

Article

Hybrid Deep Learning Predictor for Smart Agriculture Sensing Based on Empirical Mode Decomposition and Gated Recurrent Unit Group Model

Xue-Bo Jin ^{1,2,3,*} , Nian-Xiang Yang ^{1,2,3}, Xiao-Yi Wang ^{1,2,3,*}, Yu-Ting Bai ^{1,2,3} , Ting-Li Su ^{1,2,3} and Jian-Lei Kong ^{1,2,3}

¹ School of Computer and Information Engineering, Beijing Technology and Business University, Beijing 100048, China; yangnianxiang@st.btbu.edu.cn (N.-X.Y.); baiyuting@btbu.edu.cn (Y.-T.B.); sutingli@btbu.edu.cn (T.-L.S.); kongjianlei@btbu.edu.cn (J.-L.K.)

² China Light Industry Key Laboratory of Industrial Internet and Big Data, Beijing Technology and Business University, Beijing 100048, China

³ Beijing Key Laboratory of Big Data Technology for Food Safety, Beijing Technology and Business University, Beijing 100048, China

* Correspondence: jinxuebo@btbu.edu.cn (X.-B.J.); wangxy@btbu.edu.cn (X.-Y.W.)

Received: 18 January 2020; Accepted: 26 February 2020; Published: 29 February 2020



Abstract: Smart agricultural sensing has enabled great advantages in practical applications recently, making it one of the most important and valuable systems. For outdoor plantation farms, the prediction of climate data, such as temperature, wind speed, and humidity, enables the planning and control of agricultural production to improve the yield and quality of crops. However, it is not easy to accurately predict climate trends because the sensing data are complex, nonlinear, and contain multiple components. This study proposes a hybrid deep learning predictor, in which an empirical mode decomposition (EMD) method is used to decompose the climate data into fixed component groups with different frequency characteristics, then a gated recurrent unit (GRU) network is trained for each group as the sub-predictor, and finally the results from the GRU are added to obtain the prediction result. Experiments based on climate data from an agricultural Internet of Things (IoT) system verify the development of the proposed model. The prediction results show that the proposed predictor can obtain more accurate predictions of temperature, wind speed, and humidity data to meet the needs of precision agricultural production.

Keywords: sensing data prediction; EMD; convolution operation; GRU; smart sensing; IoT

1. Introduction

Smart agriculture has been capable of offering many solutions to the modernization of agriculture [1]. With the development of Internet of Things (IoT) technology, smart agricultural applications have developed greatly [2–4] in recent years. Thanks to IoT systems, in which a wireless sensor network collects data from sensors deployed at various nodes and sends data over a wireless protocol, the massive data from the IoT agriculture system can be collected, such as temperature, wind speed, and humidity, which can provide information about environmental factors, enabling climate predictions.

The agricultural industry is susceptible to the climate, and a comprehensive understanding of future climate information can generate more benefits for smart agricultural development. The climate prediction has high reference value. Small climate stations in agricultural areas monitoring and

predicting climate changes in real-time can help farmers quickly adjust a planting plan to minimize losses. For the prevention of, and resistance to, risks in the agricultural industry, the support of climate prediction technology is indispensable. The estimation and prediction of climate changes are often based on mathematical models. Some of the predicted models can be established through certain parameter estimation methods [5–8], some use input–output representations [9–11], while others use state–space models [12] or network models [13,14].

To apply the IoT technology to the development of modern agriculture, many researchers have made many outstanding contributions [15–17]. One example is an agricultural Internet of Things system that is used to help monitor plant production processes and enable early warning and rapid diagnosis of major pests and diseases [18]. Hao et al. [19] studied the relationship between farmland environmental factors and crop growth cycle and crop yield based on agricultural big data. Zou et al. [20] proposed a new service-based, grid-based approach to building agricultural IoT systems. Agricultural material information can be used in agricultural engineering and has obvious advantages.

Figure 1 shows a schematic diagram of data collection, communication, prediction, and application layers in agricultural IoT systems. Because the planting is outdoor, battery-powered wireless sensors are constructed to collect the data in the IoT system. The sensors are used to measure the temperature, wind speed, and humidity, then the data is transmitted to the data server for storage by 4G/5G communication. Furthermore, a large quantity of stored data is used to train the deep learning model and to then give the predictions of future temperature, wind speed, and humidity. The climate prediction information is then sent to the manager, and a suggestion for a future planting plan is also given.

The prediction and evaluation of climate trends can assure the quality of agricultural products; in addition, the improvement of climate prediction accuracy in agricultural production areas will help to better protect the development of agriculture. Prediction based on climate data is a difficult problem because data collected from sensors have complex nonlinear relationships with multiple components and are polluted by noise. Therefore, it is impossible to make accurate long-term predictions, but researchers are also convinced that accurate medium-term predictions are possible.

On the other hand, thanks to their high sensing frequency, large-scale data can be collected and stored by IoT systems, making it possible to analyze sensory data, discover new information, and gain insights [21] by using artificial intelligence methods such as deep learning.

This study focuses on medium-term prediction in an agricultural IoT system by processing the collected sensing data with artificial intelligence methods. Medium-term prediction means predicting 20 to 30 steps ahead. As for the climate data collected by the agricultural IoT system shown in Figure 1, we provide a prediction of the temperature, wind speed, and humidity 24 steps ahead based on the deep learning method. To overcome the highly complex nonlinearity, the climate data are first decomposed, and then the deep learning network is used to model component groups. The proposed method can accurately predict changes in the next 24 hours to meet the needs of precision agricultural production the next day.

The remainder of this study is organized as follows. Section 2 introduces the current prediction technology, discusses its advantages and disadvantages, and explains our innovations. Section 3 describes each part of the proposed predictor. Experimental results of the proposed predictor are shown and evaluated in Section 4, and finally Section 5 presents our conclusion.

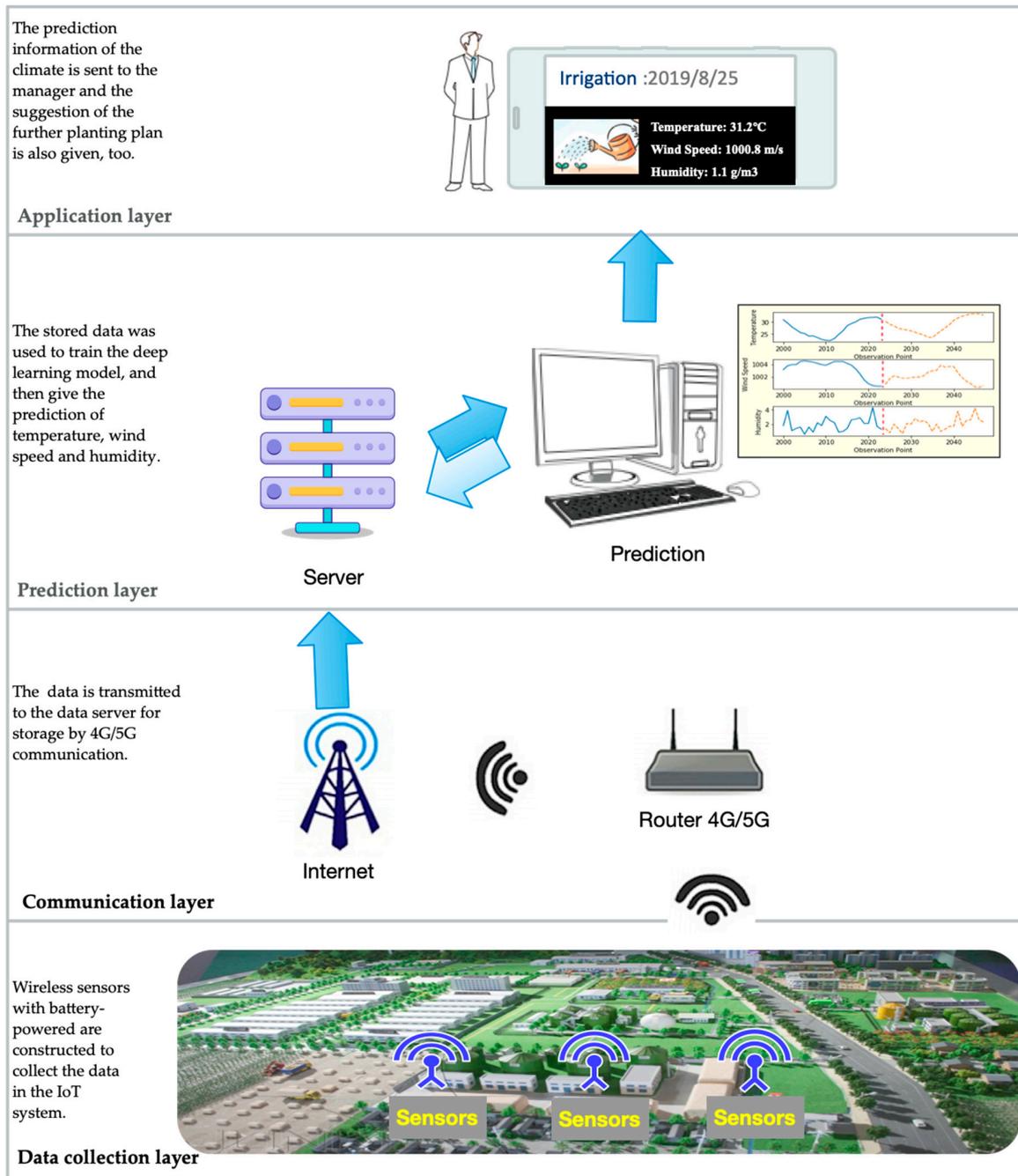


Figure 1. Data collection and prediction in an agricultural Internet of Things (IoT) system.

2. Related Work

As a key data processing technology, prediction methods have been widely used in science, engineering, and other fields. However, since the dynamics of collected sensing data are usually strongly nonlinear, the prediction methods still face enormous challenges.

Time series prediction methods can be roughly divided into two categories: single methods and combined methods. We discuss these 2 kinds of prediction methods in the following subsection, and then based on the current research results, we propose our innovative method.

2.1. Single Methods

Traditional methods include physical analysis models and prediction models based on statistical theory. In the physical models, the equations are established based on the physical relationships between variables, and then the prediction models analyze and predict time series data [22]. Obviously, a suitable mathematical model should be built to fit the historical curve, and the future trend is predicted according to the model trend curve. Statistical models used include the autoregressive moving average (ARMA) [23–25], autoregressive integrated moving average (ARIMA) [26], threshold autoregressive (TAR) model [27], and the hidden Markov model (HMM) [28]. For example, Kumarmahto et al. [29] selected the ARIMA (1,1,2) model for the prediction of agriculture commodity prices.

Based on theoretical assumptions and prior knowledge, model parameters have to be set for these traditional methods. However, due to a lack of background knowledge, this is often difficult. Therefore, traditional methods are less commonly used in complex agricultural applications. Unlike traditional prediction methods, machine learning methods do not require prior physical information, meaning prediction models can be built based on learning algorithms and historical data. Prediction is obtained based on mathematical models, and some parameter identification methods can be used, such as iterative algorithms [30–32], particle-based algorithms [33,34], and recursive and learning algorithms [35–39].

Regression models were initially used for prediction tasks. Bhuriya et al. [40] applied linear regression methods to predict stock prices. Oteros et al. [41] established a multiple regression model by using different factors of pollen concentration to take into account extreme climate events in the Mediterranean climate. Donnelly et al. [42] proposed a method for predicting air quality based on multiple linear regression. These linear regression models face challenges in prediction tasks with highly nonlinear time series data.

Artificial neural networks (ANNs) play a key role in solving nonlinear problems. Mao et al. [43] developed a short-term wind prediction system based on a back-propagation (BP) neural network for numerical climate prediction data, such as wind speed, wind direction, temperature, relative humidity, and atmospheric pressure data. Theoretically, the BP neural network has been proven to enable fit any persistent nonlinear function, however problems exist, such as falling into local minimum values, over-fitting, and excessive training times. These problems can be corrected by optimization. For example, in [44], the particle swarm optimization (PSO) algorithm was combined with the BP network for short-term load prediction of power grids. The study showed that the prediction performance of the proposed model based on the PSO algorithm was better than that of a traditional BP neural network. Bai et al. studied the combined prediction method of a shallow nonlinear autoregressive network (NAR) on the basis of BP [45] and proposed the prediction method from time and space dimensions by using shallow networks [46].

Later studies compared the optimized results with the results of regression models (e.g., [47,48]), proving the effectiveness of the optimization algorithm and promoting the combination of optimization algorithms and neural networks as predictive tools. A recurrent neural network (RNN) [49] for time series prediction has attracted extensive attention from researchers because it could capture the high nonlinearity of time series data. Yadav et al. [50] used climate data to predict average solar radiation through RNN and proposed an adaptive learning rate for RNN. As an improved version of RNN, long short-term memory (LSTM) replaced it and became a popular time series data prediction technology [51,52]. A gated recurrent unit (GRU) [53] inherits the advantages of LSTM, can automatically learn features and model long-term dependent information, and also shows an improvement in calculation speed.

This intelligent method has been applied to intelligent agriculture. Alreshidi [54] explored artificial intelligence technologies for smart, sustainable agriculture. Pantazi et al. [55] used machine learning to predict wheat yield based on online multilayer soil data and satellite imagery crop growth characteristics, and showed that the supervised Kohonen networks had the best overall performance. To utilize scalability for yield prediction, Oliveira et al. [56] developed the geographically weighted

gradient boosting machine (GW-GBM), which was essential to agriculture stakeholders. In [57], authors studied comparisons of deep learning with other existing popular techniques with respect to differences in classification or regression performance. The findings indicated that deep learning provided high accuracy, outperforming commonly used image processing techniques.

Deep learning is used to extract accurate sensor data information from IoT devices deployed in complex environments [58]. In [59], the performance of IoT deep learning applications was optimized by edge computing. As a review, [21] provided a thorough overview of the use of a deep learning method to facilitate analysis and learning in the IoT domain, and pointed out that deep learning models produce high-level abstraction and insight that is fed into the IoT systems to fine-tune and improve services. Fast data analysis and medium- or long-term prediction of collected sensing data are still challenging research aspects.

2.2. Combined Methods

In recent years, combined methods based on machine learning have been proven effective at improving prediction performance, and various hybrid models have been introduced to predict nonlinear time series data. Differing from the so-called single methods in Section 2.1, a combined method here means that several methods are used together in a parallel or serial structure. For example, Yahata et al. [60] combined machine learning techniques to construct sensing methods in an agricultural cyber-physical system, in which big data of agricultural plants and environmental information (e.g., temperature, humidity, solar radiation, soil condition, etc.) were analyzed to mine useful rules for appropriate cultivation. Wu et al. [61] proposed a mixed model, which combined the ARIMA model and regression method based on time and space factors, and produced warnings for daily price changes using neural networks.

A seasonal trend decomposition procedure based on loess, the sequential two-level method, was used to model pollen time series in the air, and this was then used to predict the daily pollen concentration for the next period [62]. The authors explained that analyzing each component of the data separately can identify the source of change in data more accurately than the original undecomposed series. Xiong et al. [63] proposed a novel hybrid method combining STL and extreme learning machines (ELMs) to predict seasonal vegetable prices, which contributed to the development of agriculture. Liu et al. [64] implemented a hybrid method to predict wind speed, in which the wavelet transform was used to decompose the wind speed signal into two components and the approximated signal (one of two components) was modeled by a support vector machine. Zhi et al. [65] selected empirical mode decomposition (EMD) as the decomposition method for the time series, and components with different features in the original hydrological time series were decomposed. Yaslan et al. [66] predicted power load demand using a combined EMD and support vector regression (SVR) model. The decomposition process of EMD was regarded as a denoising procedure for the training data, and the prediction results were compared with those of the SVR algorithm based on different feature vectors. The results showed the superiority of the model in power load prediction.

Through EMD, the mode function (IMF) component is a local characteristic signal. The IMF is based on different time scales of the original time series itself, representing each frequency component in the original signal, which are arranged independently from each other in order from high to low. The EMD decomposition process is a simplification of complex time series. For example, the first high-frequency IMF sequence is treated as a noise term and discarded [67,68].

Qiu et al. [69] proposed a hybrid method based on EMD and deep learning, in which the load demand sequence is decomposed into IMF. The extracted IMF is then modeled using a deep belief network (DBN) to accurately predict the evolution of each IMF. Finally, the predictions of each model are combined by addition to obtain the total output of the load demand. Wang et al. [70] introduced a feedforward neural network (FNN) into an EMD-based prediction framework, proposed a weighted recombination strategy, and performed single-step prediction experiments on four nonlinear, nonstationary sequences. Bedi et al. [71] combined EMD with LSTM to estimate the power demand over

a given time interval. The performance of this method was evaluated by comparing it the prediction results of the EMD of RNN and RMD models. EMD can result in different numbers of trained and online predicted models, but [69–71] does not explain how to solve this problem. Significantly differing from previous studies, we will combine IMF components to achieve the unification of the training model and prediction model in practical applications.

We continue the combination of EMD together with deep learning networks, and our innovative contributions are highlighted as follows.

- (1) The obtained IMF components are analyzed with frequency characteristics, and all the components are divided into 3 groups using a learning network based on convolution operation. Differing from [69–71], this method can solve the “different number problem” of IMF components.
- (2) We present a general agricultural IoT system framework for predicting climate data and obtain accurate medium-term predictions that can meet the needs of precision agricultural production.

3. Hybrid Deep Predictor

The proposed predictor has a hybrid structure, for example using EMD decomposition to reduce the nonlinear complexity and dividing IMFs into 3 groups using Convolution Neural Networks (CNN) networks. For each group, the deep learning network GRU is used for modeling and prediction, and finally all the predictions of the GRU are added to obtain the prediction result.

3.1. Decomposition and Analysis for Time Series

The EMD method decomposes complex signals into a finite number of IMFs based on the frequency characteristics, which should satisfy the following conditions: (1) the absolute value of the difference between the number of zero crossings and extreme points is equal to 0 or 1; and (2) the mean value of the envelope constructed by local maxima and minima must be zero at any point. EMD is an adaptive data processing or mining method and is essentially a smoothing process for time series data.

Assume D_t is the time series to be decomposed and h_e is the expected decomposition result to be obtained. The decomposition process is as follows [72]:

- (1) Fit the maximum and minimum points of D_t with the cubic spline interpolation function to form the upper and lower envelope.
- (2) Calculate the mean of the upper envelope and the lower envelope, denoted as m_e .
- (3) Subtract the mean of m_e by D_t to obtain a new data sequence h_e : $h_e = D_t - m_e$.
- (4) Repeat steps 1-4 until one of the following stop criteria is met: (1) the preset maximum number of iterations is reached; (2) the last separated IMF is small; (3) the maximum or minimum value of the signal is less than 2; (4) h_e is a monotonic curve.
- (5) Treat h_e as an IMF, and calculate the remainder $R_t = D_t - h_e$.
- (6) Use R_t as the new D_t , and repeat steps (1)–(6) until all IMFs are obtained.

We take temperature data as an example to give the results of EMD decomposition. In Figure 2a, all the obtained IMFs are shown over time (from IMF-0 to IMF-8), and correspondingly each sub-picture on the right-hand side (b) is the frequency component. It can be found that each IMF has a specific time and frequency domain, and the frequency components contained in the IMF are reduced from top to bottom.

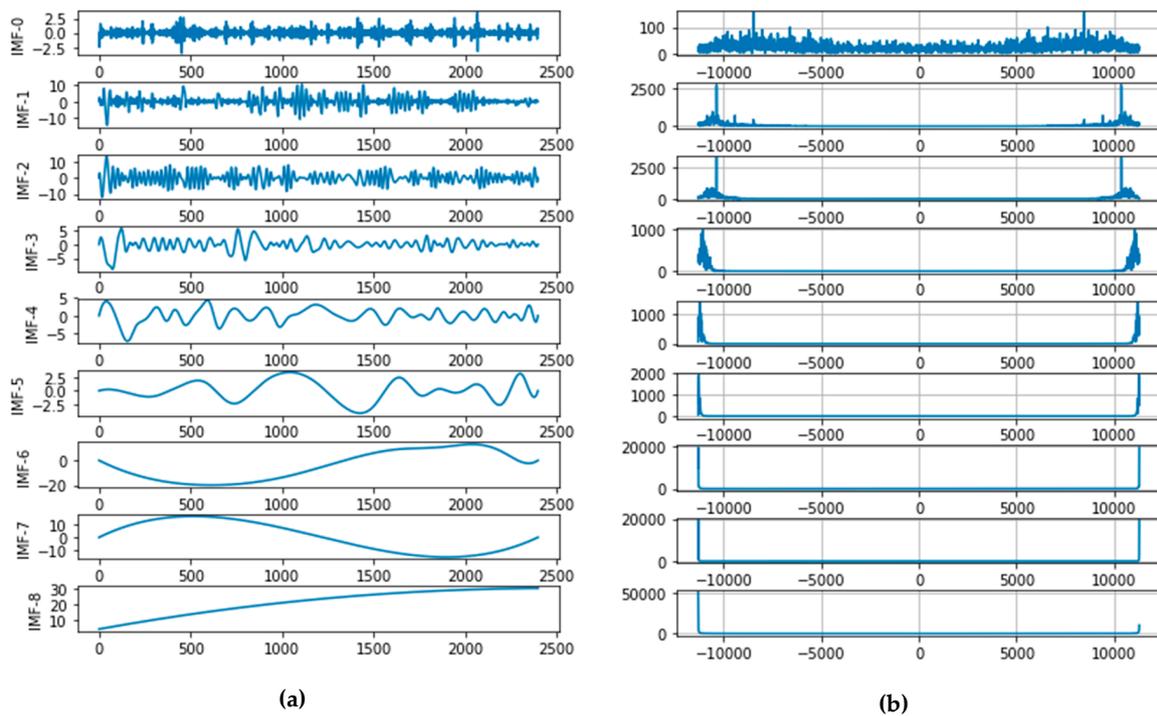


Figure 2. Correspondence of each mode function (IMF) between time and frequency domains after decomposition. IMFs in the (a) time domain and (b) frequency domain.

Moreover, we found that the number of IMFs varies for different periods. As shown in Figure 3, we performed EMD on the three different data intervals $[0, 2400)$, $[2400, 4800)$, and $[4800, 7200]$ of the temperature data, and the number of IMFs obtained was 9, 7, and 8, respectively.

We can note that the number of trained prediction sub-models will be different from the number of IMF components in the training prediction interval. Therefore, it is necessary to combine IMFs into a fixed number according to frequency characteristics.

In this study, according to the frequency characteristics of IMFs, we combined all the IMFs into 3 groups, meaning the decomposition components with similar frequency characteristics will be labeled, grouped, and added together, then for each group, one model will be trained. Therefore, the number of models in each prediction interval will be fixed.

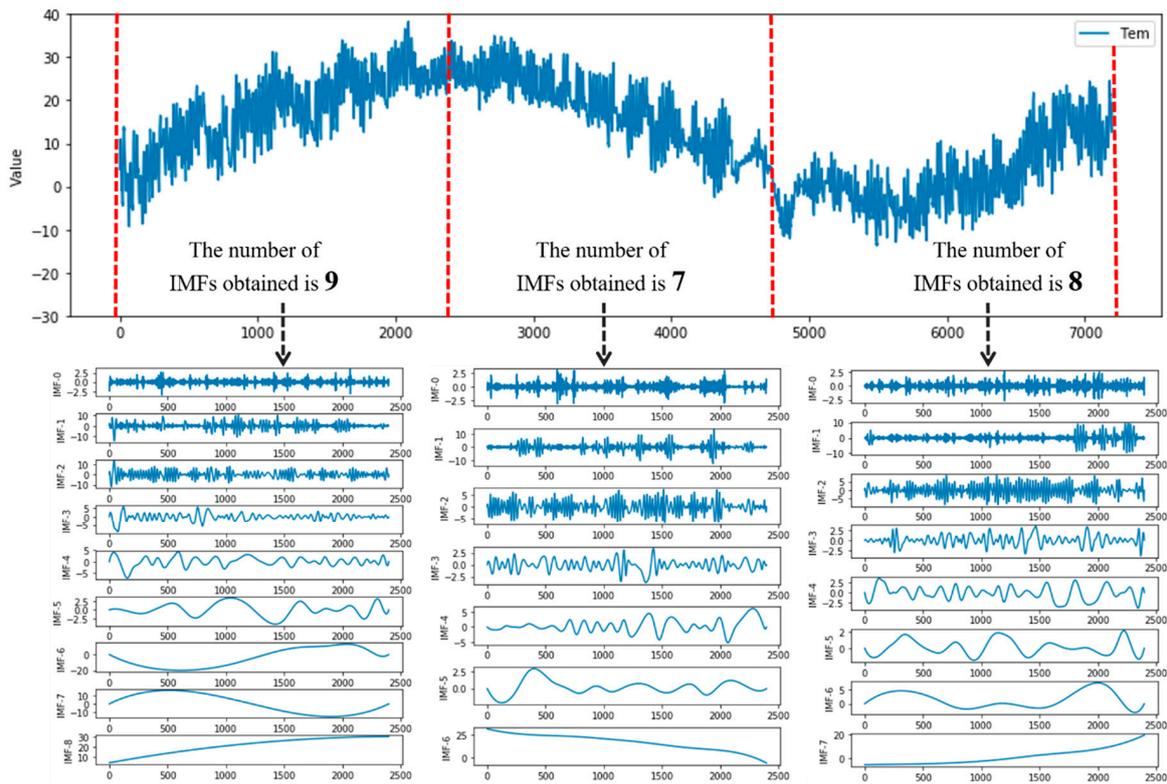


Figure 3. The number of IMFs within different time intervals of temperature data.

3.2. Classification and Combination for IMFs

Figure 4 shows that a one-dimensional convolution operation is calculated for each IMF in the frequency domain to illustrate the different frequency components, with the convolution formula as follows:

$$f(x) * g(x) = \int_{-\infty}^{+\infty} f(\tau) * g(x - \tau) d\tau \tag{1}$$

where $f(x)$ is the convolved function and $g(x)$ is the convolution kernel function. The result of the one-dimensional convolution is equal to the integral of the integrand function $f(\tau) * g(x - \tau)$ on the interval $(-\infty, +\infty)$ and the convolution kernel $g(x)$ is selected as a Gaussian kernel function.

The results in Figure 4 show that IMF-0 contains a wider frequency band. In contrast, IMF-1 and IMF-2 are significantly reduced, but there are still long tails in the cutoff band. Differing from IMF-3 and IMF-4, the downhill is significantly steeper, indicating that fewer frequency components are included. Furthermore, for IMF-5–8, the downslope is almost vertical, and we find that the fluctuations of these components on the time domain map (Figure 2a) are relatively flat.

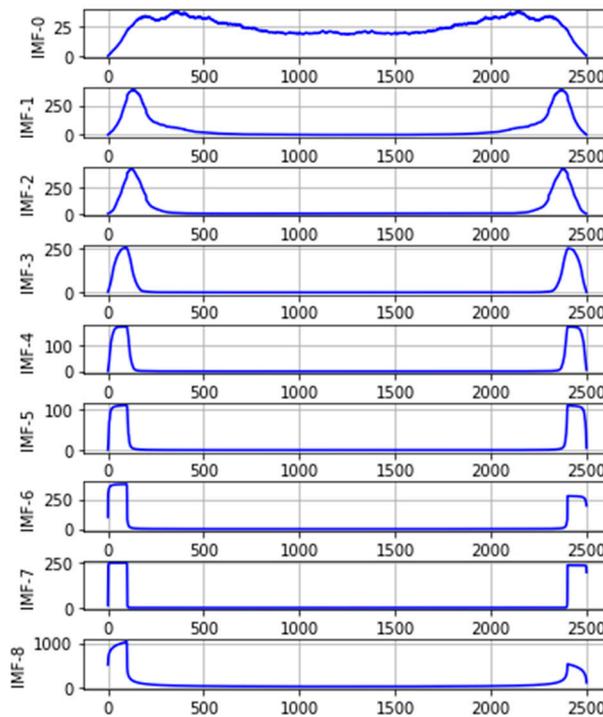


Figure 4. Convolution results for each IMF in the frequency domain.

Figure 4 has shown that the convolution operation can capture the dynamic change of data, so we use the CNN neural network to classify and group IMFs. A one-dimensional CNN is used for feature extraction from IMF sequences. Given an input IMF sequence $X_t, t = 1, \dots, n$ and the convolution kernel function, the filters sequentially perform a local convolution operation on the input features of the previous layer. The output of the convolution is as follows:

$$x_t = \sum_{l=1}^m k_l \times X_{t-l+1}. \quad (2)$$

The rectified linear unit (ReLU) with fast convergence speed is selected as the activation function as follows:

$$f(x_t) = \begin{cases} 0, & x_t \leq 0 \\ x_t, & x_t > 0 \end{cases} \quad (3)$$

Then using flattening and full connection processes [73], a one-dimensional CNN is used to extract the frequency characteristics of the IMFs, using the Softmax classifier to classify the features, finally achieving the network output (i.e., the labels for each IMF). The schematic of the one-dimensional CNN is shown in Figure 5.

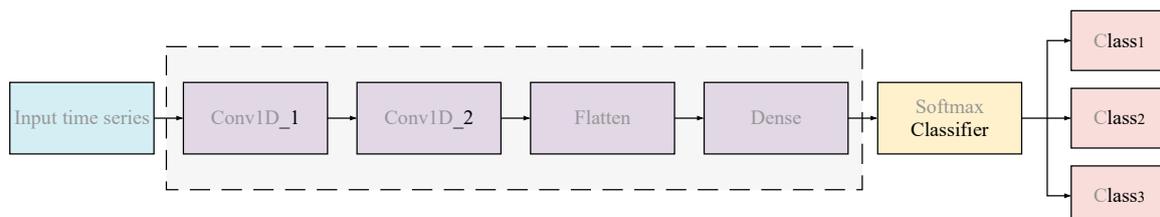


Figure 5. Schematic of a one-dimensional convolutional neural network (CNN).

3.3. Deep Prediction Network for Combined IMFs

Using the known input and output data, the GRU is trained by the stochastic gradient descent algorithm and the optimal weight can be obtained. The GRU network was trained on the sum of IMF sequences in each group. The GRU network consisted of multiple GRU cells, and here the number of hidden layers is set as 2. Shown as Figure 6, $S_t, t = 1, 2, \dots, n$ is the input of the GRU network and $P_t, t = 1, 2, \dots, n$ is the output.

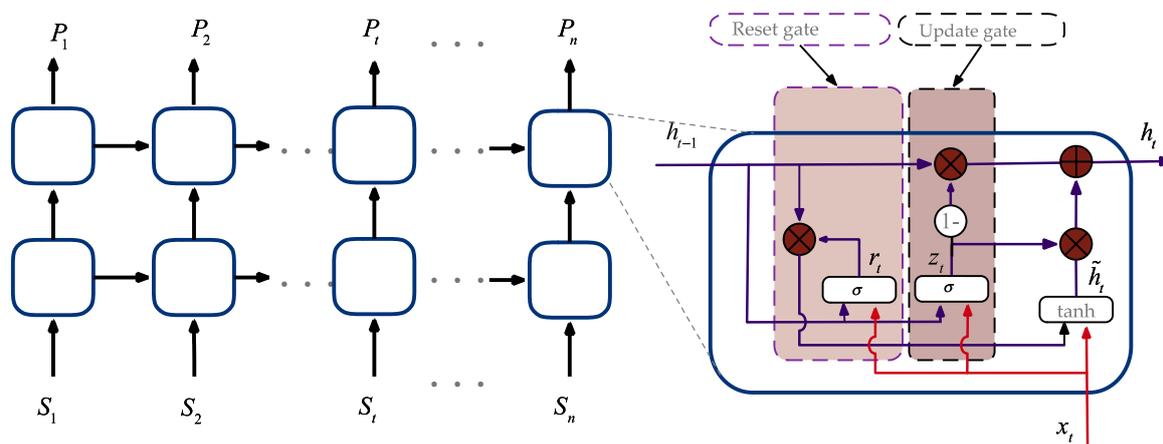


Figure 6. The network structure of the gated recurrent unit (GRU).

The GRU cell uses the update gate to control the degree to which the state information of the previous moment is brought into the current state. The larger the value of the update gate, the more the state information is brought in at the previous moment. The reset gate is similar to the forget gate of LSTM, which is used to control the degree to which the state information of the previous moment is ignored. The smaller the reset gate value, the more information is neglected.

The forward propagation formulas in each GRU cell are as follows [74]:

$$\begin{aligned}
 z_t &= \sigma(a_t U^z + h_{t-1} W^z + b^z) \\
 r_t &= \sigma(a_t U^r + h_{t-1} W^r + b^r) \\
 \tilde{h}_t &= \tanh(a_t U^h + (h_{t-1} \circ r_t) W^h + b^h) \\
 h_t &= (1 - z_t) \circ \tilde{h}_t + z_t \circ h_{t-1}
 \end{aligned} \tag{4}$$

where $a_t \in R^d$ is the input vector of each GRU cell; z_t, r_t, \tilde{h}_t , and h_t stand for the update gate, reset gate, candidate state of the current hidden node, and the active state of the current hidden node output at time t , respectively; U and W are weight matrices to be learned during model training; b represents bias vectors; \circ is an element-wise multiplication; and σ and \tanh are activation functions. The GRU is trained by the gradient descent algorithm and the parameters are continually updated until convergence. The methods proposed in this paper can be applied to other fields, such as water environment prediction and management control systems [75], IoT intelligent systems [76–78], and wireless sensor networks [79–81].

3.4. Model Framework for Smart Agriculture Sensing

In conclusion, based on the discussion in Sections 3.1–3.3, the proposed deep learning predictor is shown in Figure 7. The number of groups is fixed to 3. The model includes the two processes of training and prediction. The first process trains the CNN and GRU based on the IMFs decomposed by EMD. The data is decomposed into IMFs by EMD and the labels are set for each IMF, which are then separated into 3 groups based on frequency characteristics and assigned as Groups 1-3. Then, the CNN

is trained by IMFs and labels, and the sequences are added to each group. Finally, GRU models are trained for each group to obtain three GRU sub-predictors.

The prediction process predicts the future trends of climate data using the trained networks, and it is implemented by summing all the group predictions of GRU models based on IMF groups. The details are given in Figure 7.

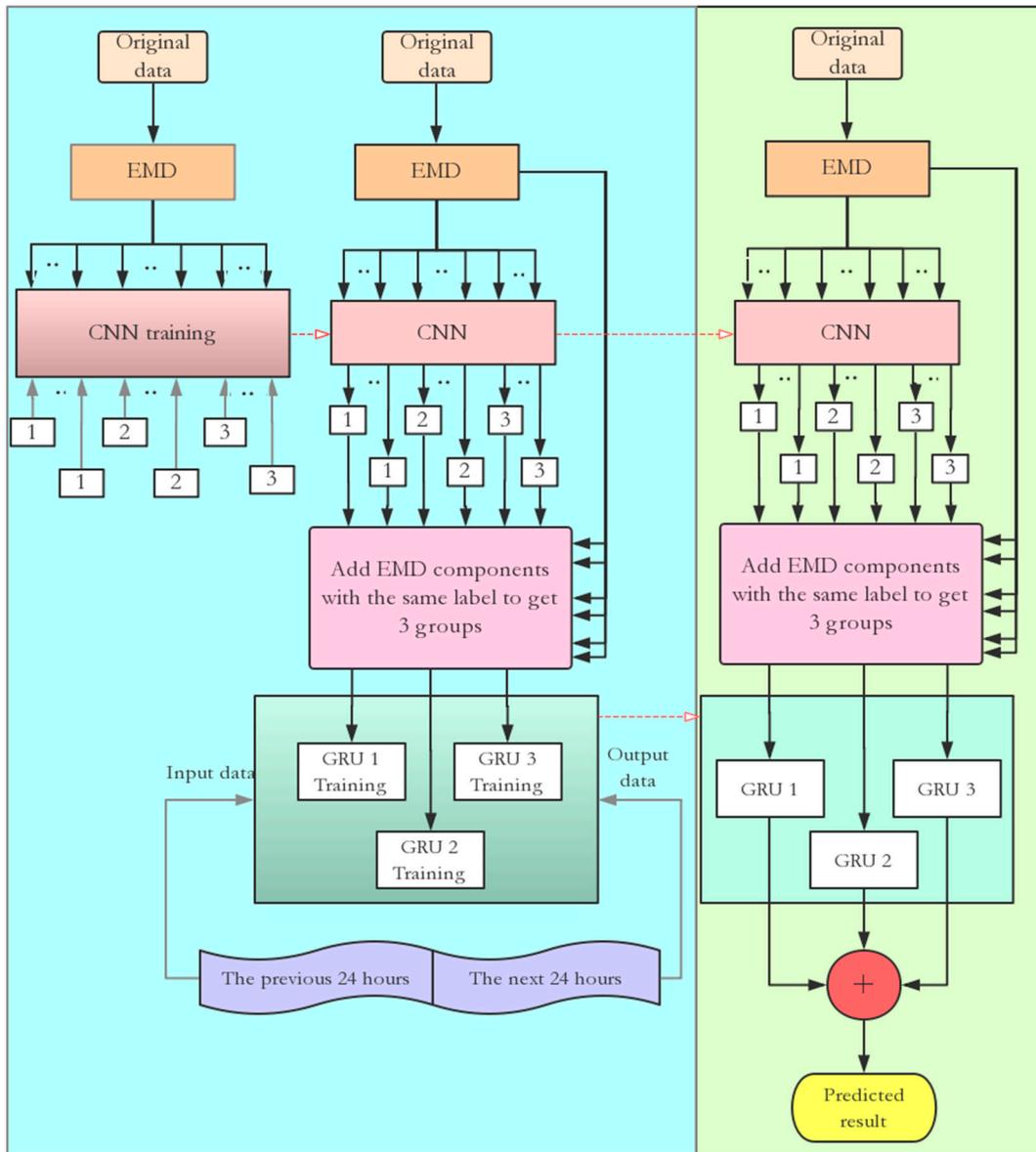


Figure 7. Flowchart of a hybrid predictor for smart agriculture sensing.

4. Experiment Results and Discussion

4.1. Dataset and Experimental Setup

Our experiments are based on the collected sensing data of an agricultural IoT system in Beijing, which collected hourly climate data, including temperature, wind speed, and humidity. The data was obtained from 2016 to 2018 and consisted of 20,013 time series data points. This dataset will be used to train and test the proposed model and other models in Sections 4.2 and 4.3. In the experiments,

the ratio of training to testing sets was 7:3. We need to predict temperature, wind speed, and humidity 24 h ahead of time.

The open-source deep learning library Keras, based on TensorFlow, was used to build all the learning models. All the experiments were performed on a PC server with an Intel CORE CPU i5-4200U at 1.60 GHz, with 4 GB of memory.

To effectively model the deep neural network, hyper-parameters were set based on experience from experiments. The default parameters in Keras were used for deep neural network initialization (e.g., weight initialization). The activation function of the CNN models is ReLU. The CNN had 2 convolutional layers, with 32 convolution kernels in each layer and a logistic loss function. The kernel size was set as 5. The input and output lengths of the CNN were 24. The identification labels were set by one-hot encoding.

The GRU was designed with two layers and the number of neurons was set at 24. For GRU, the Adam method is used by optimizing a predetermined objective function and Huber loss is used to obtain the robust prediction result, because the sensing data always contain noise from the IoT system. The subsequences were trained using GRU to establish sub-predictive models. Tanh is used as the activation function of GRU. In the experiments, the prediction was considered 24 steps ahead, in which we used the climate data during the previous 24 h to predict the next 24 h.

In Case 1 and Case 2, the root mean square error (RMSE) (shown in Equation (5)) was used to measure the difference between the prediction and the collected data.

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (x_{pre}(i) - x_{obs}(i))^2}{N}} \quad (5)$$

where N is the number of the predictive datasets; x_{obs} represents the collected data and x_{pre} is predicted value.

4.2. Case 1: Prediction Performance Analysis of Different Predictors

In this experiment, 6 models are used for comparison with the proposed method, which are RNN [49], LSTM [51], GRU [53], EMDCNN_RNN [49] and EMDCNN_LSTM [51] (which are obtained by decomposing the data using EMD and with classification of the CNN) as the sub-predictors, and finally the sequential two-level method (STL) method from [17]. The temperature, wind speed, and humidity data introduced in Section 4.1 are used to show the prediction result.

In Table 1, a comparison between the proposed method and RNN [49], LSTM [51], GRU [53], EMDCNN_RNN [49] (EMD and CNN-based RNN [49]), EMDCNN_LSTM [51] (EMD and CNN-based LSTM [51]), and STL [17] in terms of RMSE.

Table 1. Comparison of root mean square error (RMSE) of prediction results with the different predictors.

Element	RMSE						
	recurrent neural network(RNN) [40]	long short-term memory(LSTM) [42]	gated recurrent unit(GRU) [56]	sequential two-level method (STL)[17]	EMD and CNN-based RNN(EMDCNN_RNN)	EMD and CNN-based LSTM(EMDCNN_LSTM)	The Proposed Method
Temperature	3.8273	3.8442	3.2939	2.6672	2.5992	2.2688	2.1310
Wind speed	1.3472	1.3499	1.3154	1.3241	1.3249	1.1599	1.1533
Humidity	4.8143	4.8578	4.3844	3.9811	3.9215	3.5128	2.5189

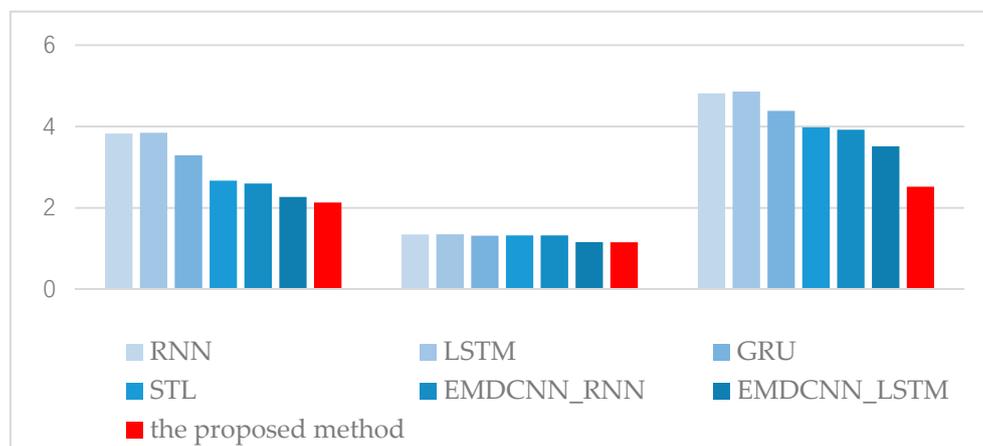
Table 2 gives comparisons the means between the proposed method and RNN [49], LSTM [51], and GRU [53]; between the proposed method and STL, EMDCNN_RNN, and EMDCNN_LSTM; and between the proposed method and EMDCNN_RNN and EMDCNN_LSTM. Figures 8 and 9 give the histogram of RMSEs in Tables 1 and 2.

Table 2. Comparisons of the means of RMSE.

Element	RMSE			
	Mean of RNN [40], LSTM [42], and GRU [56]	Mean of STL [17], EMDCNN_RNN, EMDCNN_LSTM, and the proposed method	STL [17]	Mean of EMDCNN_RNN, EMDCNN_LSTM, and the proposed method
Temperature	3.6551	2.4165	2.6672	2.333
Wind speed	1.3375	1.2405	1.3241	1.2127
Humidity	4.6855	3.4836	3.9811	3.3177

The results show that firstly, the decomposition is necessary as the RMSEs can be significantly reduced. We obtained the mean RMSEs of the decomposition methods of EMD and STL as 2.4165, 1.24055, and 3.48356 for temperature, wind speed, and humidity, respectively. These values are much less than those obtained for single models for RNN, LSTM, and GRU, which were 3.6551, 1.3375, and 4.6855, respectively.

Moreover, EMD outperforms the STL method. The mean RMSEs of all the EMD methods were 2.333, 1.2127, and 3.3177, respectively, which were less than the results of STL. Further, we find that GRU is the best choice as the sub-predictor in Table 1. The prediction RMSE values of the proposed method for temperature were approximately 18.01% and 6.07% lower, respectively, than those of EMDCNN_RNN and EMDCNN_LSTM; for wind speed, the RMSEs were about 12.95% and 0.57% lower, respectively; and for humidity, the RMSEs were about 35.77% and 28.29% lower, respectively. This shows that the proposed method, which used GRU as the sub-predictor, gave the best performance. These improvements are beneficial to agricultural IoT systems. The development of the proposed method means that the temperature prediction accuracy is increased by about 1 degree, which is important for agricultural production.

**Figure 8.** Histogram of RMSE values for predictions of temperature, wind speed, and humidity.

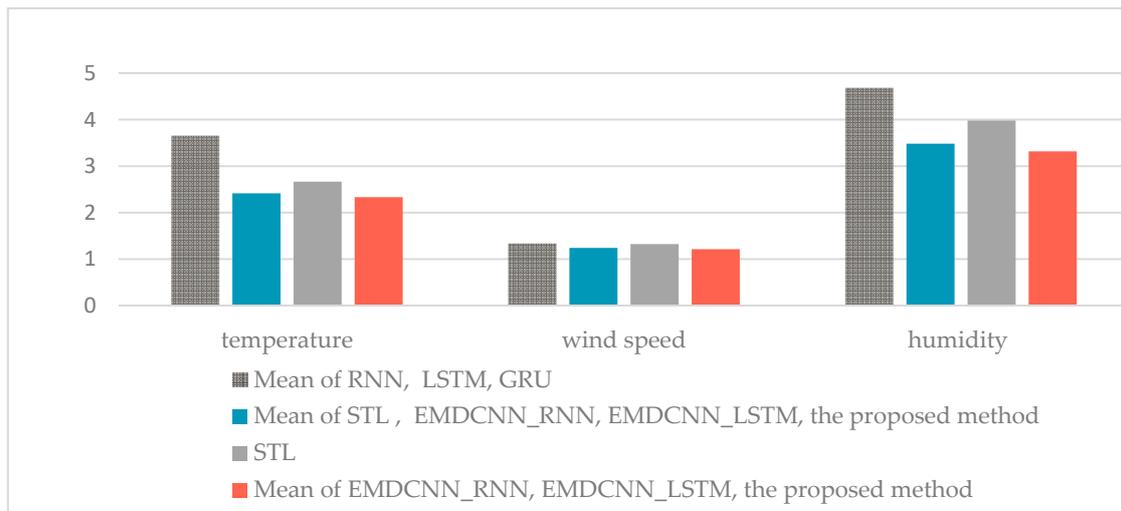


Figure 9. The histogram of different RMSE values.

4.3. Case 2: Prediction Performance Analysis of Different Combinations for IMFs

In this experiment, the collected hourly climate data, including temperature, wind speed, and humidity, in Beijing from January 2016 to January 2018 are used to show the prediction result. We conclude that mode no. 5 (which includes 3 groups, namely Group1: {IMF 0-2}, Group2: {IMF 3-4}, and Group3: {IMF 5-8}) is the suitable mode for the proposed system.

The 10 different modes with different groups are shown in Figure 10 using different color blocks. For example, in mode no. 1, IMF0 is discarded, and the other IMFs are added together as Group 1. As for mode no. 2, all the IMFs are added together as Group 1. Further, mode no. 3 has two groups (i.e., Group 1 (with IMF 0) and Group 2 (the sum of IMF 1 to IMF 8)) and we use 2 color blocks in the third line of Figure 9. Similarly, mode no. 10 has 8 groups, namely Group 1 (with IMF 0), Group 2 (with the sum of IMF 1 and IMF 2), Group 3 (with IMF 3), Group 4 (with IMF 4), Group 5 (with IMF 5), Group 6 (with IMF 6), Group 7 (with IMF 7), and Group 8 (with IMF 8).

Combination Mode	Different Grouping									
Mode No.1 With 1 group	IMF 0	IMF 1	IMF 2	IMF 3	IMF 4	IMF 5	IMF 6	IMF 7	IMF 8	<div style="display: flex; flex-direction: column; gap: 5px;"> <div style="border: 1px solid black; background-color: #f8d7da; padding: 2px;">Group 1</div> <div style="border: 1px solid black; background-color: #fff3cd; padding: 2px;">Group 2</div> <div style="border: 1px solid black; background-color: #d4edda; padding: 2px;">Group 3</div> <div style="border: 1px solid black; background-color: #d1ecf1; padding: 2px;">Group 4</div> <div style="border: 1px solid black; background-color: #c6e0b4; padding: 2px;">Group 5</div> <div style="border: 1px solid black; background-color: #bee5eb; padding: 2px;">Group 6</div> <div style="border: 1px solid black; background-color: #d6d8db; padding: 2px;">Group 7</div> <div style="border: 1px solid black; background-color: #fff3cd; padding: 2px;">Group 8</div> </div>
Mode No.2 With 1 group	IMF 0	IMF 1	IMF 2	IMF 3	IMF 4	IMF 5	IMF 6	IMF 7	IMF 8	
Mode No.3 With 2 groups	IMF 0	IMF 1	IMF 2	IMF 3	IMF 4	IMF 5	IMF 6	IMF 7	IMF 8	
Mode No.4 With 2 groups	IMF 0	IMF 1	IMF 2	IMF 3	IMF 4	IMF 5	IMF 6	IMF 7	IMF 8	
Mode No.5 With 3 groups	IMF 0	IMF 1	IMF 2	IMF 3	IMF 4	IMF 5	IMF 6	IMF 7	IMF 8	
Mode No.6 With 4 groups	IMF 0	IMF 1	IMF 2	IMF 3	IMF 4	IMF 5	IMF 6	IMF 7	IMF 8	
Mode No.7 With 5 groups	IMF 0	IMF 1	IMF 2	IMF 3	IMF 4	IMF 5	IMF 6	IMF 7	IMF 8	
Mode No.8 With 6 groups	IMF 0	IMF 1	IMF 2	IMF 3	IMF 4	IMF 5	IMF 6	IMF 7	IMF 8	
Mode No.9 With 7 groups	IMF 0	IMF 1	IMF 2	IMF 3	IMF 4	IMF 5	IMF 6	IMF 7	IMF 8	
Mode No.10 With 8 groups	IMF 0	IMF 1	IMF 2	IMF 3	IMF 4	IMF 5	IMF 6	IMF 7	IMF 8	

Figure 10. Different modes with different groups shown by different color blocks.

For each mode, we used the different combination modes to train the CNN and obtained different numbers of groups. Then, GRUs were trained for each group. Table 3 gives the comparison of RMSEs of prediction results with different groupings, and Figure 11 shows the histogram of numerical comparisons of RMSEs.

The data in Table 3 and Figure 11 can be interpreted as a large difference in performance across the different groups. For the temperature data, the RMSE of {IMF 1-8} in mode no. 1 is 6.56% lower than mode no. 2, which indicates that removing the first component (noise item) decomposed by EMD is helpful for prediction. The RMSE decreases from mode no. 2 to mode no. 7, but prediction increases in mode no. 8 and mode no. 10. We can note that the performance will be worse if the number of groups exceeds 5.

Table 3. Comparison of RMSEs of prediction results with different groupings.

Combination Mode	Number of Groups	RMSE		
		Temperature	Wind Speed	Humidity
Mode No. 1	1 group	3.2354	2.1989	4.0798
Mode No. 2	1 group	3.4626	2.4560	4.1343
Mode No. 3	2 groups	3.2558	2.3054	3.6562
Mode No. 4	2 groups	2.5474	1.5152	2.9345
Mode No. 5	3 groups	2.1310	1.1533	2.5189
Mode No. 6	4 groups	2.1156	1.1321	2.5166
Mode No. 7	5 groups	2.1093	1.1102	2.5101
Mode No. 8	6 groups	2.9550	1.8350	3.2859
Mode No. 9	7 groups	2.8293	1.7685	3.2855
Mode No. 10	8 groups	3.0985	2.1026	3.5113

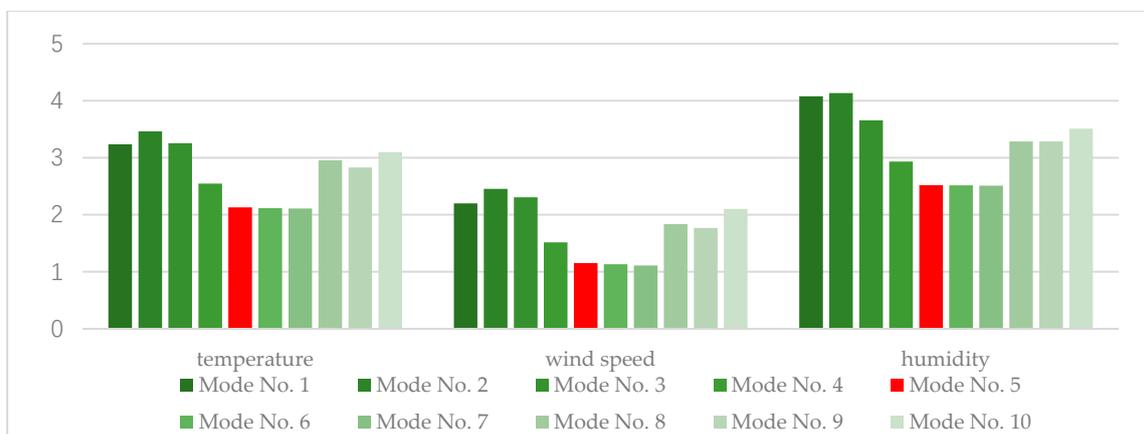


Figure 11. Histogram of numerical comparisons of RMSEs for different combination modes.

Mode no. 5, mode no. 6, and mode no. 7 have similar prediction performance, with RMSE reaching 2.1310, 2.1556, and 2.1093, respectively. Compared with mode no. 5, the RMSE of mode no. 6 or mode no. 7 is slightly reduced, but we have to train 4 or 5 GRUs, so the training parameters are increased by one-third and two-thirds, respectively. In addition to the accuracy of the predictions, we believe that the amount of computation of the IoT system is also important. To reduce the cost of training, a method that requires less calculation should be selected. Therefore, to ensure prediction performance and maintain the cost of parameters, mode no. 5 with 3 groups is a good choice for application of temperature data in agricultural IoT systems.

For wind speed and humidity data, we obtained similar results, so in this study 3 group modes are used for the proposed predictor.

5. Conclusions

In recent years, predictions based on collected sensing data have become an important aspect of IoT applications in many fields, and deep learning has been the most widely used method in many IoT systems because it has excellent nonlinear modeling capabilities.

Through the establishment of climate monitoring stations in agricultural areas, various climate data are collected. In actual IoT agricultural systems, IoT technology has been widely used to predict key climate factors in real time. A three-dimensional network of agro-meteorological data has been built, which has greatly promoted the development of agriculture.

This study proposes a hybrid deep learning predictor based on a learning EMD and GRU group model. The proposed method decomposes data by EMD and extracts local feature components using CNN, then uses a learning network based on convolution operation to classify IMFs based on the frequency feature and trains the GRU as the sub-predictor. The prediction results of the sub-predictor are finally added to obtain the final prediction result.

The proposed predictor has been used to predict temperature, wind speed, and humidity data in an agricultural IoT system. In practical applications, the proposed predictor can obtain accurate predictions for the following 24 hours, providing sufficient climate information for precision production.

Author Contributions: Conceptualization, X.-B.J. and N.-X.Y.; data curation, Y.-T.B. and T.-L.S.; formal analysis, J.-L.K.; methodology, X.-B.J. and N.-X.Y.; software, N.-X.Y.; supervision, X.-Y.W., Y.-T.B., T.-L.S., and J.-L.K.; validation, X.-Y.W., Y.-T.B., and T.-L.S.; validation, N.-X.Y.; writing—original draft, X.-B.J. and N.-X.Y.; writing—review and editing, X.-B.J. and N.-X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China, grant numbers 61673002, 61903009, and 61903008; National Key Research and Development Program of China, grant number 2017YFC1600605, Beijing Municipal Education Commission, grant numbers KM201910011010 and KM201810011005; Young Teacher Research Foundation Project of BTBU, grant number QNJJ2020-26; and Beijing Excellent Talent Training Support Project for Young Top-Notch Teams (2018000026833TD01).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bu, F.; Wang, X. A smart agriculture IoT system based on deep reinforcement learning. *Future Gener. Comput. Syst.* **2019**, *99*, 500–507. [[CrossRef](#)]
2. Hamdan, O.; Shanableh, H.; Zaki, I.; Al-Ali, A.R.; Shanableh, T. IoT-based interactive dual mode smart home automation. In Proceedings of the IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 11–13 January 2019; pp. 1–2.
3. Adriano, D.B.; Budi, W.A. Iot-based Integrated Home Security and Monitoring System. *J. Phys. Conf. Ser.* **2018**, *1140*. [[CrossRef](#)]
4. Farahani, B.; Firouzi, F.; Chang, V.; Badaroglu, M.; Constant, N.; Mankodiya, K. Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare. *Future Gener. Comput. Syst.* **2018**, *78*, 659–676. [[CrossRef](#)]
5. Zhang, X.; Ding, F.; Yang, E.F. State estimation for bilinear systems through minimizing the covariance matrix of the state estimation errors. *Int. J. Adapt. Control Signal Process.* **2019**, *33*, 1157–1173. [[CrossRef](#)]
6. Ma, H.; Pan, J.; Ding, F.; Xu, L.; Ding, W. Partially-coupled least squares based iterative parameter estimation for multi-variable output-error-like autoregressive moving average systems. *IET Control Theory Appl.* **2019**, *13*, 3040–3051. [[CrossRef](#)]
7. Li, M.H.; Liu, X.M.; Ding, F. The filtering-based maximum likelihood iterative estimation algorithms for a special class of nonlinear systems with autoregressive moving average noise using the hierarchical identification principle. *Int. J. Adapt. Control Signal Process.* **2019**, *33*, 1189–1211. [[CrossRef](#)]
8. Liu, S.Y.; Ding, F.; Xu, L.; Hayat, T. Hierarchical principle-based iterative parameter estimation algorithm for dual-frequency signals. *Circuits Syst. Signal Process.* **2019**, *38*, 3251–3268. [[CrossRef](#)]

9. Liu, L.J.; Ding, F.; Xu, L.; Pan, J.; Alsaedi, A.; Hayat, T. Maximum likelihood recursive identification for the multivariate equation-error autoregressive moving average systems using the data filtering. *IEEE Access* **2019**, *7*, 41154–41163. [[CrossRef](#)]
10. Ding, F.; Zhang, X.; Xu, L. The innovation algorithms for multivariable state-space models. *Int. J. Adapt. Control Signal Process.* **2019**, *33*, 1601–1608. [[CrossRef](#)]
11. Wan, L.J.; Ding, F. Decomposition- and gradient-based iterative identification algorithms for multivariable systems using the multi-innovation theory. *Circuits Syst. Signal Process.* **2019**, *38*, 2971–2991. [[CrossRef](#)]
12. Zhang, X.; Ding, F.; Xu, L.; Yang, E.F. State filtering-based least squares parameter estimation for bilinear systems using the hierarchical identification principle. *IET Control Theory Appl.* **2018**, *12*, 1704–1713. [[CrossRef](#)]
13. Bai, Y.; Wang, X.; Jin, X.; Zhao, Z.; Zhang, B. A neuron-based kalman filter with nonlinear autoregressive model. *Sensors* **2020**, *20*, 299. [[CrossRef](#)] [[PubMed](#)]
14. Wang, L.; Zhang, T.; Wang, X.; Jin, X.; Xu, J.; Yu, J.; Zhang, H.; Zhao, Z. An approach of improved Multivariate Timing-Random Deep Belief Net modelling for algal bloom prediction. *Biosyst. Eng.* **2019**, *177*, 130–138. [[CrossRef](#)]
15. Zheng, Y.Y.; Kong, J.L.; Jin, X.B. CropDeep: The Crop Vision Dataset for Deep-Learning-Based Classification and Detection in Precision Agriculture. *Sensors* **2019**, *19*, 1058. [[CrossRef](#)]
16. Elijah, O.; Rahman, T.A. An Overview of Internet of Things (IoT) and Data Analytics in Agriculture: Benefits and Challenges. *IEEE Internet Things J.* **2018**, *5*, 3758–3773. [[CrossRef](#)]
17. Jin, X.; Yu, X.; Wang, X.; Bai, Y.; Su, T.; Kong, J. Deep Learning Predictor for Sustainable Precision Agriculture Based on Internet of Things System. *Sustainability* **2020**, *12*, 1433. [[CrossRef](#)]
18. Bing, F. The research of IOT of agriculture based on three layers architecture. In Proceedings of the International Conference on Cloud Computing and Internet of Things (CCIOT), Dalian, China, 22–23 October 2016.
19. Hao, F.; Luo, X.; Mu, C. Research on Key Technologies of Intelligent Agriculture Based on Agricultural Big Data. In Proceedings of the International Conference on Smart City & Systems Engineering, Hunan, China, 25–26 November 2016.
20. Zou, Y.; Quan, L. A new service-oriented grid-based method for AIoT application and implementation. *Mod. Phys. Lett. B* **2017**, *31*, 1740064. [[CrossRef](#)]
21. Mohammadi, M.; Al-Fuqaha, A.; Sorour, S.; Guizani, M. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Commun. Surv. Tut.* **2018**, *20*, 2923–2960. [[CrossRef](#)]
22. Vezzani, A.; Barkai, E.; Burioni, R. Single Big Jump Principle in Physical Modelling. *Phys. Rev. E* **2019**, *100*, 012108. [[CrossRef](#)]
23. Ma, H.; Pan, J.; Lv, L.; Xu, G.; Ding, F.; Alsaedi, A.; Hayat, T. Recursive algorithms for multivariable output-error-like ARMA systems. *Mathematics* **2019**, *7*, 558. [[CrossRef](#)]
24. Kocak, C. ARMA (p,q) type high order fuzzy time series forecast method based on fuzzy logic relations. *Appl. Soft Comput.* **2017**, *58*, 92–103. [[CrossRef](#)]
25. Perez, E.G.; Ceballos, R.F. Malaria Incidence in the Philippines: Prediction using the Autoregressive Moving Average Models. *Int. J. Eng. Future Tech.* **2019**, *16*, 1–10. [[CrossRef](#)]
26. Ruby-Figueroa, R.; Saavedra, J.; Bahamonde, N.; Cassano, A. Permeate flux prediction in the ultrafiltration of fruit juices by ARIMA models. *J. Membr. Sci.* **2017**, *524*, 108–116. [[CrossRef](#)]
27. Aero, O.; Ogundipe, A. Fiscal Deficit and Economic Growth in Nigeria: Ascertaining a Feasible Threshold. *Soc. Sci. Elec. Publ.* **2018**, *8*, 296–306. [[CrossRef](#)]
28. Guo, H.; Pedrycz, W.; Liu, X. Hidden Markov Models-Based Approaches to Long-term Prediction for Granular Time Series. *IEEE. Trans. Fuzzy Syst.* **2018**, *26*, 2807–2817. [[CrossRef](#)]
29. KumarMahto, A.; Biswas, R.; Alam, M.A. Short Term Forecasting of Agriculture Commodity Price by Using ARIMA: Based on Indian Market. In *Advances in Computing and Data Sciences*; Singh, M., Gupta, P., Eds.; Singapore Communications in Computer and Information Science; Springer: Berlin/Heidelberg, Germany, 2019.
30. Ding, F.; Pan, J.; Alsaedi, A.; Hayat, T. Gradient-based iterative parameter estimation algorithms for dynamical systems from observation data. *Mathematics* **2019**, *7*, 428. [[CrossRef](#)]
31. Ding, F.; Lv, L.; Pan, J.; Wan, X.; Jin, X.B. Two-stage gradient-based iterative estimation methods for controlled autoregressive systems using the measurement data. *Int. J. Control Autom. Syst.* **2020**, *18*. [[CrossRef](#)]

32. Xu, L.; Ding, F. Iterative parameter estimation for signal models based on measured data. *Circuits Syst. Signal Process.* **2018**, *37*, 3046–3069. [[CrossRef](#)]
33. Ding, J.; Chen, J.; Lin, J.X.; Wan, L.J. Particle filtering based parameter estimation for systems with output-error type model structures. *J. Frankl. Inst.* **2019**, *356*, 5521–5540. [[CrossRef](#)]
34. Ding, J.; Chen, J.Z.; Lin, J.X.; Jiang, G.P. Particle filtering-based recursive identification for controlled auto-regressive systems with quantised output. *IET Control Theory Appl.* **2019**, *13*, 2181–2187. [[CrossRef](#)]
35. Ding, F.; Xu, L.; Meng, D.D.; Jin, X.B.; Alsaedi, A.; Hayat, T. Gradient estimation algorithms for the parameter identification of bilinear systems using the auxiliary model. *J. Comput. Appl. Math.* **2020**, *369*, 112579. [[CrossRef](#)]
36. Cui, T.; Ding, F.; Jin, X.B.; Alsaedi, A.; Hayat, T. Joint multi-innovation recursive extended least squares parameter and state estimation for a class of state-space systems. *Int. J. Control Autom. Syst.* **2020**, *18*, 1–13. [[CrossRef](#)]
37. Wang, X.; Zhou, Y.; Zhao, Z.; Wang, L.; Xu, J.; Yu, J. A novel water quality mechanism modeling and eutrophication risk assessment method of lakes and reservoirs. *Nonlinear Dyn.* **2019**, *96*, 1037–1053. [[CrossRef](#)]
38. Wang, Z.; Jin, X.; Wang, X.; Xu, J.; Bai, Y. Hard decision-based cooperative localization for wireless sensor networks. *Sensors* **2019**, *19*, 4665. [[CrossRef](#)]
39. Wang, L.; Zhang, T.; Jin, X.; Xu, J.; Wang, X.; Zhang, H.; Yu, J.; Sun, Q.; Zhao, Z.; Xie, Y. An approach of recursive timing deep belief network for algal bloom forecasting. *Neural Comput. Appl.* **2020**, *32*, 163–171. [[CrossRef](#)]
40. Bhuriya, D.; Kaushal, G.; Sharma, A.; Singh, U. Stock market predication using a linear regression. In Proceedings of the International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 20–22 April 2017.
41. Oteros, J.; García-Mozo, H.; Hervás, C.; Galán, C. Biowater and autoregressive indices for predicting olive pollen intensity. *Int. J. Biometeorol.* **2013**, *57*, 307–316. [[CrossRef](#)]
42. Donnelly, A.; Misstear, B.; Broderick, B. Real time air quality forecasting using integrated parametric and non-parametric regression techniques. *Atmos. Environ.* **2015**, *103*, 53–65. [[CrossRef](#)]
43. Mao, J.D.; Zhang, X.J.; Li, J. Wind Power Forecasting Based on the BP Neural Network. *Appl. Mech. Mater.* **2013**, *341*, 1303–1307. [[CrossRef](#)]
44. Zhaoyu, P.; Li, S.; Zhang, H.; Zhang, N. The Application of the Pso Based BP Network in Short-Term Load Forecasting. *Phys. Procedia* **2012**, *24*, 626–632. [[CrossRef](#)]
45. Bai, Y.; Jin, X.; Wang, X. Compound Autoregressive Network for Prediction of Multivariate Time Series. *Complexity* **2019**, *2019*, 9107167. [[CrossRef](#)]
46. Bai, Y.; Wang, X.; Sun, Q. Spatio-Temporal Prediction for the Monitoring-Blind Area of Industrial Atmosphere Based on the Fusion Network. *Int. J. Environ. Res. Public Health* **2019**, *16*, 3788. [[CrossRef](#)]
47. Elons, A.S.; Magdi, D.A.; Elgendy, M.Y. A proposed model for predicting the drilling path based on hybrid Pso-Bp neural network. In Proceedings of the SAI Computing Conference (SAI), London, UK, 13–15 July 2016.
48. Zheng, B.H. Material procedure quality forecast based on genetic BP neural network. *Mod. Phys. Lett. B* **2017**, *31*, 19–21. [[CrossRef](#)]
49. Wang, Y.; Wang, Y.; Lui, Y.W. Generalized Recurrent Neural Network accommodating Dynamic Causal Modeling for functional MRI analysis. *Neuroimage* **2018**, *178*, 385–402. [[CrossRef](#)]
50. Yadav, A.P.; Kumar, A.; Behera, L. RNN based solar radiation forecasting using adaptive learning rate. In Proceedings of the International Conference on Swarm, Evolutionary, and Memetic Computing, Chennai, India, 19–21 December 2013.
51. Lin, H.; Shi, C.; Wang, B.; Chan, M.F.; Ji, W. Towards real-time respiratory motion prediction based on long short-term memory neural networks. *Phys. Med. Biol.* **2019**, *64*, 085010. [[CrossRef](#)] [[PubMed](#)]
52. Zhang, D.; Lindholm, G.; Ratnaweera, H. Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring. *J. Hydrol.* **2018**, *556*, 409–418. [[CrossRef](#)]
53. Rui, Z.; Wang, D.; Yan, R.; Mao, K.; Fei, S.; Wang, J. Machine Health Monitoring Using Local Feature-Based Gated Recurrent Unit Networks. *IEEE T. Ind. Electron.* **2017**, *65*, 1539–1548.
54. Alreshidi, E.A. Smart sustainable agriculture (SSA) solution underpinned by internet of things (IoT) and artificial intelligence (AI). *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 93–102. [[CrossRef](#)]

55. Pantazia, X.E.; Moshoua, D.; Alexandridis, T.; Whetton, R.L.; Mouazenc, A.M. Wheat yield prediction using machine learning and advanced sensing techniques. *Comput. Electron. Agric.* **2016**, *121*, 57–65. [[CrossRef](#)]
56. Oliveira, I.; Cunha, R.L.F.; Silva, B.; Netto, M.A.S. A Scalable Machine Learning System for Pre-Season Agriculture Yield Forecast. *arXiv* **2018**, arXiv:1806.09244.
57. Andreas, K.; Francesc, X.P. Deep Learning in Agriculture: A Survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90.
58. Zheng, Y.Y.; Kong, J.L.; Jin, X.B. Probability Fusion Decision Framework of Multiple Deep Neural Networks for Fine-Grained Visual Classification. *IEEE Access* **2019**, *7*, 122740–122757. [[CrossRef](#)]
59. Li, H.; Ota, K.; Dong, M. Learning IoT in edge: Deep learning for the Internet of Things with edge computing. *IEEE Netw.* **2018**, *32*, 96–101. [[CrossRef](#)]
60. Yahata, S.; Onishi, T.; Yamaguchi, K.; Ozawa, S.; Kitazono, J.; Ohkawa, T.; Yoshida, T.; Murakami, N.; Tsuji, H. A hybrid machine learning approach to automatic plant phenotyping for smart agriculture. In Proceedings of the International Joint Conference on Neural Networks, Anchorage, AK, USA, 14–19 May 2017.
61. Wu, H.; Wu, H.; Zhu, M.; Chen, W.; Chen, W. A new method of large-scale short-term forecasting of agricultural commodity prices: Illustrated by the case of agricultural markets in Beijing. *J. Big Data* **2017**, *4*, 1. [[CrossRef](#)]
62. Jin, X.B.; Yang, N.; Wang, X.; Bai, Y.; Su, T.; Kong, J. Integrated predictor based on decomposition mechanism for PM2.5 long-term prediction. *Appl. Sci.* **2019**, *9*, 4533. [[CrossRef](#)]
63. Xiong, T.; Li, C.; Bao, Y. Seasonal forecasting of agricultural commodity price using a hybrid STL and ELM method: Evidence from the vegetable market in China. *Neurocomputing* **2018**, *275*, 2831–2844. [[CrossRef](#)]
64. Liu, D.; Niu, D.; Wang, H.; Fan, L. Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm. *Renew. Energy* **2014**, *62*, 592–597. [[CrossRef](#)]
65. Wang, Z.Y.; Qiu, J.; Li, F.F. Hybrid models combining EMD/EEMD and ARIMA for Long-term streamflow forecasting. *Water* **2018**, *10*, 853. [[CrossRef](#)]
66. Yaslan, Y.; Bican, B. Empirical mode decomposition based denoising method with support vector regression for time series prediction: A case study for electricity load forecasting. *Measurement* **2017**, *103*, 52–61. [[CrossRef](#)]
67. Kumar, S.; Panigrahy, D.; Sahu, P.K. Denoising of Electrocardiogram (ECG) signal by using empirical mode decomposition (EMD) with non-local mean (NLM) technique. *Biocybern. Biomed. Eng.* **2018**, *38*, 297–312. [[CrossRef](#)]
68. Wang, J.; Wei, Q.; Zhao, L.; Tao, Y.; Rui, H. An improved empirical mode decomposition method using second generation wavelets interpolation. *Digit. Signal Process.* **2018**, *79*, 164–174. [[CrossRef](#)]
69. Qiu, X.; Ren, Y.; Suganthan, P.N.; Amaratunga, G.A.J. Empirical Mode Decomposition based ensemble deep learning for load demand time series forecasting. *Appl. Soft Comput.* **2017**, *54*, 246–255. [[CrossRef](#)]
70. Wang, J.; Tang, L.; Luo, Y.; Peng, G. A weighted EMD-based prediction model based on TOPSIS and feed forward neural network for noised time series. *Knowl. Based Syst.* **2017**, *132*, 167–178.
71. Bedi, J.; Toshniwal, D. Empirical Mode Decomposition Based Deep Learning for Electricity Demand Forecasting. *IEEE Access* **2018**, *6*, 49144–49156. [[CrossRef](#)]
72. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.C.; Chi, C.T.; Liu, H.H. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. Math. Phys. Eng. Sci.* **1998**, *454*, 903–995. [[CrossRef](#)]
73. Salamon, J.; Bello, J.P. Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Process. Lett.* **2017**, *24*, 279–283. [[CrossRef](#)]
74. Yang, W.; Zuo, W.; Cui, B. Detecting malicious urls via a keyword-based convolutional gated-recurrent-unit neural network. *IEEE Access* **2019**, *7*, 29891–29900. [[CrossRef](#)]
75. Yu, J.; Deng, W.; Zhao, Z.; Wang, X.; Xu, J.; Wang, L.; Sun, Q.; Shen, Z. A hybrid path planning method for an unmanned cruise ship in water quality sampling. *IEEE Access* **2019**, *7*, 87127–87140. [[CrossRef](#)]
76. Yang, Y.; Bai, Y.; Wang, X.; Wang, L.; Jin, X.; Sun, Q. Group Decision-Making Support for Sustainable Governance of Algal Bloom in Urban Lakes. *Sustainability* **2020**, *12*, 1494. [[CrossRef](#)]
77. Zhao, Z.; Yao, P.; Wang, X.; Xu, J.; Wang, L.; Yu, J. Reliable flight performance assessment of multirotor based on interacting multiple model particle filter and health degree. *Chin. J. Aeronaut.* **2019**, *32*, 444–453. [[CrossRef](#)]

78. Wang, X.; Zhou, Y.; Zhao, Z.; Wei, W.; Li, W. Time-Delay System Control Based on an Integration of Active Disturbance Rejection and Modified Twice Optimal Control. *IEEE Access* **2019**, *7*, 130734–130744. [[CrossRef](#)]
79. Wang, F.; Su, T.; Jin, X.; Zheng, Y.; Kong, J.; Bai, Y. Indoor Tracking by RFID Fusion with IMU Data. *Asian J. Control* **2019**, *21*. [[CrossRef](#)]
80. Bai, Y.; Wang, X.; Jin, X.; Su, T.; Kong, J. Adaptive filtering for MEMS gyroscope with dynamic noise model. *ISA Trans.* **2020**. [[CrossRef](#)] [[PubMed](#)]
81. Jin, X.; Yang, N.; Wang, X.; Bai, Y.; Su, T.; Kong, J. Deep Hybrid Model Based on EMD with Classification by Frequency Characteristics for Long-Term Air Quality Prediction. *Mathematics* **2020**, *8*, 214. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).