mathematics

Article

Adaptive Broad Echo State Network for Nonstationary Time Series Forecasting

Wen-Jie Liu [1,2,3], Yu-Ting Bai [1,2,3,*], Xue-Bo Jin [1,2,3,*], Ting-Li Su [1,2,3] and Jian-Lei Kong [1,2,3]

1 School of Artificial Intelligence, Beijing Technology and Business University, Beijing 100048, China
2 Beijing Laboratory for Intelligent Environmental Protection, Beijing Technology and Business University, Beijing 100048, China
3 State Environmental Protection Key Laboratory of Food Chain Pollution Control, Beijing Technology and Business University, Beijing 100048, China
* Correspondence: baiyuting@btbu.edu.cn (Y.-T.B.); jinxuebo@btbu.edu.cn (X.-B.J.)

Abstract: Time series forecasting provides a vital basis for the control and management of various systems. The time series data in the real world are usually strongly nonstationary and nonlinear, which increases the difficulty of reliable forecasting. To fully utilize the learning capability of machine learning in time series forecasting, an adaptive broad echo state network (ABESN) is proposed in this paper. Firstly, the broad learning system (BLS) is used as a framework, and the reservoir pools in the echo state network (ESN) are introduced to form the broad echo state network (BESN). Secondly, for the problem of information redundancy in the reservoir structure in BESN, an adaptive optimization algorithm for the BESN structure based on the pruning algorithm is proposed. Thirdly, an adaptive optimization algorithm of hyperparameters based on the nonstationary test index is proposed. In brief, the structure and hyperparameter optimization algorithms are studied to form the ABESN based on the proposed BESN model in this paper. The ABESN is applied to the data forecasting of air humidity and electric load. The experiments show that the proposed ABESN has a better learning ability for nonstationary time series data and can achieve higher forecasting accuracy.

Keywords: time series forecasting; echo state network; broad learning system; adaptive optimization algorithm

MSC: 68T07

Citation: Liu, W.-J.; Bai, Y.-T.; Jin, X.-B.; Su, T.-L.; Kong, J.-L. Adaptive Broad Echo State Network for Nonstationary Time Series Forecasting. *Mathematics* **2022**, *10*, 3188. https://doi.org/10.3390/math10173188

Academic Editor: Paweł Kliber

Received: 4 July 2022
Accepted: 30 August 2022
Published: 3 September 2022

1. Introduction

Nonstationary time series represent a set of data in which the mean values change with time. They exist widely in the real world [1], such as in the social sciences [2,3], meteorological industry [4–6], financial markets [7–9], modern agriculture [10–14], and electric power field [15,16]. It is vital to analyze the regularity of time series data and predict their trend for system management and control, by collecting and organizing historical data in social phenomena, identifying and filtering these data, and finding the trend of the social phenomenon over time by analyzing the time series, thereby obtaining a more accurate model structure, and using this model to predict the future changes of the social phenomenon. Time series forecasting plays a crucial part in the advanced perception and control of various systems.

Statistical methods and neural networks have been widely used in the field of time series forecasting [17–19]. The classical methods include the autoregression moving average model [20], radial basis function neural network [21], relevant vector machine [22], long short-term memory (LSTM) [23], gated recurrent unit (GRU) [24], recurrent neural network (RNN) [25], echo state network (ESN) [26], broad learning system (BLS) [27], and fuzzy model [28]. For the methods above, the statistical models cannot fully exploit the internal correlation information in complex nonstationary time series, which disables it in real-world applications. Deep learning networks can achieve a strong learning ability by stacking layers with gradient optimization algorithms, but the time and space costs in deep learning are large. The BLS is proposed to improve the training efficiency with similar accuracy to deep learning networks [27]. Meanwhile, the ESN has attracted wide attention as a recurrent neural network model, which is a lightweight structure. Many scholars have studied forecasting models based on BLS and ESN. The BLS achieves the horizontal expansion of network units rather than the vertical deepening, which is the biggest structural difference from deep learning networks [29]. In the BLS, the original data features are extracted by the mapping layer, and the mapping layer's output is set as the input of the enhancement layer. The final output weight matrix is solved by the pseudo-inverse method. If the model performance is poor, incremental learning algorithms can be applied to optimize the mapping or enhancement layers. The ESN can relatively solve the problem of slow convergence and local minima in RNN. The ESN uses typical reservoir computing, which consists of a dynamical reservoir pool with sparsely connected neurons. It has a certain short-term memory capability because of the echo state property. It only needs to train the output weight matrix, which is a simple linear regression process. The existing studies show that BLS and ESN have certain advantages in data feature extraction and regression. Meanwhile, there are disadvantages in the two networks, in which the reservoir structure may be redundant and hyperparameters are difficult to determine. Therefore, how to merge different networks to the extent of the learning performance for the time series forecasting problem has been an important issue.

To improve the performance of a single machine learning model on nonstationary time series prediction as described above, the ABESN is proposed by fusing the BLS with ESN and introducing the optimization algorithms. In the framework of BLS, the reservoir of ESN is introduced to the enhancement layer. The two models are merged structurally to form the new BESN model. For the reservoir characteristics, the adaptive optimization algorithm of the BESN structure based on the pruning algorithm is proposed. It can solve the redundancy problem of reservoir information and reduce the structural complexity of the model. For the fitting problem of nonstationary time series for the BESN, the adaptive optimization algorithm is proposed for the hyperparameter. It is based on the index analysis of nonstationary time series with the ADF test. The optimization algorithms of the structure and hyperparameter can adjust the BESN adaptively. The fused structure and the optimization algorithms form the ABESN which can be applied in time series forecasting. Relative to the general optimization algorithm, the optimization algorithm of the ABESN model contains a structural optimization algorithm and a parametric optimization algorithm, aimed at improving the prediction performance of the network model on nonstationary time series by optimizing from two different perspectives.

This paper is organized as follows: Section 2 introduces the related methods in time series forecasting; Section 3 presents the proposed adaptive broad echo state network structure and the concrete optimization algorithms; Section 4 presents the experimental results and analysis; Section 5 summarizes the method.

2. Related Works

2.1. Time Series Forecasting Methods

This section introduces the classical methods used in the field of time series forecasting.

The early forecasting methods are mainly the statistical methods based on the stochasticity theory, including the autoregressive (AR) model [30], the autoregressive moving average (ARMA) model [31], and the autoregressive integrated moving average (ARIMA) model [32]. The AR uses a regression equation which is built by the correlation between historical and current data. ARMA and ARIMA mainly solve the problem of stochastic variation terms.

With the rapid development of computer hardware, machine learning begins to play an important role. The classical machine learning models include the support vector machine (SVM) [33], long short-term memory (LSTM) [34], gated recurrent unit (GRU) [35], ESN, and BLS. The SVM is less affected by the noise in small sample datasets. It can cover complex decision boundaries, but the process of adjusting parameters is complicated. Both the LSTM and the GRU are improvements of the RNN. They solve the gradient disappearance and explosion caused by the accumulation of too much error in the RNN. The GRU and LSTM introduce the concept of the gate. The LSTM consists of three gates, namely, the input, output, and forget gates. The GRU consists of reset and update gates. The structure of the GRU is simpler than that of the LSTM, and the GRU has been the mainstream in time series forecasting. The most obvious feature of the ESN is the reservoir computing and the echo state attribute which can quickly train the network. The BLS is proposed relative to the deep learning network. The neurons in the BLS are expanded horizontally rather than in the vertical direction.

Statistical methods can analyze stationary and linear time series data. Their forecasting results are usually worse when the time series are nonstationary and nonlinear. Machine learning methods mainly rely on the characteristics of the data, the structure of the model, and computer resources. Scholars have tried to solve these problems by designing new structures. In this paper, we propose the ABESN to improve the forecasting accuracy by considering the occupied resources for the training.

2.2. Standard Broad Learning System

The BLS is based on the random vector functional link network [29], which is designed to promote the network training speed and keep comparable the forecasting accuracy compared to the deep learning network model [27]. The BLS contains a mapping layer, an enhancement layer, and an output layer. The mapping layer can extract the features of the original data. The enhancement layer is designed to reinforce the nonlinear fitting ability. The incremental learning algorithm can help determine the number of nodes in the enhancement and mapping layers. Meanwhile, some scholars have optimized the BLS in terms of the feature extraction ability and the objective function. The classical improvements include the robust version of BLS with L1 regularization, L2 regularization, and elastic net regularization. They optimize the objective function on the premise that both the training error and the weights obey the Laplace distribution. Then they can reduce the effect of noises and avoid the risk of overfitting [36]. The least p-norm-based BLS maintains the robustness to different types of noises based on the adaptive filtering theory [37]. Hierarchical broad learning combines deep learning with BLS with structural optimization [38]. It adds a hidden layer to the mapping layer of BLS, which improves the fitting accuracy, but the generalization performance declines. The recurrent BLS and gated BLS are the networks derived from the recurrent solution of the RNN and the gate idea of the LSTM. They indirectly eliminate the problem of gradient disappearance in RNN and LSTM and improve the network training efficiency [39].

The horizontal extension structure and incremental learning algorithm help the BLS become an efficient network. However, the internal features of the data cannot be well extracted using only the mapping and enhancement layer when the time series data are strongly nonstationary and nonlinear. Meanwhile, the hyperparameters of the BLS also need to be determined using an optimal method.

2.3. Echo State Network

The ESN can avoid the problems of slow convergence and a local minimum in the RNN training [40]. The property of the echo state promotes the dynamic ability of the network, which is suitable for nonlinear system modeling and time series forecasting.

Some scholars have tried to improve the ESN by taking its advantages and optimization algorithms. The adaptive elastic ESN [41] was proposed to solve the covariance problem and the sparse solution of multivariate time series. The multi-reservoir ESN based on sparse Bayesian [42] is aimed at the random initialization of the connection matrix of the reservoir. A fast subspace decomposition ESN was presented to solve the ill-posed problem of multivariate time series forecasting.

The main feature of the ESN is the sparsely connected reservoir which maps the raw data to a high-altitude reservoir [43]. The output connection weights are trained in the high-latitude state space using linear regression, while the other random connection weights remain constant. It brings in the short-term memory capability for the ESN with the echo state property and state update method. However, ESN often suffers from ill-conditioning problems during training, especially when multicollinearity exists in the network. Moreover, the parameter optimization of ESN usually adopts the batch gradient descent method which has been the basic algorithm for unconstrained optimization, but the results are usually not optimal.

It can be found from the literature review above that there is redundant information in the reservoir of ESN sparse connections. The random initialization of connection weights may lead to network instability and poor generalization ability. The enhancement layer of the BLS uses simple nonlinear operations to obtain the output, which cannot fully extract the features of nonstationary and nonlinear time series data. In response to the problems above, the main contributions of this paper are as follows:

1. A novel structure of the BESN is built with the integration of the ESN reserve pool and the BLS enhancement layer;
2. An adaptive structure optimization algorithm based on the pruning algorithm is applied to the enhancement layer of ABESN to reduce the confidence redundancy of the reservoir and reduce the complexity of the ABESN network;
3. For the characteristics of nonstationary time series, the ADF test is introduced to determine the nonstationary indicators, and a hyperparameter-based adaptive optimization algorithm is proposed to improve the prediction ability of the ABESN model in nonstationary time series.

3. Adaptive Broad Echo State Network

3.1. ABESN Framework

In this paper, the reservoir structure of the ESN is introduced into the enhancement layer of the BLS framework. Meanwhile, adaptive optimization algorithms are proposed for the reservoirs in the enhancement layer and network hyperparameters. The proposed network for time series forecasting is called the adaptive broad echo state network (ABESN). The framework of the ABESN forecasting method is shown in Figure 1.

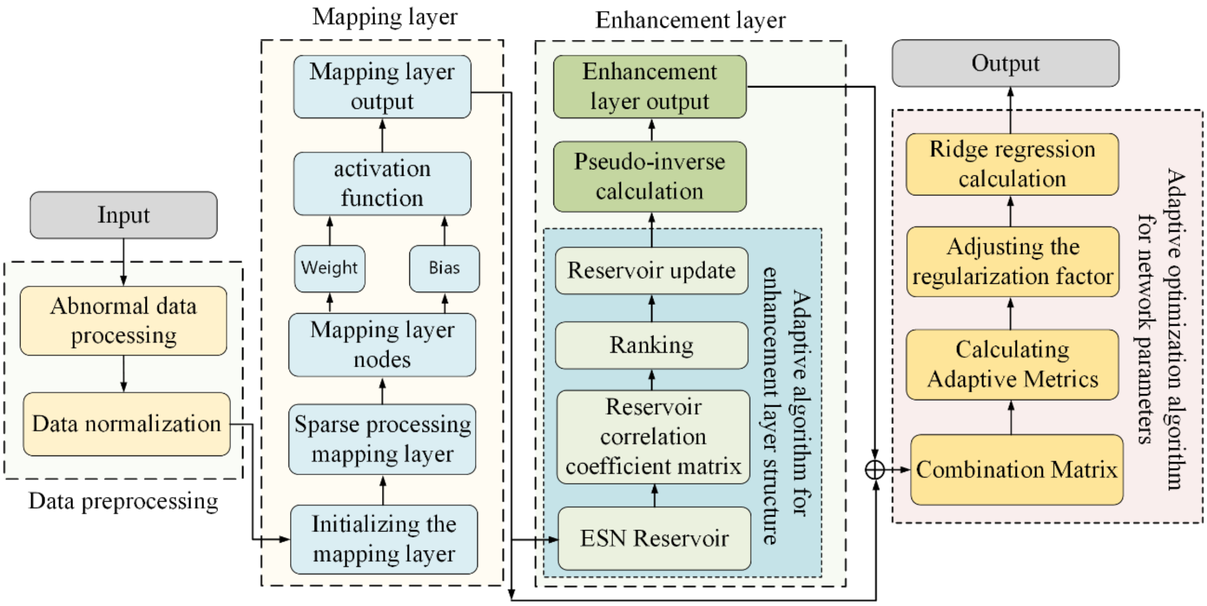

Figure 1. The framework of the ABESN model.

As shown in Figure 1, the ABESN forecasting method consists of four parts: the input layer, mapping layer, enhancement layer, and output layer. The input layer processes the outliers of the input data, and the normalized data are imported to the mapping layer. The mapping layer initializes the neural nodes first with sparse processing. Then, the connection weights are randomly initialized. A nonlinear activation function is set to obtain the mapping layer output. For the enhancement layer, the reservoirs are introduced, and their parameters are initialized. The reservoir correlation coefficient matrix is obtained using an adaptive structural optimization algorithm, through which the weight coefficients are set to 0 for the several groups of nodes with the greatest correlation. Meanwhile, the number of reservoirs is determined by the incremental learning algorithm. The output of the enhancement layer is obtained by the pseudo-inverse method. For the output of the model, the outputs of the mapping layer and enhancement layer are combined and imported. The hyperparameter adaptive optimization algorithm is introduced for model training.

As seen in the framework, adaptive optimization algorithms are proposed for the enhancement layer structure and the output hyperparameters, which are presented in Sections 3.2 and 3.3. The novel ABESN model makes each reservoir light and reduces the information redundancy in the reservoir neurons. The fitting ability can be improved with the optimization of the regularization coefficients. In brief, the ABESN model is built in view of the training scale and the data extracting ability.

The concrete structure of BESN in the ABESN forecasting method is shown in Figure 2. The BESN is derived from the basic structure of the BLS, in which the enhancement layer nodes are replaced with ESN reservoirs. Then, the fitting capability of the nonlinear data can be strengthened. The BESN inherits the rapid training of the BLS and the echo state property of the ESN.

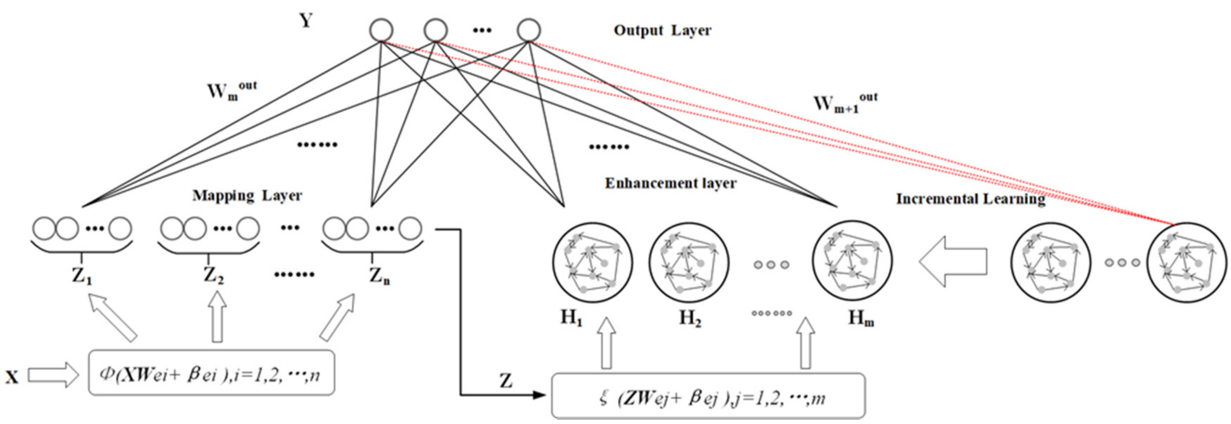

Figure 2. Structure of the BESN.

In the BESN shown in Figure 2, the enhancement layer consists of ESN cells, $\boldsymbol{Z} = [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n]$, and $\boldsymbol{H} = [\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_m]$. The output of the enhancement layer is denoted as follows:

$$\boldsymbol{H}_i(k) = (1-\alpha)\boldsymbol{H}_i(k-1) + \alpha f(\boldsymbol{W}_{hi}\boldsymbol{Z}(k) + \boldsymbol{W}_i\boldsymbol{H}_i(k-1)), \tag{1}$$

where $k = 1, 2, \ldots, n$ is the number of samples, $\boldsymbol{W}_{hi}$ denotes the weight connection matrix between the reservoir and the mapping layer, and $\boldsymbol{W}_i$ denotes the reservoir weight connection matrix, $f(\cdot)$ is a nonlinear activation function. The output formulas of ABESN and the output weight matrix are:

$$\begin{aligned} \boldsymbol{Y} &= [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n | \boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_m] \boldsymbol{W}_m{}^{out} \\ &= [\boldsymbol{Z}|\boldsymbol{H}] \boldsymbol{W}_m{}^{out} = \boldsymbol{A}^m \boldsymbol{W}_m{}^{out} \end{aligned}, \tag{2}$$

$$\boldsymbol{W}_m{}^{out} = \left(\lambda \boldsymbol{I} + \boldsymbol{A}^m (\boldsymbol{A}^m)^T\right)^{-1} (\boldsymbol{A}^m)^T \boldsymbol{Y}. \tag{3}$$

$\boldsymbol{W}_m{}^{out}$ can be solved using the ridge regression algorithm. In the ABESN, the incremental algorithm is applied to dynamically increase the ESN reservoir. The incremental algorithm does not need to retrain the network. It can complete the training only by processing the matrix operation of new nodes. The incremental algorithm of the ABESN proceeds as described below.

(1) Update the reservoir state matrix $\boldsymbol{A}^{m+1}$ after adding a new ESN cell,

$$\boldsymbol{A}^{m+1} = [\boldsymbol{A}^m | \boldsymbol{H}_{m+1}]. \tag{4}$$

(2) Calculate the pseudo-inverse matrix $\left(\boldsymbol{A}^{m+1}\right)^*$ of

$$\left(\boldsymbol{A}^{m+1}\right)^* = \begin{bmatrix} (\boldsymbol{A}^m)^* - \boldsymbol{D}\boldsymbol{B}^T \\ \boldsymbol{B}^T \end{bmatrix}. \tag{5}$$

(3) Calculate the updated output weight matrix $\boldsymbol{W}_{m+1}{}^{out}$,

$$\boldsymbol{W}_{m+1}{}^{out} = \begin{bmatrix} \boldsymbol{W}_m{}^{out} - \boldsymbol{D}\boldsymbol{B}^T\boldsymbol{Y} \\ \boldsymbol{B}^T\boldsymbol{Y} \end{bmatrix}, \tag{6}$$

where $\boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D}$ are obtained as follows:

$$\begin{aligned} &\boldsymbol{D} = (\boldsymbol{A}^m)^* \boldsymbol{H}_{m+1} \\ &\boldsymbol{B}^T = \begin{cases} (\boldsymbol{C})^*, \boldsymbol{C} \neq 0 \\ \left(1 + \boldsymbol{D}^T\boldsymbol{D}\right)^{-1} \boldsymbol{B}^T (\boldsymbol{A}^m)^*, \boldsymbol{C} = 0 \end{cases} \\ &\boldsymbol{C} = \boldsymbol{H}_{m+1} - \boldsymbol{A}^m \boldsymbol{D} \end{aligned} \tag{7}$$

The incremental algorithm can largely save the network training time and realize the dynamic update of the output weight matrix. In practical training, an evaluation index is chosen as the cutoff condition for the incremental algorithm.

3.2. Adaptive Structure Optimization for Enhancement Layers

The ESN reservoir structure is adopted in the enhancement layer neurons of the proposed ABESN. The ESN reservoir can be regarded as a nonlinear dynamic filter, which maps the input signals into a space of the bit through the high-dimension mapping. It greatly increases the extraction capability of the time series. However, the neuron connections in the reservoirs are highly sparse, leading to the different contributions of neurons to the enhancement layer. Meanwhile, the sparse connections expand the internal size of the neurons, increasing the training time and the calculation resources. Then, an optimization solution is studied in terms of the reservoir scale. An optimization algorithm based on the pruning algorithm is proposed to adaptively determine the enhancement layer structure.

The training mechanism of the pruning algorithm relies on the correlation degrees between the neurons in the reservoir. The neural nodes with high correlation are removed. The remaining neural nodes are recalculated using the regression algorithm to output the weight matrix. The reservoir can finally be built with a suitable size. In the solution, the network structure can be adjusted adaptively to achieve a lightweight model.

The determination basis of the pruning algorithm is the correlation between neurons. The correlation coefficient matrix of neurons in the enhancement layer is calculated as

$$r_{nm} = \frac{\sum\limits_{i=1}^{T} (s_{ni} - \bar{s}_n)(s_{mi} - \bar{s}_m)}{\sqrt{\sum\limits_{i=1}^{T} (s_{ni} - \bar{s}_n)^2 \sum\limits_{i=1}^{T} (s_{mi} - \bar{s}_m)^2}}, \tag{8}$$

where r_{nm} denotes the correlation coefficient between the $n'th$ and the $m'th$ reservoir nodes. T is the number of state vectors. $\bar{s}_n$ and $\bar{s}_m$ denote the mean value of the state vector of the $n'th$ and the $m'th$ reservoir nodes, calculated as follows:

$$\bar{s}_n = \frac{1}{T}\sum_{i=1}^{T} s_{ni}. \tag{9}$$

The correlation coefficient matrix R is obtained using Equation (10). The nodes with the largest correlation coefficient can be determined through the one-dimensional conversion of R and sorting. The selected nodes are pruned. The relationship matrix R is shown as follows:

$$R = \begin{bmatrix} 1 & r_{12} & \cdots & r_{1T} \\ r_{21} & 1 & \cdots & r_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ r_{T1} & r_{T2} & \cdots & 1 \end{bmatrix}. \tag{10}$$

On the basis of the calculations above, the adaptive algorithm for the enhancement layer structure is summarized in Algorithm 1.

Algorithm 1. Adaptive algorithm for the enhancement layer structure

Input: Number of mapping layer windows, initial value of the number of nodes in the enhancement layer, RMSE threshold, maximum number of pruning, ESN initialization reservoir size, and pruning step l.
Output: Predicted value, training time, and test time.

Step 1: Initialize the mapping and enhancement layers of ABESN, including random initialization $W_{ej}, W_{hi}, \beta_{ej}, \beta_{hi}$.
Step 2: Record the mapping layer vector $\boldsymbol{Z}$, and the enhancement layer vector $\boldsymbol{H}$.
Step 3: Calculate $\boldsymbol{W}_m{}^{out}$ and $\boldsymbol{A}^m$ according to Equations (2) and (3).
Step 4: The largest group of subscripts in the relationship matrix R, calculated according to Equation (10), is recorded as $R_r = [(r_{n1}, r_{m1})_1, (r_{n2}, r_{m2})_2, \ldots, (r_{nl}, r_{ml})_l]$.
Step 5: Set the connection weight matrix of the reservoir neural nodes corresponding to the subscript number to 0 to achieve pruning optimization.
Step 6: The current pruned state is recorded, while the parameter optimization of the ABESN network is updated using the parameter adaptive optimization algorithm shown in Figure 3.
Step 7: If the current forecasting result RMSE is greater than the set RMSE threshold, the number of ESN reservoirs is dynamically increased using the incremental algorithm, as expressed in Equations (4)–(7).
Step 8: Repeat Step 2 to Step 7 until the current forecasting result RMSE is less than the set RMSE threshold, then skip to Step 9.
Step 9: Record forecasting results, including predicted values, time, and other indicators.

The Algorithm 1 shows that the optimization object is the ESN reservoir unit. The weight of the most relevant nodes in the reservoir is set to 0. This optimization can reduce the unnecessary connection in the reservoir. The network structure is simplified to speed up the model training.

3.3. Adaptive Optimization for Hyperparameters

The nonstationary trend is one of the most important features of the time series data, which mainly affects the forecasting accuracy; the analysis methods for nonstationary time series include statistical tools such as correlogram is a statistical tool [44,45], statistical tests for multiple detrended correlation coefficients [46], multifractal detrended cross-correlation analysis [47], ADF tests [48], and many other mathematical analysis methods based on statistical foundations. In this paper, we focus on the nonstationary degree of the time series data and introduce the nonstationary index into the model training process. In the adaptive parameter optimization algorithm, the ABESN model is based on the difference between the degree of nonstationary fluctuation of the predicted data and the degree of nonstationary fluctuation of the real data as the basis for adjustment, while the evaluation index also takes into account the point-to-point error between the predicted data and the real data, with the former globally adjusting the degree of fluctuation between the predicted data and the real data, and the latter locally adjusting the local error value.

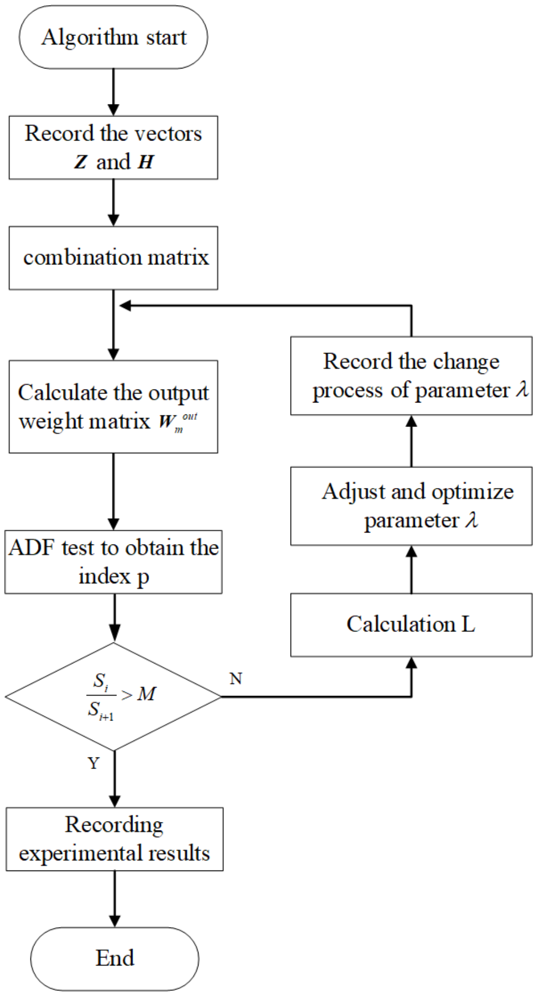

Figure 3. Process diagram of parameter adaptive algorithm.

In existing studies [49,50], ADF is widely used in the detection of nonstationary time series, especially in difference nonstationary data. In the test, the probability value (p) is a very important evaluation metric, which can be obtained from the Akaike information criterion (AIC). The probability value is considered a vital parameter in the optimization.

In this paper, an optimization method is proposed for the regularization coefficient λ in the BESN. Firstly, the starting condition for hyperparametric optimization is proposed on the basis of the nonstationary index,

$$\frac{S_i}{S_{i+1}} > M, \tag{11}$$

where S is the forecasting performance index proposed in this paper, i denotes the number of iterations, and M is the change rate threshold. The forecasting performance index S is calculated as follows:

$$S_i = |\hat{p} - p| \cdot \frac{1}{N} \sqrt{\sum_{k=1}^{N} (\hat{y}(k) - y(k))^2}, \tag{12}$$

where $\hat{p}$ and p denote the probability values of predicted and true data in ADF, respectively. $\hat{y}(k)$ and $y(k)$ denote the predicted and true values, respectively. N denotes the number of samples.

If a certain iteration in the model training meets the index in Equation (12), the regularization factor λ is adjusted and optimized as follows:

$$\lambda_{i+1} = \begin{cases} \alpha \cdot \lambda_i, L \geq 0 \\ \beta \cdot \lambda_i, L < 0 \end{cases}, \tag{13}$$

where λ_i denotes the parameter after the $i'th$ time. α and β are self-defined coefficients. L is the judgment condition for the direction of iterative adjustment.

$$L = \sum_{k=1}^{N} \hat{y}(k) - y(k). \tag{14}$$

The adaptive optimization algorithm of the ABESN hyperparameter is shown in Figure 3.

The definition in Equation (12) shows that S_i contains the description of the global error and the local error. A smaller value of S_i indicates a better forecasting effect of the model. When L is set to a different value range, the regularization scaling factor is adjusted to different degrees. Then, the network can obtain a certain degree of adaptive capability. The adaptive hyperparameter optimization algorithm and adaptive structure optimization algorithm form conjointly the adaptive optimization for the ABESN, which will adjust and optimize both the network structure and the hyperparameters.

The data preprocessing, training process, and testing process of ABESN are shown in Figure 4, which shows the adaptive structure optimization algorithm and adaptive parameter optimization steps, as well as the final testing step.

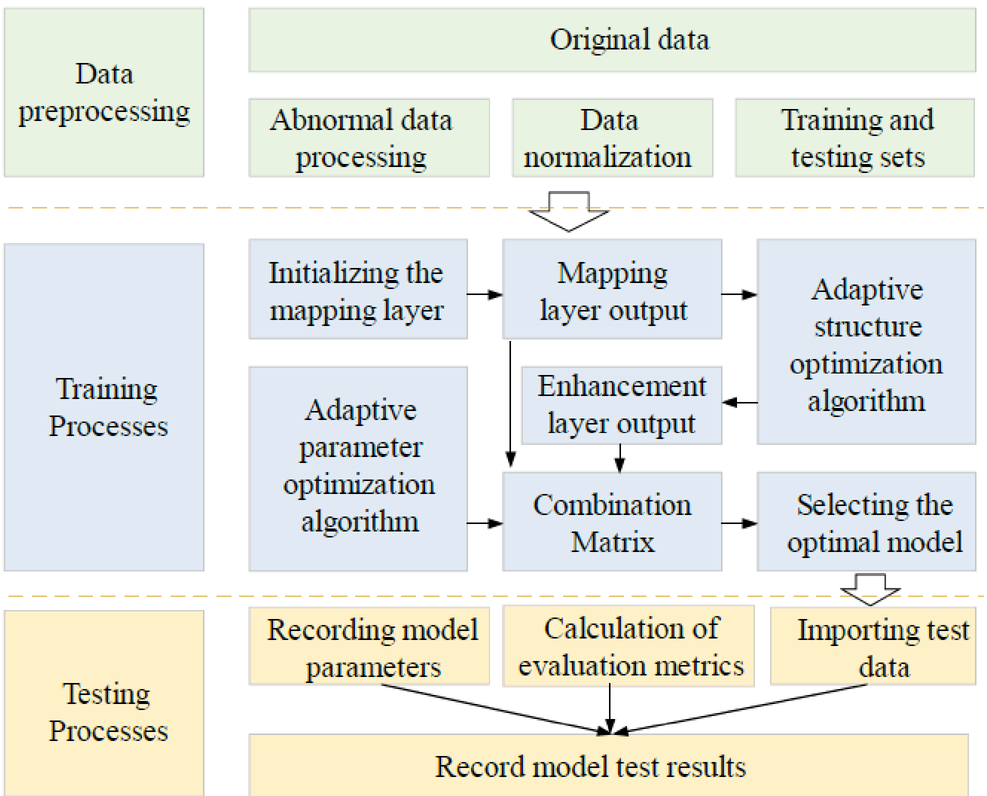

Figure 4. Training and testing process of ABESN model.

4. Experiment and Result

4.1. Datasets

In the experiments, the forecasting methods were validated with two datasets from real-world systems, the Beijing air humidity dataset and the US electric load dataset. The ADF test method could obtain the data characteristics of the two datasets to verify the characteristics of the data et itself such as non-stationarity. The ADF test is a more scientific judgment method based on whether the mean and variance of time series change over time. Meanwhile, the probability values (p), test statistics (TS), 1% critical value (CV_1), 5% critical value (CV_5), and 10% critical value (CV_{10}) are generated in a standard test. The p-value is the probability, which reflects the likelihood of an event occurring. The p-value obtained in the ADF test according to the significance test method is generally significant at $p < 0.05$ and highly significant at $p < 0.01$, which means that the probability that the difference between samples is due to sampling error is less than 0.05 or 0.01; conversely, $p \geq 0.05$ means that it is not significant, thus rejecting the original hypothesis. The p-value is an important basis for testing decisions; hence, it is chosen as the index in the optimization. To make the data more convincing, it is necessary to determine the numerical relationship among the statistics, CV_1, CV_5, and CV_{10}. According to the Akaike information criterion (AIC), the test time series is nonstationary when $p \geq 0.05$, TS > CV_1, TS > CV_5, TS > CV_{10}, and the null hypothesis is not denied.

4.1.1. Beijing Air Humidity Dataset

The first dataset was a meteorological dataset. The dataset of the air humidity monitoring data in Beijing was selected.

The sampling period of the Beijing humidity dataset is 1 h, with a total of 21,600 records. The data of 720 days were selected as the training set, while the data of 180 days were selected as the test set, strictly following the ratio of training to test sets of 4:1. The ADF test yielded $p = 0.1451$, TS = -3.3838, $CV_1 = -4.4059$, $CV_5 = -3.8500$, and $CV_{10} = -3.5642$. The original hypothesis could not be rejected, indicating the non-stationarity of the data. In the experiment, each set of data contained 12 samplings, in which the data of the first 11 h were set as the input and the data of the last 1 h were set as the output. The general distribution of the Beijing humidity dataset is shown in Figure 5.

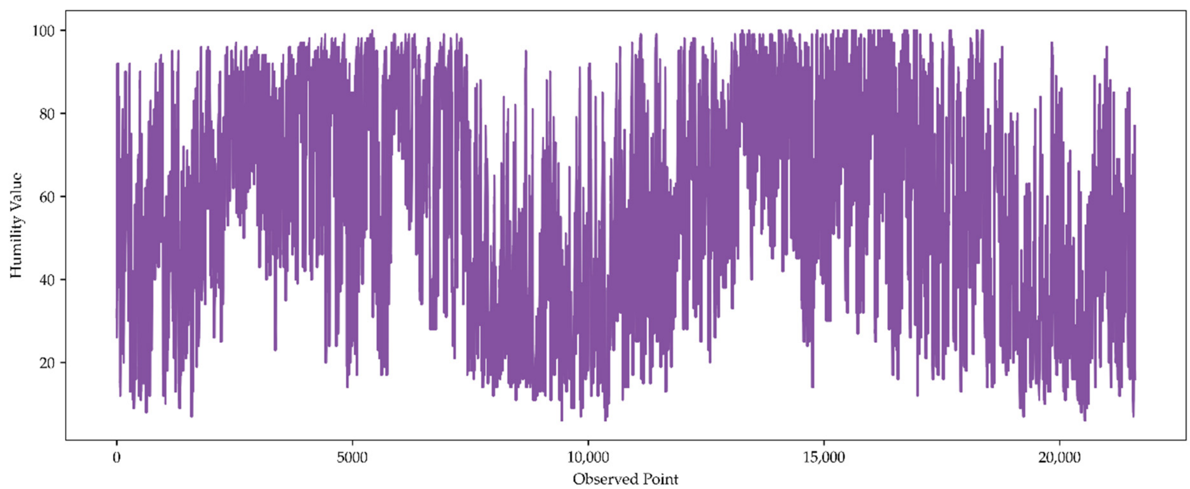

Figure 5. Distribution of air humidity data in Beijing.

4.1.2. US Electricity Load Dataset

The other dataset used in the experiments was the electric load dataset in the US. The data were collected from 1 January 2017 to 1 January 2020. The sampling interval is 1 h, and the total number of records is 26,280. In the experiment, the anomalous data in the original records were deleted first. Then, the consecutive data of 625 days were selected. A total of 15,000 sets were used as the experimental dataset. In each set of samples, the data of the first 23 h were set as the input and the data of the last 1 h were set as the output. The training set accounted for 80% and the test set accounted for 20% of the data. The ADF test yielded $p = 0.3550$, TS = -2.4469, $CV_1 = -3.9602$, $CV_5 = -3.4109$, and $CV_{10} = -3.1272$, which shows that the data were non-stationary. The total sample distribution of the electric load dataset is shown in Figure 6.

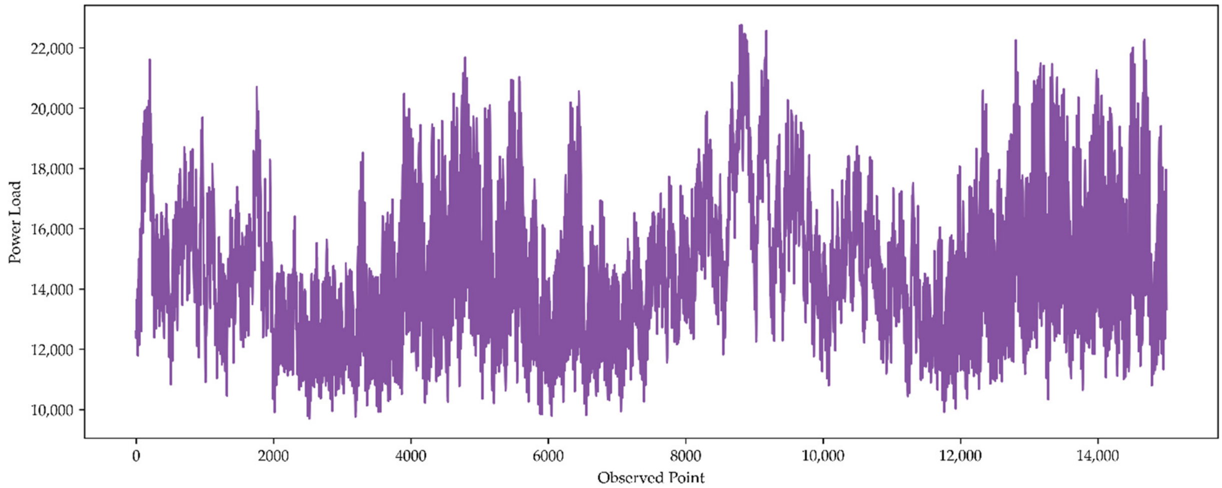

Figure 6. Distribution of US electricity load data.

4.2. Experimental Environment and Evaluation Metrics

4.2.1. Experimental Environment and Setup

The platform in the experiment was based on a 64 bit Windows 10 system with 16 GB of RAM, an AMD R7 4800 H (2.9 GHz) processor, a software support framework in Keras 2.4.3, and a programming language of Python 3.7.

To verify the ABESN model proposed in this paper, the relevant forecasting models were selected as the comparisons, including the GRU, LSTM, ESN, BLS, and BESN. The ESN, BLS, and BESN were selected because the ABESN is improved in the framework of BLS and ESN and optimized from the BESN. The broad learning methods are studied to solve the problems in deep learning; hence, the GRU and LSTM were necessary as comparison models.

For the two datasets selected in the experiments, the comparison models above were tuned to reach the relative optimal status. The parameters of the models are recorded in Tables 1 and 2. The parameters include the number of neural nodes in the mapping layer, the number of neural nodes in the enhancement layer, the size of the reservoir, the spectral radius (which is generally set between 0 and 1 to ensure the echo state property), the leakage rate, and the sparsity.

Table 1. Configuration of each model in Beijing air humidity dataset cited.

Model	Number of Mapping Layer Nodes	Number of Enhancement Layer Nodes	Reservoir Size	Spectral Radius Rate	Leaking Rate	Sparseness
BLS	10–30	2–30	NA	NA	NA	NA
ESN	NA	NA	300–800	0.90	0.10	0.09
BESN	10–30	2–30	300–800	0.90	0.10	0.09
ABESN	10–30	2–30	300–800	0.90	0.10	0.09

Table 2. Configuration of each model in the US electric load dataset.

Model	Number of Mapping Layer Nodes	Number of Enhancement Layer Nodes	Reservoir Size	Spectral Radius Rate	Leaking Rate	Sparseness
BLS	10–40	2–30	NA	NA	NA	NA
ESN	NA	NA	300–1000	0.90	0.10	0.09
BESN	10–40	2–30	300–1000	0.90	0.10	0.09
ABESN	10–40	2–30	300–1000	0.90	0.10	0.09

4.2.2. Time Series Forecast Error Metrics

The accuracy of forecasting depends on the forecasting error. The use of correct error forecasting measures is of vital importance. An error not only affects the model's optimization but also distorts the choice of the most suitable model. There are many categories for measuring forecasting errors depending on the nature of the data. According to Hyndman and Koehler (2006) [51], the common metrics used are described below.

Statistical forecasting measures are dependent on scale. When we refer to scale, this means that error measures are expressed in the same units. This category consists of the mean absolute error (MAE) [52], mean squared error (MSE), and root-mean-squared error (RMSE) [53]. The MAE should not be used if there are outliers (see Hyndman, 2006) [51]. The MSE is suitable when there are big errors. Chai and Drexler (2014) [54] suggested the RMSE over the MAE for model optimization, as well as for the assessment of different models where the error distribution is expected to be Gaussian. The above three indices depend on the measured units of the variables. Thus, they are used as measures for comparison only when we have the same variable on different models.

Statistical forecasting measures of percentage error are scale-free and are used for the comparison of forecasting among different time series. However, they are greatly affected the zero values in a time series. In such cases, they become infinite or undefined, thus being nonexplanatory (see Hyndman 2006) [51]. This category consists of the mean absolute percentage error (MAPE) and symmetric mean absolute percentage error (SMAPE) [55]. The MAPE cannot be used if there are zero values on the examined data. The SMAPE avoids the problem of large errors of the MAPE when data values are close to zero.

Scale-free error metrics (SFEMs) escalate the error on the basis of the mean absolute error (MAE) within the sample from a naïve forecasting method (random walk). In this category, we have the mean absolute scaled error (MASE), which is suitable for time series and zero data.

Lastly, the Theil U statistic is a measure of accuracy that compares the forecasted results with the results of forecasting with minimal historical data. It also squares the deviations to give more weight to large errors and to exaggerate errors, which can help eliminate methods with large errors.

Each category has its pros and cons. Therefore, their use depends on each independent variable and the data. It is important not to examine an individual error measure during the assessment of a model. If all time series are on the same scale, the procedures of preprocessing are accomplished, and the aim was to assess the forecasting, then the MAE must be chosen because it is easier to be explained (see Shcherbakov et al., 2013) [56]. Chai and Draxler (2014) [54] suggested the RMSE over the MAE when the error distribution is expected to be Gaussian. In case the data contain outliers, the application of escalating measures is recommended, e.g., the mean absolute scale error (MASE). In this case, the time horizon must be large enough but there must not be repeated values and the normalized factor must be equal to zero (Shcherbakov et al., 2013) [56].

In summary, the evaluation indicators selected in this paper included the MAE, RMSE, and SMAPE, and each evaluation indicator is defined as follows:

$$MAE = \frac{1}{N}\sum_{k=1}^{N}\left|y_r(k) - y_p(k)\right|, \tag{15}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y_r(k) - y_p(k))^2}, \tag{16}$$

$$SMAPE = \frac{1}{N}\sum_{k=1}^{N}\frac{|y_r(k) - y_p(k)|}{y_r(k) + y_p(k)}, \tag{17}$$

where $y_r(k)$ denotes the true value of the $k'th$ ample, $y_p(k)$ denotes the predicted value of the $k'th$ sample, and N denotes the number of samples.

4.3. Results

4.3.1. Experimental Results of Beijing Air Humidity Data

The forecasting results of each model on the air humidity data are shown in Figure 7. The curves in different colors indicate the forecasting results of different models. The forecasting results of the first 70 sets of data are shown. In the figure, an enlarged view is given for a clear and intuitive comparison. The curve in red denotes the real data, and the curve in blue denotes the forecasting results of the ABESN model. It can be found that the results of the ABESN were the closest to the real data among all forecasting curves.

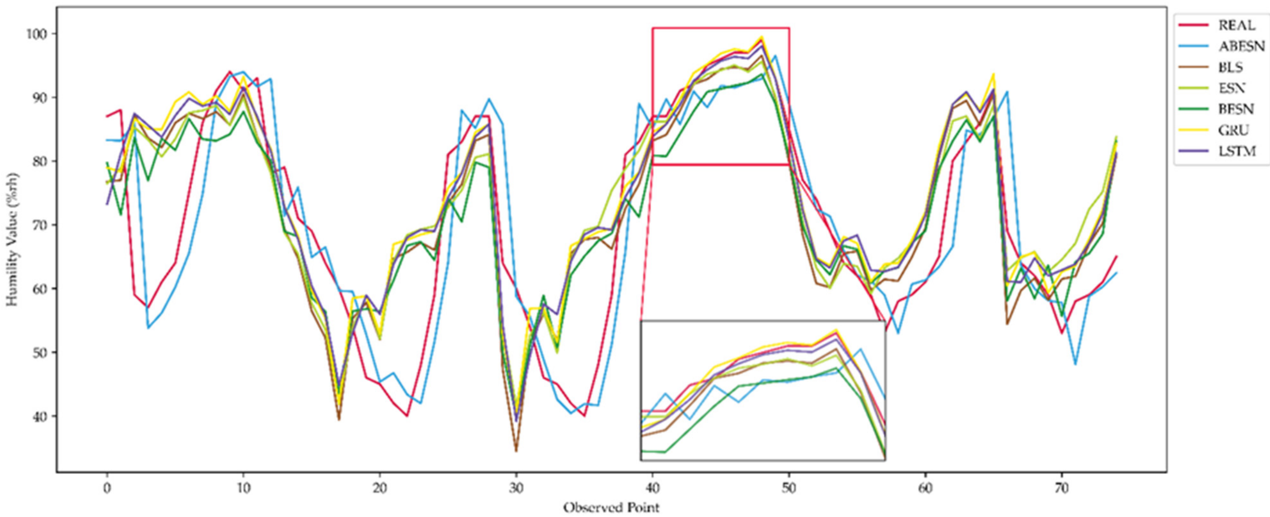

Figure 7. Forecasting results of each model on the Beijing air humidity dataset.

The evaluation metrics of the ABESN model and the comparison models on the test set of the humidity data and the training time are shown in Table 3. It can be seen that the deep learning networks of the GRU and LSTM improved the accuracy relative to the BLS and ESN, but the training time became long. The ABESN had the best performance in each evaluation metric, and the training speed declined relative to the GRU and LSTM.

Table 3. Evaluation index of each model in the Beijing air humidity dataset.

Model	Training Time (s)	SMAPE	MAE	RMSE
ESN	24.4139	0.0429	5.1367	7.4954
BLS	0.0480	0.0407	4.8979	7.5339
GRU	175.5832	0.0355	4.4311	7.3018
LSTM	132.0319	0.0372	4.5892	7.3669
BESN	84.9047	0.0414	5.1941	7.3062
ABESN	**105.4334**	**0.0403**	**4.3693**	**6.7875**

The forecasting errors of each model are shown in Figure 8. The bars indicate the difference between the forecasting results and the real data. As seen in Figure 8, the bars of the BLS and GRU were taller, showing a larger error. The blue bars of the ABESN were lower than others, indicating smaller errors.

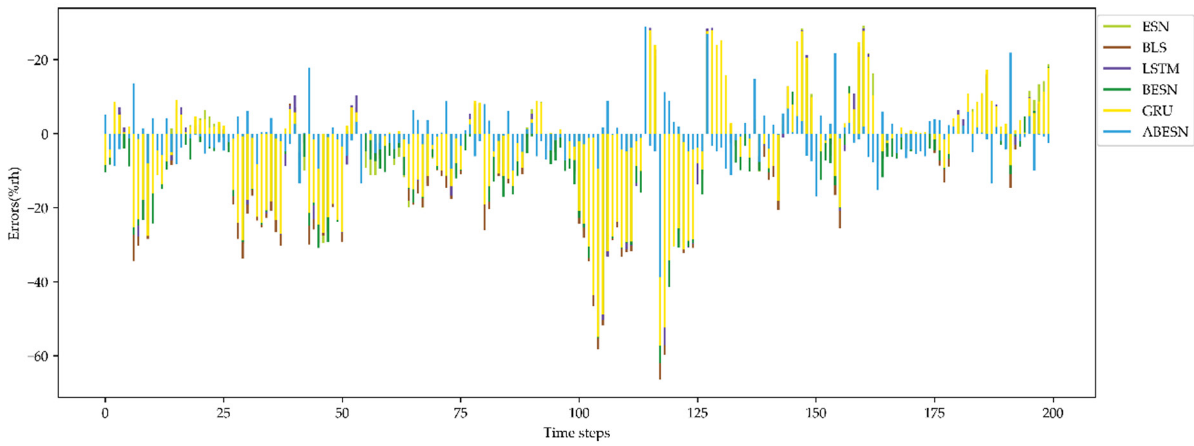

Figure 8. Forecasting error of each model in the Beijing air humidity dataset.

The data distribution of the forecasting results of each model is shown in Figure 9. The boxplots of the models are compared with the real data. The horizontal lines and star dots in each box indicate the median and mean of the forecasting results. The plot shows that the distribution of the ABESN results was closest to the real data.

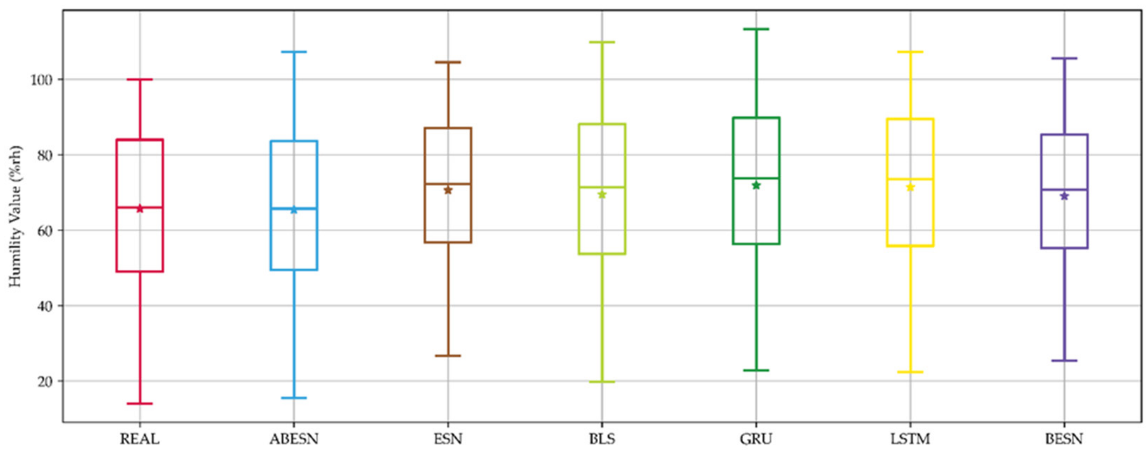

Figure 9. Boxplot diagram of forecasting results of each model (* indicates the average value of the data).

The adaptive tuning process of the hyperparameter in the ABESN is shown in Figure 10. The changes in the regularization coefficient and the RMSE in the tuning are plotted, where the red dotted line means the RMSE is 0. After 23 iterations, the hyperparameter λ reached a more suitable value, and the corresponding RMSE was minimized.

The scatter distributions of the forecasting results of models are shown in Figure 11. The plots contain the scatter distribution of the forecasting values, the fitting line of the real data, and the fitting line of the predicted data. The horizontal axis indicates the true value, and the vertical axis indicates the predicted value. The dots in blue indicate the scattered points of the predicted values. The straight lines in black and red are the fitting results of the real and predicted values, of which the slope is 1 and the intercept is 0. The slopes of the ABESN, BESN, GRU, LSTM, BLS, and ESN were 0.90, 0.48, 0.49, 0.43, 0.42, and 0.43, respectively. The intercepts were 5.68, 38.89, 38.77, 38.72, 36.79, and 39.53, respectively. It can be seen that the forecasting results of ABESN were the closest to the real data with the smallest errors. The results in Figure 11 are consistent with the evaluation metrics in Table 3.

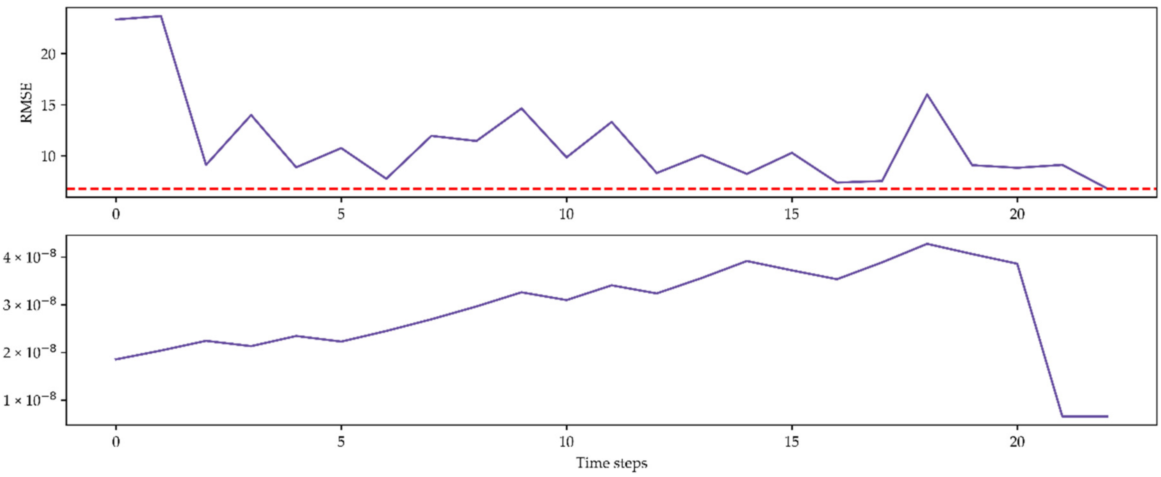

Figure 10. Adaptive parameter change process of ABESN model in Beijing air humidity dataset.

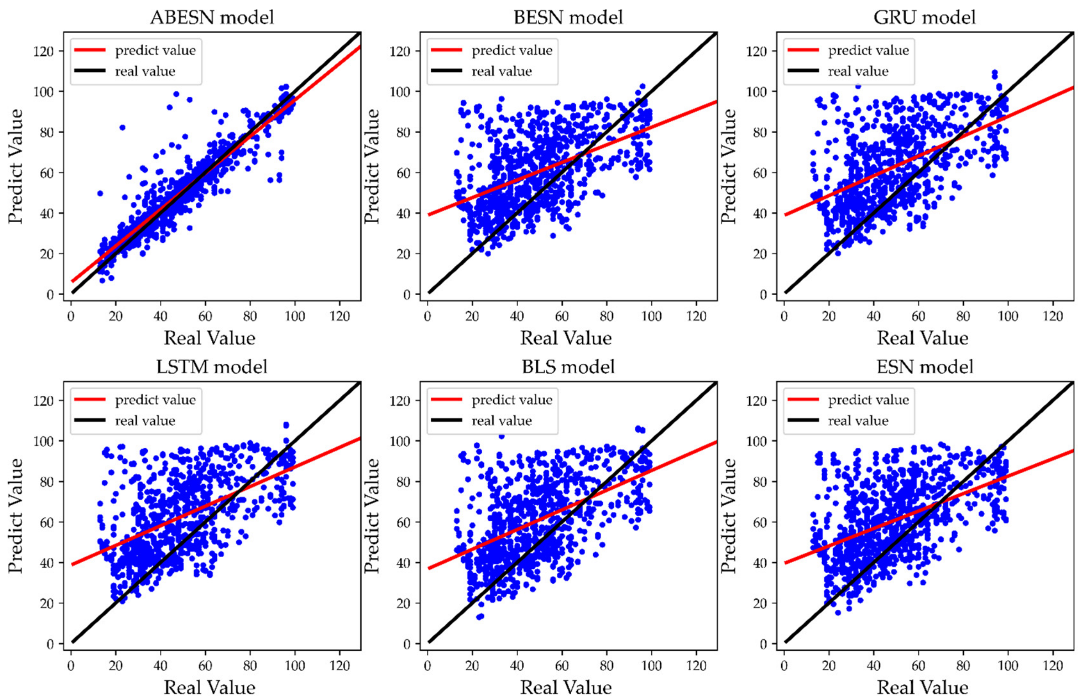

Figure 11. Scatter plot of each model in the Beijing air humidity dataset.

To more strongly verify the predictive power of the model, Figure 12 shows the prediction difference in values between timesteps. Through the slope and intercept of different models on time-differenced data, it can be found that the slope of the ABESN model was closest to 1 and the intercept was closest to 0. Instead, the slope of the ESN model was farthest from 1 and the intercept of the BLS model is the largest; therefore, from this prediction using the time-differenced data perspective, we can also get that the prediction ability of the ABESN model was stronger than that of other comparable models.

4.3.2. Experimental Results of US Electricity Load Data

The forecasting results of the electric load dataset in the US are shown in Figure 13, which contains a range of 80 sequential points. In the figure and the enlarged view, the curve in blue was closest to the red one, indicating that the results of the ABESN were closest to the real value.

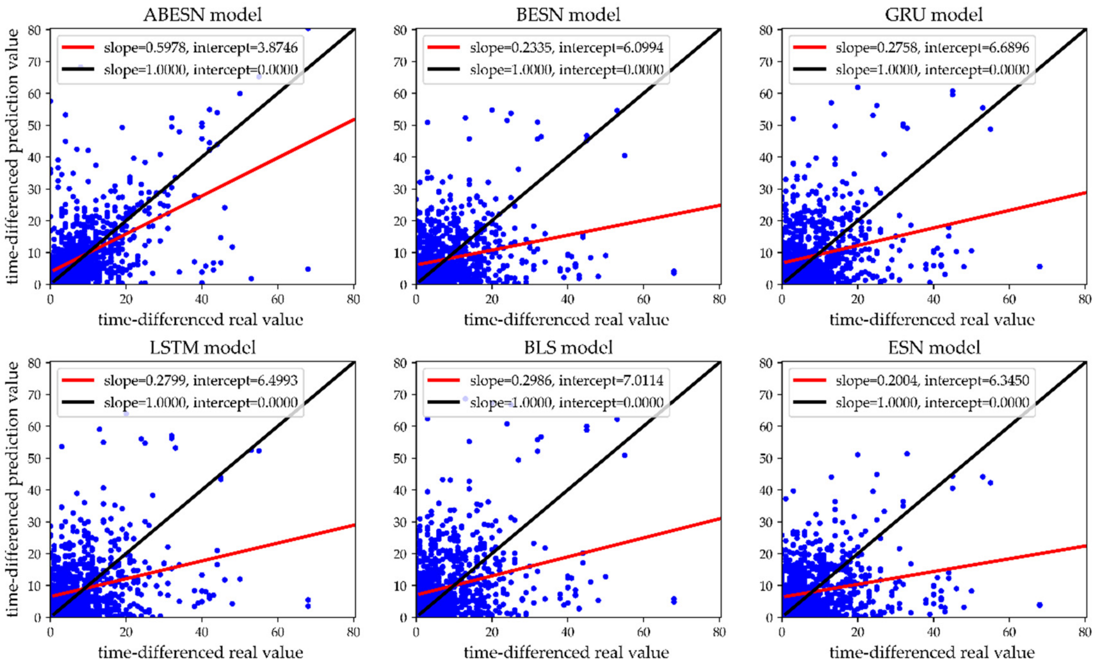

Figure 12. Scatter plot of predictions of different models on time-differenced data in the Beijing air humidity dataset.

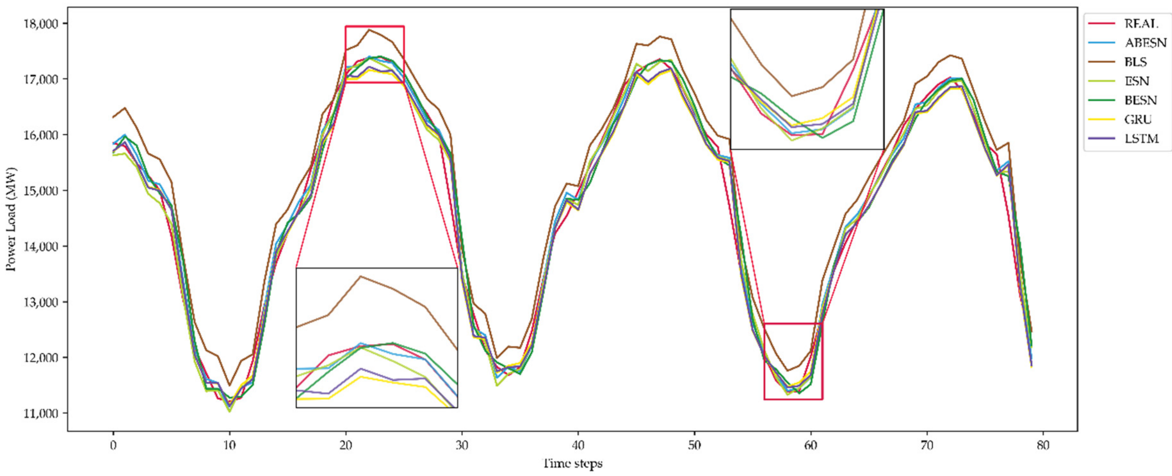

Figure 13. Forecasting results for each model on the US electric load dataset.

Table 4 shows the evaluation metrics and model training time for each model on the electric load test dataset, and Figure 14 shows the gap between the predicted and real data for each model. Table 4 and Figure 14 can show the specific and intuitional performance of different forecasting models.

Table 4. Results of evaluation metrics for each model on the U.S. electric load dataset.

Model	Training Time (s)	SMAPE	MAE	RMSE
ESN	20.0324	0.0073	227.3705	359.3435
BLS	0.0323	0.0117	346.1871	416.3548
GRU	176.5443	0.0083	266.4652	335.5402
LSTM	155.5230	0.0073	231.4214	293.2511
BESN	98.6472	0.0072	224.6844	286.7904
ABESN	**115.6756**	**0.0057**	**173.3812**	**233.4863**

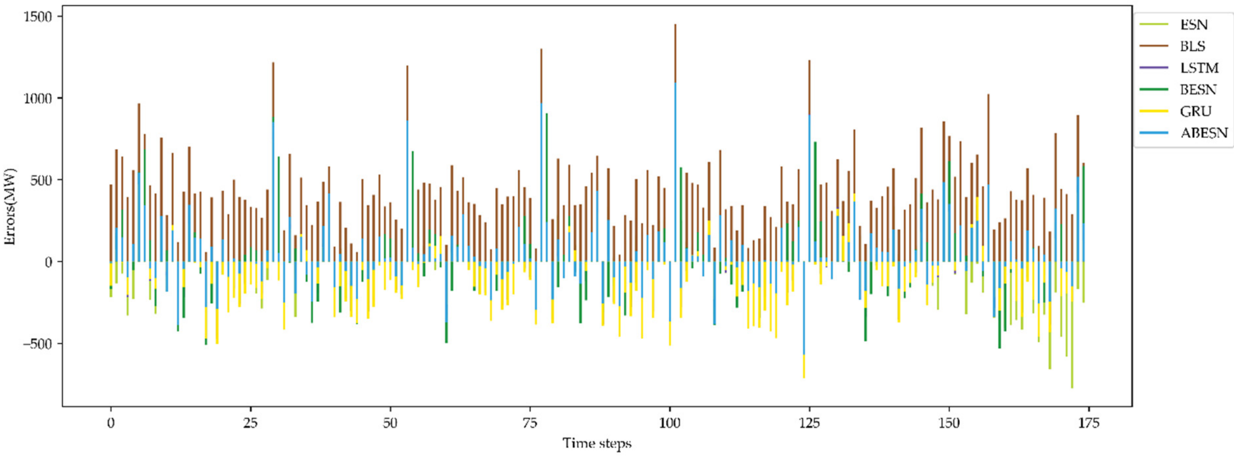

Figure 14. Errors of each model on the US electric load dataset.

The tuning process of the ABESN parameter is shown in Figure 15. The hyperparameter λ changed along with the RMSE, which reached the minimum value after 38 iterations of the adaptive tuning. The plot shows that λ increased in the first 33 iterations before converging until the 38th iteraation.

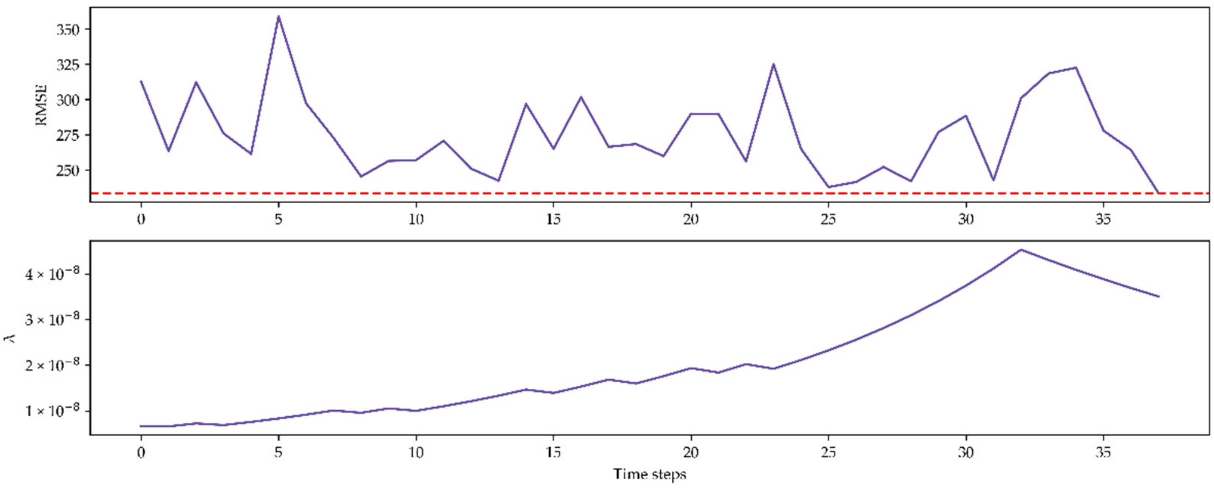

Figure 15. Adaptive parameter change process of the ABESN on the US electric load dataset.

The scatter distribution of the results on the US electric load dataset is shown in Figure 16. The slopes of the ABESN, BESN, GRU, LSTM, BLS, and ESN were 0.98, 0.97, 0.93, 0.95, 0.96 and 0.97 with intercepts of 262.05, 377.83, 904.65, 619.51, 776.44, and 384.44, respectively. It is proven from the values of slope and intercepts that the ABESN obtained the best fitting.

Predicting the time-differenced data rather than the raw data is a much stronger indication of the predictive power of the model. Figure 17 shows the prediction difference in values between timesteps. Although the intercept of the ABESN model on the time-differenced data was not the smallest, only larger than the intercepts of the BESN model, GRU model, and LSTM model, the slope of the ABESN model was the closest to 1. Overall, the predictive ability of the ABESN model was still excellent.

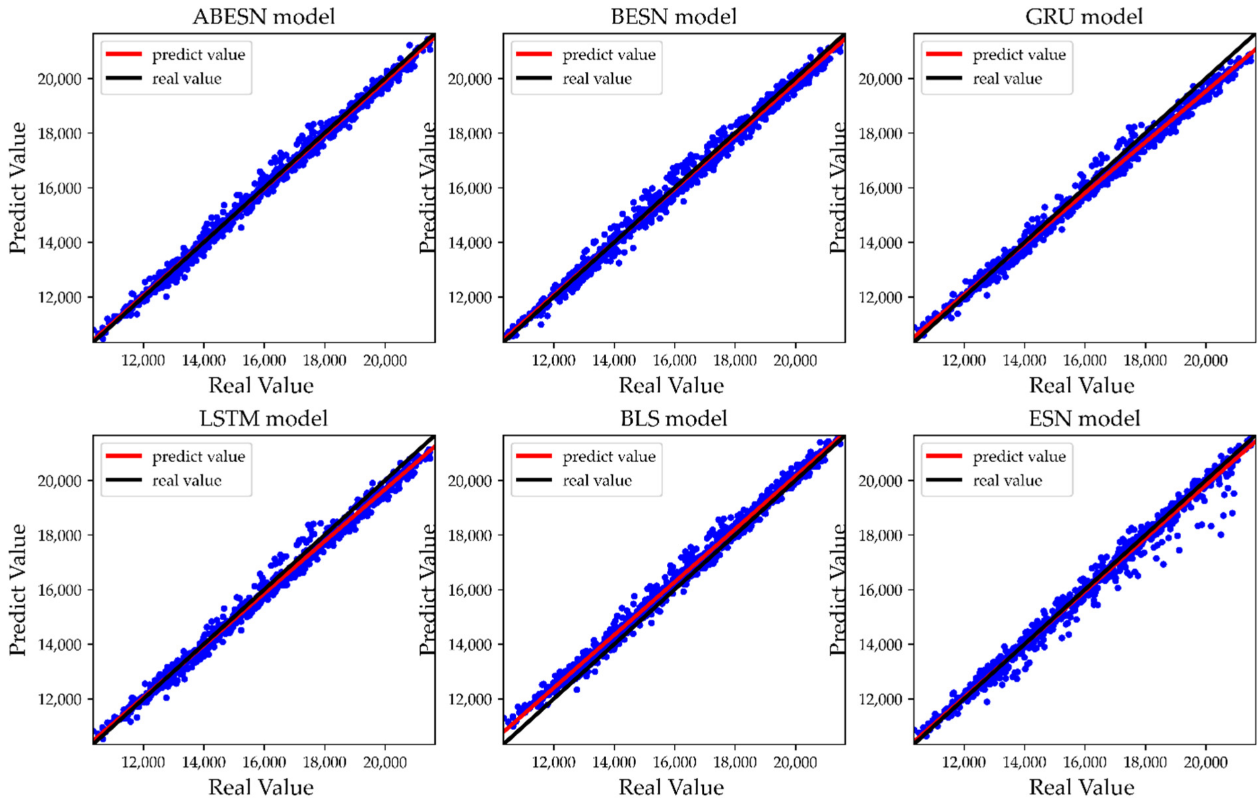

Figure 16. Scatter plot of each model in the US electric load dataset.

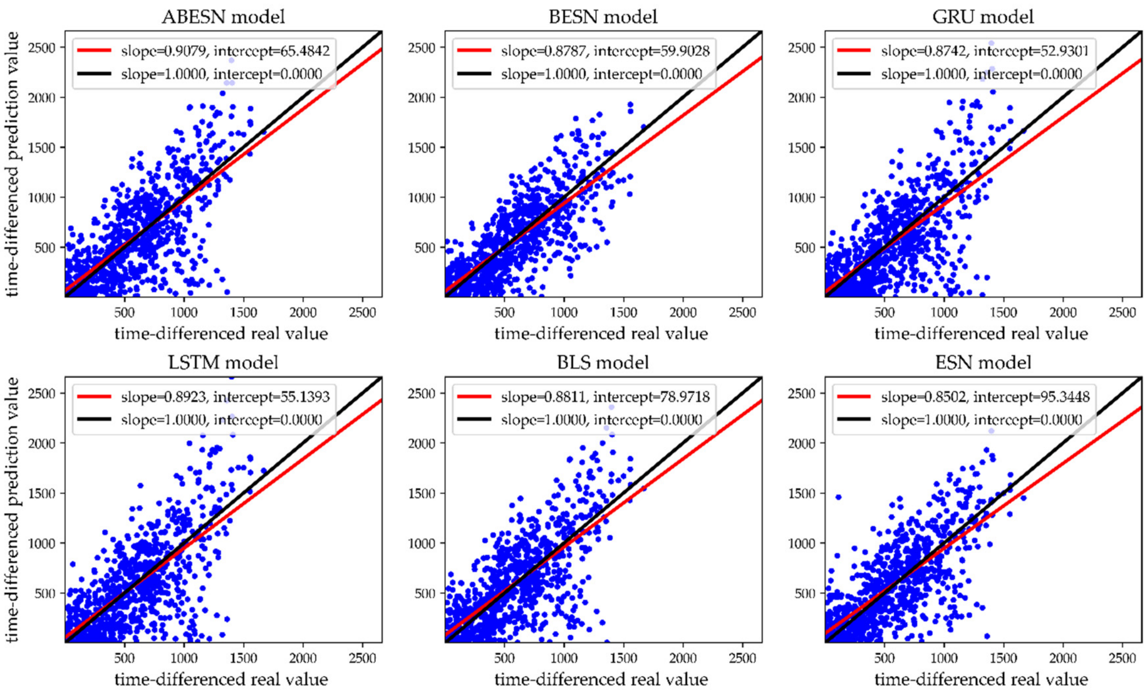

Figure 17. Scatter plot of predictions of different models on time-differenced data in the US electric load dataset.

4.4. Discussion

It can be seen from the results above that the BESN performed better than the single BLS and ESN models. It is proven that the proposed fusion network benefitted from the broad learning and the reservoir structure.

The RMSE, MAE, and SMAPE of the ABESN model were the smallest among the evaluation metrics in Tables 3 and 4, indicating the best fitting ability. On the air humidity dataset, the RMSE was decreased by 9.44%, 9.91%, 7.04%, 7.87%, and 7.10% relative to the ESN, BLS, GRU, LSTM, and BESN, respectively. On the electric load dataset, the RMSE was decreased by 35.02%, 43.92%, 30.41%, 20.38%, and 18.59% relative to the ESN, BLS, GRU, LSTM, and BESN, respectively. A similar performance comparison can be found in the error plots and the result distributions. At the same time, the ABESN model demonstrated its excellent prediction ability in nonstationary time series by both predicting time-differenced data and directly predicting numerical values.

This proves that the ABESN surpassed the classical models of the BLS, ESN, GRU, and LSTM on the two datasets. Moreover, the optimization algorithms were proven to work because the ABESN had better results than the BESN.

In the experiments of the two datasets, the ABESN performed better than the deep learning networks of the GRU and LSTM in terms of training speed, benefiting from the structure of the broad framework and the reservoirs. Due to the combination of the structure, the training of the ABESN was slower than that of the BLS and ESN. Therefore, the forecasting accuracy of the ABESN was improved at the expense of the structural complexity to some extent.

5. Conclusions

In this paper, the ABESN model was proposed for the forecasting of nonstationary time series. The ABESN is a novel network based on the combination of broad learning and the echo state structure, as well as the adaptive optimization of the structure and hyperparameters. The model was verified on two datasets representing natural and social systems with nonstationary time series. It is proven that the proposed model benefited from the fusion of the BLS and ESN. Meanwhile, the optimizations were efficient in terms of both the network structure and the hyperparameters. The training time of the ABESN was longer than the simple BLS and ESN, but it was decreased compared to the deep learning networks. The proposed model aims to balance the forecasting accuracy with the model complexity, and it achieved a certain effect. In future work, the determination of the neuron amounts should be explored for the mapping layer in ABESN. Meanwhile, the training speed can be improved by introducing other optimization methods.

Author Contributions: Conceptualization, W.-J.L. and Y.-T.B.; methodology, W.-J.L. and Y.-T.B.; writing—original draft preparation, W.-J.L., Y.-T.B. and X.-B.J.; writing—review and editing, W.-J.L., J.-L.K. and T.-L.S.; funding acquisition, X.-B.J., J.-L.K. and T.-L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key Research and Development Program of China (No. 2020YFC1606801), the National Natural Science Foundation of China (Nos. 62173007, 61903009, and 62006008), and the 2022 Graduate Research Ability Improvement Program of Beijing Technology and Business University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Horvatic, D.; Stanley, H.E.; Podobnik, B. Detrended cross-correlation analysis for non-stationary time series with periodic trends. *EPL Europhys. Lett.* **2011**, *94*, 18007. [CrossRef]
2. Harris, E.; Coleman, R. The social life of time and methods: Studying London's temporal architectures. *Time Soc.* **2020**, *29*, 604–631. [CrossRef]

3. Kong, J.; Yang, C.; Wang, J.; Wang, X.; Zuo, M.; Jin, X.; Lin, S. Deep-stacking network approach by multisource data mining for hazardous risk identification in iot-based intelligent food management systems. *Comput. Intell. Neurosci.* **2021**, *2021*, 1194565. [CrossRef] [PubMed]
4. Xu, L.; Li, Q.; Yu, J.; Wang, L.; Xie, J.; Shi, S. Spatio-temporal forecastings of SST time series in China's offshore waters using a regional convolution long short-term memory (RC-LSTM) network. *Int. J. Remote Sens.* **2020**, *41*, 3368–3389. [CrossRef]
5. Shi, Z.; Bai, Y.; Jin, X.; Wang, X.; Su, T.; Kong, J. Parallel deep forecasting with covariance intersection fusion on non-stationary time series. *Knowl. Based Syst.* **2021**, *211*, 106523. [CrossRef]
6. Praveen, B.; Talukdar, S.; Mahato, S.; Mondal, J.; Sharma, P.; Islam, A.R.M.; Rahman, A. Analyzing trend and forecasting of rainfall changes in India using non-parametrical and machine learning approaches. *Sci. Rep.* **2020**, *10*, 1–21. [CrossRef]
7. Cao, J.; Li, Z.; Li, J. Financial time series forecasting model based on CEEMDAN and LSTM. *Phys. A Stat. Mech. Its Appl.* **2019**, *519*, 127–139. [CrossRef]
8. Gkillas, K.; Gupta, R.; Pierdzioch, C. Forecasting realized oil-price volatility: The role of financial stress and asymmetric loss. *J. Int. Money Financ.* **2020**, *104*, 102137. [CrossRef]
9. Bukhari, A.H.; Raja MA, Z.; Sulaiman, M.; Islam, S.; Shoaib, M.; Kumam, P. Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting. *IEEE Access* **2020**, *8*, 71326–71338. [CrossRef]
10. Kong, J.; Wang, H.; Wang, X.; Jin, X.; Fang, X.; Lin, S. Multi-stream hybrid architecture based on cross-level fusion strategy for fine-grained crop species recognition in precision agriculture. *Comput. Electron. Agric.* **2021**, *185*, 106134. [CrossRef]
11. Kong, J.L.; Wang, H.X.; Yang, C.C.; Jin, X.B.; Zuo, M.; Zhang, X. Fine-grained pests & diseases recognition via Spatial Feature-enhanced attention architecture with high-order pooling representation for precision agriculture practice. *Agriculture* **2022**, *2022*, 1592804.
12. Yang, X.; Shu, L.; Chen, J.; Ferrag, M.A.; Wu, J.; Nurellari, E.; Huang, K. A survey on smart agriculture: Development modes, technologies, and security and privacy challenges. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 273–302. [CrossRef]
13. Jin, X.; Zhang, J.; Kong, J.; Su, T.; Bai, Y. A reversible automatic selection normalization (RASN) deep network for predicting in the smart agriculture system. *Agronomy* **2022**, *12*, 591. [CrossRef]
14. Kong, J.; Yang, C.; Lin, J.; Ma, K.; Zhu, Q.A. Graph-Related High-Order Neural Network Architecture via Feature Aggregation Enhancement for Identification Application of Diseases and Pests. *Comput. Intell. Neurosci.* **2022**, *2022*, 4391491. [CrossRef]
15. Jin, X.B.; Zheng, W.Z.; Kong, J.L.; Wang, X.Y.; Bai, Y.T.; Su, T.L.; Lin, S. Deep-learning forecasting method for electric power load via attention-based encoder-decoder with bayesian optimization. *Energies* **2021**, *14*, 1596. [CrossRef]
16. Wang, H.; Fang, Y.P.; Zio, E. Risk assessment of an electrical power system considering the influence of traffic congestion on a hypothetical scenario of electrified transportation system in New York state. *IEEE Trans. Intell. Transp. Syst.* **2019**, *22*, 142–155. [CrossRef]
17. Jin, X.B.; Gong, W.T.; Kong, J.L.; Bai, Y.T.; Su, T.L. A variational Bayesian deep network with data self-screening layer for massive time-series data forecasting. *Entropy* **2022**, *24*, 335. [CrossRef]
18. Jin, X.B.; Gong, W.T.; Kong, J.L.; Bai, Y.T.; Su, T.L. PFVAE: A planar flow-based variational auto-encoder forecasting model for time series data. *Mathematics* **2022**, *10*, 610. [CrossRef]
19. Chen, J.L.; Li, G.; Wu, D.C.; Shen, S. Forecasting seasonal tourism demand using a multiseries structural time series method. *J. Travel Res.* **2019**, *58*, 92–103. [CrossRef]
20. Tang, W.H.; Röllin, A. Model identification for ARMA time series through convolutional neural networks. *Decis. Support Syst.* **2021**, *146*, 113544. [CrossRef]
21. Karamichailidou, D.; Kaloutsa, V.; Alexandridis, A. Wind turbine power curve modeling using radial basis function neural networks and tabu search. *Renew. Energy* **2021**, *163*, 2137–2152. [CrossRef]
22. Kardani, N.; Bardhan, A.; Kim, D.; Samui, P.; Zhou, A. Modelling the energy performance of residential buildings using advanced computational frameworks based on RVM, GMDH, ANFIS-BBO and ANFIS-IPSO. *J. Build. Eng.* **2021**, *35*, 102105. [CrossRef]
23. Shahid, F.; Zameer, A.; Muneeb, M. A novel genetic LSTM model for wind power forecast. *Energy* **2021**, *223*, 120069. [CrossRef]
24. Javed, A.R.; Ur Rehman, S.; Khan, M.U.; Alazab, M.; Reddy, T. CANintelliIDS: Detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 1456–1466. [CrossRef]
25. Guo, L.; Li, N.; Jia, F.; Lei, Y.; Lin, J. A recurrent neural network based health indicator for remaining useful life forecasting of bearings. *Neurocomputing* **2017**, *240*, 98–109. [CrossRef]
26. Jaeger, H.; Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* **2004**, *304*, 78–80. [CrossRef] [PubMed]
27. Chen, C.L.P.; Liu, Z. Broad learning system: An effective and efficient incremental learning system without the need for deep architecture. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 10–24. [CrossRef]
28. Jian-Qin, M.; Jian, Y.; Hai-Shan, D. Chaotic time series forecasting based on fuzzy tree. *Acta Phys. Sin.* **2009**, *58*, 2220–2230. [CrossRef]
29. Liu, W.; Bai, Y.; Jin, X.; Wang, X.; Su, T.; Kong, J. Broad Echo State Network with Reservoir Pruning for Nonstationary Time Series Forecasting. *Comput. Intell. Neurosci.* **2022**, *2022*, 3672905.
30. Creal, D.; Koopman, S.J.; Lucas, A. Generalized autoregressive score models with applications. *J. Appl. Econom.* **2013**, *28*, 777–795. [CrossRef]

31. 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]
32. Yuan, C.; Liu, S.; Fang, Z. Comparison of China's primary energy consumption forecasting by using ARIMA (the autoregressive integrated moving average) model and GM (1, 1) model. *Energy* **2016**, *100*, 384–390. [CrossRef]
33. Liu, B.; Li, C.C.; Yan, K. DeepSVM-fold: Protein fold recognition by combining support vector machines and pairwise sequence similarity scores generated by deep learning networks. *Brief. Bioinform.* **2020**, *21*, 1733–1741. [CrossRef]
34. Zhang, N.; Shen, S.L.; Zhou, A.; Jin, Y.F. Application of LSTM approach for modelling stress–strain behaviour of soil. *Appl. Soft Comput.* **2021**, *100*, 106959. [CrossRef]
35. Zhao, R.; Wang, D.Z.; Yan, R.Q.; Mao, K.Z.; Shen, F.; Wang, J.J. Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Trans. Ind. Electron.* **2017**, *65*, 1539–1548. [CrossRef]
36. Jin, J.W.; Chen, C.L.P. Regularized robust broad learning system for uncertain data modeling. *Neurocomputing* **2018**, *322*, 58–69. [CrossRef]
37. Zheng, Y.; Chen, B. Least p-norm based broad learning system. *Pattern Recognit. Artif. Intell.* **2019**, *32*, 1–6.
38. Chauhan, V.; Tiwari, A. On the Construction of Hierarchical Broad Learning Neural Network: An Alternative Way of Deep Learning. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; IEEE: New York, NY, USA, 2018; pp. 182–188.
39. Du, J.; Vong, C.M.; Chen CL, P. Novel efficient RNN and LSTM-like architectures: Recurrent and gated broad learning systems and their applications for text classification. *IEEE Trans. Cybern.* **2020**, *51*, 1586–1597. [CrossRef]
40. Chatzis, S.P.; Demiris, Y. The copula echo state network. *Pattern Recognit.* **2012**, *45*, 570–577. [CrossRef]
41. Xu, M.; Han, M. Adaptive elastic echo state network for multivariate time series forecasting. *IEEE Trans. Cybern.* **2016**, *46*, 2173–2183. [CrossRef]
42. Han, M.; Mu, D. Multi-reservoir echo state network with sparse Bayesian learning. In *International Symposium on Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 450–456.
43. Lukoševičius, M. A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 659–686.
44. Zebende, G.F.; Guedes E, F. Detrended correlogram method for non-stationary time-series analysis. *Fluct. Noise Lett.* **2022**, *21*, 2250012. [CrossRef]
45. Brito, A.A.; Araújo, H.A.; Zebende, G.F. Detrended multiple cross-correlation coefficient applied to solar radiation, air temperature and relative humidity. *Sci. Rep.* **2019**, *9*, 1–10.
46. Da Silva Filho, A.M.; Zebende, G.F.; de Castro, A.P.N.; Guedes, E. Statistical test for multiple detrended cross-correlation coefficient. *Phys. A Stat. Mech. Its Appl.* **2021**, *562*, 125285. [CrossRef]
47. Gu, D.; Huang, J. Multifractal detrended cross-correlation analysis of high-frequency stock series based on ensemble empirical mode decomposition. *Fractals* **2020**, *28*, 2050035. [CrossRef]
48. Fotopoulos, S.B.; Ahn, S.K. Rank based Dickey–Fuller test statistics. *J. Time Ser. Anal.* **2003**, *24*, 647–662. [CrossRef]
49. Worden, K.; Iakovidis, I.; Cross, E.J. On stationarity and the interpretation of the ADF statistic. In *Dynamics of Civil Structures, Volume 2*; Springer: Cham, Switzerland, 2019; pp. 29–38.
50. Székely, G.J.; Rizzo, M.L. Partial distance correlation with methods for dissimilarities. *Ann. Stat.* **2014**, *42*, 2382–2412. [CrossRef]
51. Hyndman, R.J.; Koehler, A.B. Another look at measures of forecast accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688. [CrossRef]
52. Ni, T.; Wang, L.; Zhang, P.; Wang, B.; Li, W. Daily tourist flow forecasting using SPCA and CNN-LSTM neural network. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e5980. [CrossRef]
53. Liao, Y.; Li, H. Deep echo state network with reservoirs of multiple activation functions for time-series forecasting. *Sādhanā* **2019**, *44*, 1–12. [CrossRef]
54. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [CrossRef]
55. Chen, C.; Twycross, J.; Garibaldi, J.M. A new accuracy measure based on bounded relative error for time series forecasting. *PLoS ONE* **2017**, *12*, e0174202. [CrossRef] [PubMed]
56. Shcherbakov, M.V.; Brebels, A.; Shcherbakova, N.L.; Tyukov, A.P.; Janovsky, T.A.; Kamaev, V.A.E. A survey of forecast error measures. *World Appl. Sci. J.* **2013**, *24*, 171–176.