

Article

A Prediction Method with Data Leakage Suppression for Time Series

Fang Liu , Lizhi Chen, Yuanfang Zheng and Yongxin Feng *

School of Information Science & Engineering, Shenyang Ligong University, Shenyang 110159, China

* Correspondence: fengyongxin@sy.lu.edu.cn

Abstract: In view of the characteristics of the collected time series, such as being high noise, non-stationary and nonlinear, most of the current methods are designed to smooth or denoise the whole time series at one time and then divide the training set and testing set, which will lead to using the information of the testing set in the training process, resulting in data leakage and other problems. In order to reduce the impact of noise on time series prediction and prevent data leakage, a prediction method with data leakage suppression for time series (DLS) is proposed. This prediction method carries out multiple variational mode decomposition on the time series by overlapping slicing and improves the noise reduction threshold function to perform noise reduction processing on the decomposed time series. Furthermore, the idea of deep learning is introduced to establish a neural network multi-step prediction model, so as to improve the performance of time series prediction. The different datasets are selected as experimental data, and the results show that the proposed method has a better prediction effect and lower prediction error, compared with the common multi-step prediction methods, which verifies the superiority of the prediction method.

Keywords: time series; data leakage; overlapping slicing; noise reduction threshold function



Citation: Liu, F.; Chen, L.; Zheng, Y.; Feng, Y. A Prediction Method with Data Leakage Suppression for Time Series. *Electronics* **2022**, *11*, 3701. <https://doi.org/10.3390/electronics11223701>

Academic Editor: Dionysios Kehagias

Received: 12 October 2022
Accepted: 9 November 2022
Published: 11 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A set of observed values produced in chronological order can be called a time series, and these time series widely exist in many fields. Analyzing the observed values of time series and predicting the series values at future time points using various methods is called time series prediction. Time series prediction has great significance. For example, in the financial field, the development trend of financial time series can be understood by predicting financial data; in the power field, energy distribution can be guided through power load prediction; in the medical field, the transmission of diseases can be prevented by predicting disease incidence.

Time series is closely related to human activities and has high noise, non-stationary and nonlinear characteristics, etc., which results in the accuracy of time series prediction generally not being high. Therefore, a lot of research on time series prediction methods has been carried out by domestic and foreign scholars. Traditional time series prediction methods are mostly limited to a fixed model framework and have stricter assumptions, and these methods use statistical knowledge to construct the model, according to the development rule of time series, which extends the time series, so as to predict the subsequent time series. The popular traditional time series prediction methods, such as random walk model, autoregressive moving average model [1,2] and generalized autoregressive conditional heteroskedasticity model [3], have higher requirements for data and cannot get better prediction results for complex non-linear time series. In recent years, modern time series prediction methods have mainly used machine learning and deep learning techniques, such as support vector machine [4,5], artificial neural network [6–9] and so on. The neural network prediction method pays more attention to the data itself and deals with non-linear problems through activation function, so it can better handle time series prediction problem

and provide more accurate prediction results. Recurrent neural network (RNN) [10] has a strong memory function and has certain advantages in dealing with time series problems, and it can use historical data to theoretically solve the long-term dependence problem of time series. However, RNNs are learned by back propagation, so the gradient will disappear or decrease when the input time series is long. Hochreiter et al. [11] proposed a long short-term memory (LSTM) network model to improve the problems in the RNN network, which can well learn and process the long short-term dependencies of data.

The collected data in practice often have the characteristics of being high noise, non-stationary, nonlinear and so on, which means that a single model cannot achieve better results. Therefore, some scholars consider using signal decomposition [12] and noise reduction methods [13,14] to process the original time series. In order to solve the noise problem of time series, various noise reduction methods have been proposed by domestic and foreign scholars, which can be roughly divided into noise reduction methods based on singular spectrum analysis (SSA), wavelet transform (WT) and empirical mode decomposition (EMD). For example, Dai Hailiang et al. [15] proposed a non-linear motion modeling method combining wavelet multi-scale decomposition with singular spectrum analysis, which can more accurately extract useful information, such as trend and period from the finite scale time series with noise. Ma Jun et al. [16] proposed a method based on wavelet transform and using information entropy theory to eliminate colored noise to improve the model prediction effect. Pham et al. [17] proposed a hybrid method combining singular spectrum analysis with the deep learning neural network for short-term load demand forecast. To improve the estimation accuracy and reliability of wind power, Saroha et al. [18] proposed a linear time-delay neural network based on wavelet transform to carry out probability wind power prediction under the time series framework. Chacon et al. [19] proposed a method to improve the prediction ability of financial time series by using the complete set empirical mode decomposition of the adaptive noise and the intrinsic sample entropy. Zhao Yangyang et al. [20] proposed a short-time metro passenger flow prediction model based on empirical mode decomposition and long short-term memory network, which provides more experience for subsequent research. The key to the noise reduction method of singular spectrum analysis is to find the boundary point between the noise and the useful components, but it is difficult to determine the number of singular values of the useful components for the actual time series, which limits the noise reduction effect. The noise reduction method of the wavelet transform depends on the selection of the wavelet basis function and the number of decomposition layers, which often requires prior knowledge, resulting in a great reduction in its applicability. Due to the abuses of EMD and its improved algorithms, the decomposition is not accurate enough, and the effect is limited. Variational mode decomposition (VMD) [21] is a decomposition estimation method proposed by Dragomiretskiy, which improves empirical mode decomposition and has a stronger theoretical grounding and more accurate decomposition than EMD.

Although the use of noise reduction methods is very common, most of the current noise reduction methods are designed to smooth or denoise the whole time series at one time, then divide the time series into training set and testing set, which will lead to using the information of the testing set in the training process, resulting in data leakage and other problems. Aimed at the above problems, using overlapping slicing method to process the time series several times is more in line with the actual situation, and a prediction method with data leakage suppression for time series (DLS) is proposed. The method carries out multiple attempts at VMD processing on the time series through overlapping slicing and improves the noise reduction threshold function to retain the decomposed low-frequency sequence and denoises the decomposed high-frequency sequence. Further, the modeling idea of deep learning is introduced to establish the neural network multi-step prediction model [22], so as to improve the prediction performance of time series. Additionally, the prediction accuracy of this method is then compared with a traditional prediction method [23], VMD prediction method based on overlapping slicing, overall VMD prediction method [24], and overall VMD noise reduction prediction method [25].

2. DLS Prediction Method

Considering the data leakage problem in existing time series smoothing or noise reduction processing, the noise reduction threshold function is improved, the noise reduction processing of VMD compromising threshold based on overlapping slicing is constructed and applied to the time series prediction model, and then the DLS prediction method is proposed.

Collecting of the time series. Sequence $x(n)$ with length N is contaminated by noise $u(n)$, $n = 1, 2, \dots, N$, and thus collected sequence with noise can be given as:

$$y(n) = x(n) + u(n) \tag{1}$$

VMD processing of time series $y(n)$. VMD can decompose the input sequence $y(n)$ into different numbers of subsequences with limited bandwidth, and these subsequences are the intrinsic mode function (IMF) components, which can reproduce the original input sequence according to their sparsity as shown in (2) and (3):

$$Y_w(n) = F[y(n)]^A, w = 1, 2, \dots, K \tag{2}$$

$$A = \begin{bmatrix} K & \alpha \\ \tau & \varepsilon \end{bmatrix} \tag{3}$$

where $Y_w(n)$ is the decomposed w -th IMF components; $F[\cdot]$ is the VMD decomposition course; A is a parameter matrix containing the decomposition scale K , penalty factor α , noise margin τ and discriminant accuracy ε .

The experimental demonstration shows that the values of the parameters τ and ε have little influence on the decomposition result, usually set $\tau = 0$ and $\varepsilon = 1 \times 10^{-7}$. Therefore, the selection of the decomposition scale K and penalty factor α in VMD is mainly analyzed. Based on the observed center frequency, the value of K can be determined, and the value of K is set from small to large positive integer values. When the last IMF component maintains a relatively stable center frequency, the value of K at this time is considered to be the best value, and $K \in \mathbf{N}^*$. After the value of K is determined, the impact of different α on VMD run time is observed. With the gradual increase in α , the appropriate value of α is obtained when the run time reaches the first minimum value of time, and $\alpha \in \mathbf{N}^*$. According to this experience, set $K \in [2, 15]$ and $\alpha \in [200, 3000]$, and for this study set, $K = 7$ and $\alpha = 1000$.

By setting the decomposition scale K and penalty factor α , the Wiener filtering noise reduction and the alternating multiplication operator processing are carried out to obtain and update the K center frequencies, and then IMF components are obtained according to the different center frequencies. After several calculations, the IMF components are matched to the optimal center frequency and achieve the effective decomposition of the original sequence.

Low-frequency component $Y_l(n)$ and high-frequency component $Y_h(n)$ in $Y_w(n)$ can be determined by permutation entropy, where $l = 1, \dots, d$, $h = d + 1, \dots, K$, $d \in \mathbf{Z}$. First, the phase space reconstruction is performed on the component $Y_w(n)$. $K = 1$ is taken as an example, the delay time θ and embedding dimension m are determined in the phase space reconstruction method, and the component $Y_1(n)$ is decomposed into $n - (m - 1)\theta$ m -dimensional vectors, as shown in (4).

$$\begin{aligned} \tilde{Y}_{11} &= (Y_1(1), \dots, Y_1(1 + (f - 1)\theta)) \\ \tilde{Y}_{12} &= (Y_1(2), \dots, Y_1(2 + (f - 1)\theta)) \\ &\dots \\ \tilde{Y}_{1i} &= (Y_1(i), \dots, Y_1(i + (f - 1)\theta)) \\ &\dots \\ \tilde{Y}_{1(n-(m-1)\theta)} &= (Y_1((n - (m - 1)\theta)), \dots, Y_1(n)) \end{aligned} \tag{4}$$

Each reconstructed component is rearranged in ascending order, and the column index j_1, j_2, \dots, j_m of the position of each element in the vectors can be obtained as follows:

$$Y_1(i + (j_1 - 1)\theta) \leq \dots \leq Y_1(i + (j_m - 1)\theta) \tag{5}$$

A symbol sequence $S(c)$ that reflects the size order of the elements can be obtained for any reconstructed vector, as shown in (6). The different symbol sequences $\{j_1, j_2, \dots, j_m\}$ mapped by m -dimensional phase space are a total of $m!$. $S(c)$ is one arrangement form of symbol sequences, and each reconstruction component updates $\{j_1, j_2, \dots, j_m\}$ in ascending order after being arranged.

$$S(c) = \{j_1, j_2, \dots, j_m\}, c = 1, 2, \dots, R, R \leq m! \tag{6}$$

The occurrence number of each symbol sequence divided by the total occurrence number of $m!$ of different symbol sequences gives the occurrence probability of the symbol sequence, that is $\{V_1, V_2, \dots, V_R\}$.

The permutation entropy of the IMF components is calculated by using the probability $\{V_1, V_2, \dots, V_R\}$.

$$H_u = - \sum_{c=1}^R V_c \ln(V_c) \tag{7}$$

The value of the maximum permutation entropy is $\ln(m!)$, and the permutation entropy is normalized as follows:

$$0 \leq H'_u = \frac{H_u}{\ln(m!)} \leq 1 \tag{8}$$

The size of the permutation entropy indicates the random degree of the IMF component: the smaller the entropy value, the simpler and more regular the sequence; conversely, the larger the entropy, the more complex and random the sequence. The threshold value is set according to the permutation entropy of the time series $y(n)$ and the empirical value (0.7–0.85) of the high- and low-frequency nodes, and the value of d is the number of IMF components whose permutation entropy is less than the threshold value, and then the low-frequency component $Y_1 \sim Y_d$ and the high-frequency component $Y_{d+1} \sim Y_K$ can be determined.

The VMD noise reduction processing is based on overlapping slicing for time series $y(n)$. This study chooses a fixed threshold value, and the noise reduction threshold function is improved to construct a compromised threshold function, as shown in (9).

$$Y'_{kt} = \begin{cases} \text{sign}(Y_{kt}) \left(|Y_{kt}| - \beta \lambda e^{-|Y_{kt}| \lambda} \right), & |Y_{kt}| \geq \lambda \\ 0, & |Y_{kt}| < \lambda \end{cases} \tag{9}$$

where β is the compromise factor and $\beta \in [0, 1]$; Y_{kt} represents the value of the decomposed component Y_k at t time; Y'_{kt} represents the value of the denoised component Y'_k at t time; $\text{sign}(\)$ represents the symbol function; the threshold of Y_k is $\lambda = \sigma_k \sqrt{2 \ln Q}$; Q is the slice size; $\sigma_k = \frac{\text{median}(|Y_k|)}{0.6745}$; $\text{median}(\)$ represents the median function.

Time series $y(n)$ is processed in unit time step by using a slice of fixed size. The time series $y(n)$ is processed by VMD after slicing; the low-frequency component is preserved, and the high-frequency component is processed by the noise reduction in compromising threshold. The slice size Q is 120 and the slice moves at a step of 1, there will then be $M = N - Q + 1$ slices in total. The time series $y(n)$ is divided into M slices, as shown in (10).

$$F_b = \{y(b), y(b + 1), \dots, y(b + Q - 1)\} \tag{10}$$

where $b = 1, 2, \dots, M$.

The first set of time series slices is extracted as $(y(1), y(2), \dots, y(Q))$, and VMD processing is performed on this sequence to obtain the first decomposition sequence $(y_1(1), y_1(2), \dots, y_1(Q))$, then the low-frequency component of the decomposition sequence is preserved, and the high-frequency component of the decomposition sequence is processed by noise reduction, so as to obtain the first noise reduction sequence $(y'_1(1), y'_1(2), \dots, y'_1(Q))$. Similarly, remaining time series slices of each set continue to be extracted, and the same processing is carried out until time series slices $(y(M), y(M+1), \dots, y(Q+M-1))$ of the last set are processed.

Preparation of training set and testing set. The data from the past P time steps are used to predict the data in the future P time steps, $P \leq P'$, and P' is the sum of the last P steps of each slice. The sequence after decomposition and noise reduction is integrated, then the last P data of each slice are taken to form the input dataset, which is organized into the suitable data format for the input of neural network, and the input dataset is shown in (11):

$$\mathbf{Z}_b = \{y'_b(Q-P+b), y'_b(Q-P+b+1), \dots, y'_b(Q+b-1)\} \quad (11)$$

where $b = 1, 2, \dots, M$.

The label dataset uses the sliding window to process time series $y(n)$, and the window moves one unit time step at a time. The label dataset is shown in (12).

$$\mathbf{G}_b = \{y(Q+b), y(Q+b+1), \dots, y(Q+b+P-1)\} \quad (12)$$

where $b = 1, 2, \dots, M$.

The data in \mathbf{Z}_b that removes the last five rows of slices as training set input \mathbf{Z}_{ir} are selected, and the data in \mathbf{G}_b that removes the last five rows of slices as training set label \mathbf{Z}_{or} are selected.

The data of the last row of slices in \mathbf{Z}_b as testing set input \mathbf{Z}_{ie} are selected, and the data of the last row of slices in \mathbf{G}_b as testing set label \mathbf{Z}_{oe} are selected.

The training set input and label form the training set \mathbf{Z}_{train} , and the testing set input and label form the testing set \mathbf{Z}_{test} . Additionally, then the training set and the testing set are integrated into a complete dataset \mathbf{Z} .

Preprocessing of time series. The neural network algorithm needs to standardize the time series data to prevent the influence between the current values of each variables being too large. At the same time, standardization can also improve the model prediction accuracy and convergence rate. Therefore, before the model training, the maximum and minimum values of training set data are selected to normalize the training and testing data, as shown in (13).

$$\mathbf{Z}' = \frac{\mathbf{Z} - \min(\mathbf{Z}_{train})}{\max(\mathbf{Z}_{train}) - \min(\mathbf{Z}_{train})} \quad (13)$$

where \mathbf{Z}' is the normalized data.

The normalized training set input, training set label, testing set input, testing set label are $\mathbf{Z}'_{ir}, \mathbf{Z}'_{or}, \mathbf{Z}'_{ie}, \mathbf{Z}'_{oe}$, respectively.

Then neural network is trained and appropriate hyper-parameters are selected for prediction. The long short-term memory network structure is adopted based on attention mechanism, and the model parameters of LSTM are set as follows: the number of hidden layers is 1, the number of neurons is 64, the tanh function is set as the activation function, the number of iterations is set as 300 times to ensure the experiment effect, the early stop mechanism is adopted, the step size of LSTM is set as 5, the batch size is set as 32, the mean-square error (MSE) of the normalized predicted value and true value as the loss function is used, Adam optimization algorithm to update the parameters is used, and the dimension of the output layer is 5.

The normalized training set input \mathbf{Z}'_{ir} is input into the neural network, and the prediction data $\tilde{\mathbf{Z}}_{or}$, which is output from the neural network, is shown in (14).

$$\tilde{\mathbf{Z}}_{or} = L[\mathbf{Z}'_{ir}]^{\text{MSE}} \quad (14)$$

where $L[\cdot]$ is the neural network function and MSE is the evaluating indicator.

The parameters of back propagation are updated by using MSE as the evaluating indicator and training is stopped after MSE no longer drops or reaches the maximum number of iterations.

Prediction of future data. The trained neural network is used to predict the future data, the normalized testing set input Z'_{ie} is input into the trained neural network, and the prediction data \tilde{Z}_{oe} obtained is shown in (15).

$$\tilde{Z}_{oe} = L[Z'_{ie}]^{\text{MSE}} \quad (15)$$

The prediction data \tilde{Z}_{oe} is anti-normalized to get the predicted P step future data \tilde{T} .

3. Experiments

3.1. Effectiveness Analysis of Noise Reduction

The simulated mixed signal sequence is constructed, as shown in (16–19).

$$x_1(t) = 2 \times \sin\left(50 \times \pi \times t + \frac{\pi}{2}\right) \quad (16)$$

$$x_2(t) = (t + 1) \times \sin\left(20 \times \pi \times t + \frac{\pi}{4}\right) \quad (17)$$

$$x(t) = x_1(t) + x_2(t) + r(t) \quad (18)$$

$$x_0(t) = x_1(t) + x_2(t) \quad (19)$$

where $t = [0, 0.99975]$; the sampling interval is 0.00025 s; $r(t)$ is the random noise signal sequence, that is gaussian white noise; $x(t)$ is the mixed simulation signal sequence with noise; $x_0(t)$ is the original noise-free signal sequence; $x_1(t)$ and $x_2(t)$ are the signal sequence components of $x_0(t)$.

The original noise-free signal sequence $x_0(t)$ and the mixed simulation signal sequence $x(t)$ with noise are shown in Figures 1 and 2.

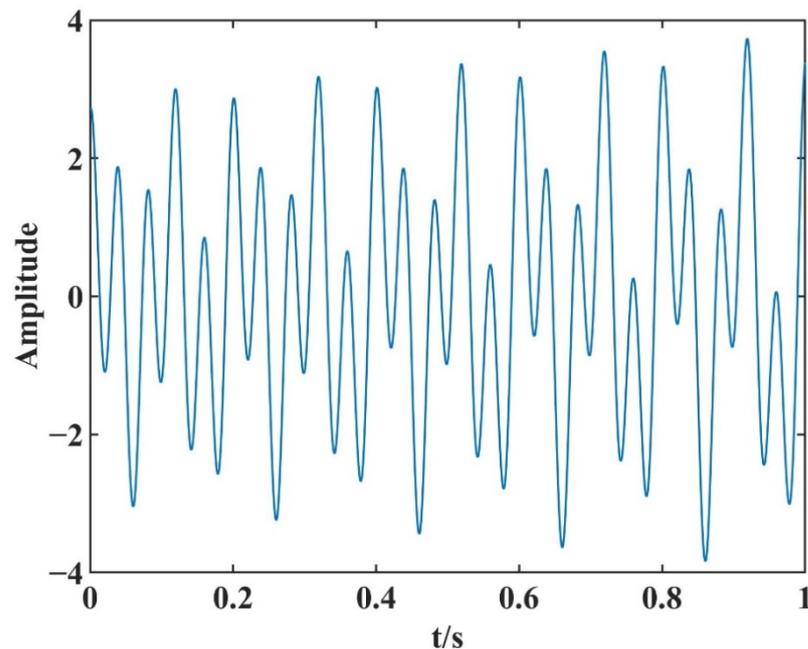


Figure 1. Original noise-free signal sequence $x_0(t)$.

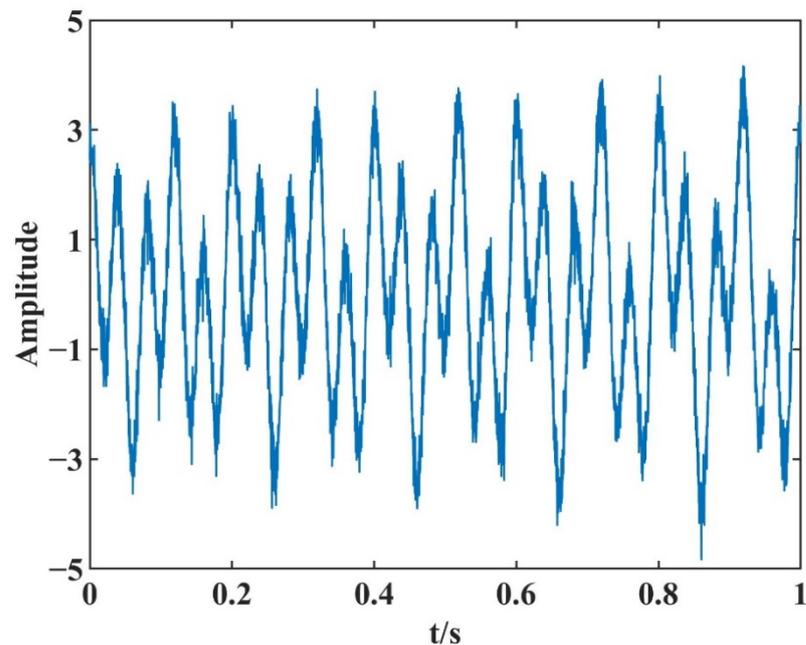


Figure 2. Mixed simulation signal sequence $x(t)$ with noise.

The mixed simulation signal sequence $x(t)$ with noise is carried out with the noise reduction processing of wavelet soft threshold function and VMD compromising threshold, respectively, and two noise reduction results of the signal sequences are obtained, as shown in Figures 3 and 4.

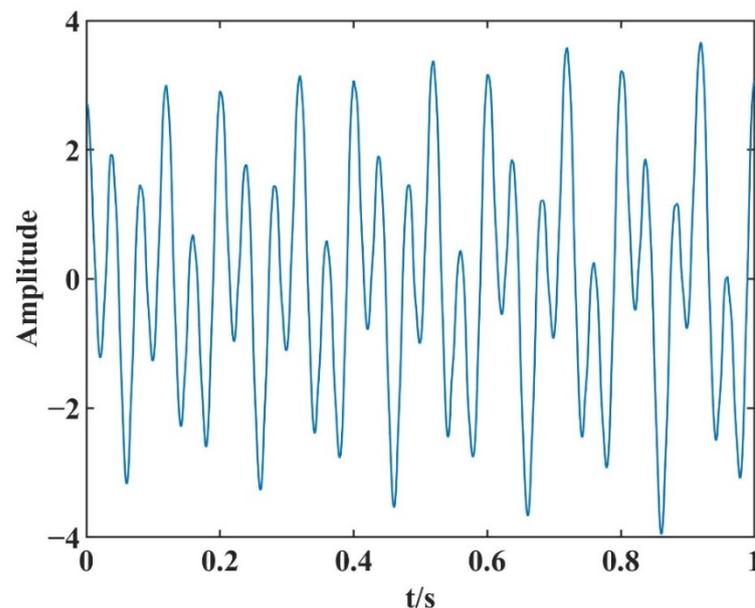


Figure 3. Noise reduction result of wavelet soft threshold function.

The morphological characters of the signal sequences before and after noise reduction can be visually compared from Figures 3 and 4. The signal-to-noise ratio (SNR) and the root mean squared error (RMSE) are selected as the noise reduction evaluating indicators for the noise reduction effects of the different examples of processing, as shown in (20) and (21). After calculation, the evaluating indicators are SNR = 25.9460 and RMSE = 0.0897 for the noise reduction processing of wavelet soft threshold function; the evaluating indicators are SNR = 28.2607 and RMSE = 0.0687 for the noise reduction processing of VMD compromise threshold. Through the comparison diagrams and evaluating indicators of noise reduction,

it shows that the noise reduction processing of the VMD compromise threshold has better noise reduction effect, and its validity is verified.

$$\text{SNR} = 10 \times \lg \frac{\sum_{i=1}^N x_{0i}^2}{\sum_{i=1}^N (x_{0i} - x'_{0i})^2} \quad (20)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_{0i} - x'_{0i})^2} \quad (21)$$

where x'_{0i} is the noise-free signal sequence; x_{0i} is the signal sequence after noise reduction; N is the length of the sequence.

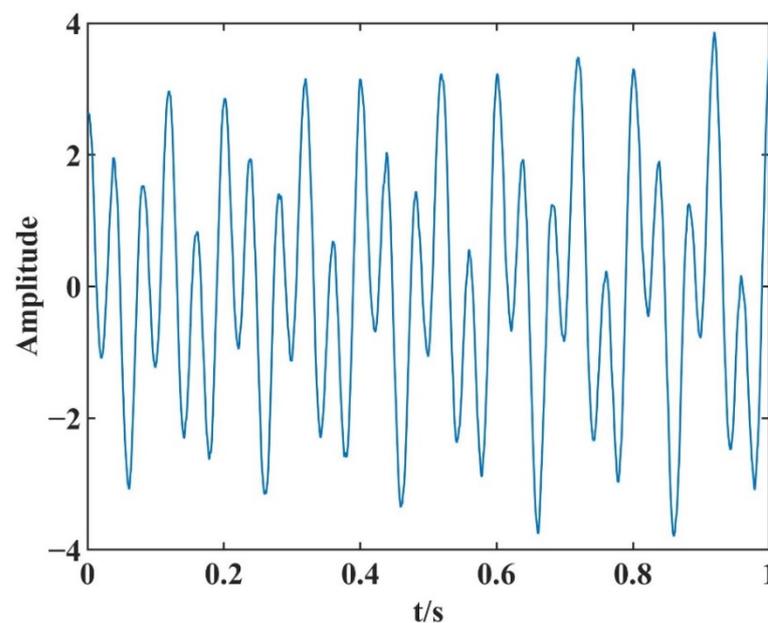


Figure 4. Noise reduction result of VMD compromising threshold.

3.2. Predicted Results and Analysis

By decomposing the data, the DLS method retains the decomposed low-frequency components directly and reduces the noise of the decomposed high-frequency components. Therefore, this method is suitable for time series prediction, where the high-frequency information is not dominant and the low-frequency information is dominant. This test selects datasets with such characteristics as the main test objects, among which the satellite clock error data and the stock data are more representative. In order to ensure the reliability of the experimental results and avoid the contingency of the experimental results, multi-group datasets are set up for test analysis. The selected data can be divided into two types, the first is GPS satellite clock error data from IGS, and the second is stock data from financial circulation. The first type uses the final satellite clock error data with a sampling interval of 30 s, and the compressed clock error file `igs21526.clk_30s` can be downloaded from the website <ftp://garner.ucsd.edu/pub/products/> (accessed on 8 June 2022). The file is the clock error data with a sampling interval of 30 s on 10 April 2021, and the satellite clock error sequences of G05 and G24 are extracted as experimental data. The second type uses the closing price transaction dataset of the Shanghai and Shenzhen 300 Index (CSI300), the closing price trading dataset of the Shanghai Composite Index and the closing price trading dataset of the Shenzhen Component Index, which can be downloaded from the website <https://money.163.com/stock/> (accessed on 8 June 2022). The closing price trading dataset of the CSI 300 is the daily closing price trading data from July 2005 to June 2021, the closing

price trading dataset of the Shanghai Composite Index is the daily closing price trading data from October 2005 to April 2022 and the closing price trading dataset of the Shenzhen Component Index is the daily closing price trading data from October 2005 to April 2022.

Therefore, the DLS method is tested and analyzed based on these two types of data, and four comparison methods are set up:

- (1) Traditional prediction method (LSTM). First, the dataset is divided into training set and testing set. Then, the training sample and testing sample are processed using the sliding window after normalizing the dataset and the data of the next five days is predicted based on the data from the past five days. The labels of both training sample and testing sample are the original data.
- (2) VMD prediction method based on overlapping slicing (P-VMD-LSTM). The first 120 data in the dataset are obtained by overlapping slicing each time they carried out VMD processing and processing the last 5 data of the processed 120 data. Then, the training sample and testing sample are processed by sliding window after normalizing the dataset, and the data of the next five days is predicted based on the data from the past five days. The labels of both the training sample and testing sample are the original data.
- (3) Overall VMD prediction method (VMD-LSTM). First, the dataset is divided into training set and testing set, and the whole training set is carried out with VMD processing. Then, the training sample and testing sample are processed using the sliding window after normalizing the dataset, and the data of the next five days is predicted based on the data from the past five days. The labels of both the training sample and testing sample are the original data.
- (4) Overall VMD noise reduction prediction method (VMD-LSTM-NR). First, the dataset is divided into training set and testing set, and the whole training set is carried out with the VMD noise reduce processing. Then, the training sample and testing sample are processed using the sliding window after normalizing the dataset, and the data of the next five days is predicted based on the data from the past five days. The labels of both the training sample and testing sample are the original data.

The prediction results based on the G05 and G24 using each method are shown in Figures 5 and 6.

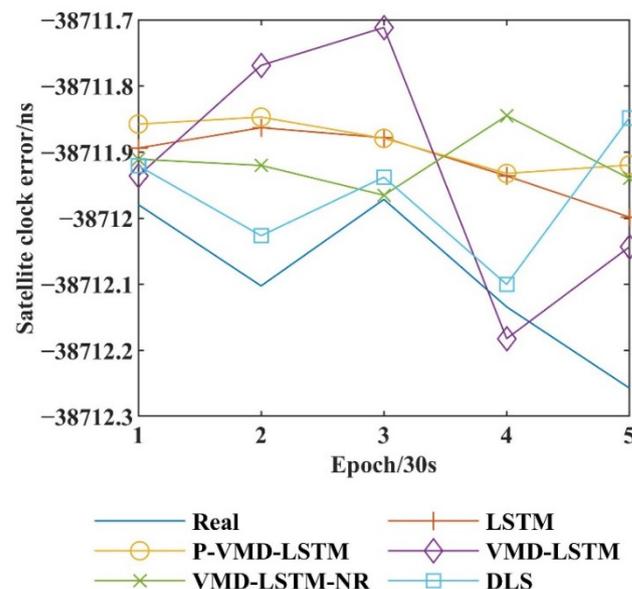


Figure 5. Comparison of prediction results based on the G05.

In Figure 5, the effect of the DLS method is significantly better than the other comparison methods. Except for the fact that there are deviations of the prediction value and the

direction trend in the last step, the prediction results of the other steps have good effects. However, other comparison methods can only have good results in the first half or the second half of the prediction, which shows that these methods are significantly worse than the DLS method.

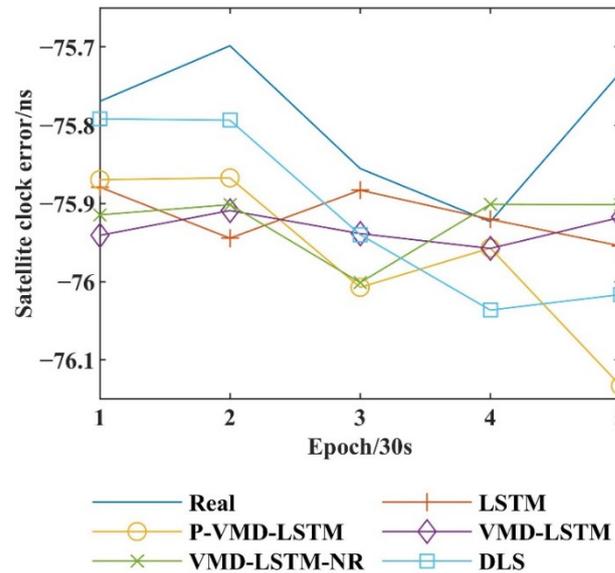


Figure 6. Comparison of prediction results based on the G24.

It can be seen from Figure 6 that all the methods have good prediction results on the overall direction trend of prediction, but careful comparison shows that each method has different effects on different prediction time steps. In order to better analyze the prediction effectiveness, further discussion is needed through the evaluation indicators.

The comparison curves of the prediction value and the true value of the CSI300 are shown in Figure 7. It can be seen that the DLS method is superior to other comparison methods in fitting most of the data, and the daily fluctuation trend can be consistent. Except for the fact that the LSTM method can also meet the daily fluctuation trend, all of other comparison methods show deviations on some days. Therefore, whether discussing the fitting or the fluctuation trend of the predicted value and the true value, the DLS method is superior to other comparison methods and has better prediction effect.

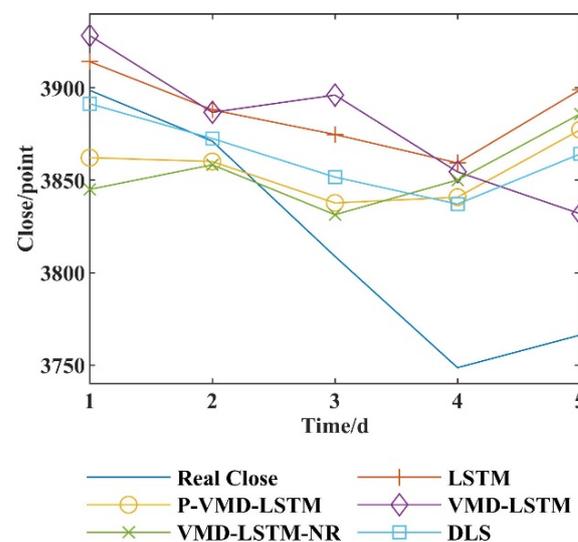


Figure 7. Comparison of prediction results based on the CSI 300.

The comparison curves of the predicted value and the true value of the Shanghai Composite Index and the Shenzhen Component Index are shown in Figures 8 and 9. By comparison result of the predicted value and the true value in Figure 8, it shows that the predicted value of the DLS method is closer to the true value, and the daily fluctuation trend can be consistent, while some of comparison methods show deviations from the daily fluctuation trend. Although the DLS method does not maintain the same daily fluctuation trend in Figure 9, it still has better data fitting, compared with other comparison methods.

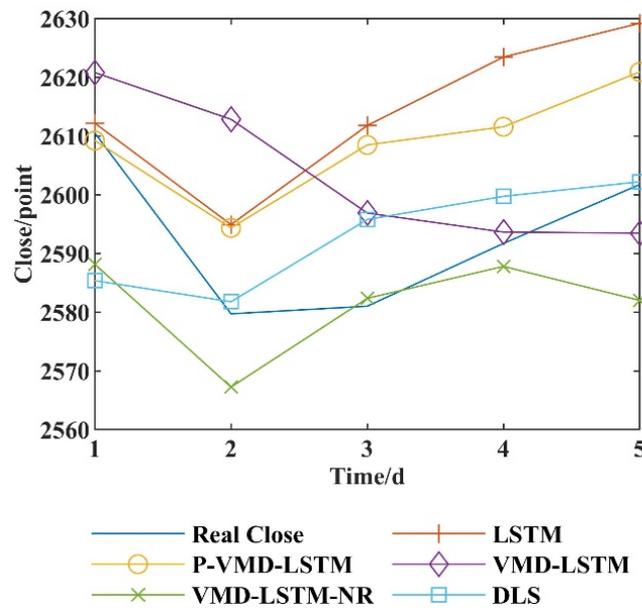


Figure 8. Comparison of prediction results based on the Shanghai Composite Index.

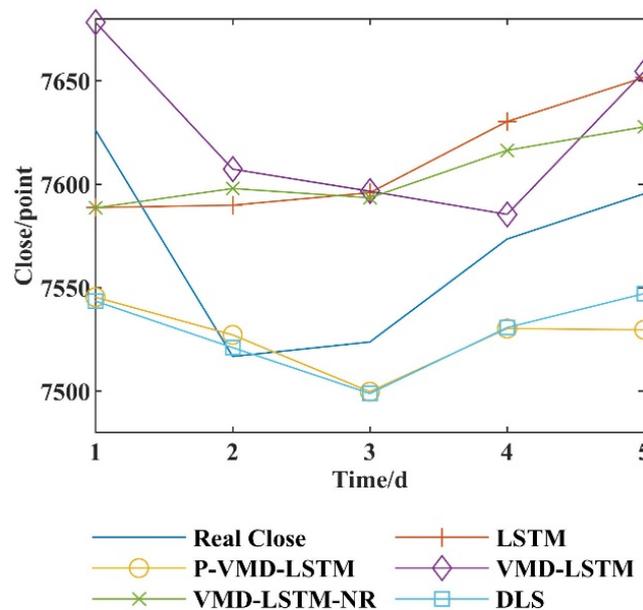


Figure 9. Comparison of prediction results based on the Shenzhen Component Index.

Mean square error (MSE), mean absolute error (MAE), mean absolute percentage error (MAPE) and symmetric mean absolute percentage error (SMAPE) are used as prediction evaluating indicators to judge the degree of fitting between predicted values and true values, as shown in (22) to (25).

$$\text{MSE} = \frac{1}{P} \sum_{i=1}^P [z'_{oe}(i) - \tilde{z}_{oe}(i)]^2 \quad (22)$$

$$\text{MAE} = \frac{1}{P} \sum_{i=1}^P |z'_{oe}(i) - \tilde{z}_{oe}(i)| \quad (23)$$

$$\text{MAPE} = \frac{100\%}{P} \sum_{i=1}^P \left| \frac{z'_{oe}(i) - \tilde{z}_{oe}(i)}{z'_{oe}(i)} \right| \quad (24)$$

$$\text{SMAPE} = \frac{100\%}{P} \sum_{i=1}^P \frac{|z'_{oe}(i) - \tilde{z}_{oe}(i)|}{(|z'_{oe}(i)| + |\tilde{z}_{oe}(i)|)/2} \quad (25)$$

where P is the length of the testing set label; $z'_{oe}(i)$ is the true value; $\tilde{z}_{oe}(i)$ is the predicted value.

The evaluating indicators of the prediction results on the G05 and G24 datasets are shown in Tables 1 and 2, respectively. It can be seen from Table 1 that all the evaluation indicators need to be analyzed together. The LSTM method and DLS method have the best results on MSE and MAE, respectively, and the MSE of the DLS method is not significantly inferior to the LSTM method. Moreover, the MAE of the DLS method is reduced by at least 29%, compared with other methods, and the MAPE and SMAPE of the DLS are superior to other methods and are reduced by 29% at least, which indicates that the DLS method has a better prediction effect. The results of the evaluating indicators in Table 2 show that all the four evaluating indicators of the DLS method are the lowest, indicating that the DLS method has better prediction effectiveness, and the MSE, MAE, MAPE and SMAPE are reduced by at least 2%, 1%, 1% and 1%, compared with other methods, respectively.

Table 1. Evaluating indicators of prediction results based on the G05.

Method	MSE	MAE	MAPE	SMAPE
LSTM	0.000008498	0.002692	0.0004520	0.0004520
P-VMD-LSTM	0.00001152	0.003105	0.0005212	0.0005212
VMD-LSTM	0.00001084	0.002767	0.0004645	0.0004645
VMD-LSTM-NR	0.00001052	0.002658	0.0004461	0.0004461
DLS	0.000008479	0.001882	0.0003159	0.0003159

Table 2. Evaluating indicators of prediction results based on the G24.

Method	MSE	MAE	MAPE	SMAPE
LSTM	0.00002110	0.003570	0.1605	0.1603
P-VMD-LSTM	0.00003844	0.005019	0.2256	0.2253
VMD-LSTM	0.00002000	0.004017	0.1806	0.1804
VMD-LSTM-NR	0.00001939	0.004021	0.1808	0.1806
DLS	0.00001896	0.003515	0.1580	0.1578

The comparison results of the prediction evaluating indicators of the CSI300 are shown in Table 3. It can be seen that all the four evaluating indicators of the DLS method are the lowest, and MSE, MAE, MAPE and SMAPE are, respectively, reduced by 16%, 14%, 14% and 14%, compared with the P-VMD-LSTM method, which indicates that the DLS method has better prediction ability than other comparison methods.

The comparison results of the prediction evaluating indicators of the Shanghai Composite Index and the Shenzhen Component Index are shown in Tables 4 and 5. It can be seen from Table 4 that all the four evaluating indicators of the DLS method are the lowest, and MSE, MAE, MAPE and SMAPE are, respectively, reduced by 13%, 15%, 15% and 15%, compared with the VMD-LSTM-NR method. It can be seen from Table 5 that all the four evaluating indicators of the DLS method are the lowest, and MSE, MAE, MAPE and

SMAPE are, respectively, reduced by 13%, 9%, 9% and 9%, compared with the P-VMD-LSTM method. Therefore, it indicates that the DLS method has good prediction ability.

Table 3. Evaluating indicators of prediction results based on the CSI 300.

Method	MSE	MAE	MAPE	SMAPE
LSTM	0.0002718	0.01352	1.8078	1.7837
P-VMD-LSTM	0.0001809	0.01106	1.4766	1.4624
VMD-LSTM	0.0001898	0.01202	1.6033	1.5865
VMD-LSTM-NR	0.0002204	0.01226	1.6356	1.6201
DLS	0.0001512	0.009416	1.2615	1.2481

Table 4. Evaluating indicators of prediction results based on the Shanghai Composite Index.

Method	MSE	MAE	MAPE	SMAPE
LSTM	0.00002656	0.004540	0.8250	0.8206
P-VMD-LSTM	0.00001564	0.003503	0.6369	0.6343
VMD-LSTM	0.00001379	0.002952	0.5370	0.5349
VMD-LSTM-NR	0.000009589	0.002541	0.4601	0.4617
DLS	0.000008301	0.002144	0.3885	0.3890

Table 5. Evaluating indicators of prediction results based on the Shenzhen Component Index.

Method	MSE	MAE	MAPE	SMAPE
LSTM	0.00001477	0.003754	0.7826	0.7799
P-VMD-LSTM	0.00001080	0.002846	0.5909	0.5932
VMD-LSTM	0.00001599	0.003637	0.7583	0.7549
VMD-LSTM-NR	0.00001268	0.003347	0.6982	0.6959
DLS	0.000009379	0.002576	0.5346	0.5366

Because the data distribution and data range of every dataset are different, each prediction method has certain changes on the prediction results and prediction indicators, and the prediction accuracy difference of each method also show changes. By testing the above five datasets that satisfy the dominance of low-frequency information, the test results show that the DLS method can not only effectively suppress the occurrence of the data leakage problem and avoid the use of future data but also has better prediction results than the other four comparison methods. Thus, the DLS method has the best performance on five datasets. Although MSE, MAE, MAPE and SMAPE have slightly different performances due to the data quality, the DLS method still has a better performance than other methods, which proves that the DLS method has certain applicability and superiority.

4. Conclusions

Data leakage can possibly lead to worsened prediction results. When data leakage occurs, it will lead to false high evaluation results. “False high” means that when the model is evaluated, the test information is used in the training processing and it performs well on the testing set, but when the model is deployed to the production environment to solve practical business problems, the performance will be very poor. Additionally, this data leakage does not meet the actual application needs and practical significance. In this study, a processing mechanism based on overlapping slicing is established to avoid data leakage, so that test information will not be used in the training process and the experiment on the testing set meets the actual application requirements. The DLS prediction method is proposed to solve the data leakage problem. The validity analysis of noise reduction verifies that the proposed noise reduction processing of VMD-compromising threshold has a better noise reduction effect. By comparison tests of different datasets, MSE, MAE, MAPE and SMAPE are used to evaluate the prediction results, and the results show that both the

data fitting and the rise and fall trend have a better prediction effect, which verifies the superiority of the proposed method. This method is suitable to be multi-step prediction system and device of time series, which has noise characteristics and is dominated by low-frequency information, has no data leakage problem and has more suitability for practical application.

Author Contributions: Conceptualization, Y.Z.; data curation, Y.Z.; formal analysis, F.L. and Y.F.; investigation, L.C. and Y.Z.; methodology, F.L., L.C. and Y.F.; resources, Y.Z.; software, L.C.; supervision, F.L. and Y.F.; validation, F.L., L.C. and Y.F.; visualization, F.L., L.C. and Y.F.; writing—original draft preparation, L.C. and Y.Z.; writing—review and editing, F.L. and Y.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the National Natural Science Foundation of China (Grant No. 61971291, 61501309), the Central Government Leads Local Science and Technology Development Projects (Grant No. 2022020128-JH6/1001).

Data Availability Statement: <ftp://garner.ucsd.edu/pub/products/> (accessed on 8 June 2022), <https://money.163.com/stock/> (accessed on 8 June 2022).

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

References

1. Yuan, S.; Yang, L.; Shang, B.; Li, X.; Zhang, H. Short term power network gateway load forecasting algorithm based on ARMR model. In Proceedings of the 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference, Chongqing, China, 20–22 May 2016; pp. 497–501.
2. Mbara, G. A double mixture autoregressive model of commodity prices. *Commun. Stat. Case Stud. Data Anal. Appl.* **2021**, *7*, 249–270. [[CrossRef](#)]
3. Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. *J. Econom.* **1986**, *31*, 307–327. [[CrossRef](#)]
4. Li, Y.; Wang, J.; Sun, X.; Li, Z.; Liu, M.; Gui, G. Smoothing-aided support vector machine based nonstationary video traffic prediction towards B5G networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 7493–7502. [[CrossRef](#)]
5. Jaramillo, J.; Velasquez, J.D.; Franco, C.J. Research in financial time series forecasting with SVM: Contributions from literature. *IEEE Lat. Am. Trans.* **2017**, *15*, 145–153. [[CrossRef](#)]
6. Wang, J.Z.; Wang, J.J.; Zhang, Z.G.; Guo, S.P. Forecasting stock indices with back propagation neural network. *Expert Syst. Appl.* **2011**, *38*, 14346–14355. [[CrossRef](#)]
7. Guresen, E.; Kayakutlu, G.; Daim, T.U. Using artificial neural network models in stock market index prediction. *Expert Syst. Appl.* **2011**, *38*, 10389–10397. [[CrossRef](#)]
8. Hu, Y.; Sun, X.; Nie, X.; Li, Y.; Liu, L. An enhanced LSTM for trend following of time series. *IEEE Access* **2019**, *7*, 34020–34030. [[CrossRef](#)]
9. Liu, Y.; Liu, S.; Wang, Y.; Lombardi, F.; Han, J. A survey of stochastic computing neural networks for machine learning applications. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 2809–2824. [[CrossRef](#)]
10. Sadeghi-Niaraki, A.; Mirshafiei, P.; Shakeri, M.; Choi, S.M. Short-term traffic flow prediction using the modified elman recurrent neural network optimized through a genetic algorithm. *IEEE Access* **2020**, *8*, 217526–217540. [[CrossRef](#)]
11. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
12. Yang, Y.; Yang, Y. Hybrid method for short-term time series forecasting based on EEMD. *IEEE Access* **2020**, *8*, 61915–61928. [[CrossRef](#)]
13. Bao, W.; Yue, J.; Rao, Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE* **2017**, *12*, e0180944. [[CrossRef](#)] [[PubMed](#)]
14. Sugiartawan, P.; Pulungan, R.; Sari, A.K. Prediction by a hybrid of wavelet transform and long-short-term-memory neural network. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 326–332. [[CrossRef](#)]
15. Dai, H.L.; Sun, F.P.; Jiang, W.P.; Xiao, K.; Zhu, X.; Liu, J. Application of wavelet decomposition and singular spectrum analysis to GNSS station coordinate time series. *Geomat. Inf. Sci. Wuhan Univ.* **2021**, *46*, 371–380.
16. Ma, J.; Cao, C.D.; Jiang, W.P.; Zhou, L. Elimination of colored noise in GNSS station coordinate time series by using wavelet packet coefficient information entropy. *Geomat. Inf. Sci. Wuhan Univ.* **2021**, *46*, 1309–1317.
17. Pham, M.H.; Nguyen, M.N.; Wu, Y.K. A Novel Short-Term Load Forecasting Method by Combining the Deep Learning with Singular Spectrum Analysis. *IEEE Access* **2021**, *9*, 73736–73746. [[CrossRef](#)]
18. Saroha, S.; Aggarwal, S.K. Wind power forecasting using wavelet transforms and neural networks with tapped delay. *CSEE J. Power Energy Syst.* **2018**, *4*, 197–209. [[CrossRef](#)]
19. Chacón, H.D.; Kesici, E.; Najafirad, P. Improving financial time series prediction accuracy using ensemble empirical mode decomposition and recurrent neural networks. *IEEE Access* **2020**, *8*, 117133–117145. [[CrossRef](#)]

20. Zhao, Y.Y.; Xia, L.; Jiang, X.G. Short-term metro passenger flow prediction based on EMD-LSTM. *J. Traffic Transp. Eng.* **2020**, *20*, 194–204.
21. Dragomiretskiy, K.; Zosso, D. Variational mode decomposition. *IEEE Trans. Signal Process.* **2013**, *62*, 531–544. [[CrossRef](#)]
22. Liu, H.; Long, Z.H.; Duan, Z.; Shi, H. A new model using multiple feature clustering and neural net-works for forecasting hourly PM2.5 concentrations, and its applications in China. *Engineering* **2020**, *6*, 944–970. [[CrossRef](#)]
23. Zhao, Y.; Su, D.; Zou, L. Rogue wave predic-tion based on LSTM neural network. *J. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.)* **2020**, *48*, 47–51.
24. Yang, J.X.; Zhang, S.; Liu, J.C. Short-term photovoltaic power prediction based on variational mode decomposition and long short-term memory with dual-stage attention mechanism. *Autom. Electr. Power Syst.* **2021**, *45*, 174–182.
25. Lu, T.D.; Xie, J.X. Deformation monitoring data de-noising method based on variational mode de-composition combined with sample entropy. *J. Geod. Geodyn.* **2021**, *41*, 1–6.