Time Series Prediction Method Based on E-CRBM

Huixin Tian 1,2,* and Qiangqiang Xu 1,2

1 Key Laboratory of Advanced Electrical Engineering and Energy Technology, Tiangong University, Tianjin 300387, China; 1830041175@tiangong.edu.cn
2 The School of Control Science and Engineering, Tiangong University, Tianjin 300387, China
* Correspondence: tianhuixin@tiangong.edu.cn

Abstract: To solve the problems of delayed prediction results and large prediction errors in one-dimensional time series prediction, a time series prediction method based on Error-Continuous Restricted Boltzmann Machines (E-CRBM) is proposed in this paper. This method constructs a deep conversion prediction framework, which is composed of two E-CRBMs and a neural network (NN). Firstly, the E-CRBM models of the original input sequence and the target prediction sequence are trained, respectively, to extract the time features of the two sequences. Then the NN model is used to connect and transform the time features. Secondly, the feature sequence H1 is extracted from the original input sequence of test data through E-CRBM1, which is used as input of NN to obtain feature transformation sequence H2. Finally, the target prediction sequence is obtained by reverse reconstruction of feature transformation sequence H2 through E-CRBM2. The E-CRBM in this paper introduces the residual sequence of NN feature transformation in the hidden layer of CRBM, which increases the robustness of CRBM and improves the overall prediction accuracy. The classical time series data (sunspot time series) and the actual operation data of reciprocating compressor are selected in the experiment. Compared with the traditional time series prediction method, the results verify the effectiveness of the proposed method in single-step prediction and multi-step prediction.

Keywords: time series prediction; continuous restricted Boltzmann machines; neural network; multi-step prediction

1. Introduction

Time series refers to the sequence of values of the same statistical index in the order of their occurrence time. The current research directions of time series data include classification, clustering, and regression prediction [1–3]. This paper studies the regression prediction of time series. The time series prediction algorithm mainly analyzes the phenomenon characteristics, direction and trend of historical data, summarizes, and infers, so as to predict the data of the next moment or the next time period. Time series prediction algorithms are of great significance in the fields of economy, engineering, and natural technology such as finance, transportation, stocks, regional precipitation [4–8], etc.

At present, there are two main types of time series modeling and prediction methods. The first type is the classical time series analysis methods and their variants, which mainly use the principle of mathematical statistics to analyze time series. The classical time series prediction models mainly include autoregressive (AR) model [9], moving average (MA) model [10], autoregressive moving average (ARMA) model [11], autoregressive integrated moving average (ARIMA) model [12], autoregressive conditional heteroscedasticity (ARCH) model [13], generalized autoregressive conditional heteroscedasticity (GARCH) model [14], etc. The traditional time series prediction method depends on the selection of parameter model. The accuracy of prediction results is largely determined by whether the parameter model can be correctly selected. These models usually ignore the nonlinear characteristics of data. With the increase of prediction period, the limitations of linear models become increasingly prominent. This makes it difficult for the traditional statistical analysis method to effectively predict the time series in practical application.
The second type of time series modeling and prediction methods are data-driven models, such as traditional machine learning models and deep learning models. Based on the historical input and output data, the machine learning method establishes the input-output relationship model through various learning rules. The model is further used to predict the future output. The traditional machine learning models for time series prediction mainly include linear regression (LR) [15], support vector regression (SVR) [16], artificial neural network (ANN) [17], and extreme learning machine (ELM) [18], etc. With the advent of the era of big data and the complexity and diversity of data itself, time series data present the characteristics of large amount of data and diverse and complex structure. Therefore, it is difficult for the machine learning model with simple structure to obtain effective feature information, which makes its prediction accuracy unable to meet the actual situation.

Compared with traditional machine learning, deep learning has a more complex and larger model structure, such as deep belief networks (DBN) [19], recurrent neural network (RNN) [20], long short-term memory (LSTM) [21], echo state network (ESN) [22], convolutional neural network (CNN) [23], fuzzy neural network (FNN) [24], etc. The deep learning model can be regarded as a multi-layer perceptron composed of multi-layer hidden layers. Therefore, it is essentially a deep nonlinear neural network. Through its own complex network structure, it can combine the low-level features to generate more abstract deep features, thus having a more comprehensive description of complex and abstract concepts. Among them, CRBM is a kind of deep learning model [25]. It is not only suitable for data feature extraction for classification and recognition [26,27], but also for modeling continuous data and applying to time series prediction [28]. Compared with the traditional machine learning model, the deep learning model has more powerful learning ability and adaptive ability and can better model and analyze the nonlinear system.

Although technically improved methods have greatly improved the prediction accuracy, the existing methods have not effectively solved the two shortcomings of delayed prediction results and large prediction errors. The guiding significance of the prediction results is still limited. Therefore, a time series prediction method based on E-CRBM is proposed in this paper. The E-CRBM is used to obtain the abstract time features automatically to the maximum extent. The feature vector can not only eliminate the autocorrelation of time series, but also be easier to predict than the original time series. The experimental results showed that this method can not only eliminate the lag of prediction results, but also improve the prediction accuracy. The innovations of this paper are as follows:

1. A time series depth prediction architecture based on continuous restricted Boltzmann machine and neural network is proposed.
2. In order to improve the robustness of the continuous restricted Boltzmann machine, an error-continuous restricted Boltzmann machine (E-CRBM) is proposed.

The rest of this paper is organized as follows. The relevant theoretical knowledge is introduced in Section 2. In Section 3, the method proposed in this paper is introduced in detail. The Section 4 is the experimental verification and analysis. Finally, the conclusion of this paper is summarized in Section 5.

2. Background

2.1. Restricted Boltzmann Machine

The restricted Boltzmann machine (RBM) is the basic component unit of DBN and a typical neural network [29]. Its structure is shown in Figure 1. The RBM consists of two layers of neurons: One layer is a visible layer for training input data and the other layer is a hidden layer for feature extraction. The neuron nodes of the visible layer and the hidden layer are not connected in the same layer, and the two layers are connected to each other. The weight matrix of the connection is represented by \( W \), and \( a \) and \( b \) are the bias of the visible layer and the hidden layer, respectively [30].
Suppose there are \( n \) visible layer nodes and \( m \) hidden layer nodes in an RBM. The node states of the visual layer and the hidden layer are represented by vectors \( v \) and \( h \), and the elements \( v_i \) and \( h_j \) are binary variables, that is, \( v_i \in \{0, 1\} \) \((i = 1, 2, \ldots, n)\), \( h_j \in \{0, 1\} \) \((j = 1, 2, \ldots, m)\), and the RBM energy function is defined as:

\[
E(v, h; \theta) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n} \sum_{j=1}^{m} v_i w_{ij} h_j
\]

where \( \theta = \{w, a, b\} \), \( v_i \) is the state of the \( i \)-th visible unit, and \( h_j \) is the state of the \( j \)-th hidden layer unit. The \( a_i \) is the bias of the visible unit \( i \), \( b_j \) is the bias of the hidden unit \( j \), and \( w_{ij} \) is the weight connecting the visible unit \( i \) and the hidden unit \( j \).

The joint probability of the visible unit and the hidden unit can be expressed as:

\[
P(v, h; \theta) = \frac{1}{Z(\theta)} \exp(-E(v, h; \theta))
\]

\[
Z(\theta) = \sum_{v,h} \exp(-E(v, h; \theta))
\]

Since the states of hidden units in RBM are independent of each other, the conditional probability of the hidden layer and the visible layer is:

\[
P(h_j = 1|v; \theta) = 1/[1 + \exp(-b_j - \sum_i v_i w_{ij})]
\]

\[
P(v_i = 1|h; \theta) = 1/[1 + \exp(-a_i - \sum_j w_{ij} h_j)]
\]

For the above RBM model, this paper uses the contrast divergence (CD) method to solve the negative gradient of the log-likelihood function to obtain the optimal value of the parameter [31]. Firstly, the training data are taken as the state \( v \) of the visible unit, and then the state \( h \) of the hidden unit is calculated according to Equation (4). Then the updated and reconstructed state \( v' \) of the visible unit is calculated according to Equation (5). Finally, the updated and reconstructed state \( h' \) of the hidden unit is recalculated according to Equation (4). Its parameter update formula can be expressed as:

\[
\begin{align*}
\Delta w_{ij} &= \varepsilon_{\text{CD}}(E(v|h_j) - E(v|h'_j)) \\
\Delta a_i &= \varepsilon_{\text{CD}}(E(v_i) - E(v'_i)) \\
\Delta b_j &= \varepsilon_{\text{CD}}(E(h_j) - E(h'_j))
\end{align*}
\]

where \( \varepsilon_{\text{CD}} \) is the learning rate of the contrast divergence gradient descent algorithm. \( E(.) \) represents the mathematical expectation of the variable.

In the traditional RBM model, the input of visual layer is limited to binary number (0 or 1), which is very inconvenient for continuous values such as most actual time series. Therefore, the CRBM model is proposed. In this model, the continuous value of independent Gaussian distribution is added to the linear element to simulate the real data,
so that the traditional RBM can process continuous input vectors. Its energy function is changed to:

$$E(v, h; \theta) = -\sum_{i=1}^{n} \left( \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n} \sum_{j=1}^{m} v_i w_{ij} h_j \right)$$  \hspace{1cm} (7)$$

Among them, $v_i$ represents the real value of the visible layer and $\sigma_i$ is the standard deviation vector of the Gaussian function. According to the energy Equation (7), their conditional probability distribution is obtained as:

$$P(v_i|h) = N(b_i + \sigma_i \sum_{j=1}^{n} h_j w_{ij}, \sigma_i^2)$$  \hspace{1cm} (8)$$

$$P(h_j|v) = \text{sign} \left( \sum_{i=1}^{n} v_i w_{ij} + a_j \right)$$  \hspace{1cm} (9)$$

The update method and training process of parameters in CRBM are the same as traditional RBM, and CD algorithm can be used to adjust parameters.

The traditional random binary unit RBM is not suitable for feature extraction of continuous values. CRBM is a continuous random generation model that can model continuous data. It realizes the modeling of continuous data by adding a Gaussian unit with variance $\sigma^2$ and mean value 0 to the Sigmoid function of the visual layer of RBM. Therefore, CRBM is selected for time series feature extraction.

2.2. Predictive Model Architecture Based on CRBM

CRBM model can not only effectively simulate high-dimensional and complex non-linear data, but also can well simulate continuous data to extract data features. On this basis, a prediction model combining CRBM and NN is proposed, which is applied to continuous time series prediction. CRBM can automatically obtain abstract time features, and the conversion of feature vectors is easier than the conversion of the original time series. The time features of the original input sequence and the target prediction sequence are connected and transformed by NN, and then the new target sequence is reconstructed by using CRBM in reverse direction to achieve the purpose of prediction. The Figure 2 shows the basic flow of the proposed method.

![Figure 2. CRBM-NN flow chart.](image-url)

As shown in Figure 2, CRBMs share weights (associative weights) between bottom-up and top-down. Given weight parameter matrices $W_1$ and $W_2$, for CRBM1 and CRBM2, each bottom-up coding function $f_1(t)$ and $f_2(t)$ can be expressed as:

$$f_i(t) = \sigma(W_i t), i \in \{1, 2\}$$  \hspace{1cm} (10)$$
where $\sigma(t)$ represents the activation function.

Similarly, given the feature vector of the hidden layer and the decoding function from $f^{-1}(t)$, the top to the bottom is used to reconstruct the original sequence:

$$f^{-1}(t) = \sigma(W^T_1 t)$$

(11)

In this method, the $(L + 1)$ layer NN is used to transform the input vector into a new feature vector. The new feature vector is obtained by Equation (12), once the weight parameter $W$ $(L = 1, 2, ..., L)$ is estimated in advance, and the input vector can be transformed into:

$$\xi(t) = \bigcirc_{l=1}^L \xi_l(t)$$

(12)

$$\xi_l(t) = \sigma(W_l t)$$

(13)

where $\bigcirc_{l=1}^L$ is the composition of $L$ functions. For example, if there are two hidden layers in the NN model, there is $\xi(t) = \bigcirc_{l=1}^2 \xi_l(t) = \sigma(W^2 \sigma(W^1 t))$. $W_l$ represents an element in a set of weight parameters $W$.

$$W = \{W_l\}_{l=1}^L = \{W_1, W^{(1)}, \cdots, W^{(L)}, W_2\}$$

(14)

where $W_1$ represents the weight of the CRBM1 model, $W^{(1)}, \cdots, W^{(L)}$ represents the weight of the NN model, and $W_2$ represents the weight of the CRBM2 model.

3. Method

Traditional time series prediction methods ignore the nonlinearity of data itself and rely on the accurate selection of model parameters. Traditional machine learning is usually not enough to effectively simulate the complex spatio-temporal features of high-dimensional nonlinear time series data. Compared with the traditional machine learning model, deep learning model has more powerful learning ability and adaptive ability and can better model and analyze nonlinear system.

Therefore, in this paper, E-CRBM is used to extract the features of time series, which not only eliminates the autocorrelation of time series, but also transforms high-dimensional time series into low-dimensional time feature series. Then, NN is used to connect and transform the original input sequence features and the target prediction sequence features, and finally complete the model training. The flow chart is shown in Figure 3.

The learning process based on E-CRBM and NN prediction model is as follows:

1. Data division: The time series is divided into three parts: training set 1, training set 2, and test set.
2. Data processing: The data in each data set are divided into the original input sequence and the target prediction sequence, and the appropriate training samples are constructed.
3. Model training: Training set 1 is used as model pretraining data to obtain feature conversion error sequences, and training set 2 is used as E-CRBM training data to obtain E-CRBM-NN prediction model.
4. Model validation: Test set data are used for validation, calculation of relevant indicators, and analysis of results.

The process of E-CRBM-NN prediction algorithm is shown in Algorithm 1.
Algorithm 1 E-CRBM-NN prediction algorithm

Input: data set Data1, data set Data2, test set Data, data length $L$, the sequence reconstruction step length $m$, and prediction step length $p$.
Output: E-CRBM model, NN model, prediction result $v_2$.

1. The original time series data set is divided into three parts: training set 1, training set 2, and test set. Each partial data set is divided and constructed according to Equations (16) and (17).
2. In data processing, it is necessary to normalize the time series to the range of $[0,1]$.
3. CRBM is used to extract the time features of the original input sequence and the target prediction sequence in training set 1, while NN is used to transform the features. The feature error sequence is preserved.
4. The feature error sequence is analyzed and the probability distribution function $f(\theta)$ is obtained. It is added to the hidden layer of CRBM as a noise function. The original input sequence and target prediction sequence of training set 2 are used for E-CRBM training. The E-CRBMB1 and E-CRBMB2 models are obtained, and the features are extracted. Then, NN is used to transform the features, and NN model is obtained.
5. In the model validation, the original input sequence of test set is extracted by E-CRBMB1 model and then transformed by trained NN model. The prediction result $v_2$ is obtained by the output of NN through the reverse input of E-CRBMB2.

3.1. Time Series Division

The essence of time series prediction is to calculate the value of time series at $T+1$ time according to the observation data of the first $T$ time. The prediction method in this paper needs to divide the data set into training set 1, training set 2, and test set. Training set 1 is used to obtain the feature conversion error sequence in the NN network. The error sequence is introduced into CRBM, and E-CRBM is obtained using training set 2. The test set is used to verify the performance of the prediction method. Among them, each data set needs to be divided into original input sequence and target prediction sequence with the same size, and then they are reconstructed into appropriate input forms, as shown in Figure 4.
Figure 4. Schematic diagram of time series division.

The time series \( T \) is firstly divided into the original input sequence \( T_1 \) and the target prediction sequence \( T_2 \) and then reconstructed into appropriate samples according to Equation (16).

\[
T = [T_1, T_2]
\] (15)

\[
T'_i = \text{Rrshape}(T_i, m, n), i = 1, 2
\] (16)

where \( T' \) is the constructed sample and \( m \) and \( n \) represent the size of the sample.

The original input sequence \( T_1 \) is used as the model input, and the target prediction sequence \( T_2 \) is used as the model output, where \( p = 2 \) is the single-step prediction and \( p > 2 \) is the multi-step prediction.

3.2. E-CRBM Model

In Section 2.2, the basic framework of the prediction model in this paper was introduced. The error of this prediction model mainly consists of two parts: the reconstruction error of CRBM and the feature transformation error of NN network. For the error caused by NN network, the appropriate parameters can be found to compensate. However, the improvement of the overall prediction accuracy is very limited. Therefore, we turn to the study of allowing NN feature transformation error within a certain error range, which can ensure that the reconstruction error of CRBM is as small as possible, so as to improve the overall prediction accuracy. Based on the analysis of the feature of error sequence and the principle of CRBM, a CRBM method based on error compensation was proposed, which is called E-CRBM. E-CRBM is to analyze the feature error sequence obtained by NN network. Noise is added to the hidden layer of CRBM, and the noise probability distribution is fitted by the probability distribution of error sequence, which makes CRBM more robust. In the E-CRBM-NN prediction model, the preparation of E-CRBM is divided into the following two steps.

The first step is to obtain the feature error sequence. The dimension of original input sequence \( X_1 \) and target prediction sequence \( Y_1 \) of training set 1 data are reduced by CRBM, and time features \( h_1 \) and \( h_2 \) are extracted. Taking \( h_1 \) and \( h_2 \) as inputs and outputs of NN, the model output \( h'_2 \) is obtained, and then the feature error sequence \( e = h'_2 - h_2 \) is obtained, as shown in Figure 5.

The second step is to fit the probability density function of feature error sequence. Through the analysis of multiple sets of feature error sequence samples of training data set 1, we found that their distribution basically conformed to the Gaussian distribution, as shown in Figure 6. In this paper, we mainly determine whether the feature error conforms to Gaussian distribution through numerical measurement and graphic analysis. The data are measured and analyzed by Jarque–Bera hypothesis test. Graphic analysis method is to judge whether the number conforms to Gaussian distribution by analyzing the normal fitting degree of data, such as a histogram. Gaussian distribution is the most common. Although a few feature error sequences do not fit Gaussian distribution, most of them fit Gaussian distribution better.
Therefore, only a few unknown parameters $\theta$ need to be estimated based on the known probability density function. When it is Gaussian distribution, the maximum likelihood estimation is used to solve the unknown parameters. Suppose the sample set is $X = [x_1, x_2, \ldots, x_N]$ and the unknown parameter is $\theta = [\theta_1, \theta_2]$, $\theta_1 = \mu$, $\theta_2 = \sigma^2$, the probability density function is:

$$f(x|\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp \left[ -\frac{1}{2} \left( \frac{x - \mu}{\sigma} \right)^2 \right] \quad (17)$$

The parameters obtained by maximum likelihood estimation are:

$$\hat{\mu} = \theta_1 = \frac{1}{N} \sum_{i=1}^{N} x_i \quad (18)$$

$$\hat{\sigma}^2 = \theta_2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2 \quad (19)$$

The fitting probability density function can be obtained by substituting it into Equation (17).

After the probability distribution function of error sequence is obtained, the data of training data set 2 are taken as the training data of E-CRBM. After noise is introduced into the hidden layer of traditional CRBM, the structure of E-CRBM model is still the same as that of traditional CRBM. Therefore, the parameters’ training of E-CRBM model also adopts the traditional CRBM training method.

E-CRBM is to solve the problem that after feature extraction of time series, when the feature is damaged, it can still be reconstructed back to the original data. The minimization target of traditional CRBM is $L(x, g(f(x)))$, where $L$ is the loss function, $x$ is the original data, and $g$ and $f$ are the data reconstruction function and feature extraction function, rese-
tively. The minimization goal of E-CRBM is \( L(g(\hat{f}(x))) \), where \( \hat{f}(x) = f(x) + p(f(x)|\theta) \) and \( p(f(x)|\theta) \) are the probability distribution functions of the error sequence. The structure of E-CRBM is shown in Figure 7.

![Figure 7. Structure diagram of E-CRBM.](image)

The training process of E-CRBM is as follows:

A damage process \( C(\hat{h}|h) \) is added to CRBM, which means that the extracted feature \( h \) is introduced into the noise function \( p(f(x)|\theta) \).

1. A training sample \( x \) is collected from the training data, and the feature \( h = f(x) \) is extracted by \( f(x) \) function.
2. After the damage process of feature \( h \), the noise function \( p(f(x)|\theta) \) is introduced. Then \( \hat{h} = h + p(h|\theta) \).
3. The original data \( \hat{x} = g(\hat{h}) \) are reconstructed from the damaged feature \( \hat{h} \) by \( g(\hat{h}) \) function, and the loss function \( L(x, \hat{x}) \) is calculated.

After E-CRBM training is completed, the test set data are input into the prediction model to test the effect of the model on time series prediction, as shown in Figure 8.

![Figure 8. Prediction model based on E-CRBM-NN.](image)

The E-CRBM-NN time series prediction method proposed in this paper can effectively restrain the lag of prediction results and improve the prediction accuracy. This is because CRBM can automatically obtain abstract time features, which not only eliminates the autocorrelation of time series but also is easier to predict than the original time series. At the same time, the proposed E-CRBM introduces the error feature sequence into the hidden layer of CRBM, which enhances the robustness of E-CRBM and improves the overall prediction accuracy. However, compared with the traditional neural network prediction method, the training process of this method is more complex, so it takes longer time. This paper mainly considers the off-line modeling and prediction. Obviously, the accuracy is more important.

4. Experiment

In order to verify the effectiveness of the proposed method, sunspot time series data and the actual operation data of the reciprocating compressor of China National Offshore Oil Corporation (CNOOC) offshore natural gas production platform were tested.
The results were compared with ARIMA, multilayer perceptron (MLP), DBN, and LSTM. For quantitative comparison, root mean square error (RMSE) and mean absolute error (MAE) were introduced to measure the prediction results.

\[
RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{X}_i - X_i)^2}
\]

\[
MAE = \frac{1}{N} \sum_{i=1}^{N} |\hat{X}_i - X_i|
\]

where \(\hat{X}\) is the predicted value and \(X\) is the real value.

4.1. Data Set Description

In this paper, two data sets were selected for experiments. In order to show the universal applicability of this method, the commonly used representative time series data (sunspot data) were selected for verification. In order to verify the effectiveness of this method in time series prediction, this paper selected the compressor vibration signal data for experimental verification. The sunspot time series data used in this paper were from the smooth monthly mean sunspot number data from January 1753 to December 2001 provided by solar influences data analysis center (SIDC). The data length was 2998 and the data interval was 1 month, which were recorded as Data1. Another compressor operation parameter data were the actual operation status data of reciprocating compressor during the actual natural gas production process of a platform of CNOOC from March 2016 to April 2016. The data interval was 1 min and the data length was 43,200, which were recorded as Data2. Because of the numerous parameters of the compressor, this paper selected the vibration signal of the compressor for verification and analysis.

4.2. Single-Step Prediction

For sunspot time series data, the first 2500 data were selected as training data, and the rest of the data were used as test data. There were many compressor data, 20,000 of which were selected for the test, with the first 19,000 as the training data and the rest as the test data.

The experimental setup followed the general method of this kind of experiment and adjusted the depth and width (the number of hidden units) for each neural network to obtain the optimal prediction effect. Therefore, for each neural network, the number of hidden layer units was selected in the set \{10, 15, 20, 25\}. The selection range of network depth was \{1, 2\}. The network parameter settings of the relevant methods in this paper are shown in Table 1.

| Parameter               | MLP  | DBN  | LSTM | E-CRBM | NN   |
|-------------------------|------|------|------|--------|------|
| Learning rate           | 0.01 | 0.01 | 0.005| 0.05   | 0.01 |
| Number of hidden layers | 1    | 2    | 2    | 1      | 1    |
| Network structure       | [10 20 1] | [10 20 15 1] | [10 20 20 1] | [10 5] | [5 10 5] |
| Batch size              | 20   | 10   | 20   | -      | -    |
| Number of iterations    | 500  | 200  | 150  | 150    | 100  |

The single-step prediction results of each prediction method are shown in Figure 9. The input of the prediction model was the value of the first 10 time samples, and the target output was the value of the 11th time sample. The Figure 9 shows the prediction results of MLP, DBN, LSTM, and the method proposed in this paper. In the figure, blue represents the true value and red represents the predicted value. The size of the display prediction data was 300, in which the 60–120 time period data were enlarged.
Figure 9. Single-step prediction results.

Due to the randomness of neural network in the training process, the results of each training were different. In order to ensure the validity of the experimental results, all neural network models were trained for 10 times and the average value was taken. The prediction results of each method are shown in Table 2.
Table 2. Comparison of prediction results of various methods under different data sets.

| Data Sets | Data1  | Data2  |
|-----------|--------|--------|
| Error     | MAE    | RMSE   | MAE    | RMSE   |
| MLP       | 10.1215| 13.1104| 0.0514 | 0.0597 |
| DBN       | 9.3157 | 12.0942| 0.0476 | 0.0511 |
| LSTM      | 8.0974 | 10.0836| 0.0302 | 0.0391 |
| Proposed  | 4.7857 | 6.0397 | 0.0186 | 0.0254 |

In this paper, E-CRBM-NN was introduced into the problem of single dimension time series prediction, and experiments were carried out to verify whether the model can get good results. The prediction performance was compared with MLP, DBN, and LSTM. The Table 2 shows the results of each model for sunspot data and compressor vibration signal single-step prediction error measurement index. The analysis results showed that:

1. LSTM is more suitable for time series prediction than other neural networks. Therefore, LSTM is better than MLP and DBN in the two data sets in this paper. The prediction method proposed in this paper combined E-CRBM with NN, and its prediction effect was obviously better than the previous three methods.
2. E-CRBM can be used to predict time series, and the prediction result was better than LSTM by combining with NN neural network.
3. The prediction accuracy of ARIMA was poor and the experiment took a long time. Especially, the vibration sequence of reciprocating compressor with a large amount of data could not be effectively predicted, so it is not shown in the figure.

4.3. Multi-Step Prediction

In order to further verify the effectiveness of the proposed method in time series prediction, multi-step prediction experiments were carried out on two data sets to test the performance in multi-step prediction. The process of multi-step prediction experiment was almost the same as that of single-step prediction experiment, but it was different from that of single-step prediction experiment when constructing model input samples. As described in Section 3.1, the prediction step was adjusted by changing the size of p. When p > 2, the model was multi-step prediction. The same parameter setting method was adopted, and the related parameter settings of each method are shown in Table 3.

Table 3. Parameter settings of each network.

| Parameter          | MLP     | DBN     | LSTM    | E-CRBM  | NN      |
|--------------------|---------|---------|---------|---------|---------|
| Learning rate      | 0.01    | 0.01    | 0.005   | 0.05    | 0.01    |
| Number of hidden layers | 1 | 2 | 2 | 1 | 1 |
| Network structure  | [10 15 1]| [10 15 10 1]| [10 15 15 1]| [10 5]| [5 10 5]|
| Batch size         | -       | 20      | 10      | 20      | -       |
| Number of iterations| 500    | 200     | 100     | 100     | 100     |

The above two data sets were still used for the test. The multi-step prediction experiment was carried out in 1–10 steps, and the five-step prediction is analyzed in this paper. The input was the value of the first 10 time samples, and the target output was the value of the 15th time sample. The prediction results are shown in Figure 10. It can be seen from the experimental results that the prediction step size affected the prediction error. In the two data sets, MLP, DBN, and LSTM had significantly increased prediction error when the step size was 5, while E-CRBM-NN had small data volume and large fluctuation for the first data set and its prediction effect was still acceptable although it was weakened. For the second largest data set, the prediction accuracy of E-CRBM-NN could still be maintained in a certain range without significant decline, and the prediction effect of E-CRBM-NN was better than other methods.
Table 3. Parameter settings of each network.

Parameter | MLP | DBN | LSTM | E-CRBM | NN
---|---|---|---|---|---
Batch size | | 20 | 10 | 20 | -
Learning rate | 0.01 | 0.01 | 0.005 | 0.05 | 0.01
Number of hidden layers | 1 2 2 1 1 | | | | |
Number of iterations | 500 200 100 100 100 | | | | |
Network structure | [10 15 1] [10 15 10 1] [10 15 15 1] [10 5] [5 10 5] | | | | |
Error MAE | LSTM | 12.8223 | 15.7214 | 0.0447 | 0.0589 |
RMSE | DBN | 14.3169 | 18.6578 | 0.0712 | 0.0799 |
MLP | 15.5373 | 19.9258 | 0.0785 | 0.0897 |

Figure 10. Multi-step prediction results.

Similarly, in order to ensure the validity of the experimental results, all neural network models were trained for 10 times and the average value was taken. The prediction results of each method are shown in Table 4.
Table 4. Comparison of prediction results of various methods under different data sets.

| Data Sets | Data1          | Data2          |
|-----------|----------------|----------------|
|           | Error          | MAE            | RMSE          | Error          | MAE            | RMSE          |
| MLP       | 15.5373        | 19.9258        | 0.0785        | 0.0897         |
| DBN       | 14.3169        | 18.6578        | 0.0712        | 0.0799         |
| LSTM      | 12.8223        | 15.7214        | 0.0447        | 0.0589         |
| Proposed  | 6.03541        | 7.8175         | 0.0253        | 0.0341         |

In the multi-step prediction experiment, the prediction accuracy within 10 steps was acceptable. However, although the prediction error after 10 steps was smaller than that of the traditional time series method, there was still not much practical guiding significance. The results showed that the method had good performance in short-term prediction and the learning effect was better when the data sample was sufficient.

5. Conclusions

In order to improve the prediction accuracy of time series, this paper combined E-CRBM with NN to predict single dimensional time series. In this method, CRBM was used to extract the time feature of the sequence, and the autocorrelation of the sequence was eliminated. High-dimensional data were transformed into low-dimensional features to reduce the difficulty of NN model training. In addition, E-CRBM introduced feature error sequences in the hidden layer of CRBM, which made E-CRBM more robust and improved the overall prediction accuracy. However, compared with the traditional neural network prediction method, the training process was more complex. In the experiment, it was found that although LSTM is more suitable for time series prediction and its effect is better than MLP and DBN, its prediction accuracy is still not ideal. Compared with traditional time series prediction methods ARIMA and machine learning methods MLP, DBN, and LSTM, this method has better prediction accuracy. From the experimental results of single-step prediction, it can be seen that this method can effectively suppress the lag of the prediction results and improve the prediction accuracy. The experimental results of multi-step prediction show that the performance is superior in short-term prediction.

Author Contributions: Conceptualization, H.T. and Q.X.; Data creation, Q.X.; Funding acquisition, H.T.; Investigation, H.T. and Q.X.; Methodology, H.T. and Q.X.; Supervision, H.T.; Validation, H.T.; Visualization, H.T. and Q.X.; Writing—original draft, Q.X.; Writing—review and editing, H.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (grant number: 61703406, 71602143), Natural Science Foundation of Tianjin (grant number: 18 JCYBJC22000), and Tianjin Science and Technology Correspondent Project (grant number: 19 JCTPJC47600).

Data Availability Statement: Data available in a publicly accessible repository. The data presented in this study are openly available in http://www.sidc.be/silso/datafiles (accessed on 31 January 2021).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References
1. Wang, Z.H.; Zhang, W.; Yuan, J.D.; Liu, H.-Y. A novel lazy time series classification algorithm based on the shapelets. Chin. J. Comput. 2019, 42, 29–43.
2. Yu, C.C.; Wu, Z.J.; Tan, L. Multivariate time series fuzzy clustering segmentation mining algorithm. J. Univ. Sci. Technol. Beijing 2014, 36, 260–265.
3. Xu, R.C.; Yan, W.W.; Wang, G.L.; Yang, J.C.; Zhang, X. Time series forecasting based on seasonality modeling and its application to electricity price forecasting. Acta Autom. Sin. 2020, 46, 1136–1144.
4. Wang, W.B.; Fei, P.S.; Yi, X.M. Prediction of China stock market based on EMD and neural network. Syst. Eng.-Theory Pract. 2010, 30, 1027–1033.
5. Zhou, B.; Yan, H.S. Financial time series forecasting based on wavelet and multi-dimensional Taylor network dynamics model. *Syst. Eng.-Theory Pract.* **2013**, *33*, 2654–2662.

6. Chen, Z.; Liu, J.; Li, C.; Ji, X.; Li, D.; Huang, Y.; Di, F. Ultra short-term power load forecasting based on combined LSTM-XGBoost model. *Power Syst. Technol.* **2020**, *44*, 614–620.

7. Peng, L.; Niu, R.Q.; Wu, T. Time series analysis and support vector machine for landslide displacement prediction. *J. Zhejiang Univ.* **2015**, *47*, 1672–1679.

8. Wang, Y.; Gu, Y.; Ding, Z.; Li, S.; Wan, Y.; Hu, X. Charging demand forecasting of electric vehicle based on empirical mode decomposition-fuzzy entropy and ensemble learning. *Autom. Electr. Power Syst.* **2020**, *44*, 114–124.

9. Yule, G.U. On a method of investigating periodicities in disturbed series, with special reference to Wolfer’s sunspot numbers. *Philos. Trans. R. Soc. Lond.* **1927**, *226*, 267–298.

10. Walker, G. On periodicity in series of related terms. *Proc. R. Soc. Lond.* **1931**, *131*, 518–532. [CrossRef]

11. Box, G.E.; Jenkins, G.M. The likelihood function for stationary autoregressive-moving average models. *Biometrika* **1979**, *66*, 265–270.

12. Caginalp, G.; Constantine, G. Statistical inference and modelling of momentum in stock prices. *Appl. Math. Financ.* **1995**, *2*, 225–242. [CrossRef]

13. Engle, R.F. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econom. J. Econom. Soc.* **1982**, *50*, 987–1007. [CrossRef]

14. Bollerslev, T. Generalized autoregressive conditional heteroscedasticity. *EERI Res. Pap.* **1986**, *31*, 307–327.

15. Goldberg, D.E.; Holland, J.H. Genetic Algorithms and Machine Learning. *Mach. Learn.* **1988**, *3*, 95–99. [CrossRef]

16. Palmer, A.; Montano, J.J.; Albert, S. Designing an artificial neural network for forecasting tourism time series. *Tour. Manag.* **2006**, *27*, 781–790. [CrossRef]

17. Nikolaev, N.Y.; Smirnov, E.; Stamate, D.; Zimmer, R. A regime-switching recurrent neural network model applied to wind time series. *Appl. Soft Comput.* **2019**, *80*, 723–734. [CrossRef]

18. Chen, H.; Murray, A.F. Continuous restricted Boltzmann machine with an implementable training algorithm. *IEEE Access* **2017**, *5*, 1527–1554. [CrossRef]

19. Zhang, T.B.; Murray, A.F. Adaptive sensor modelling and classification using a continuous restricted Boltzmann machine (CRBM). *Neurocomputing* **2006**, *70*, 1198–1206. [CrossRef]

20. Peng, L.; Niu, R.Q.; Wu, T. Time series analysis and support vector machine for landslide displacement prediction. *J. Zhejiang Univ.* **2015**, *47*, 1672–1679.

21. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [CrossRef]

22. Wang, Q.; Wang, L.; Liu, Y.; Zhao, J.; Wang, W.C. Time series prediction with incomplete dataset based on deep bidirectional echo state network. *IEEE Access* **2019**, *7*, 152533–152544. [CrossRef]

23. Abedinia, O.; Amjady, N.; Shafie-Khah, M.; Catalão, J.P. Electricity price forecast using combinatorial neural network trained by a new stochastic search method. *Energy Convers. Manag.* **2015**, *105*, 642–654. [CrossRef]

24. Chou, C.C.; Lin, K.S. A fuzzy neural network combined with technical indicators and its application to Baltic Dry Index forecasting. *J. Mar. Eng. Technol.* **2019**, *18*, 82–91. [CrossRef]

25. Hinton, G.E.; Osindero, S.; The, Y.W. A fast learning algorithm for deep belief nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef]

26. Salakhutdinov, R.; Hinton, G.E. Deep boltzmann machines. *J. Mach. Learn. Res.* **2009**, *5*, 448–455.

27. Hinton, G.E. Training products of experts by miminizing contrastive divergence. *Neural Comput.* **2002**, *14*, 1771–1800. [CrossRef] [PubMed]