Hindawi
Complexity
Volume 2024, Article ID 1124822, 19 pages
https://doi.org/10.1155/2024/1124822

Research Article

A Multilevel Wavelet Decomposition Network Hybrid Model Utilizing Cyclic Patterns for Stock Price Prediction

H. R. Wen, Mingchuan Yuan, Shuxin Wang, Lixin Liang, and Xianghua Fu

Shenzhen Technology University, Shenzhen, China

Correspondence should be addressed to Shuxin Wang; wangshuxin@sztu.edu.cn

Received 16 September 2022; Revised 31 October 2023; Accepted 27 November 2023; Published 17 January 2024

Academic Editor: M. De Aguiar

Stock price prediction is an important and complex time-series problem in academia and financial industries. Stock market prices are voted by all kinds of investors and are influenced by various factors. According to the literature studies, such as Elliott's wave theory and Howard's market cycle investment theory, the cyclic patterns are significant characteristics of the stock market. However, even several studies that do consider cyclic patterns (or similar concepts) suffered from the data leakage or boundary problems, which could be impractical for real applications. Inspired by the abovementioned, we propose a hybrid deep learning model called mWDN-LSTM, which correctly utilizes the cyclic patterns' information to predict stock price while avoiding the data leakage and alleviating boundary problems. According to the experiments on two different datasets, our model mWDN-LSTM outperforms the well-known benchmarks such as CNN-LSTM on the same experimental setup and demonstrates the effectiveness of utilizing cyclic patterns in stock price prediction.

1. Introduction

Stock price prediction is a very important and complex problem in the field of financial time-series prediction [1]. Stock price fluctuations are influenced by corporate fundamentals, business cycles, stock market trading rules, international political events, investor sentiments, and other various factors. Due to the abovementioned reasons, stock price prediction is a challenging problem that has attracted more and more researcher's attention.

The main methods of stock price prediction could be classified into two major classes: traditional statistical methods and machine learning methods [2]. The traditional statistical methods have the advantages of solid statistical theory as support. Various statistical methods for stock price prediction such as exponential smoothing model (ESM) [3], vector autoregression (VAR) [4], autoregression integrated moving average (ARIMA) model [5], generalized autoregression conditional heteroskedasticity (GARCH) model [6], and radial basis function (RBF) [3] were proposed and widely adopted in econometrics. However, most of these methods are linear based on hand-crafted factors and are limited by the statistical assumptions that the data are smooth and normally distributed. In this case, such methods may face challenges when analyzing financial series data with large volume, highly noisy, nonlinear, and nonstationary characteristics [7].

Among machine learning methods, deep learning methods based on neural networks are more popular and of better performance [8]. The main advantage of deep learning models is their ability to learn representations from raw data without feature engineering conducted by experienced practitioners, and this advantage makes deep learning models especially suitable for complex systems such as the stock market. Moreover, deep learning models can provide a general approximation of functions for complex, nonlinear, and nonsmooth processes [9]. Therefore, deep learning models are quite worth exploring to deal with financial time-series data.

In addition to the main two classes of methods, some researchers proposed stock prediction methods based on fuzzy system theory. Wu et al. presented a fuzzy momentum contrarian uncertain characteristic system for the classification and quantification of stock characteristics [10]. Based on the suitability index (SI) derived from fuzzy-set theory, Syu et al. presented a stock selection system called TripleS [11].

According to literature [12], the time series consists of four components: trend, cycle, seasonal, and irregular component. Trend is a long-run tendency characterizing the time series. It may be a linear increase or decrease in level over time. It may be stochastic, a result of a random process, or deterministic, a result of a prescribed mathematical function of time. Seasonal components or signals, by contrast, are distinguishable patterns of regular annual variations in a series. These may be due to changes in the precipitation or temperature and so on. Cycles are recurrent data movement patterns over periods. It is also a more or less regular long-range fluctuation above or below some equilibrium level or trend line. They have upswings, peaks, downswings, and troughs. They are studied for their turning points, duration, frequencies, depths, phases, and effects on related phenomena. For example, business cycles are postulated recurrent patterns of prosperity, recession, depression, and recovery. And what is left over after these components are extracted from the series is the irregular or error components.

In this paper, we refer to the cycles (or similar concepts) in the time series of stock prices generally as cyclic patterns. As far as we all know, signal decomposition methods in the data signal processing field can generate features of different frequencies from series data and perform time-frequency analysis on the series data. These features that contain upswings, peaks, downswings, troughs, and cycle information (or frequency information) can be considered as cyclic patterns. In order to capture cyclic patterns, we found, among the signal decomposition methods, discrete wavelet transform (DWT, or discrete wavelet decomposition) and empirical mode decomposition (EMD), which are considered to be effective methods for obtaining cyclic patterns [13].

However, we found that inappropriate procedures for applying the wavelet decomposition to time-series data easily lead to data leakage [14], which uses unobserved data, and its forecasting results would be of extremely high precision, and predictions based on these methods are unreliable. We also found that in order to curb the data leakage, a sliding window mechanism was proposed; however, the wavelet coefficients vary near the endpoint of the transformation window with its shifts and cause boundary problems. The boundary problem causes the generated subseries to be distorted, and the constructed hybrid models are less effective than simple prediction methods in prediction. In this study, we carefully investigate the calculation mechanism of wavelet decomposition and multilevel wavelet decomposition network methods to resolve the two problems mentioned above.

Currently, most hybrid neural network models that utilize cyclic pattern information did not take into consideration the data leakage and boundary problem during the utilization of signal decomposition techniques. Therefore, we would like to propose a hybrid neural network model that utilizes cyclic patterns to predict stock prices, avoiding the data leakage and alleviating boundary problems. We propose the mWDN-LSTM stock price prediction hybrid model, which utilizes the mWDN network to generate cyclic patterns and then uses the LSTM model to make time-series predictions.

The rest of this paper is organized as follows. Section 2 introduces the related research work. Section 3 introduces the proposed mWDN-LSTM in detail. Then, Section 4 and Section 5 present the experimental setup and discuss the experimental results. At last, Section 6 concludes this paper.

The main contributions of this paper are as follows:

(1) We propose a solution that avoids data leakage while alleviating the boundary problem. A multilevel wavelet decomposition neural network and its variants are investigated, which can adaptively adjust the wavelet coefficients.

(2) A new hybrid model combining the wavelet decomposition network and LSTM is proposed, which can effectively utilize the cyclic patterns, and experimental results demonstrate the effectiveness of our proposed model.

2. Related Work

Due to the success of deep learning in recent years, models based on neural networks have gained more and more attention for stock price prediction problems [15]. In 2010, Naeini et al. applied two neural networks, a feedforward multilayer perceptron (MLP) and an Elman recurrent network, to predict a company's stock value based on its history stock price [16]. In 2013, Ticknor proposed the model of feedforward neural networks with Bayesian regularization to predict stock prices, thereby reducing the possibility of model overfitting [17]. In 2015, Rather et al. achieved a high accuracy based on the RNN model for the prediction of 6 stock prices from NSE [18]. In 2016, Di Persio and Honchar employed CNN to predict the S&P 500 price movement. The results showed that CNN achieved better results for financial time series compared to MLP and RNN models [19]. In 2017, Selvin constructed several deep learning models to predict stock prices in the Indian stock market, and in this paper, the following neural network models were employed: deep recurrent neural network (RNN), long short-term memory (LSTM) neural network, and convolutional neural network (CNN). The results of the empirical analysis showed that these models achieved reasonable prediction accuracy for stock prices, and among them, LSTM performed the best [20]. In 2020, Lu et al. proposed a CNN-LSTM-based stock price prediction method. Meanwhile, the prediction models such as MLP, CNN, RNN, LSTM, and CNN-RNN were used to predict the SSE (Shanghai Stock Exchange) stock index, and the proposed CNN-LSTM model demonstrated the best results on MAE, RMSE, and R^2 evaluation criteria [21]. In 2021, Wu and Ming-Tai proposed the SACLSTM stock price prediction algorithm, which constructs a sequence array of historical data and its leading indicators and uses the array as the input image of the CNN framework, and this algorithm has achieved excellent forecasting results for Taiwan and American stocks [22], which is similar to the work proposed by the authors in reference [23]. An LSTM-GA stock trading suggestion system in IOT was proposed, based on historical data and leading indicators [24]. In 2022, Zhang et al. proposed the novel transformer encoder-based attention network (TEANet) framework, which realizes the effective processing and analysis of stock prices to improve the accuracy of stock movement prediction [25].

Some researchers have constructed hybrid models based on signal decomposition techniques and neural networks to exploit the cyclic patterns in the stock market. However, the vast majority of researchers did not take into account the data leakage and boundary problem implicit in the utilization of signal decomposition techniques such as DWT for time-series prediction tasks. For example, in 2019, Qiu et al. decomposed the historical stock price time series using DWT and EMD and then analyzed the obtained subseries and generated prediction by the RVFL model [26]. Chandar decomposed the financial time series using DWT and subsequently inputted the decomposed subseries into ANFIS to predict closing prices [27]. In 2020, Li and Tang proposed the WT-FCD-MLGRU model and chose four major stock indices, S&P 500, IXIC, DJI, and SSE, to test the model performance [28]. In 2021, Wu et al. proposed a combination of ELM and DWT-based models to predict the stock price movements of 400 stocks in China [29]. In the abovementioned study that employed the signal decomposition techniques, the data decomposition of the whole data series, including the training and test sets, was performed before the model was trained. This decomposition operation leads to the problem of data leakage of future data. Therefore, the final results are unrealistic, and similar effects cannot be achieved in practical applications. In addition, in 2018, Hasumi and Kajita found that wavelet-based time-series predictions cannot even outperform a simple prediction when the time series is properly processed due to boundary problems [30]. Since the data leakage and boundary problem may lead to unreliable results, we will explain them in detail in Section 3.2.

There were also research works utilizing cyclic patterns (or similar concepts) in other time-series tasks. In 2018, Wang designed the mWDN network which implemented a multilevel discrete wavelet decomposition process through a neural network called mWDN. This model has a better prediction performance than SAE, RNN, and LSTM in cellphone user numbers and in ECG time-series prediction tasks [31]. In 2020, Zhang proposed a hybrid neural network model based on mWDN in an industrial productivity prediction task that was able to effectively improve the accuracy and granularity of the prediction [32].

3. Model

In this section, first, we introduce how to generate cyclic patterns in the stock market by the discrete wavelet decomposition techniques. Second, we explain the two major problems during the wavelet decomposition procedure that need to be overcome. Third, we introduce the proposed model and each of its components in detail. At last, the training and prediction of our model are introduced.

3.1. MDWD and Cyclic Patterns

3.1.1. MDWD. Multilevel discrete wavelet decomposition (MDWD), a typical discrete signal analysis method, is commonly applied to numerical analysis, time-frequency analysis, denoising, and so on. The process of multilevel discrete wavelet decomposition mainly includes convolution operation and downsampling. The convolution operation can decompose the series into low-frequency and high-frequency subseries. Downsampling was designed to reduce the redundancy of the data, and at the same time, it can keep the total amount of decomposed data consistent with the original data. However, if the translation-invariance of the decomposition process needs to be maintained (the length of each subseries obtained from the decomposition is equal to the length of the original series), this step can be left out.

The multilevel discrete wavelet decomposition process is shown in Figure 1, and the related parameters are shown in Table 1. The implementation steps are as follows:

(1) In the 1-th level of decomposition, the input series $\mathbf{x}$ will do convolution operations with the low-pass filter $\mathbf{l}$ and the high-pass filter $\mathbf{h}$ and generate the intermediate variable series $\mathbf{a}^l(1)$ and $\mathbf{a}^h(1)$, respectively. This step can also be represented in the form of matrix operations.

The formula for the i-th level convolution operation is as follows:

$$\begin{aligned} a_n^l(i+1) &= \sum_{k=1}^{K} x_{n+k-1}^l(i) * l_k, \\ a_n^h(i+1) &= \sum_{k=1}^{K} x_{n+k-1}^l(i) * h_k, \end{aligned} \tag{1}$$

where $x_n^l(i)$ is the n-th element of the low-frequency subseries in the i-th level and $\mathbf{x}^l(0)$ is set as the input series $\mathbf{x}$.

(2) 1/2 downsampling of the intermediate variable series $\mathbf{a}^l(1)$ and $\mathbf{a}^h(\mathbf{1})$ is performed to obtain the low-frequency and high-frequency subseries $\mathbf{x}^l(1)$ and $\mathbf{x}^h(1)$ of the 1-th level decomposition.

(3) The low-frequency subseries $\mathbf{x}^l(1)$ is set as the input series for the next level of decomposition.

(4) After pooling i times for step (1) to step (3), the decomposition result $\boldsymbol{\chi}(i)$ of the i-th level decomposition is obtained.

3.1.2. Cyclic Patterns. In order to utilize the cycle characteristic in the stock market, the first step is to generate cyclic patterns from the raw dataset. For example, discrete wavelet decomposition methods are employed to generate cyclic pattern information. The subseries obtained by discrete wavelet decomposition contain cyclic information (or frequency information), such as cycle fluctuation depth, fluctuation duration, and fluctuation turning point, rendering it consistent with the definition of a cyclic pattern in a time series.

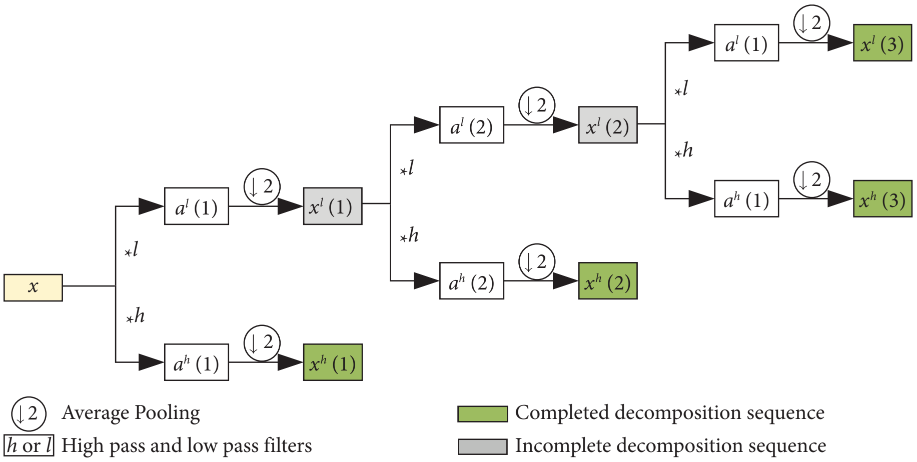

Figure 1: Multilevel discrete wavelet decomposition process.

Table 1: Parameters in the MDWD implementation process.

Parameter	Symbol
Input time series	$\mathbf{x} = \{x_1, \ldots x_t, \ldots, x_T\}$
Low and high subseries in the i-th level	$\mathbf{x}^l(i), \mathbf{x}^h(i)$
Low-pass filter	$\mathbf{l} = \{l_1, \ldots, l_k, \ldots, l_K\}, K \ll T$
High-pass filter	$\mathbf{h} = \{h_1, \ldots, h_k, \ldots, h_K\}, K \ll T$
Low-frequency intermediate variable series	$\mathbf{a}^l(i) = \{a_1^l(i), a_2^l(i), \ldots\}$
High-frequency intermediate variable series	$\mathbf{a}^h(i) = \{a_1^h(i), a_1^h(i), \ldots\}$
i-th level subseries/decomposed result	$\boldsymbol{\chi}(i) = \mathbf{x}^h(1), \mathbf{x}^h(2), \ldots, \mathbf{x}^h(i), \mathbf{x}^l(i)$

For example, we choose the series with a length of 200 and a decomposition level of 2 in Figure 2 to illustrate it. As shown in Figure 3, first, we can notice that the generated series fluctuates more or less regularly at the level of value 0, showing a cyclic fluctuation pattern. Although this series is not a rigorous cyclic series, it is a discrete combination of several cyclic series. Second, the series contains upward fluctuations and downward fluctuations, with the maximum upward fluctuations from $t = 5$ to $t = 18$ and the minimum downward fluctuations from $t = 95$ to $t = 100$. These unidirectional fluctuations are half of the cyclic fluctuations. Therefore, the series contains cyclic fluctuations with a minimum cycle of 10 days and a maximum cycle of 26 days. In addition, the series contains peaks and troughs, and the highest peak in this series is (195, 50.69) and the lowest trough is (5, −54.75). Therefore, the depth of the cyclic fluctuations in this series is between −54.75 and 50.69.

In conclusion, we can get the cycle information (or frequency information), cycle fluctuation depth information, fluctuation duration information, and fluctuation turning point information in the stock series data from the subseries decomposed by discrete wavelet decomposition methods, and it is consistent with the definition of a cycle in time series, so the subseries obtained by discrete wavelet decomposition is the cyclic pattern in the stock market that we need. In our model, the low-frequency subseries are long-term cyclic patterns and the high-frequency subseries are short-term cyclic patterns. We argue that cyclic patterns could be an enhancement for stock market prediction.

3.2. Data Leakage and Boundary Problem.

We find that data leakage and boundary problems are two major problems when applying discrete wavelet decomposition in real stock price prediction applications. In the following, we describe these two problems in detail and introduce our method.

3.2.1. Data Leakage.

Data leakage is the use of information in the model training process which would not be expected to be available at the prediction time, causing the predictive scores (metrics) to overestimate the model's utility when run in a production environment. We include the results of a method with data leakage in our experiments to demonstrate its easily overestimated effect.

When employing DWT with translation-invariance property, the length of the subseries is equal to the length of the original series, which makes many researchers mistakenly believe that they can decompose the original series in a one-time manner and then divide the dataset into the subseries. Based on this, model training and prediction are performed. This process is shown in Figure 4, and it contains data leakage. This is because the wavelet transform works by computing the convolution operation of the time series with

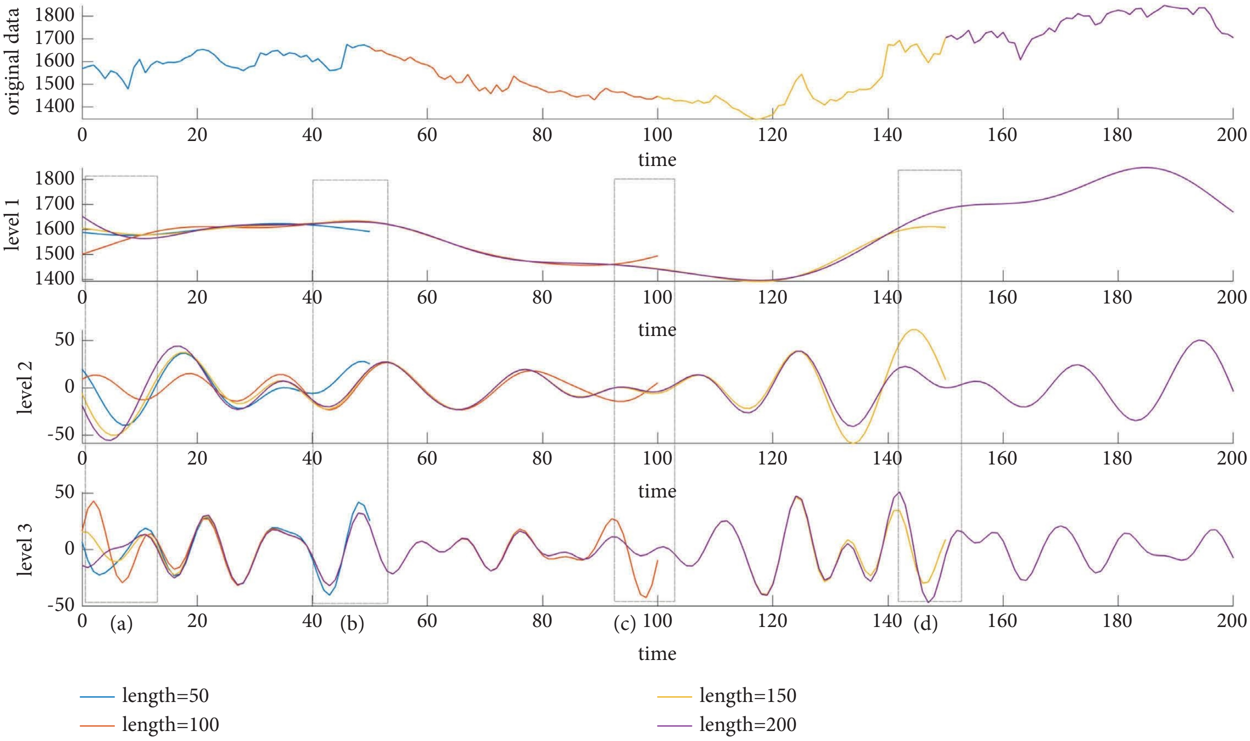

Figure 2: Comparison of series of different lengths after decomposition. (a) is the original data, (b) is the high-frequency subseries of level 3 decomposition, (c) is the high-frequency subseries of level 2 decomposition, and (d) is the high-frequency subseries of level 1 decomposition. The abovementioned decomposition has not been downsampled.

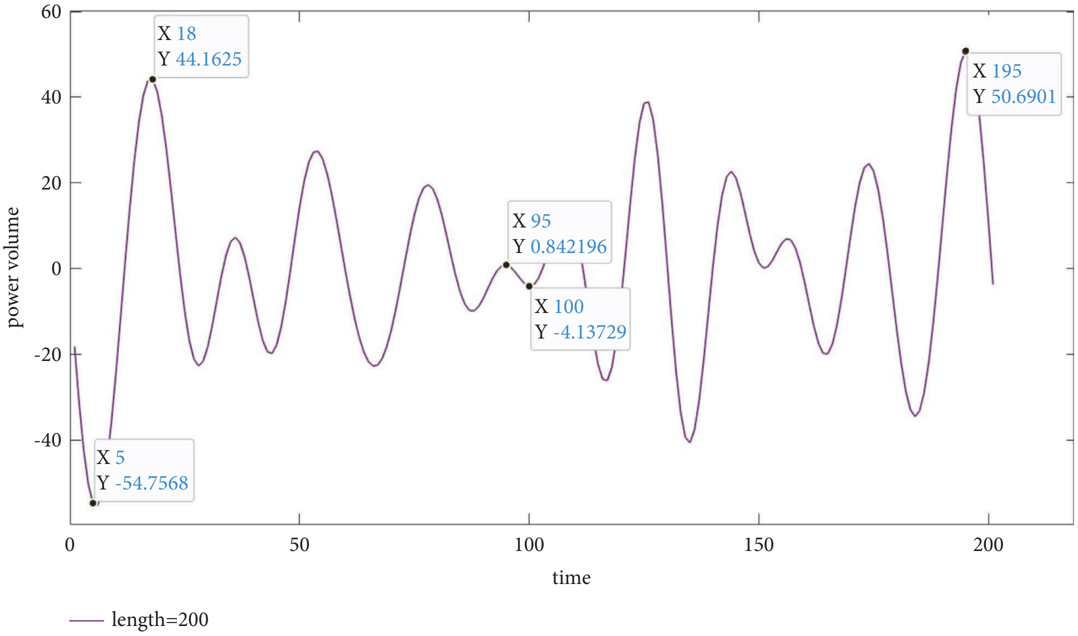

Figure 3: Level 2 decomposition series of length 200 series.

the selected wavelet function. When calculating the output of a point in a time series, it is necessary to convolve the wavelet function with that point and several points before and after it. As shown in the case of Figure 5, x_0 to x_9 are the time-series data arranged in a chronological order. The output of the data point x_1 needs to be obtained by convolution calculation with x_0, x_2, x_3, x_8, and x_9. The output of the data point x_3 needs to be obtained by convolution calculation with x_0, x_1, x_2, x_4, and x_5. So, the output of the convolution operation is a local combination of data points, and the decomposed components involve historical and future data. This is a typical data leakage problem.

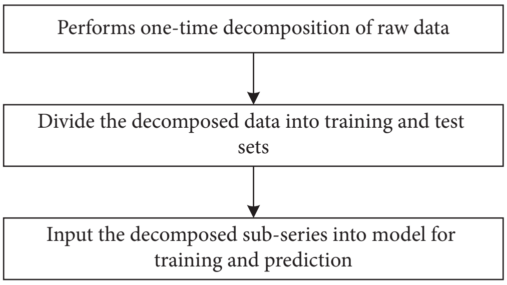
Performs one-time decomposition of raw data
Divide the decomposed data into training and test sets
Input the decomposed sub-series into model for training and prediction

Figure 4: Wavelet decomposition-based prediction process in the existence of data leakage.

$$\underset{W}{\begin{bmatrix} h_3 & h_4 & h_5 & 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 \\ h_2 & h_3 & h_4 & h_5 & 0 & 0 & 0 & 0 & h_0 & h_1 \\ h_1 & h_2 & h_3 & h_4 & h_5 & 0 & 0 & 0 & 0 & h_0 \\ h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & 0 & 0 & 0 & 0 \\ 0 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & 0 & 0 & 0 \\ 0 & 0 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & 0 & 0 \\ 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & 0 \\ 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 \\ h_5 & 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 & h_4 \\ h_4 & h_5 & 0 & 0 & 0 & 0 & h_0 & h_1 & h_2 & h_3 \end{bmatrix}} \underset{X}{\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix}} = \underset{Y}{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \end{bmatrix}}$$

Figure 5: Schematic diagram of wavelet transform convolution calculation. W is the regular wavelet convolution matrix case, X is the series data, and Y is the output.

This warns us that data decomposition should not involve prediction points and their subsequent data.

3.2.2. Boundary Problem. When we take measures to precisely control the decomposed series to avoid data leakage, such as sliding windows, the prediction results are significantly affected by the boundary problem and the model cannot generate accurate predictions.

In order to illustrate the boundary problem, we plot Figure 2 to show the difference in decomposing time series of different lengths. It also shows the difference of the corresponding output between whether the same point in time is at the boundary. The data are the SSE Composite Index data which are a part of the experimental dataset. We apply discrete wavelet decomposition to decompose the data into three components by expanding the number of data points from 50 to 200.

As can be seen from Figure 2, in the results of discrete wavelet decomposition of series data with lengths of 50, 100, 150, and 200, there are huge differences in the calculated results at the boundaries of the series. The subseries at the boundary are off-track and distorted. The subseries at the boundary are off-track and distorted, such as the four areas A, B, C, and D in Figure 2. This is caused by the assumption of circularity that the computation of the boundary involves the data at the other edge of the window. For example, the output of x_9 in Figure 5 needs to be calculated together with x_0 and x_1. The prediction of future data should be performed with the most recent data possible, rather than data from the other end of the sliding window, which would cause large biases in the prediction results.

We avoid data leakage by applying the sliding window mechanism with mWDN to replace the one-time wavelet decomposition. By establishing a new wavelet convolutional operation matrix and incorporating an adaptive adjustment mechanism for mWDN parameters, we aim to mitigate the impact of boundary problem on the prediction accuracy. This is elucidated in both the input layer and the mWDN component.

3.3. mWDN-LSTM Model. Our model mWDN-LSTM can be divided into four components: input layer, mWDN component, LSTM component, and output component. The model structure diagram is shown in Figure 6. In the first component, the input layer is designed to set up the sliding window and normalize the data. In the second component, the mWDN component is designed to implement wavelet decomposition and decompose the series data to generate cyclic patterns. In the third component, the LSTM component is utilized to learn and memorize long-term and short-term information and to make predictions. In the fourth component, the output component is a fully connected network that is utilized to convert the output vector into the final prediction.

3.3.1. Input Layer. In order to avoid data leakage and make the solution practically feasible, we can only decompose in real-time and predict while decomposing, so we use a sliding window mechanism as shown in Figure 7. The window is set up in front of the prediction point and moves forward one unit at a time until all data points are covered. mWDN also only decomposes the data within the window. This mechanism ensures that the decomposition process is real-time and does not include future data, which makes the prediction results realistic and reliable and can be deployed in real investment scenarios.

3.3.2. mWDN Component. In order to obtain cyclic patterns using discrete wavelet decomposition while alleviating the influence of boundary problems on prediction, we set up a new convolutional operation matrix and utilize mWDN with adaptive parameter adjustment capability to implement the discrete wavelet decomposition process.

(1) Redesign of the Convolutional Operation Matrix. When adopting the regular convolutional operation matrix (similar to Figure 5), the convolutional calculation at the boundary of the window involves data on the other side of the sliding window, causing the calculation results to be distorted. Therefore, we alleviate the impact of the boundary problem by shifting the wavelet parameters in the convolutional operation matrix so that the calculation results at the boundary near the prediction point are not distorted as much as possible. The redesigned matrix is shown in Figure 8. The redesigned matrix will be applied in mWDN.

The mWDN approximately implements an MDWD under a deep neural network framework. This neural network framework mainly consists of a perceptron model and an average pooling layer. mWDN implements the convolution operation in the MDWD by replacing the weight parameter matrix in the perceptron model using the wavelet function

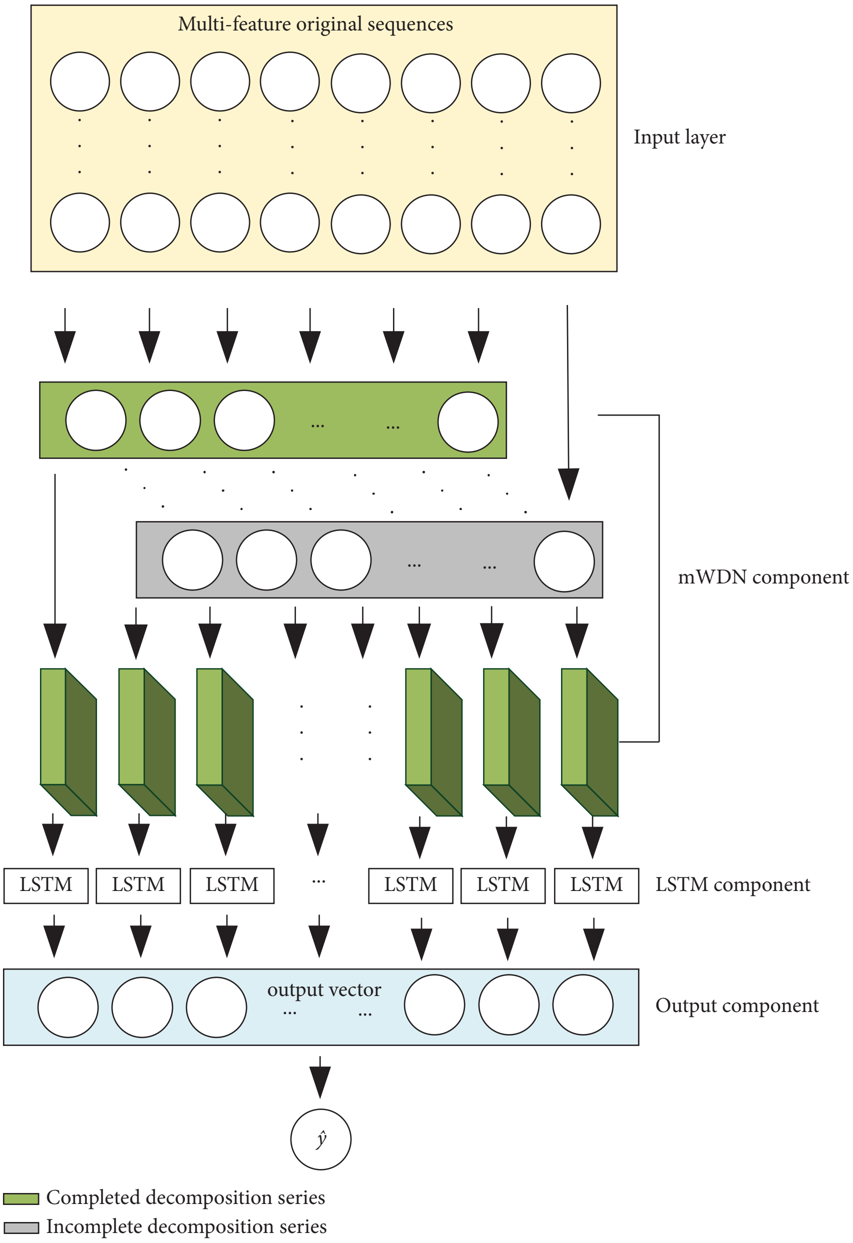

Figure 6: mWDN-LSTM structure diagram.

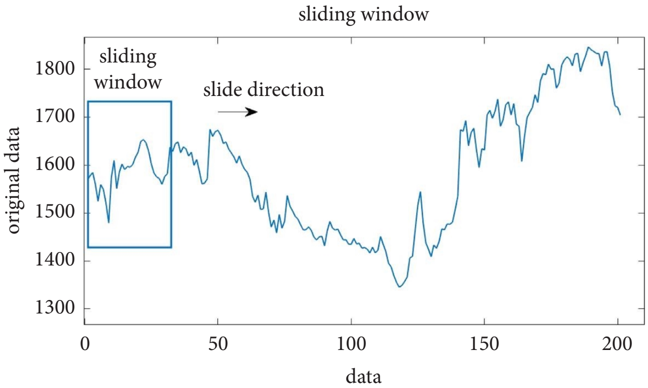

Figure 7: Diagram of the sliding window.

$$W^l(i) = \begin{bmatrix} l_K & \varepsilon & \cdots & \varepsilon & \varepsilon & \cdots & \varepsilon & \varepsilon & \varepsilon \\ l_{K-1} & l_K & \cdots & \varepsilon & \varepsilon & \cdots & \varepsilon & \varepsilon & \varepsilon \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_1 & l_2 & \cdots & l_K & \varepsilon & \cdots & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & l_1 & \cdots & l_{K-1} & l_K & \cdots & \varepsilon & \varepsilon & \varepsilon \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \varepsilon & \varepsilon & \ldots & l_3 & l_4 & \ldots & l_K & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \cdots & l_2 & l_3 & \cdots & l_{K-1} & l_K & \varepsilon \\ \varepsilon & \varepsilon & \cdots & l_1 & l_2 & \cdots & l_{K-2} & l_{K-1} & l_K \end{bmatrix}, \quad W^h(i) = \begin{bmatrix} h_K & \varepsilon & \cdots & \varepsilon & \varepsilon & \cdots & \varepsilon & \varepsilon & \varepsilon \\ h_{K-1} & h_K & \cdots & \varepsilon & \varepsilon & \cdots & \varepsilon & \varepsilon & \varepsilon \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_1 & h_2 & \cdots & h_K & \varepsilon & \cdots & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & h_1 & \cdots & h_{K-1} & h_K & \cdots & \varepsilon & \varepsilon & \varepsilon \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \varepsilon & \varepsilon & \ldots & h_3 & h_4 & \ldots & h_K & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \cdots & h_2 & h_3 & \cdots & h_{K-1} & h_K & \varepsilon \\ \varepsilon & \varepsilon & \cdots & h_1 & h_2 & \cdots & h_{K-2} & h_{K-1} & h_K \end{bmatrix}$$

FIGURE 8: Redesigned convolutional operation matrix, where $\mathbf{W}^l(i)$ and $\mathbf{W}^h(i) \in \mathbb{R}^{P \times P}$, P is the size of input series at the i-th level decomposition, ε are random values that satisfy $|\varepsilon| \ll |l|, \forall l \in l$, and $|\varepsilon| \ll |h|, \forall h \in h$.

matrix of the convolution operation. This makes mWDN different from MDWD with constant parameters and has the ability to fine-tune parameters such as convolutional operation matrix and bias vector to fit different learning tasks. Then, the downsampling process in MDWD is implemented by the average pooling layer. We hope to alleviate the impact of the wavelet decomposition boundary problem on the prediction results by optimizing the prediction effect based on the capability of mWDN to fine-tune the convolution calculation matrix and deviation vector. The schematic diagram for mWDN to implement the i-th level decomposition of the MDWD process is shown in Figure 9. The steps of the process are as follows:

(1) We set up weight matrices $\mathbf{W}^l(i)$ and $\mathbf{W}^h(i)$ according to the parameters of low-pass filter $\mathbf{l}$ and high-pass filter $\mathbf{h}$. The values of the low-pass filter and high-pass filter depend on the selected wavelet function. We initialize the bias vectors $\mathbf{b}^{\mathbf{l}}(i)$ and $\mathbf{b}^{\mathbf{h}}(i)$ as close-to-zero random values. We set the initial value of the weight matrices $\mathbf{W}^l(i)$ and $\mathbf{W}^h(i)$ at the i-th level decomposition as shown in Figure 8.

(2) We then multiply the weight matrix with the input series to implement the convolution operation described in Section 3.2.1 (1).

(3) The result of the previous step is then added to the bias vectors $\mathbf{b}^{\mathbf{l}}(i)$ and $\mathbf{b}^{\mathbf{h}}(i)$, and the addition result is input into the activation function to obtain the intermediate variable series $\mathbf{a}^l(i)$ and $\mathbf{a}^h(i)$.

The calculation process of step (2) and step (3) is shown in the following equation:

$$\begin{aligned} a^l(i) &= \sigma\left(W^l(i)x^l(i-1) + b^l(i)\right), \\ a^h(i) &= \sigma\left(W^h(i)x^l(i-1) + b^h(i)\right), \\ \mathbf{a}^l(i) &= \sigma\left(\mathbf{W}^l(i)\mathbf{x}^l(i-1) + \mathbf{b}^l(i)\right), \\ \mathbf{a}^h(i) &= \sigma\left(\mathbf{W}^h(i)\mathbf{x}^l(i-1) + \mathbf{b}^h(i)\right). \end{aligned} \tag{2}$$

(4) The intermediate variables $\mathbf{a}^l(i)$ and $\mathbf{a}^h(i)$ are downsampled using the average pooling layer as $x_j^l(i) = (a_{2j}^l(i) + a_{2j-1}^l(i))/2, x_j^h(i) = (a_{2j}^h(i) + a_{2j-1}^h(i))/2, j \in [0, P/2]$.

3.3.3. LSTM Component. In this component, we employ LSTM to model time-series data. LSTM is a special kind of recurrent neural network (RNN) proposed by Hochreiter and Schmidhuber [33]. Although RNN models can store history information by hiding states and effectively utilize history data information for prediction. However, RNNs can only learn short-term dependencies between features. The model has a problem of gradient explosion and gradient disappearance. LSTM is improved for the abovementioned problems by adding three gate structures and a memory cell on the basis of RNN. The three gates are the input gate, forget gate, and output gate. The role of these gate structures is to control the flow of information in the hidden state, learning long-term and short-term dependencies, which work quite well on time-series datasets.

3.4. mWDN-LSTM Training and Prediction Process. In order to train the mWDN-LSTM network, the training set needs to be standardized first. After standardization, the data are imported into the mWDN component for calculation based on the sliding window. The input data are decomposed into subseries of different frequencies after passing through the high-pass and low-pass filters. During decomposition, cyclic patterns are generated from the data. Then, the data are imported into the LSTM component for calculation. The input of each LSTM subnetwork is the output of the mWDN component. The LSTM component is calculated to obtain an output vector. The output vector is fed into a fully connected neural network to obtain the final prediction. After completing one model's calculation, the error function is utilized to calculate the error between the predicted value and the real value. Finally, the network is trained by propagating the calculated error values back to the network and using the optimizer to update the weights and biases of the network.

After the training is completed, the model is saved. Similarly, the test set first needs to be standardized. After standardization, the test set is imported into the saved model to calculate and obtain the predicted values. Since the obtained predicted values are standardized, standard restoration of the predicted values is required. Finally, the evaluation criteria are calculated based on the predicted and real values, and the predicted values and evaluation criteria are given as output.

The process of mWDN-LSTM training and prediction is shown in Figure 10.

4. Experiment

To demonstrate the effectiveness of mWDN-LSTM, we compare the model with MLP, CNN, RNN, LSTM, and

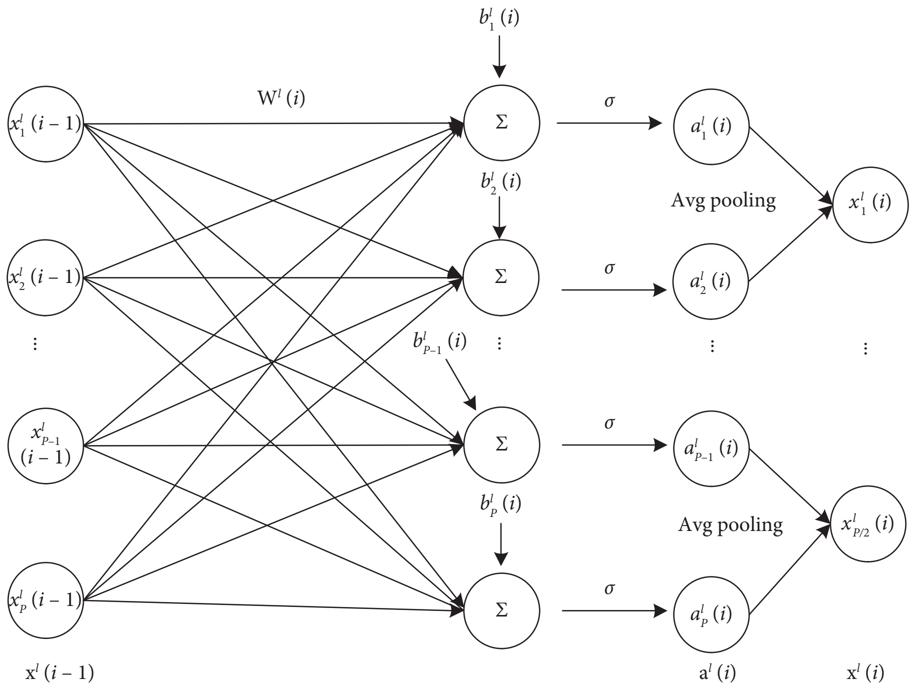

Figure 9: Schematic diagram of the mWDN implementation of MDWD process.

CNN- LSTM under the same environmental setup. In addition, we also show an experimental result of a model with data leakage (DWT-LSTM).

4.1. Dataset. The dataset settings for the experiment are as follows:

(1) Experimental subject: The experimental subject is SSE (Shanghai Stock Exchange) Composite Index (000001)

(2) Date range and data source: This includes daily transaction data for 7,127 trading days from July 1, 1991, to August 31, 2020, obtained from the wind database.

(3) Features included in each piece of data: The features include the opening price, highest price, lowest price, closing price, volume, turnover, ups and downs, and change. A sample of the data is shown in Table 2.

The dataset features can be described as follows:

(i) Opening price is the first price of any listed stock at the beginning of an exchange on a trading day.

(ii) High and low prices are the highest and lowest prices of the stock on that day. Generally, these data are applied by traders to measure the volatility of a stock.

(iii) Closing price is the price of the stock at the end of a trading day.

(iv) Volume is the total number of shares or contracts traded in the market during the day.

(v) Turnover is the total value of stocks or contracts traded in the market on that day.

(vi) Ups and downs are the values of the increase or decrease of the day's closing price relative to the previous day's closing price.

(vii) Change is the ratio of the increase or decrease of the day's closing price relative to the previous day's closing price.

(4) Prediction target: Prediction target is the closing price of the next day.

(5) Train and test set splits: We take the data of the first 6,627 trading days as the training set and the data of the last 500 trading days as the test set.

4.2. Experimental Setup. The data are standardized and restored by the z-score method. The standardization of data by using $ and the restoration of data are performed by using the following equations:

$$y_i = \frac{x_i - \overline{x}}{s}, \tag{3}$$

$$x_i = y_i * s + \overline{x}, \tag{4}$$

where x_i is the input data, x is the average of the input data, s is the standard deviation of the input data, and y_i is the standardized value.

For evaluation criteria, the mean absolute error (MAE), root mean square error (RMSE), and R-squared (R^2) are applied to evaluate the effectiveness. The MAE, RMSE, and R^2 calculation formulas are as follows:

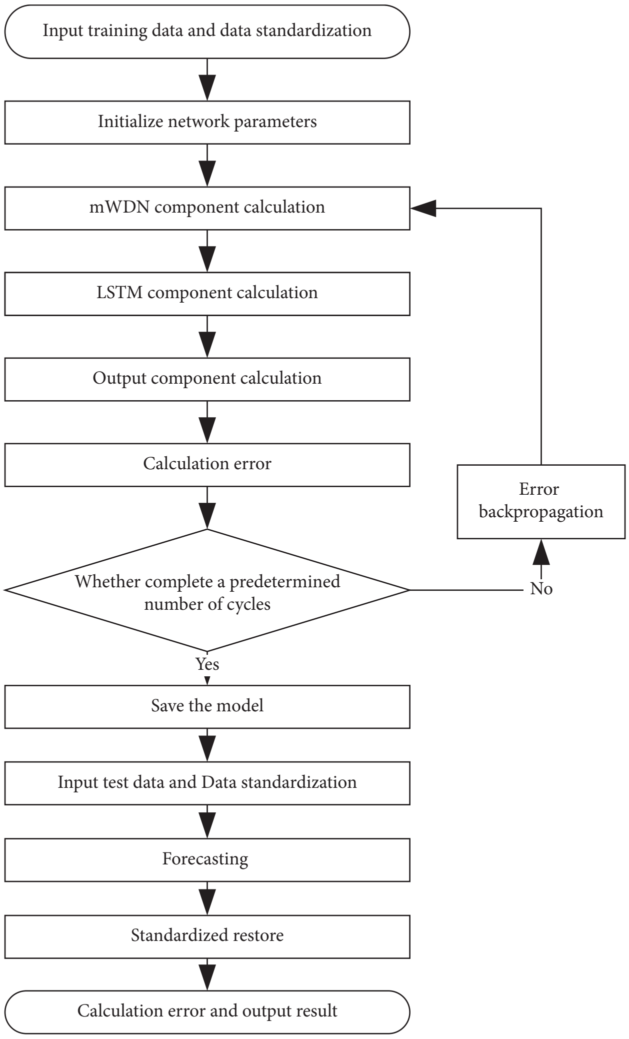
Input training data and data standardization
Initialize network parameters
mWDN component calculation
LSTM component calculation
Output component calculation
Calculation error
Error backpropagation
Whether complete a predetermined number of cycles
No
Yes
Save the model
Input test data and Data standardization
Forecasting
Standardized restore
Calculation error and output result

Figure 10: Flowchart of mWDN-LSTM training and prediction.

$$\begin{aligned}
\mathrm{MAE} &= \frac{1}{n}\sum_{i=1}^{n}\left|\widehat{y}_i - y_i\right|,\\
\mathrm{RMSE} &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\widehat{y}_i - y_i\right)^2},\\
R^2 &= 1 - \frac{\mathrm{SS}_{\mathrm{Regression}}}{\mathrm{SS}_{\mathrm{Total}}}\\
&= 1 - \frac{\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \overline{y_i}\right)^2}, \quad R^2 \in (0,\, 1),
\end{aligned} \tag{5}$$

where $\widehat{y}_i$ is the predictive value, y_i is the real value, and $\overline{y_i}$ is the average value. The closer the MAE and RMSE values are to zero, the smaller the difference between the predicted and real values is, and the higher the prediction accuracy is. The closer the R^2 is to 1, the better the fitting degree of the model is.

4.3. Implementation of mWDN-LSTM. The parameter settings of our proposed model mWDN-LSTM are tuned one by one according to cross-validation. The parameters of this experiment are shown in Table 3.

The Fejer–Korovkin 4 wavelet function is a commonly adopted wavelet function with optimal asymptotic frequency localization [34]. So, we utilize the Fejer–Korovkin 4 as a wavelet function in our experiments, and the filter coefficients are set as

$$\begin{aligned}
\mathbf{l} &= \{0.3492,\, 0.7826,\, 0.4752, -0.0996,\\
&\quad -0.1599,\, 0.0431,\, 0.0425, -0.0190\},\\
\mathbf{h} &= \{-0.0190, -0.0425,\, 0.04310,\, 0.1599,\\
&\quad -0.0996, -0.4752,\, 0.7826, -0.3492\}.
\end{aligned} \tag{6}$$

According to the parameter settings of the mWDN-LSTM network, the data dimensions of input and output in each component of mWDN-LSTM are shown in Figure 11. The model structure is as follows: according to the size of the time_step and the dimension of the input data, the data of the input layer are a three-dimensional vector (none, 32, and 8). After the data are input into the mWDN component, the data are decomposed into subseries of different frequencies. The cyclic patterns in the data are generated. After 2-level decomposition, the data of length 32 will be decomposed into one subseries of length 16 and two subseries of length 8, for a total of three subseries. Therefore, the output of the mWDN component is two four-dimensional vectors: (none, 16, 1, 8) and (none, 8, 2, 8). Each subseries feeds an LSTM subnetwork. After the LSTM component is trained, an output vector (none and 48) will be output, where 48 is the number of hidden units in the LSTM component. Finally, the vector is fed into the output component to get the final predicted value.

5. Experiment Results

In this section, we will discuss our model's effectiveness compared with other benchmarks. With regard to benchmarks, to the best of our knowledge, there is no research that utilizes the cyclic pattern correctly in the stock price prediction task. So, we choose MLP, CNN, RNN, LSTM, and CNN-LSTM models as benchmarks. DWT-LSTM is used as a case study to describe the results of hybrid models in the existence of data leakage.

Our experiments exploit the training set data to train mWDN-LSTM, MLP, CNN, RNN, LSTM, and CNN-LSTM, respectively, and then exploit the test set data to generate predictions. Based on the experimental results, we plotted the comparison figure of predicted and real values (Figures 12–19), as well as the table of evaluation criteria (Table 4) and the comparison chart of evaluation criteria performance (Figures 20 and 21).

The results of DWT-LSTM, as shown in Figure 22 and Table 4, usually cause researchers to overestimate the performance of signal decomposition techniques such as wavelet decomposition, but similar hybrid models with data

Table 2: Head data in the SSE experimental data.

Date	Opening price	Highest price	Lowest price	Closing price	Volume (share)	Turnover (RMB)	Ups and downs	Change (%)
1991-7-1	136.64	138.62	136.56	136.85	2294000	12469884	−0.71	−0.5161
1991-7-2	135.91	135.96	135.69	135.96	283800	3794100	−0.89	−0.6503
1991-7-3	135.28	135.96	134.98	135.27	271500	1818504	−0.69	−0.5075
1991-7-4	136.63	136.63	134.19	136.63	1339400	8095138	1.36	1.0054
1991-7-5	136.01	137.68	135.9	135.96	1454000	9394861	−0.67	−0.4904
1991-7-8	135.26	135.28	134.93	135.28	587400	2925933	−0.68	−0.5001
1991-7-9	136.56	136.57	134.31	134.64	844200	4174836	−0.64	−0.4731
1991-7-10	134.4	135.6	133.72	133.99	602300	2894591	−0.65	−0.4828

Table 3: Parameter setting of mWDN-LSTM.

Parameters	Value
mWDN component decomposition level	2
mWDN component wavelet function	Fejer–Korovkin 4
mWDN component activation function	Sigmoid
Number of hidden units in LSTM component	48
LSTM component activation function	tanh
Time_step	32
Batch_size	64
Learning rate	0.001
Optimizer	Adam
Loss function	MAE
Epochs	150

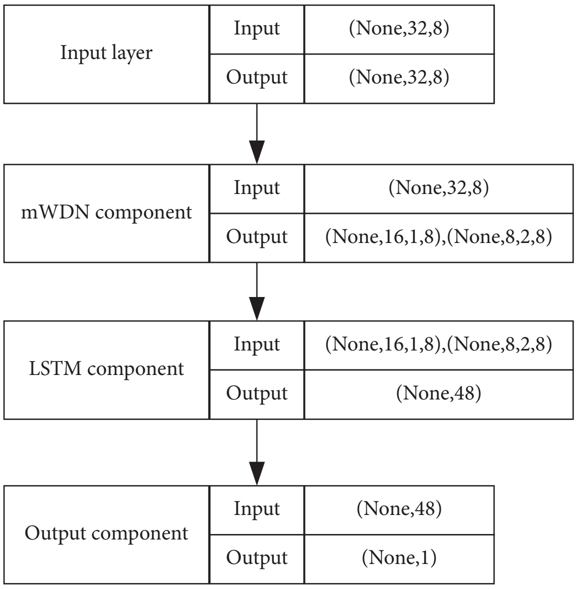

Figure 11: Data dimensions of input and output in each component of mWDN-LSTM.

leakage are unreliable in application scenarios. This is one of the motivations of our paper.

Furthermore, in order to clearly display and analyze the intersection of the constructed mWDN-LSTM stock index prediction model and the progress of the cutting-edge benchmark CNN-LSTM stock index prediction model, two time periods were selected from the test set results for enlarged display and comparison. If the first point of the test set is marked as t_1, the second point is marked as t_2, and so on, then the two time periods are "t301 to t400" and "t401 to t500". The result is as follows.

5.1. Results Demonstration. The comparison figure between predicted and real values can visually demonstrate the error between predicted and real values at the turning point and trend duration stage, as well as the degree of model fitting. From Figures 12–19, we can notice that mWDN-LSTM has the lowest error between the predicted and real values at the turning point and the trend duration phase stage compared to other models. So, the predicted value series of mWDN-LSTM has the highest degree of fitting with the real value series. Based on the diagram at the turning point and the trend duration phase, the descent order with regard to the fitting degree of all models is mWDN-LSTM, CNN-LSTM, LSTM, RNN, CNN, and MLP.

From the diagrams mentioned above, we can find that most models predict badly especially around the turning point, and our model mWDN-LSTM alleviates this problem by being guided by cyclic pattern information.

5.2. Result Analysis. The diagrams mentioned above demonstrate the prediction results visually. In this section, we calculated the evaluation criteria (MAE, RMSE, and R^2) based on the experiments carried out on the various models under the same experimental setup, so that we can more accurately evaluate the prediction error and model fitting degree. From the results presented in Table 4 and Figures 20 and 21, we can reach 3 major conclusions.

First, LSTM-based models outperform non-LSTM-based models, and this conclusion means that, generally, LSTM-based models are more suitable for time-series prediction tasks.

Among the non-LSTM-based models (CNN, RNN, and MLP), CNN and RNN have close prediction results with little differences between them, but they are significantly better than MLP. For example, compared to MLP, the MAE of CNN decreases from 37.757 to 30.397 by 19.5%, RMSE decreases from 49.371 to 41.492 by 16%, and R^2 improves by 1.59%. Therefore, the CNN and RNN models outperform the MLP.

Among the LSTM-based models (mWDN-LSTM, CNN-LSTM, and LSTM), LSTM performs worst but still, significantly improves the prediction results compared to CNN and RNN. For example, compared to CNN, the MAE of

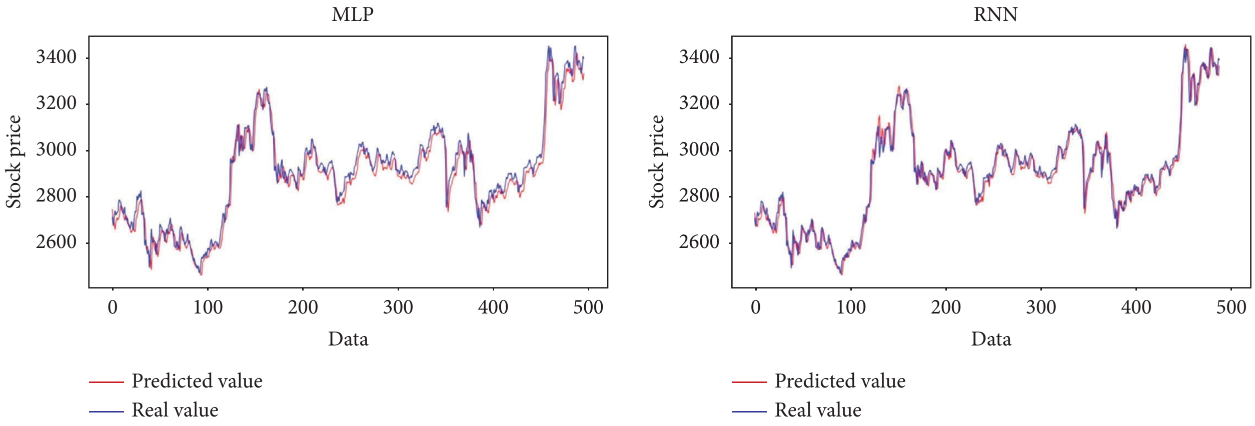

Figure 12: Comparison of the predicted value and the real value of the MLP and RNN models.

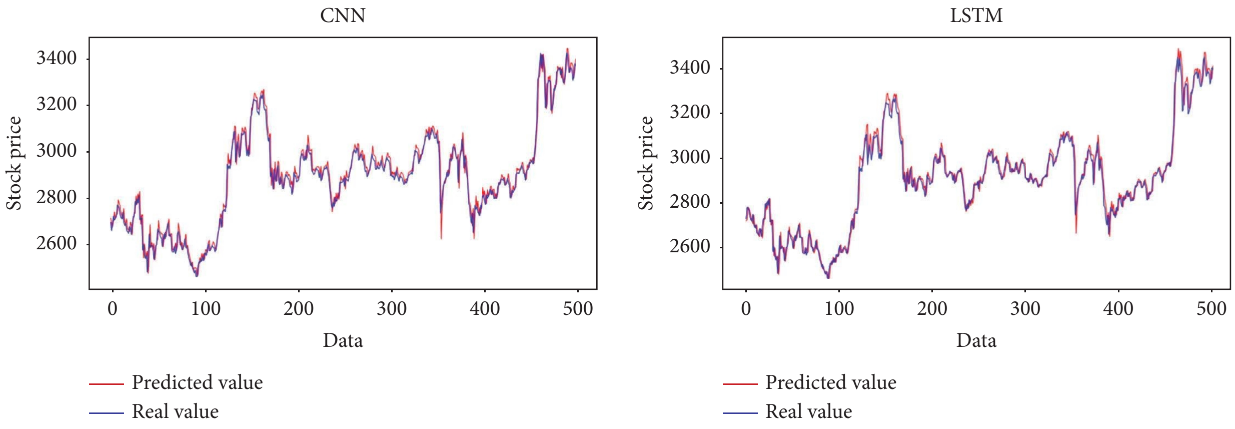

Figure 13: Comparison of the predicted value and the real value of the CNN and LSTM models.

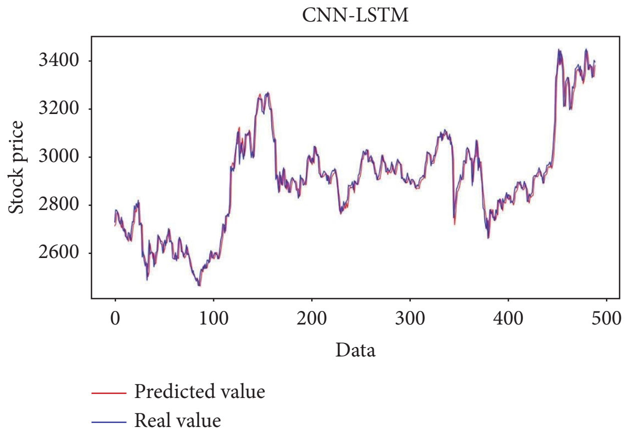

Figure 14: Comparison of the predicted value of the CNN-LSTM and the real value.

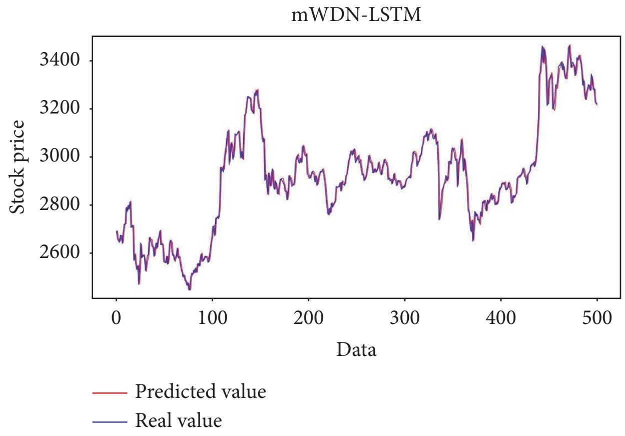

Figure 15: Comparison of the predicted value of the mWDN-LSTM and the real value.

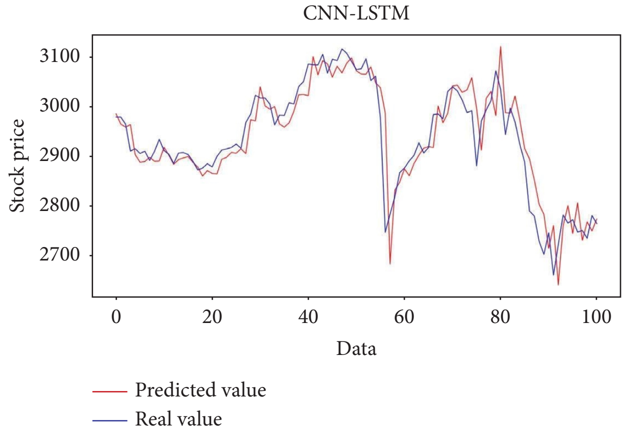

Figure 16: CNN-LSTM prediction result chart from t_{301} to t_{400}.

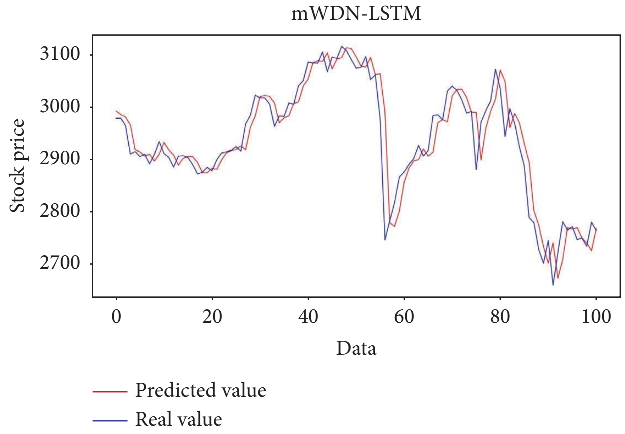

Figure 17: mWDN-LSTM prediction result chart from t_{301} to t_{400}.

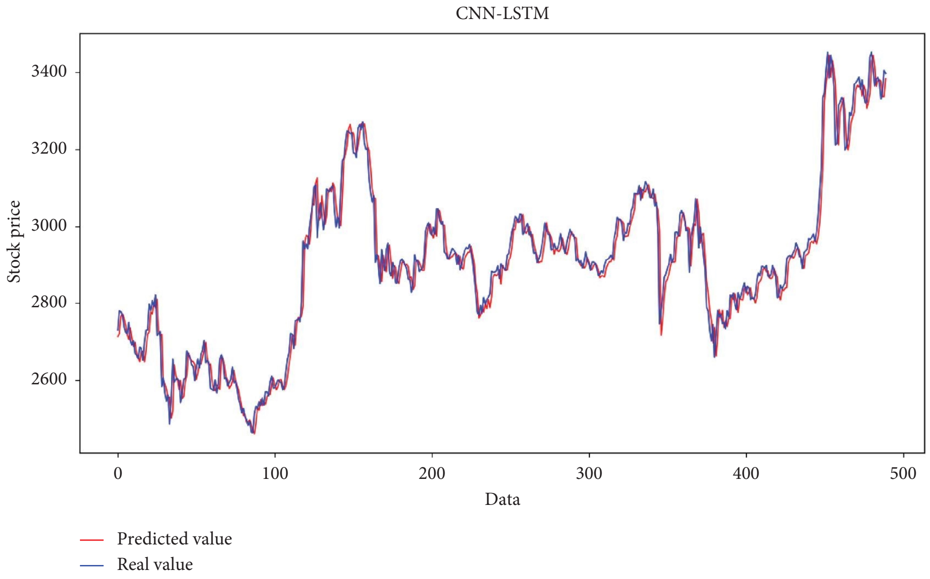

Figure 18: CNN-LSTM prediction result chart from t_{401} to t_{500}.

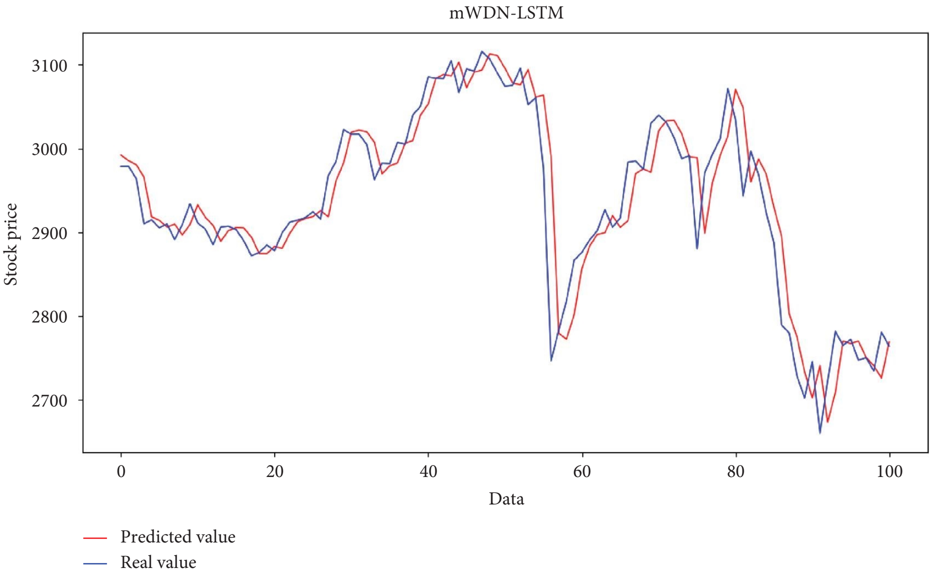

Figure 19: mWDN-LSTM prediction result chart from t_{401} to t_{500}.

LSTM decreased from 30.397 to 28.675 by 5.7%, RMSE decreased from 41.492 to 40.793 by 1.7%, and R^2 also improved by 0.13%.

Second, the hybrid models outperformed the nonhybrid model. This conclusion demonstrates that hybrid models designed for a specific task generally outperform general-purpose models.

Among all the hybrid models (CNN-LSTM and mWDN-LSTM), the CNN-LSTM model performs the worst. Among all the nonhybrid models (LSTM, RNN, CNN, and MLP), LSTM performs the best. For example, compared with LSTM, the MAE of CNN-LSTM decreased from 28.675 to 27.559 by 3.9%, and the RMSE decreased from 40.793 to 39.522 by 3.1%, and the R^2 also improved by 0.23%.

Finally, of all hybrid models, our model mWDN-LSTM performed the best. This demonstrates that correctly utilizing the cyclic patterns in a hybrid model can improve the prediction results.

We compare mWDN-LSTM with CNN-LSTM, which already achieve excellent prediction results among the benchmarks. Compared with CNN-LSTM, the MAE of the mWDN-LSTM model decreased by 4.8%, and the RMSE decreased by 3.1%, and the R^2 also improved by 0.48%.

Table 4: Comparison table of evaluation criteria.

Models	MAE	RMSE	R^2
mWDN-LSTM	26.244	38.290	0.9703
CNN-LSTM	27.559	39.522	0.9655
LSTM	28.675	40.793	0.9632
RNN	29.914	41.849	0.9612
CNN	30.397	41.492	0.9619
MLP	37.757	49.371	0.9460
DWT-LSTM	9.391	12.775	0.9964

Note. The DWT-LSTM model achieves high precision with data leakage, which is impractical in real applications.

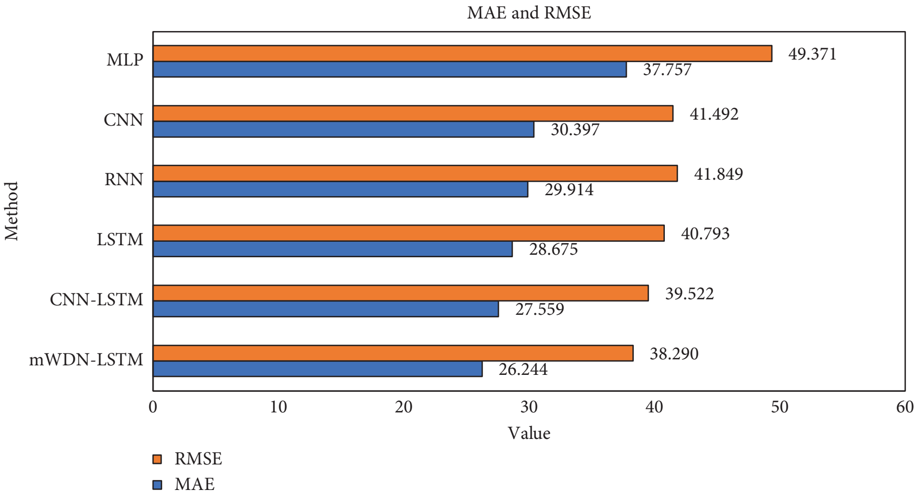

Figure 20: MAE and RMSE performance comparison chart.

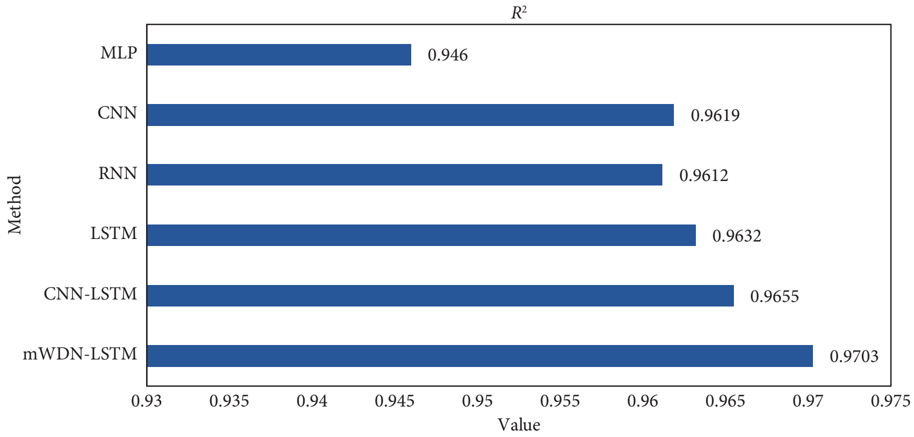

Figure 21: R^2 performance comparison chart.

5.3. Experimental Validation on Another Dataset. Furthermore, in order to validate our model, we conducted additional experiments and analysis on the Hang Seng Index (HSI) dataset in addition to the abovementioned SSE dataset. The HSI dataset has the same time frame as the SSE dataset, and the experimental setup such as dataset settings, model parameters, and experimental steps are also the same as the abovementioned experiments on the SSE dataset.

Based on the experimental results, the comparison figures of predicted values and real values (Figures 23 and 24) are plotted, as well as the table of evaluation criteria (Table 5) and the comparison charts of evaluation criteria performance (Figures 25 and 26).

From Figures 23–26 and Table 5, in experimental validation of the HSI dataset, we can see that the evaluation criteria, MAE and RMSE, of the mWDN-LSTM model are

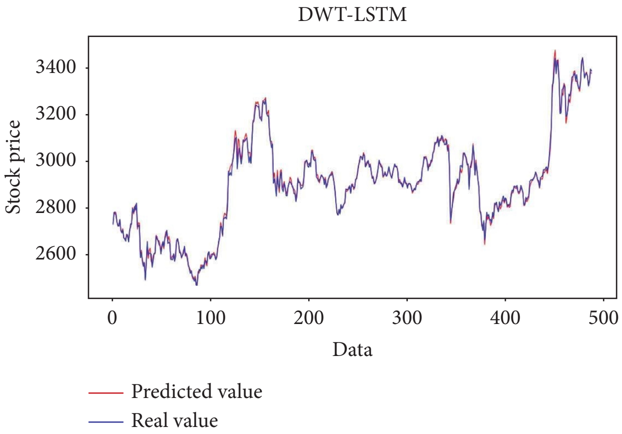

Figure 22: Experimental results with data leakage.

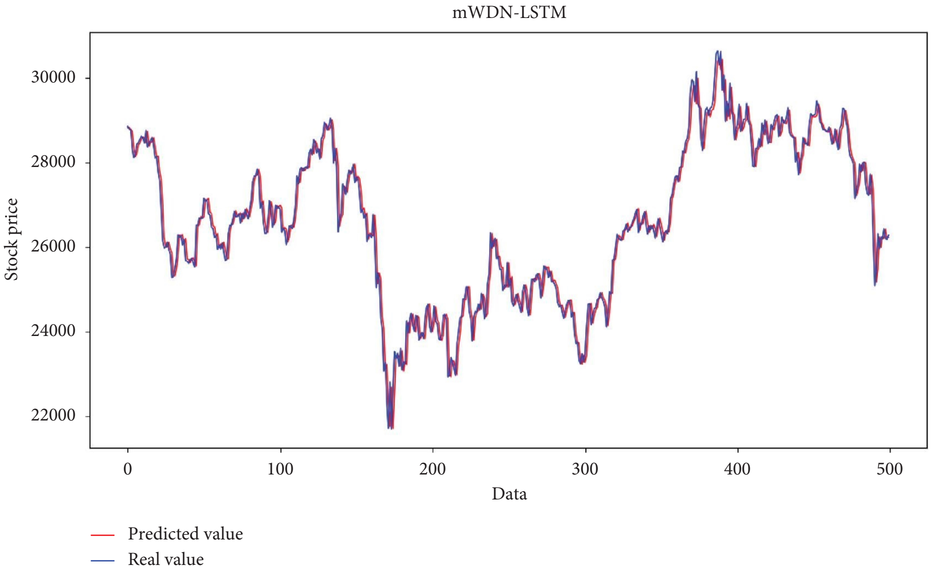

Figure 23: The comparison of the predicted value of the mWDN-LSTM and the real value (in HSI dataset's experimental validation).

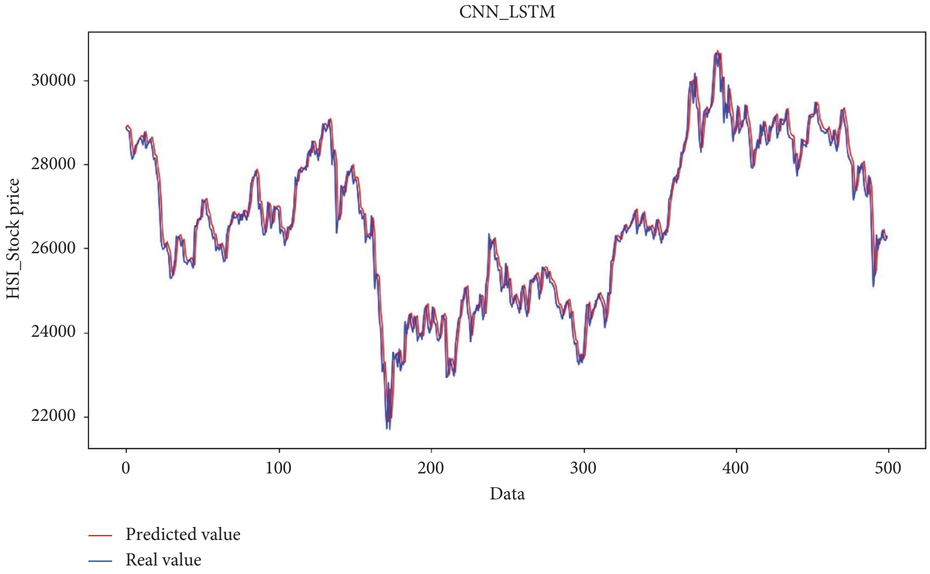

Figure 24: Comparison of the predicted value of the CNN-LSTM and the real value (in HSI dataset's experimental validation).

Table 5: Performance comparison table in HSI dataset.

Models	MAE	RMSE	R^2
mWDN-LSTM	256.488	353.775	0.9631
CNN-LSTM	263.764	363.186	0.9615
LSTM	273.402	367.061	0.9607
RNN	289.819	375.168	0.9589
CNN	290.474	376.101	0.9604
MLP	319.0930	416.251	0.9494

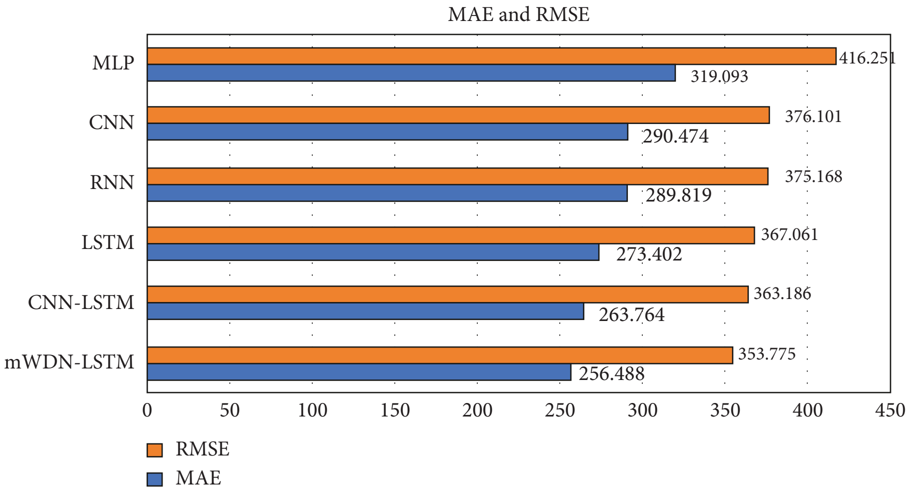

Figure 25: MAE and RMSE performance comparison chart (in HSI dataset's experimental validation).

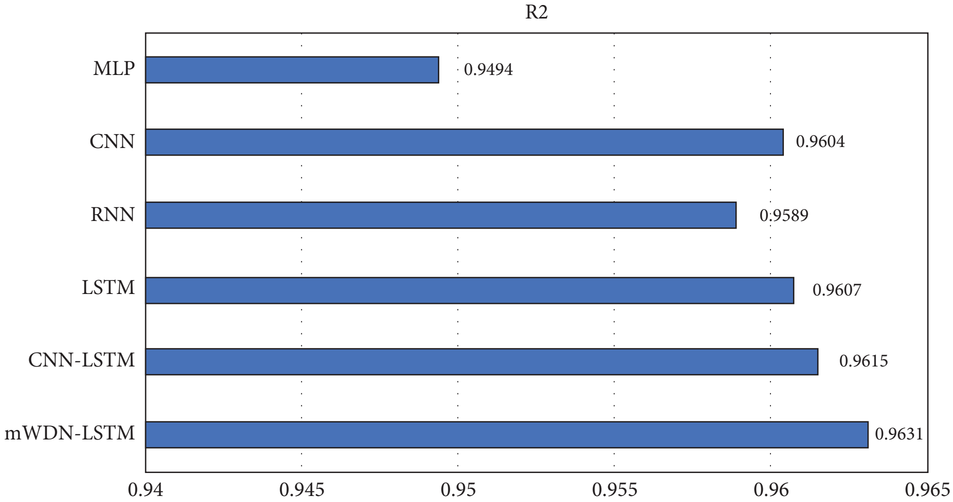

Figure 26: R^2 performance comparison chart in the HSI dataset (in HSI dataset's experimental validation).

the best, and R^2 is closest to 1; the mWDN-LSTM model also obtains excellent prediction results, and it has the highest degree of fitting compared to other benchmark models. It can be concluded that mWDN-LSTM has generalizability.

5.3.1. Summary. Our proposed mWDN-LSTM has outperformed all the other baseline models and is more effective for predicting the next day's closing price of stocks. Meanwhile, our experiments demonstrate the effectiveness of utilizing cyclic patterns while avoiding data leakage and alleviating the impact of boundary problems.

6. Conclusions

In this paper, we study the problem of stock price prediction which aims to predict the next day closing price of the stock using historical information. We have noticed that cyclic patterns are important characteristics of the stock market. From this motivation, we propose the mWDN-LSTM model based on deep neural networks, which can effectively and correctly utilize the cyclic patterns in the stock market. Unlike other DWT-based hybrid models, our mWDN-LSTM model avoids the data leakage by sliding window mechanism, and through the adaptive parameter adjustment mechanism of mWDN and redesign of convolution matrix, the impact of boundary problem in wavelet decomposition on prediction performance is alleviated. Therefore, the model is both theoretically sound and practically feasible in stock price time-series prediction.

In addition, the model generates cyclic patterns with different frequencies from stock data by applying the mWDN network and then employing the LSTM model to learn the cyclic patterns and predict the next day's closing price. We compare mWDN-LSTM with baseline models to verify its effectiveness on the datasets of the SSE Composite Index and the Hang Seng Index. The experimental results show that the evaluation criteria, MAE and RMSE, of our model are the best, and R^2 is closest to 1. This means that our model mWDN-LSTM outperforms the benchmarks and demonstrates the effectiveness of utilizing cyclic patterns in stock price prediction tasks when avoiding data leakage and alleviating the impact of boundary problems.

Data Availability

The Shanghai Composite Index data used to support the findings of this study are collected publicly and are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research work was funded by the Shenzhen Technology University.

References

[1] R. Vanaga, "Financial and capital market commission financing: aspects and challenges," *Journal of Logistics, Informatics and Service*, vol. 7, no. 1, pp. 17–30, 2020.

[2] D. Shah, H. Isah, and F. Zulkernine, "Stock market analysis: a review and taxonomy of prediction techniques," *International Journal of Financial Studies*, vol. 7, no. 2, p. 26, 2019.

[3] B. Billah, L. Maxwell, and R. D. Snyder, "Exponential smoothing model selection for forecasting," *International journal of forecasting Koehler*, vol. 22, no. 2, pp. 239–286, 2006.

[4] C. Jung and R. Boyd, "Forecasting UK stock prices," *Applied Financial Economics*, vol. 6, no. 3, pp. 279–286, 1996.

[5] A. A. Adebiyi, O. Aderemi, and C. K. Adewumi, "Stock price prediction using the arima model," in *Proceedings of the Uksim-Amss 16th International Conference on Computer Modelling and Simulation*, Cambridge, UK, March 2014.

[6] C. Zhang, X. Cheng, and M. Wang, "An empirical research in the stock market of Shanghai by GARCH model," *Operations Research and Management Science*, vol. 4, pp. 144–146, 2005.

[7] C. Shi and X. Zhuang, "A study concerning soft computing approaches for stock price fore- casting," *Axioms*, vol. 8, no. 4, p. 116, 2019.

[8] B. M. Henrique, V. A. Sobreiro, H. Kimura, and Kimura, "Literature review: machine learning techniques applied to financial market prediction," *Expert Systems with Applications*, vol. 124, pp. 226–251, 2019.

[9] G. Zhang, B. Eddy Patuwo, and M. Y Hu, "Forecasting with artificial neural networks," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[10] M. E. Wu, J. H. Syu, J. C. W. Lin, and J. M. Ho, "Effective fuzzy system for qualifying the characteristics of stocks by random trading," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 8, pp. 3152–3165, 2022.

[11] J.-H. Syu, J. C. W. Lin, C. J. Wu, and J. M. Ho, "Stock selection system through suitability index and fuzzy-based quantitative characteris tics," *IEEE Transactions on Fuzzy Systems*, vol. 31, no. 1, pp. 322–334, 2023.

[12] R. A. Yaffee and M. Mcgee, *An Introduction to Time Series Analysis and Forecasting*, Elsevier, Amsterdam, Netherlands, 2000.

[13] Z. Cui, W. Chen, and Y. Chen, "Multi-scale convolutional neural networks for time series classification," *Neural Networks*, vol. 136, 2016.

[14] R. Gao, L. Du, K. F. Yuen, and P. N. Suganthan, "Walk-forward empirical wavelet random vector functional link for time series forecasting," *Applied Soft Computing*, vol. 108, Article ID 107450, 2021.

[15] G. Duan, "Deep neural networks for stock price prediction," in *Proceedings of the 14th International Conference on Computer Research and Development (ICCRD)*, Shenzhen, China, January 2022.

[16] M. Naeini, H. Pakdaman, Taremian, H. Baradaran, and Hashemi, "Stock market value prediction using neural networks," in *Proceedings of the 2010 international conference on computer information systems and industrial management applications (CISIM)*, Krakow, Poland, October 2010.

[17] J. L. Ticknor, "A Bayesian regularized artificial neural network for stock market forecasting," *Expert Systems with Applications*, vol. 40, pp. 5501–5506, 2013.

[18] A. M. Rather, A. Agarwal, and V. Sastry, "Recurrent neural network and a hybrid model for prediction of stock returns," *Expert Systems with Applications*, vol. 42, no. 6, pp. 3234–3241, 2015.

[19] L. Di Persio and O. Honchar, "Artificial neural networks architectures for stock price prediction: comparisons and applications," *International journal of circuits, systems and signal processing*, vol. 10, pp. 403–413, 2016.

[20] S. Selvin, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, Udupi, India, September 2017.

[21] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, “A CNN-LSTM-based model to forecast stock prices,” *Complexity*, vol. 2020, Article ID 6622927, 10 pages, 2020.

[22] J. Wu and Ming-Tai, “A graph-based CNN-LSTM stock price prediction algorithm with leading indicators,” *Multimedia Systems*, vol. 29, pp. 1–20, 2021.

[23] J. M. Wu, Z. Li, G. Srivastava, M. Tasi, and J. C. Lin, “A graph-based convolutional neural network stock price prediction with leading indicators,” *Software: Practice and Experience*, vol. 51, no. 3, pp. 628–644, 2021.

[24] J. M. T. Wu, L. Sun, G. Srivastava, and J. C. W. Lin, “A novel synergetic lstm-ga stock trading suggestion system in internet of things,” *Mobile Information Systems*, vol. 2021, Article ID 6706345, 15 pages, 2021.

[25] Q. Zhang, C. Qin, Y. Zhang, F. Bao, C. Zhang, and P. Liu, “Transformer-based attention network for stock movement prediction,” *Expert Systems with Applications*, vol. 202, Article ID 117239, 2022.

[26] X. Qiu, P. N. Suganthan, G. A. J. Amaratunga, and Amaratunga, “Fusion of multiple indi- cators with ensemble incremental learning techniques for stock price forecasting,” *Journal of Banking and Financial Technology*, vol. 3, no. 1, pp. 33–42, 2019.

[27] S. K. Chandar, “Fusion model of wavelet transform and adaptive neuro fuzzy inference system for stock market prediction,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 2019, pp. 1–9, 2019.

[28] X. Li and P. Tang, “Stock index prediction based on wavelet transform and FCD-MLGRU,” *Journal of Forecasting*, vol. 39, no. 8, pp. 1229–1237, 2020.

[29] D. Wu, X. Wang, and S. Wu, “A hybrid method based on extreme learning machine and wavelet transform denoising for stock prediction,” *Entropy*, vol. 23, no. 4, p. 440, 2021.

[30] R. Hasumi and Y. Kajita, “Boundary problem and data leakage: a caveat for wavelet-based forecasting,” 2018, https://www.jcer.or.jp/english/boundary-problem-and-data-leakage-a-caveat-for-wavelet-based-forecasting.

[31] J. Wang, “Multilevel wavelet decomposition network for interpretable time series analysis,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, London, UK, July 2018.

[32] Y. Zhang, “Multivariate and multi-frequency lstm based fine-grained productivity forecasting for indus trial iot,” in *Proceedings of the 2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, Tokyo, Japan, December 2020.

[33] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[34] M. Nielsen, “On the construction and frequency localization of finite orthogonal quadrature filters,” *Journal of Approximation Theory*, vol. 108, no. 1, pp. 36–52, 2001.