to the corresponding yellow curves, we analyze Figure 2 overall. We can learn from Figure 2 that the simulation of the proposed DAX algorithm is quite close to the observed data, and the two curves almost coincide together except few points. The good fitness shows that the DAX method does well in establishing the model. We can also notice that two curves tightly fit from 2017 to 2020, which proves that the DAX method has a great predictive ability. DAX model successfully solved the problems that 10 DA models once had and made the simulation and prediction not deviate from the real value even in the area with big fluctuation in the observed real data. As a result, we can conclude that the DAX method extremely improves the precision and stability in modeling, comparing with the results of 10 DA models shown in Figure 1.

In this part, we used RMSE as the criteria for testing the performance of the DAX algorithm as well. We compared

FIGURE 1: The calculation of 10 DA and 1 traditional ARMA method.

the DAX method with 10 DA methods and a single ARMA method. Moreover, as the DAX model is a hybrid model, its weights which combine DA models also have vital importance. We investigate whether the weights would influence the predictive ability. Consequently, we also compared the RMSE of the equal-weighted average (EWA) hybrid model. The comparison is shown in Table 2.

From the comparison of DA, DAX, and EWA methods following the RMSE criteria, it is shown that the hybrid

algorithms decrease RMSE dramatically. Based on Section 3.2.1, we discovered that Haar-ARMA is the only DA model that has low RMSE in both early February and mid-February. We can conclude Haar-ARMA model is the best DA model among 10 DA models because of its accuracy and stability. According to Table 2, the RMSE of EWA and DAX decreased by 19% $((1.3952 - 1.13) / 1.3952 * 100\%)$ and 69.55% $((1.3952 - 0.4248) / 1.3952 * 100\%)$ compared to Haar-ARMA, respectively, for early February. The RMSE of EWA and

TABLE 1: The RMSE comparison of 10 DWTs-ARMA and single ARMA method.

RMSE	Early February	Mid-February
ARMA	2.2609	2.0439
Haar-ARMA	1.3952	1.2037
Bior2.4-ARMA	1.3433	2.8804
Bior1.3-ARMA	1.8536	1.3261
Coif2-ARMA	1.3798	3.6463
Coif3-ARMA	1.3819	2.5605
Db2-ARMA	2.0630	1.3397
Rbio1.3-ARMA	2.2938	1.4525
Rbio2.4-ARMA	1.1412	2.1135
Sym1-ARMA	1.5162	2.0095
Sym2-ARMA	1.8645	1.5401

(a)

(b)

FIGURE 2: The prediction of DAX and its comparison with real value.

TABLE 2: The RMSE comparison between DAX, DA, and EWA methods.

RMSE	Early February	Mid-February
ARMA	2.2609	2.0439
Haar-ARMA	1.3952	1.2037
Bior2.4-ARMA	1.3433	2.8804
Bior1.3-ARMA	1.8536	1.3261
Coif2-ARMA	1.3798	3.6463
Coif3-ARMA	1.3819	2.5605
Db2-ARMA	2.0630	1.3397
Rbio1.3-ARMA	2.2938	1.4525
Rbio2.4-ARMA	1.1412	2.1135
Sym1-ARMA	1.5162	2.0095
Sym2-ARMA	1.8645	1.5401
EWA	1.1300	1.0211
DAX	0.4248	0.4166

DAX decreased by 15.17% $((1.2037-1.0211)/1.2037 * 100\%)$ and 65.39% $((1.2037-0.4166)/1.2037 * 100\%)$ compared to Haar-ARMA, respectively, for mid-February. The precisions of EWA and DAX models also have a significant difference. The minimum RMSE of the EWA algorithm is 1.0211, while the maximum RMSE of the DAX algorithm is 0.4248, which proves that the proposed DAX model outperforms the naive EWA model. Consequently, we can conclude that DAX method improves the simulation and prediction accuracy significantly comparing with the DA method and the naive EWA models, which demonstrates the superiority of our model.

4. Conclusion

This work proposed a novel hybrid algorithm named DAX algorithm which consists of discrete wavelet transform, ARMA, and XGBoost method. This proposed algorithm has better accuracy in both simulation and prediction than the DA models and the EWA model.

According to Figures 1 and 2, we can notice that our proposed algorithm decreases the error of prediction apparently. In Figure 1, some curves are close to observed data, while others that have low accuracy highly deviate from the real data. However, in Figure 2, the curve of prediction almost coincides with the curve of real value which concludes that DAX model has high precision. Moreover, the result of RMSE also proves the former conclusion. The RMSE of the two hybrid algorithms is much smaller than 10 DA algorithms. The bigger decrease in RMSE of the DAX model compared to the EWA model represents that the proposed DAX model has significant superiority in combining models.

After all, the DAX algorithm utilizes the strength of discrete wavelet transform in denoising and dealing with edge fluctuation and thus promotes precision and stability of the model. It successfully fills the gaps in former time series methods which have low precision and proposes a novel idea in dealing with big data prediction simultaneously. It could be used for long-time time series analysis in other areas. Nevertheless, the proposed DAX model also has its

limitations. For example, there might be some nonlinear structures in temperature time series; thus, using ARMA which is a linear model to regress may cause some errors. In the building process of ARMA, the amount of data used for fitting is kind of small which is subject to the data availability. As for the XGBoost method, the initialization of parameters is subjective which might cost more time to get the optimal solution, or it will fail to reach the optimal solution. In further work, we will aim to solve these problems to obtain a more convincing model and a more reasonable explanation.

Data Availability

The data are downloaded from the National Satellite Meteorological Centre (NSMC), and the data are confidential unless permission is obtained.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] C. N. Babu and B. E. Reddy, "Predictive data mining on average global temperature using variants of ARIMA algorithms," in *Proceedings of the 2012 International Conference on Advances in Engineering, Science and Management (ICAESM)*, pp. 256–260, IEEE, Nagapattinam, Tamil Nadu, India, March 2012.
- [2] H. Zhou, H. Li, X. Sun et al., "A temperature forecasting model of Grey-Markov based on seasonal index," *Mathematics in Practice and Theory*, vol. 4, no. 46, pp. 167–173, 2016.
- [3] D. Wang, Z. Cao, B. Chen et al., "Multivariate time series local support vector regression forecast methods for daily temperature," *Journal of System Simulation*, vol. 28, no. 3, pp. 654–660, 2016.
- [4] N. E. Huang, Z. Shen, S. R. Long et al., "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, no. 1971, pp. 903–995, 1998.
- [5] Y. Wang, M. Ding, and M. Zhang, "Application of grey GM (1,1) model in debris flow activity trend prediction in wenchuan county," *Journal of Geological Hazards and Environment Preservation*, vol. 31, no. 1, pp. 23–29, 2020.
- [6] R. Polikar, L. Udpa, S. S. Udpa, and T. Taylor, "Frequency invariant classification of ultrasonic weld," *IEEE Transactions on Ultrasonics Ferroelectrics & Frequency Control*, vol. 45, no. 3, 1998.
- [7] Z. Chen, *Wavelet Analysis and Its Application in Time Series*, Southwest Jiaotong University, Chengdu, China, 2004.
- [8] S. Wu, Y. Yin, D. Zheng, and Q. Yang, "Climate changes in the Tibetan plateau during the last three decades," *Acta Geographical Sinica*, vol. 60, no. 1, pp. 3–10, 2005.
- [9] F.-J. Chang, Y.-M. Chiang, and L.-C. Chang, "Multi-step-ahead neural networks for flood forecasting," *Hydrological Sciences Journal*, vol. 52, no. 1, pp. 114–130, 2007.
- [10] C. Sudheer, R. Maheswaran, B. K. Panigrahi, and S. Mathur, "A hybrid SVM-PSO model for forecasting monthly streamflow," *Neural Computing and Applications*, vol. 24, no. 6, pp. 1381–1389, 2014.

- [11] T. Jiang, X. Liu, Y. Zhu, and H. Qu, "Complex product cost forecast based on GAN-CNN joint network," *Electronic Design Engineering*, vol. 28, no. 17, pp. 174–184, 2020.
- [12] L. Yu, W. Dai, and L. Tang, "A novel decomposition ensemble model with extended extreme learning machine for crude oil price forecasting," *Engineering Applications of Artificial Intelligence*, vol. 47, pp. 110–121, 2016.
- [13] B. Yong, F. Qiao, C. Wang, J. Shen, Y. Wei, and Q. Zhou, "Ensemble neural network method for wind speed forecasting," in *Proceedings of the 2019 IEEE International Workshop on Signal Processing Systems (SiPS)*, October 2019.
- [14] G. Shi, Z. Jing, and L. Fan, "Research on the original oil price prediction based on lasso-xgboost combination method," *Journal of Industrial Technological Economics*, vol. 37, no. 7, pp. 31–37, 2018.
- [15] L. Wang and D. Xiao, "Analysis of non-steady time-series forecast for economy based on ARMA model," *Journal of Wuhan University of Technology (Transportation Science and Engineering Edition)*, no. 1, pp. 133–136, 2004.
- [16] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.
- [17] T. Chen and C. Guestrin, "XgBoost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM, San Francisco, CA, USA, August 2016.
- [18] D. Wang, F. Wang, and C. Niu, "Analysis of wavelet decomposition for multi-step prediction of short-term wind speed," *Power System Protection and Control*, vol. 42, no. 8, pp. 82–89, 2014.