IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 March 2024, accepted 28 March 2024, date of publication 3 April 2024, date of current version 11 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3384430

RESEARCH ARTICLE

Accurate Stock Price Forecasting Based on Deep Learning and Hierarchical Frequency Decomposition

YI LI[1], LEI CHEN[1], CUIPING SUN[2], GUOXU LIU[3], CHUNLEI CHEN[3], AND YONGHUI ZHANG[3]

[1]School of Economics and Management, Weifang University, Weifang 261061, China
[2]School of Economics and Management, Weifang University of Science and Technology, Weifang 262700, China
[3]School of Computer Engineering, Weifang University, Weifang 261061, China

Corresponding author: Cuiping Sun (suncuiping@wfust.edu.cn)

This work was supported in part by Shandong Provincial Natural Science Foundation under Grant ZR2021MF085, Grant ZR2021MC044, Grant ZR2022ME155, and Grant ZR2023QC116; in part by Shandong Provincial Key Research Project on Financial Applications under Grant 2021-JRZZ-27; in part by the Doctoral Research Foundation of Weifang University under Grant 2022BS48; and in part by the Doctoral Research Foundation of Weifang University of Science and Technology under Grant 2021RWBS10.

ABSTRACT The stock market is playing an increasingly important role in the global economy. Accurate stock price forecasting not only aids the government in predicting economic trends but also helps investors anticipate higher expected returns. Nevertheless, hurdles such as nonlinearity, complexity and high volatility make it a daunting task to predict stock prices. To address this issue, this study proposes a new hybrid model, termed Hierarchical Decomposition-based Forecasting Model (HDFM), to decompose and forecast stock prices in a hierarchical fashion. The model utilizes complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) for the initial decomposition of stock price time series. To enhance the predictive efficiency, sub-series with similar sample entropy from the decomposition were combined using the K-means clustering method. Through a thorough analysis, it was found that the first combined sub-series contained more high-frequency signals. Therefore, the first combined sub-series is subjected to a second decomposition with variational mode decomposition (VMD). Subsequently, the gated recurrent unit (GRU) model was used to individually predict each sub-series. The final results were obtained by merging the prediction outcomes. The proposed model was evaluated on three different stock markets. The experimental results show that the proposed model outperforms other forecasting methods across all stock indices. Moreover, ablation studies demonstrated the effectiveness of each component within the proposed model.

INDEX TERMS
Stock price prediction, deep learning, hierarchical decomposition, clustering.

I. INTRODUCTION

With the vigorous development of the global economy, the stock market has assumed greater prominence in the global economy and is regarded as a barometer of the economic situation. The stock market not only reflects a country's economic growth, but also provides the basis for a country to formulate the next economic policy. Changes in stock price are closely related to shifts in the national market economy. Accurately forecasting stock prices can not only aid the government in predicting the economic situation and formulating macroeconomic management policies, but also guide enterprises in making financing plans. In addition, it can assist in reducing investment risks and maximizing investment returns, thus ensuring healthy growth in the national economic market. Consequently, stock price forecasting has become a common concern among academia, investors, and governments.

With the growth of economic and financial markets, stock prices are influenced by a growing number of factors,

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

such as changes in the global situation, national macro policies, corporate activities, and investor psychology. As a result, stock prices exhibit characteristics of nonlinearity, nonstationarity, high volatility, and multi-noise, which can make it very difficult to accurately forecast their values.

In recent decades, there have been two main types of stock price forecasting methods: econometric statistical models and machine learning methods. In the early stages, traditional econometric models were widely used for stock price forecasting. These include autoregression (AR), moving average (MA), autoregressive moving average (ARMA) [1], autoregressive integrated moving average (ARIMA) [2], generalized autoregressive conditional heteroskedasticity (GARCH) [3], and linear regression [4]. Although these methods have achieved reasonable predictive performance, the assumption of linearity limits their predictive ability, especially for stock prices with high volatility. Machine learning methods can overcome these deficiencies. Common techniques include support vector regression (SVR) [5], artificial neural networks (ANN) [6], hidden Markov model (HMM) [7], random forests (RF) [8] and so on. With the success of AlexNet [9] in the ImageNet competition, deep learning methods have shown great potential in feature extraction and pattern recognition and are dominant in many areas of prediction. For time series prediction, recurrent neural networks (RNN) [10], long short-term machine (LSTM) [11], and GRU [12] solve the long-term dependency problem and show outstanding performance. However, because of the complicated characteristics of stock prices discussed above, it is still a challenge for the models to predict stock prices accurately and robustly.

To mitigate these challenges, this study proposes a novel hybrid stock price forecasting model. The proposed framework borrows the idea of divide-and-conquer strategy and performs decomposition and forecasting hierarchically. Specifically, the proposed model decomposes the original stock price into multiple sub-series by using CEEMDAN. Furthermore, to improve the forecasting accuracy of high-frequency sub-series, the VMD method is adopted to re-decompose sub-series with high volatility. In addition, to improve forecasting efficiency, a clustering method is adopted to integrate the sub-series with similar sample entropy.

The main contributions of this paper can be summarized as follows:

1) A novel hierarchical decomposition-based deep learning method HDFM is proposed to predict the stock price.
2) We introduce a clustering method to integrate the decomposed sub-series to improve the prediction efficiency.
3) Extensive experiments are conducted on different stock markets. We show that the proposed method achieves better performance compared to other deep learning-based stock price prediction methods.

The remainder of this paper is organized as follows: Section II provides a literature review on traditional statistical models, machine learning models and deep learning models for stock price forecasting. Section III presents the methodology proposed in this study. In Section IV, the empirical studies are reported. Finally, Section V draws conclusions and provides directions for future work.

II. RELATED WORK

Currently, methods for predicting stock prices can be categorized into three types: traditional statistical models, machine learning methods, and deep learning methods. The following section provides a brief review of each method category.

A. TRADITIONAL STATISTICAL MODELS

In the early days, the forecasting of financial time series, such as stock prices, was mainly based on economic statistical models. With increasing contributions from scholars to the models, they are now well developed. Some widely used models include the AR, MA, ARMA [13], and ARIMA [14] models.

Challa et al. [15] employed the ARIMA model to predict daily stock returns for the S&P BSE Sensex and S&P BSE IT time series. The results showed that ARIMA had reasonable forecasting performance, indicating the efficiency of the model in predicting stock data. Saleh et al. [14] used ARIMA, SutteARIMA and Holt-Winters to forecast closing stock price trends in BRIC countries. The results demonstrated that the Sutte ARIMA and Holt-Winters models were the most suitable for forecasting the stock prices of BRIC countries compared to the ARIMA model. Instead of ARMA, Hossain and Nasser [16] combined ARMA and GARCH to predict the Nikkei 225 and S&P 500. According to the results, the ARMA-GARCH model outperformed the SVR and back propagation (BP) models in terms of directional criteria, but performed worse in the deviation performance criteria compared to the SVR and BP models.

Although statistical models have exhibited a particular ability to forecast financial time series, they have some weaknesses when faced with complex data, particularly in modern times. First, the models are constructed based on stationary assumptions. Second, these models exhibit unsatisfactory performance in long-term time series. This impedes their use in complex time series analyses.

B. MACHINE LEARNING METHODS

With the advancements in machine learning, an increasing number of researchers are utilizing diverse models for time series prediction. Guo et al. [17] presented an SVR model to forecast stock prices. To alleviate the challenge of selecting hyper-parameters, the proposed adaptive SVR adopted a dynamic mechanism and a particle swarm optimization algorithm to adjust the parameters. The experimental results showed the efficacy of the adaptive SVR. Ren et al. [18] employed a support vector machine (SVM) in combination with investor sentiment analysis to forecast the price movement of the SSE 50 index. The findings revealed that incorporating sentiment analysis can enhance the prediction accuracy of SVM. To optimize the hyper-parameter settings of SVR, Liu et al. [19] proposed a hybrid algorithm called EGWO-SVR for stock selection, which integrated grey wolf optimizer into SVR. The experimental results indicated the superior performance of the proposed hybrid model compared with the basic SVR model in stock selection. Zhang and Lou [20] applied BP neural networks to forecast stock price patterns. Simulation experiments illustrated that the BP neural network had a certain prediction ability. To address the challenge of predicting non-linear components in stock indices, Yang and Lin [21] utilized the empirical mode decomposition (EMD) algorithm to decompose the stock indices into sub-series and subsequently employed the sub-series as input for SVR in forecasting. The study revealed that the proposed method achieved better prediction results than other models.

Compared to traditional statistical models, the prediction performance of machine learning models has improved, especially when combined with other optimization or decomposition methods. However, these models show weaknesses in terms of long-term prediction.

C. DEEP LEARNING METHODS

With the breakthrough of convolutional neural networks (CNN) in the ImageNet competition [9], computer vision has entered an era of deep learning. Simultaneously, deep learning has been applied to an increasing number of other fields such as natural language [22], agriculture [23], and finance [24].

Gülmez [25] used an LSTM network to predict stock prices. The hyperparameters of the LSTM model were optimized using an artificial rabbits optimization algorithm. The experimental results showed that the proposed model outperformed other models such as artificial neural networks, naive LSTM models, and an LSTM model optimized by the genetic algorithm. Qi et al. [26] proposed combining the GRU model with CEEMDAN to improve the stock prediction accuracy. To reduce the effect of high-frequency noise, the authors applied a wavelet transform to the decomposed sub-signals obtained from CEEMDAN. The results demonstrated the superiority of the model. Yao et al. [27] proposed a hybrid MEDM-TCN model for stock index prediction. Instead of using univariate time series, the model adopted a multivariate time series for decomposition and then applied temporal convolutional networks (TCN) to the decomposed subsequences for stock index prediction. The empirical results showed the effectiveness of the novel method. Cui et al. [28] proposed a novel hybrid model that combines multi-channel input, VMD, CBAM, and BiLSTM for stock index prediction. Empirical results showed that the proposed model outperformed other methods with high reliability. Wang et al. [29] proposed a multivariate deep learning method based on XGBoost for stock price forecasting. The experimental results showed the significant superiority of the hybrid model over the other models. Jiang et al. [30] proposed a novel dual-CNN model for stock prices prediction. First, the stock information was transformed into 2-D images. Subsequently, the proposed dual-CNN model was applied to the images for prediction. The results showed that the proposed model had superior predictive ability. To focus on the long-term and short-term dependencies of stock series simultaneously, Liu et al. [31] proposed combining VMD, a self-attention LSTM and a self-attention TCN for stock price prediction. The experimental results showed that the proposed hybrid model had better robustness and generalization than other methods. Lu and Xu [32] proposed an efficient time-series RNN for stock price prediction. The model adopted an additional time-series feature extraction module to enhance the correlation between data points, which helped improve the accuracy of stock price forecasting. Sivadasan et al. [33] studied various GRU and LSTM models with different architectures and inputs for stock market forecasting. Through a careful analysis, the GRU model with OHLC (open, high, low, and close prices) and technical indicators as inputs achieved the best forecasting performance.

Although deep learning has proven to outperform traditional machine learning methods in financial time series forecasting, the complexities of stock markets still pose a challenge for models to predict stock prices accurately and robustly.

III. METHODOLOGY

In this section, the specifications of the proposed HDFM hybrid model are detailed. First, we describe the key modules, such as CEEMDAN, K-means, VMD, and GRU, which are used to implement the decomposition process, clustering of sub-sequences, further re-decomposition, and obtaining prediction results. The structure of the HDFM is then outlined. Finally, this section details the use of evaluation metrics to validate the performance of the model, focusing on standard statistical metrics such as the coefficient of determination (R^2), root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), as well as a comparative analysis using the Diebold-Mariano (DM) test.

A. CEEMDAN

The CEEMDAN algorithm was proposed by Torres et al. [34] in 2011, which was derived from ensemble empirical mode decomposition (EEMD) [35] to deal with non-linear and non-stationary signals. This overcomes the problem of EEMD, in which the reconstructed signal is incomplete and erroneous. The CEEMDAN can be described as follows:

(1) Add white noise $\varepsilon_0\delta^k(t)$ to the original signal $x(t)$ to produce K different new series $x^k(t) = x(t) + \varepsilon_0\delta^k(t), k = 1, 2, \cdots, K$. $\delta^k(t)$ is the white noise and ε_0 is the weight.

Then, the first IMF and the first residual can be obtained:

$$IMF_1(t) = \frac{1}{K}\sum_{k=1}^{K} IMF_1^k(t) \tag{1}$$

$$r_1(t) = x(t) - IMF_1(t) \tag{2}$$

where $IMF_1^k(t)$ represents the decomposed sub-series by EMD applied to the k-th new series $x^k(t)$.

(2) For $n = 2, \cdots, N$, add white noise $\varepsilon_{n-1}E_{n-1}(\delta^k(t))$ to the residual $r_{n-1}(t)$ to produce K different new residuals, $r_{n-1}^k(t) = r_{n-1}(t) + \varepsilon_{n-1}E_{n-1}(\delta^k(t)), k = 1, 2, \cdots, K$. $E_{n-1}(\cdot)$ is defined as the $(n-1)$-th IMF of a signal produced by EMD. Then, the n-th IMF and the n-th residual are obtained as follows:

$$IMF_n(t) = \frac{1}{K}\sum_{i=1}^{K} E_1\left(r_{n-1}^k(t)\right), n = 2, \cdots, N \tag{3}$$

$$r_n(t) = r_{n-1}(t) - IMF_n(t), n = 2, \cdots, N \tag{4}$$

(3) Step 2 is repeated until the obtained residual can no longer be decomposed. The final residual of CEEMDAN is obtained as follows:

$$R(t) = x(t) - \sum_{n=1}^{N} IMF_n(t) \tag{5}$$

B. K-MEANS

In this study, the IMFs produced by CEEMDAN are grouped based on their complexity or frequency, to ease the prediction process of GRU. For this purpose, the K-means clustering algorithm [36] is adopted. In addition, to quantify the complexity of different IMFs, the sample entropy [37] is selected as the similarity measurement of two IMFs for K-means.

Given a set of samples $S = s_1, s_2, \cdots, s_N$, and the predefined cluster centers $c_1, c_2, \cdots, c_K$, the objective of K-means is to minimize the following function:

$$J = \sum_{k=1}^{K}\sum_{i=1}^{N_k}\left\| s_i^{(k)} - c_k \right\|_2 \tag{6}$$

where $s_i^{(k)}$ refers to the sample s_i that belongs to cluster c_k, N_k is the number of samples in cluster c_k, and $\|\cdot\|_2$ denotes L_2 norm.

A key component of K-means is the selection of the number of centroids, that is, K, which is a hyper-parameter. In this study, the Elbow method [38] is adopted to automatically determine the number of cluters K.

The clustering steps of the K-means algorithm is summarized in Algorithm 1.

Algorithm 1 K-Means Clustering Algorithm

$\mathcal{S} = \{s_1, s_2, \cdots, s_N\}, K$
$\mathcal{S}$ is the set of samples
K is the number of clusters
Input:
Output: Cluster division: $\mathcal{C} = \{c_1, c_2, \cdots, c_K\}$

```
Initialization: C ← S ELECTR ANDOMS EEDS(S, K)
repeat
  for n = 1, 2, ···, N do
    for k = 1, 2, ···, K do
      if k == arg min_k ||s_n − c_k||_2 then
        r_nk = 1
      else
        r_nk = 0
      end if
    end for
  end for
  for each cluster c_k do
    update cluster centroids as the mean of each cluster:
      c_k = Σ r_nk · s_n / Σ r_nk
  end for
until all current cluster centroids do not change
```

C. VMD

VMD [39] is a non-recursive signal decomposition model that adaptively determines the number of mode decomposition. It suppresses the effect of mode component aliasing in EMD and is theoretically well founded. Specifically, VMD decomposes a signal by transforming the solution problem into a variational problem. Then, it adopts the alternating direction method of multipliers (ADMM) to solve the variational problem and get a set of modes with limited bandwidth and their corresponding center frequencies.

The VMD framework consists of two stages: construction and solving. In the construction stage, for a given signal f, the variational problem is solved by minimizing the sum of the estimated bandwidth for each mode u_k $(k = 1, 2, \cdots, K)$. In reverse, the modes collectively reproduce the signal f. The solving problem can be formulated as the following constrained optimization problem:

$$\begin{aligned} &\min_{\{u_k\},\{\omega_k\}} \left\{ \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) \otimes u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\} \\ &\text{s.t.} \quad \sum_{k=1}^{K} u_k = f \end{aligned} \tag{7}$$

where $\{u_k\} := \{u_1, u_2, \cdots, u_K\}$ represents the set of all modes, and $\{\omega_k\} := \{\omega_1, \omega_2, \cdots, \omega_K\}$ represents the center frequencies of each mode. $\delta(t)$ is the Dirac distribution, ∂_t

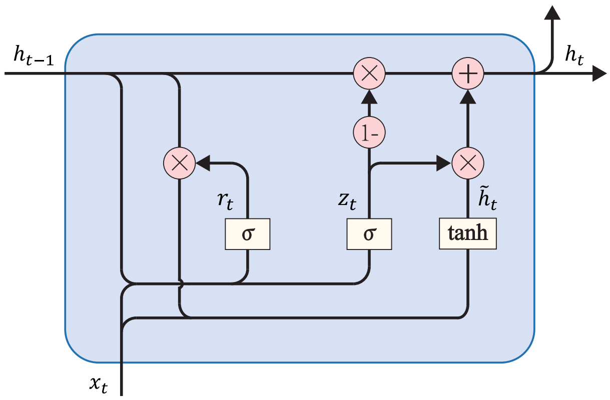


FIGURE 1. The internal structure of a GRU unit.

denotes the partial derivative of t, and $\otimes$ represents the convolutional operation. $\|\cdot\|_2$ denotes L_2 norm.

In the solving stage, the above constrained optimization problem is transformed into an unconstrained one by introducing the secondary penalty factor α and the Lagrange multiplier λ as follows:

$$\begin{aligned}\mathcal{L}(\{u_k\},\{\omega_k\},\lambda) &= \alpha\sum_{k=1}^{K}\left\|\partial_t\left[\left(\delta(t)+\frac{j}{\pi t}\right)\otimes u_k(t)\right]e^{-j\omega_k t}\right\|_2^2 \\ &\quad+\left\|f(t)-\sum_{k=1}^{K}u_k(t)\right\|_2^2+\left\langle\lambda(t),s(t)-\sum_{k=1}^{K}u_k(t)\right\rangle \end{aligned}\tag{8}$$

Equation (8) is solved by using the ADMM strategy, and the optimal solution of u_k, ω_k and λ are obtained as follows:

$$\hat{u}_k^{n+1}(\omega)=\frac{\hat{f}(\omega)-\sum_{i\neq k}a_i(\omega)+\frac{\hat{\lambda}(\omega)}{2}}{1+2\alpha(\omega-\omega_k)^2}\tag{9}$$

$$\widehat{\omega}_k^{n+1}(\omega)=\frac{\int_0^\infty\omega\left|\hat{u}_k(\omega)\right|^2d\omega}{\int_0^\infty\left|\hat{u}_k(\omega)\right|^2d\omega}\tag{10}$$

$$\hat{\lambda}^{n+1}(\omega)=\hat{\lambda}^n(\omega)+\tau\left(\hat{f}(\omega)-\sum_{k=1}^{K}\hat{u}_k^{n+1}(\omega)\right)\tag{11}$$

where $\hat{u}_k^{n+1}(\omega)$, $\hat{u}_i(\omega)$, $\hat{f}(\omega)$, and $\lambda(\omega)$ denote the Fourier transform of $u_k^{n+1}(t)$, $u_i(t)$, $f(t)$, and $\lambda(t)$, respectively. τ is the tolerance of noise.

D. GRU

The GRU is a kind of RNN model proposed by Cho et al. in 2014 [40], which aims to address the gradient vanishing problem that comes with a standard RNN in the case of long-term time series dependencies. It is a simpler version of LSTM and easier to train than LSTM with the same performance.

The GRU memory cell consists of only two parts as shown in Figure 1: the update and reset gates. In Figure 1, z_t denotes the update gate, r_t represents the reset gate, and h_{t-1} and h_t denote the previous and current hidden states, respectively. $\widetilde{h}_t$ is the candidate hidden state.

The forecasting steps of GRU are as follows:

(1) The update gate z_t determines how much of the previous hidden state h_{t-1} need to be retained, and how much will be replaced by the new information in $\widetilde{h}_t$. The computation of z_t is as below:

$$z_t=\sigma\left(w_z\left[h_{t-1},x_t\right]+b_z\right)\tag{12}$$

where w_z and b_z represent weight and bias, respectively. σ denotes the sigmoid function.

(2) The reset gate r_t determines how much of the previous hidden state h_{t-1} will be discarded, and how much will be used to calculate $\widetilde{h}_t$.

$$r_t=\sigma\left(w_r\left[h_{t-1},x_t\right]+b_r\right)\tag{13}$$

(3) This step primarily computes the candidate hidden state $\widetilde{h}_t$ that is to be mixed with the previous hidden state h_{t-1} through the update gate z_t.

$$\tilde{h}_t=\tanh\left(w_h\left[r_t\cdot h_{t-1},x_t\right]+b_h\right)\tag{14}$$

(4) The final step is to calculate the current hidden state h_t with a linear combination of the previous hidden state h_{t-1} and the candidate hidden state $\widetilde{h}_t$, weighted by the update gate z_t.

$$h_t=(1-z_t)\cdot h_{t-1}+z_t\cdot\tilde{h}_t\tag{15}$$

E. THE PROPOSED HDFM MODEL

In this section, we introduce the principles and forecasting process of the proposed HDFM model. The fundamental idea behind this model is to break down the non-linear and non-stationary stock market index price into numerous relatively simple sub-series, which are subsequently used as inputs to the forecasting model. Finally, the predictions of each sub-series are weighted summed up to derive the forecast of the stock market index. Since the decomposition process usually produces several sub-series, which makes the subsequent prediction process time-consuming, this study adopts the K-means algorithm to merge the sub-series with similar complexity before the forecasting process. Sample entropy is a measure of the probability of generating a new pattern of change in a time series and is commonly used to characterize the complexity of sequences [29], [41]. Thus, in this study, the sample entropy serves as a measure of the complexity of the decomposed IMFs and as an input to the K-means algorithm. The IMFs with the same clusters of sample entropy are aggregated by summation. Furthermore, the high-frequency component produced by the initial decomposition and reconstruction is re-decomposed to generate sub-series in a better mode, which further improves the predictive performance of the model. In this study, the sub-sequences whose sample entropy exceeds a predetermined threshold is defined as a high-frequency component. An overview of the proposed HDFM is shown in Figure 2.

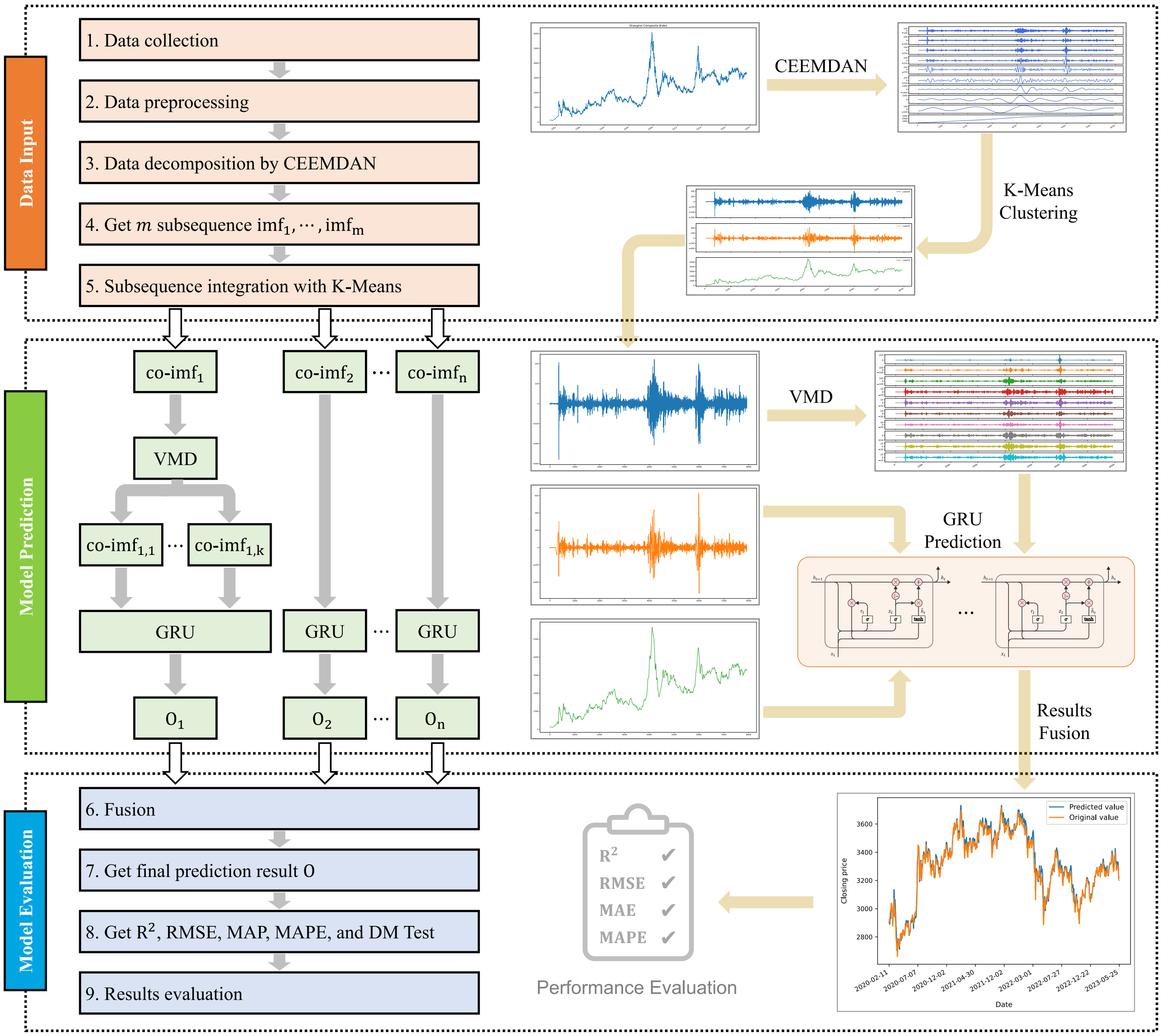


FIGURE 2. Overview of the proposed HDFM model.

F. EXPERIMENTAL SETUP

In this study, the models were implemented using Keras in Python 3.8, with a backend of TensorFlow 2.5.0. All the experiments are carried out on a PC client with Intel i7-9700K CPU@3.60 GHz, a 16GB RAM, and a NVIDIA GeForce GTX 1080Ti GPU.

To evaluate the forecasting capability of the proposed model, the standard statistical metrics of R^2, RMSE, MAE, and MAPE were adopted and defined as follows:

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \tag{16}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (\hat{y}_i - y_i)^2} \tag{17}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |\hat{y}_i - y_i| \tag{18}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} \left|\frac{\hat{y}_i - y_i}{y_i}\right| \times 100\% \tag{19}$$

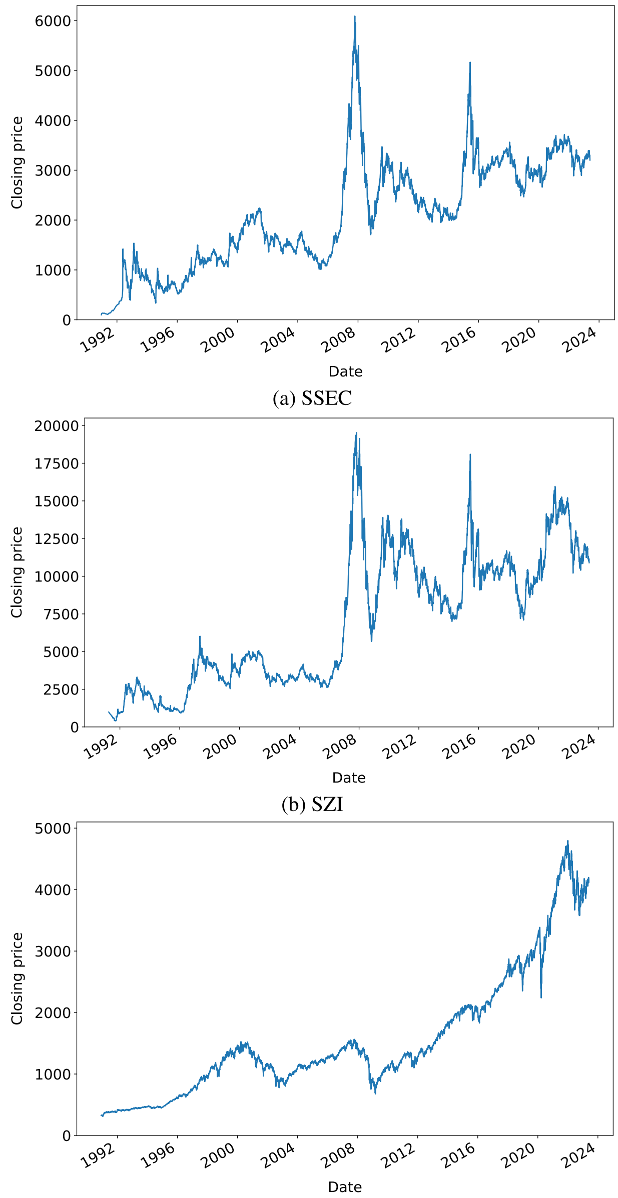
where y_i and $\hat{y}_i$ represent the actual and predicted value, respectively. $\bar{y}$ is the average of the close price of the n samples.

IV. EMPIRICAL STUDY

To evaluate the performance of the new HDFM model, this section tests the model on multiple datasets using a variety of benchmark models and evaluation criteria. First, the datasets and parameter settings of the models used

TABLE 1. The statistical analysis of three stock market indices conditions.

Stock Index	Count	Min	Max	Mean	Standard Deviation
SSEC	7922	99.98	6092.057	2090.398997	1095.930913
SZI	7879	402.5	19531.155	7014.205923	4455.95512
SPX	8184	311.49	4796.6	1567.254398	1023.730998



(a) SSEC

(b) SZI

(c) SPX

FIGURE 3. The original value of stock daily closing price for SSEC, SZI, and SPX.

in the experiment are described. Then, comparison of the proposed model and other models is conducted to verify the prediction performance. Finally, the effectiveness of each component of the model is verified through ablation studies.

A. DATASET

In this study, the daily closing prices of three stock indices were selected to evaluate the proposed method. These are the Shanghai Securities Composite Index (SSEC), Shenzhen Securities Component Index (SZI), and the Standard & Poor 500 Index (SPX). The data are obtained from Yahoo Finance (https://finance.yahoo.com). The SSEC data are from December 19,1990 to May 25, 2023. The SZI data are from April 3, 1991 to May 25, 2023, and the data of SPX are from December 3, 1990 to May 25, 2023. The total number of observations for SSEC, SZI, and SPX are 7922, 7879 and 8184, respectively. The last 800 data points are selected as the test set for each stock index, and the remaining data are used as the training set. For a better and more intuitive realization of these indices, the data are described in Table 1 and the graphs are plotted in Figure 3.

To reduce the effect of noise, such as outliers and extrema, and speed up the training process of the model, the data were normalized as follows:

$$x_{norm} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{20}$$

where x represents the sample data and $x_{\min}$ and $x_{\max}$ represent the minimum and maximum values of the sample, respectively. After prediction, the predicted values are restored according to the following equation:

$$\hat{x} = (x_{\max} - x_{\min}) * \hat{x}' + x_{\min} \tag{21}$$

where $\hat{x}'$ is the output of the forecasting model.

B. MODEL IMPLEMENTATION

Table 2 presents the hyper-parameters of all models used in the experiments. The DNN, RNN, LSTM and GRU models share the same parameters for a fair comparison. SVR and random forest regression (RFR) models are implemented using scikit-learn [42]. Furthermore, the lookback window is set to a time step of 30, that is, the closing price of one day is predicted using the closing prices of the preceding 30 days.

C. COMPARISON WITH OTHER METHODS

In this study, eight other models including Random Walk, ARIMA, SVR, RFR, LSTM, GRU, CEEMDAN-GRU and McVCsB [28] were used for comparison to validate the performance of the proposed HDFM model. It should be noted that CEEMDAN-GRU means that CEEMDAN is first used to decompose the original stock price into sub-series,

TABLE 2. The parameter setting of the models.

Algorithm	Parameter	Value(s)
ARIMA	$(p, d, q)^1$	(3, 1, 4)
	$(p, d, q)^2$	(3, 1, 4)
	$(p, d, q)^3$	(4, 1, 4)
SVR	C	1.0
	Epsilon	0.1
	Kernel	RBF
	Loss function	L1
RFR	# of estimators	100
	Max depth	50
	Loss function	MSE
DNN, RNN, LSTM, GRU	Hidden layer	3
	Cells	(128, 64, 32)
	Batch size	64
	Activation function	tanh
	Loss function	MSE
	Optimizer	Adam
	Learning rate	0.001
	GradientDecayFactor	0.9
	SquareGradientDecayFactor	0.999
	Epochs	100

Note: The superscripts $^{1, 2, 3}$ indicate parameter settings for the SSEC, SZI, and SPX datasets, respectively. Parameters without superscripts indicate that they are shared by all the datasets.

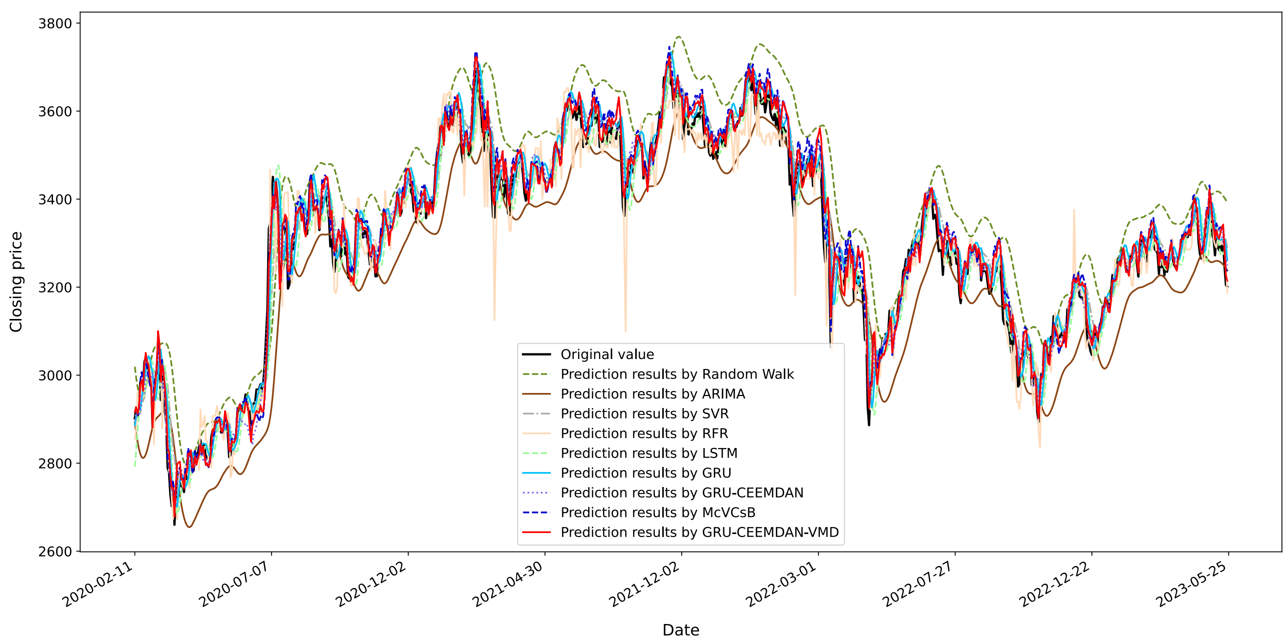


FIGURE 4. Forecasting comparison of different models for SSEC.

and then the GRU model is applied to predict each of the sub-series. Finally, all the predictions are merged to obtain the final result.

The prediction results of the models are listed in Table 3. Compared to traditional methods such as Random Walk and ARIMA, machine learning-based methods performed much

TABLE 3. Comparison of different stock index price forecasting models.

Stock Index	Models	R^2	RMSE	MAE	MAPE (%)
SSEC	Random Walk	0.733836	117.221506	95.781951	2.9149
	ARIMA	0.779042	106.803918	86.350939	2.6386
	SVR	0.939124	56.060122	40.465087	1.2394
	RFR	0.944711	53.425877	39.291353	1.1888
	LSTM	0.945714	52.939073	39.532023	1.2088
	GRU	0.952263	49.643274	37.100258	1.1304
	CEEMDAN-GRU	0.974597	36.214179	27.656854	0.8403
	McVCsB [28]	0.982210	30.305413	25.140639	0.7570
	HDFM (Ours)	0.985613	27.253625	21.262825	0.6462
SZI	Random Walk	0.730165	794.174319	646.348302	4.7330
	ARIMA	0.789279	701.811655	562.335368	4.2163
	SVR	0.930665	402.571974	313.878719	2.4428
	RFR	0.953944	328.103694	251.431450	1.9760
	LSTM	0.934517	391.229139	295.314832	2.3622
	GRU	0.942982	365.066701	280.415401	2.2494
	CEEMDAN-GRU	0.962238	297.095775	248.869239	1.9963
	McVCsB [28]	0.972950	251.448699	198.307352	1.5756
	HDFM (Ours)	0.975534	239.138236	187.725595	1.4805
SPX	Random Walk	0.706311	264.282009	248.246785	6.2603
	ARIMA	0.757341	240.227153	228.591172	5.7352
	SVR	0.929492	129.491891	107.550343	2.6104
	RFR	-1.529433	775.596148	662.914282	15.7891
	LSTM	0.943226	116.198201	99.801158	2.5357
	GRU	0.953382	105.292904	90.948408	2.3058
	CEEMDAN-GRU	0.971983	81.627592	70.669927	1.8060
	McVCsB [28]	0.982205	65.054074	59.72503	1.5091
	HDFM (Ours)	0.984663	60.394758	54.647467	1.3851

better, demonstrating the superiority of machine learning in financial time series forecasting. It can be seen that the proposed model has the best performance on all three stock market indices. In the case of SSEC, the proposed HDFM achieves a 0.35% increase in R^2, and a 10.07% decrease in RMSE compared to the second ranked McVCsB model. In the case of SZI, the proposed model has 4.90% lower RMSE than the second ranked McVCsB model. The R^2 value of HDFM is 0.26% higher than that of McVCsB. In the case of SPX, the proposed model achieves a 0.25% increase in R^2 and a 7.16% reduction in RMSE compared to the second-ranked McVCsB model. The results demonstrate the superiority of the proposed model for predicting stock prices. Qualitative visualizations of the prediction results for SSEC, SZI, and SPX are shown in Figure 4, 5, and 6, respectively.

To further demonstrate that the HDFM model proposed in this study significantly outperforms the benchmark models, we performed a DM test on the HDFM and benchmark models. The main purpose of the DM test is to test whether there is a significant difference in the time series forecasting results between the two models. If the p-value is greater than the significance level, the null hypothesis is accepted, indicating that both models have the same predictive power. Conversely, if the p-value is less than the significance level, the null hypothesis is rejected, indicating that the two models have different predictive effects. In addition, under the condition that the p-value is below the significance level, if the DM test value is greater than 0, then the benchmark model is superior to the HDFM model proposed in this study. If the DM test value is less than 0, then the HDFM model is superior to the benchmark model. The results of the DM test are shown in Table 4. It can be observed that across all datasets, the proposed HDFM model is generally superior to the benchmark models at the 1% significance level.

D. ABLATION STUDY ON DIFFERENT BASE MODELS

To determine the best base model fit for stock price forecasting, this study evaluated different deep learning

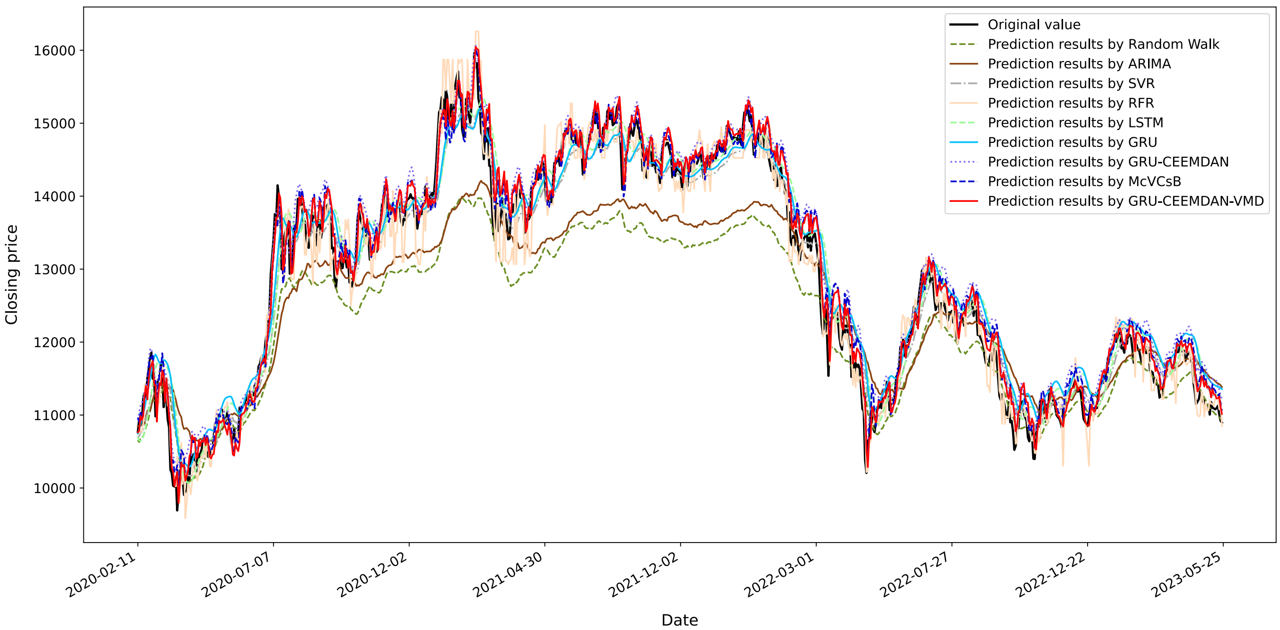


FIGURE 5. **Forecasting comparison of different models for SZI.**

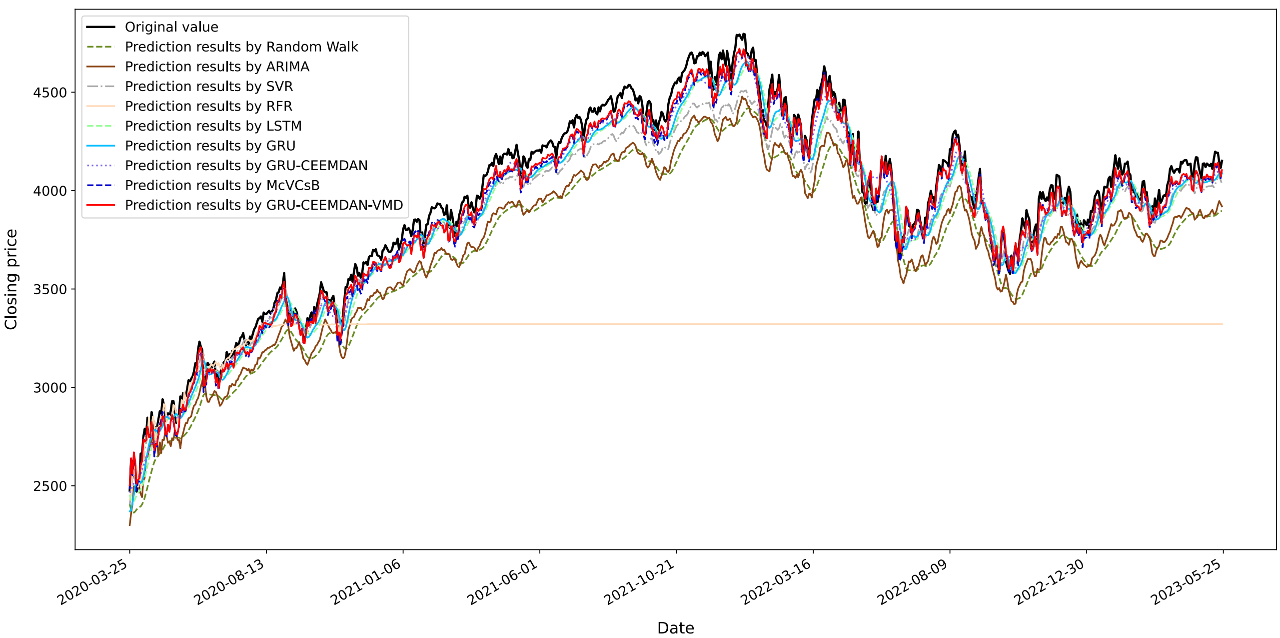


FIGURE 6. **Forecasting comparison of different models for SPX.**

models. A deep neural network (DNN), recursive neural network (RNN), LSTM and GRU were chosen as the basic forecasting networks for comparison. DNN is also known as the multilayer perceptron.

The results of the comparison are shown in Figure 7 and Table 5. From Table 5, we can see that GRU performs the best among all the models for all three stock indices. In contrast, the DNN has the worst performance on all three datasets. Specifically, in the case of SSEC, GRU obtains an R^2 of 0.952263, which is 0.65% higher than that of the second ranked LSTM model. The RMSE of GRU is 6.23% lower than that of LSTM. For SZI, the R^2 and RMSE are 0.85% higher and 6.69% lower, respectively, than those of the second-ranked LSTM. For SPX, GRU has an RMSE 9.39% lower than that of LSTM. The R^2 of the GRU is 0.953382, which is 1.02% higher than that of the LSTM. For other measures, such as MAE and MAPE, GRU's results are also the best of all three stock indices.

TABLE 4. The DM test values of benchmark models of different datasets.

Models	SSEC	SZI	SPX
Random Walk	-24.4116*	-18.3464*	-47.2520*
ARIMA	-19.3096*	-16.7186*	-51.7657*
SVR	-10.7593*	-5.8449*	-16.6940*
RFR	-9.6530*	-3.1997*	-30.8684*
LSTM	-11.1346*	-9.0286*	-15.8925*
GRU	-9.2759*	-8.4112*	-14.4942*
CEEMDAN-GRU	-6.8599*	-10.5458*	-10.7499*
McVCsB [28]	-6.0911*	-2.8999*	-11.9913*

Note: * indicates 1% level of significance.

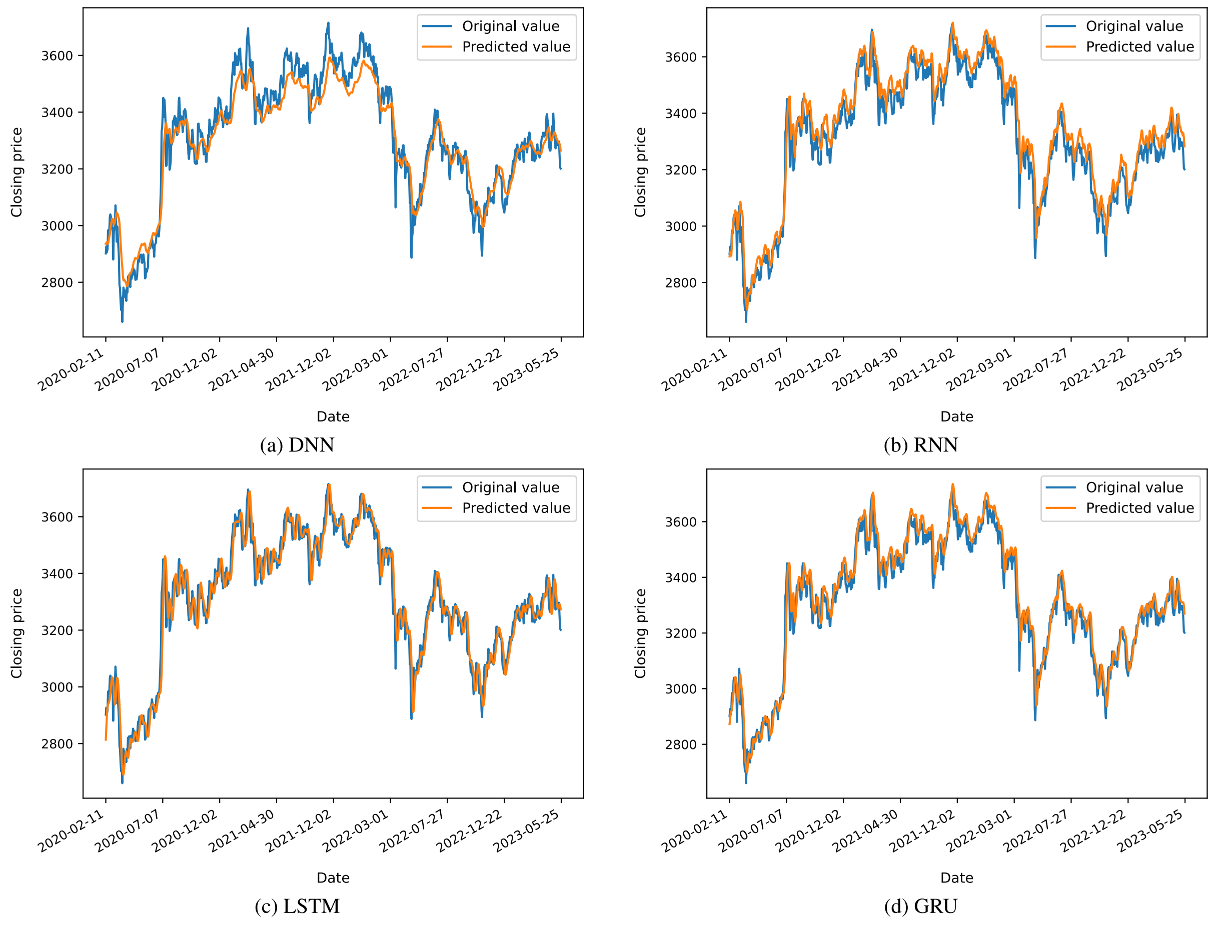


FIGURE 7. A visualization example of ablation study on different base models for SSEC.

E. ABLATION STUDY ON DIFFERENT DECOMPOSITION METHODS

Several decomposition methods have been evaluated using stock market datasets to determine the most appropriate method for predicting stock index prices. The methods considered in this study include EMD, EEMD, and CEEMDAN.

Additionally, to reduce computation time, the K-means method was employed to cluster the decomposed intrinsic mode functions (IMFs) before forecasting. For K-means, the sample entropy serves as the criterion for grouping. This means that IMFs with comparable sample entropy are merged to form fresh cooperative intrinsic mode functions (Co-IMFs). Table 6 shows the sample entropy values of each sub-series decomposed by CEEMDAN for the SSEC, SZI, and SPX datasets. The clustering results are also shown in the table, where the IMFs whose sample entropy is indicated by

TABLE 5. Ablation study on different base networks for SSEC, SZI and SPX.

Stock Index	Models	R^2	RMSE	MAE	MAPE (%)
SSEC	DNN	0.924679	62.357722	48.142256	1.4577
	RNN	0.935156	57.858308	45.351393	1.3872
	LSTM	0.945714	52.939073	39.532023	1.2088
	GRU	0.952263	49.643274	37.100258	1.1304
SZI	DNN	0.916917	440.680103	341.104727	2.7604
	RNN	0.926455	414.614627	346.296195	2.8191
	LSTM	0.934517	391.229139	295.314832	2.3622
	GRU	0.942982	365.066701	280.415401	2.2494
SPX	DNN	0.925037	133.52051	112.727832	2.9361
	RNN	0.930592	128.47827	117.273611	3.0194
	LSTM	0.943226	116.198201	99.801158	2.5357
	GRU	0.953382	105.292904	90.948408	2.3058

TABLE 6. Sample entropy of each sub-series of the CEEMDAN decomposition for different datasets.

Modal	SSEC	SZI	SPX
IMF1	1.844392*	1.744000*	1.792528*
IMF2	1.820377*	1.608018*	1.763276*
IMF3	1.391727*	1.198358*	1.394437*
IMF4	0.700810**	0.590323**	0.617028**
IMF5	0.348395**	0.302360**	0.335909**
IMF6	0.179897***	0.172095***	0.145157***
IMF7	0.037263***	0.029929***	0.033227***
IMF8	0.020775***	0.034133***	0.020805***
IMF9	0.013286***	0.009987***	0.004585***
IMF10	0.000522***	0.000385***	0.000183***

Note: *, **, and *** denote different clustering groups by K-means in each dataset, where the same symbols represent the same group.

TABLE 7. Ablation study on different decomposition methods for SSEC, SZI and SPX.

Stock Index	Decomposition Methods	R^2	RMSE	MAE	MAPE (%)
SSEC	EMD	0.963650	43.319590	34.329957	1.0412
	EEMD	0.930576	59.867071	46.814153	1.4244
	CEEMDAN	0.974597	36.214179	27.656854	0.8403
SZI	EMD	0.951253	337.550834	274.995443	2.0907
	EEMD	0.931051	401.448513	315.358002	2.3749
	CEEMDAN	0.962238	297.095775	248.869239	1.9963
SPX	EMD	0.964047	92.467734	81.792127	2.0720
	EEMD	0.933183	126.057309	107.252123	2.7204
	CEEMDAN	0.971983	81.627592	70.669927	1.8060

Note: All decomposition methods are combined with the GRU model.

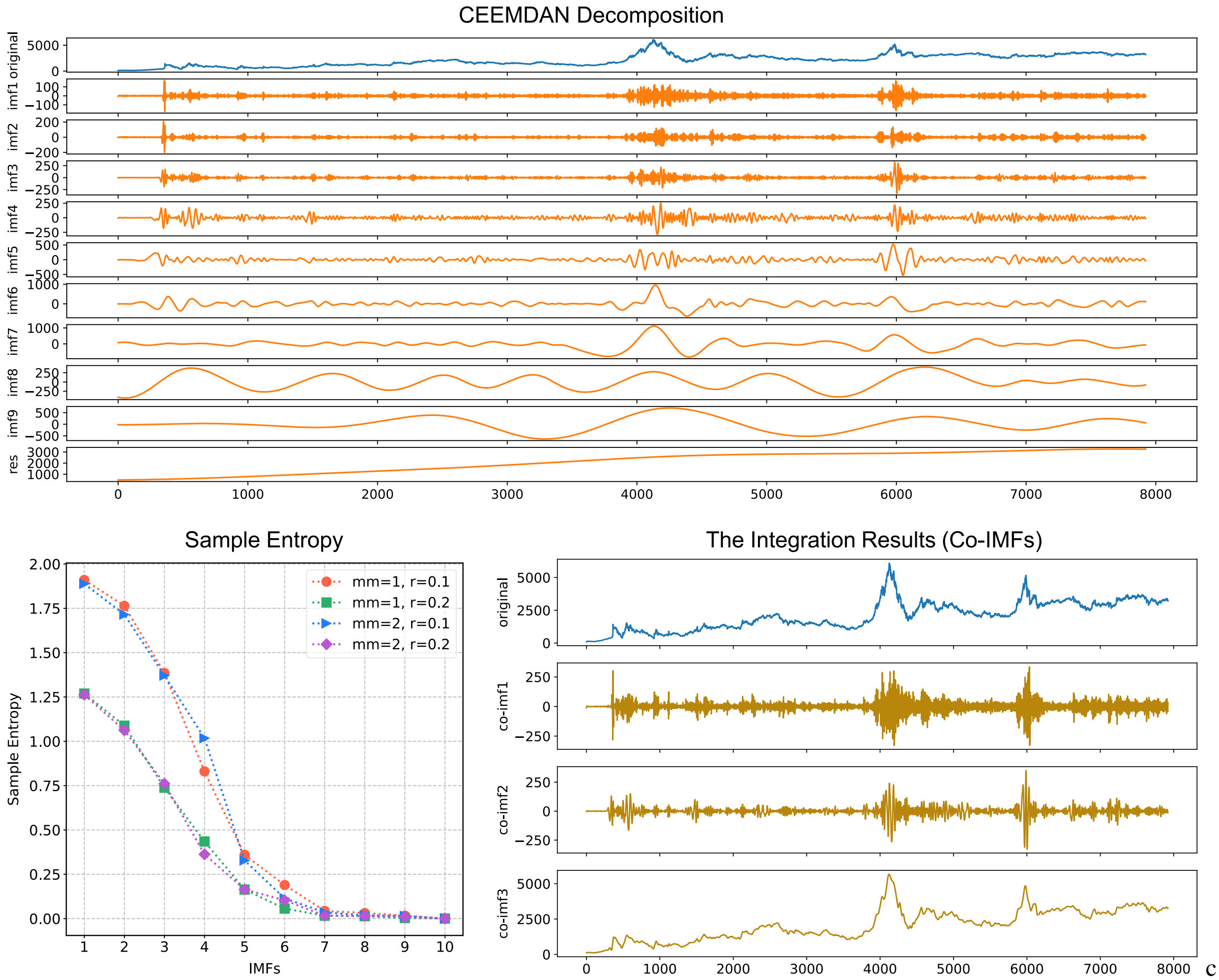
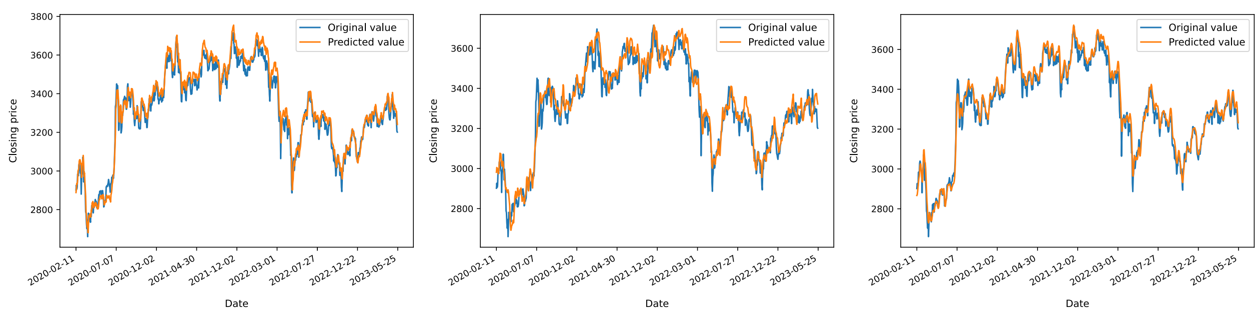


FIGURE 8. Visualization of CEEMDAN decomposition, sample entropy and integration results for SSEC.



(a) EMD-GRU (b) EEMD-GRU (c) CEEMDAN-GRU

FIGURE 9. A visualization example of ablation study on different decomposition methods for SSEC.

TABLE 8. Sample entropy of each Co-IMF for different datasets.

Modal	SSEC	SZI	SPX
Co-IMF1	1.756717	1.526462	1.657737
Co-IMF2	0.520439	0.414427	0.458080
Co-IMF3	0.024130	0.018658	0.008260

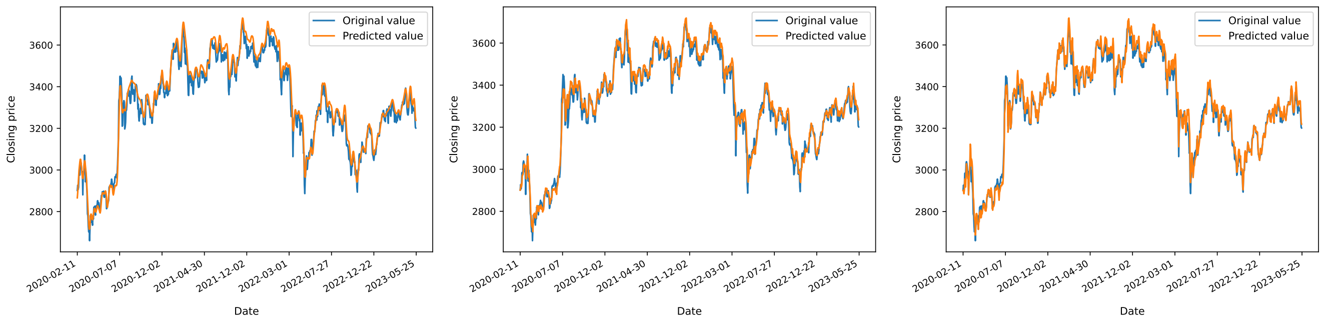
the same symbol (i.e. *, **, or ***) belong to the same cluster. Figure 8 shows the decomposition and integration process of SSEC.

Table 7 and Figure 9 present the outcomes of predicting stock index prices by applying GRU in combination with the different decomposition methods across the three datasets.

TABLE 9. Ablation study on different re-decomposition methods applied on Co-IMF1 for SSEC, SZI and SPX.

Stock Index	Re-decomposition Methods	R^2	RMSE	MAE	MAPE (%)
SSEC	EMD	0.971828	38.136466	30.312859	0.9152
	CEEMDAN	0.975591	35.498466	26.464414	0.8076
	VMD	0.985613	27.253625	21.262825	0.6462
SZI	EMD	0.964254	289.056129	238.205325	1.9010
	CEEMDAN	0.964836	286.691091	233.795311	1.7879
	VMD	0.975534	239.138236	187.725595	1.4805
SPX	EMD	0.970668	83.520435	73.957386	1.8899
	CEEMDAN	0.973861	78.844252	70.131502	1.7722
	VMD	0.984663	60.394758	54.647467	1.3851

Note: All re-decomposition methods are combined with the CEEMDAN and GRU models.



(a) CEEMDAN-EMD-GRU (b) CEEMDAN-CEEMDAN-GRU (c) CEEMDAN-VMD-GRU

FIGURE 10. A visualization example of ablation study on different re-decomoposition methods for Co-IMF1 in SSEC.

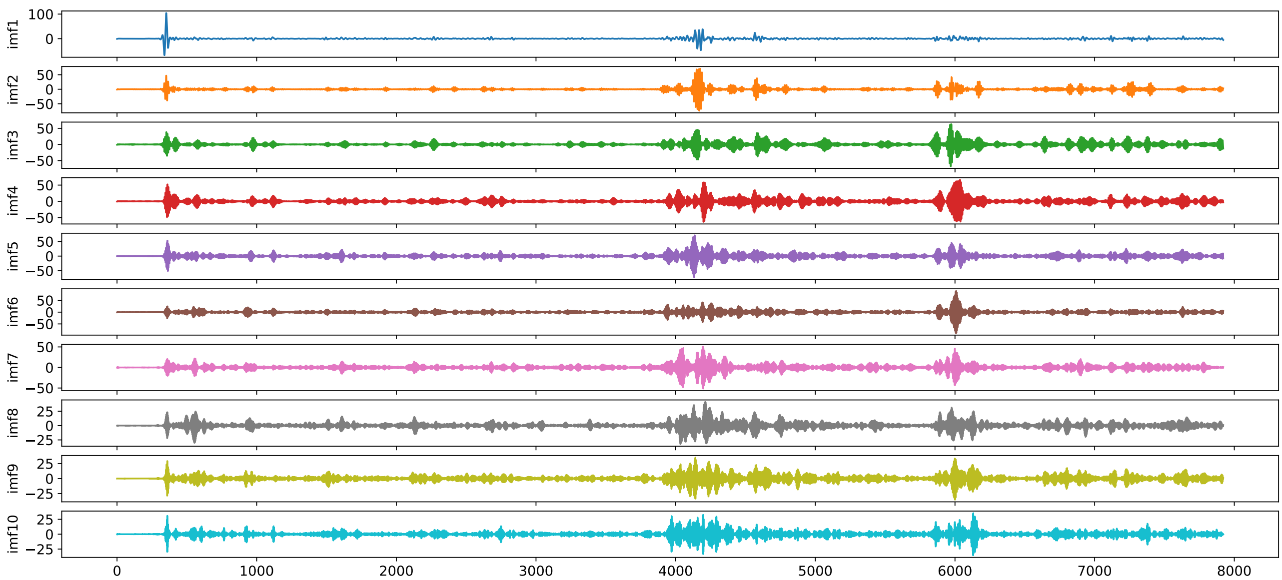


FIGURE 11. A visualization example of re-decomposition results for Co-IMF1 using VMD in SSEC.

Notably, the combination of GRU and CEEMDAN yields significantly superior forecasting outcomes for all three datasets. Specifically, in the SSEC case, the R^2 of the forecasting model utilizing CEEMDAN reaches 0.974597, which is nearly 1.1% higher than that of the second-ranked EMD-GRU. The RMSE and MAE of the model utilizing CEEMDAN display a decrease of 16.40% and 19.44%, respectively, in comparison to the EMD-GRU model. When SZI is considered, the CEEMDAN-GRU demonstrates a lower RMSE of 11.98% when compared against the EMD-GRU model in second place. The R^2 of the CEEMDAN-GRU is 1.10% higher than that of the EMD-GRU. Turning to SPX, the model incorporating CEEMDAN outperforms the second-ranked EMD-GRU with an increase of 0.79% in R^2. The CEEMDAN-GRU model achieves an 11.72% and 13.60% decrease in RMSE and MAE, respectively, compared to the EMD-GRU model. Consequently, the results indicate the efficacy of CEEMDAN in enhancing the model performance. It is worth noting that the EEMD-GRU model performs significantly worse than the EMD-GRU model. After conducting a thorough analysis, it was discovered that EEMD generates a large number of high-frequency components, rendering the forecasting process more difficult when compared to the EMD method in this study.

F. ABLATION STUDY ON DIFFERENT RE-DECOMPOSITION METHODS

Although the GRU model could make accurate predictions for the stock index price when combined with CEEMDAN, further study was conducted to explore the potential for improved predictive ability.

By analyzing the prediction results for various Co-IMFs, we found that the prediction result of Co-IMF1 is much worse than those of the other two Co-IMFs across all three datasets. Further analysis revealed that the sample entropy of the first Co-IMF (Co-IMF1) was significantly higher compared to the other two Co-IMFs across all datasets, as shown in Table 8. This observation suggests that Co-IMF1 possesses greater complexity and contains more high-frequency signals than its counterparts do. Therefore, our study focuses on enhancing the prediction performance of Co-IMF1. Our approach entails the re-decomposition of Co-IMF1 to decouple sub-series with varying frequencies and predicting Co-IMF1 using these sub-series. In this study, three re-decomposition methods, EMD, CEEMDAN, and VMD, were evaluated and compared in terms of their performances across the three datasets.

The forecasting results are presented in Tables 9 and 10, respectively. The use of VMD on Co-IMF1 for re-decomposotion yielded the best forecasting performance across all three datasets. For SSEC, utilizing VMD as re-decomposition in the forecasting model results in a decrease of 23.23% and 19.66% in RMSE and MAE, correspondingly, as compared to the second-ranked model that employs CEEMDAN. The R^2 increases by 1.00% in this scenario. For SZI, the model that adopts VMD as a re-decomposition exhibits a 1.07% increase in R^2 and a decrease of 16.59% and 19.71% in RMSE and MAE, respectively, as compared to the secondary model. Similarly, for SPX, the model using VMD for re-decomposition achieves an R^2 of 0.984663, surpassing the second-best model by 1.08%. As a result, the RMSE and MAE of the forecasting model

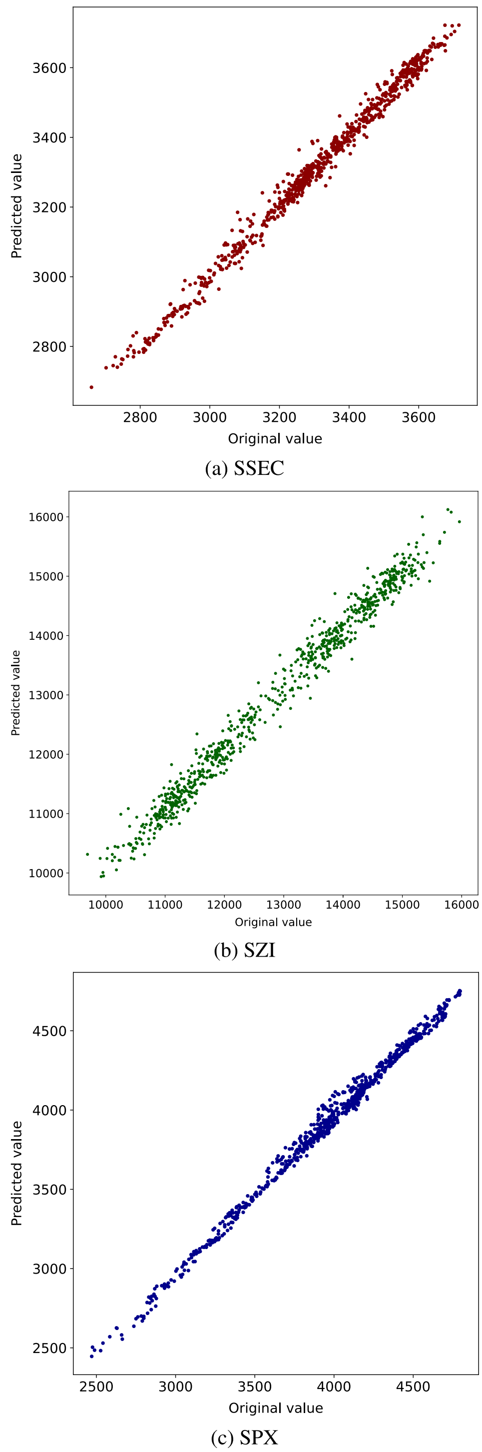


FIGURE 12. Scatter plots of original and predicted values for different stock indices using the proposed HDFM model.

adopting VMD as re-decomposition decrease by 23.40% and 22.08%, respectively, compared with the second-best model. Figure 11 depicts the re-decomposition outcomes of Co-IMF1 for SSEC using the VMD approach.

Based on these findings, it can be concluded that VMD is the most effective re-decomposition method for Co-IMF1 compared to the other two methods. This is because both EMD and CEEMDAN are derived from EMD, and utilizing them to decompose the IMF1's high-frequency components does not result in any further improvement in the performance.

G. QUALITATIVE ANALYSIS OF THE PREDICTION PERFORMANCE

Figure 12 displays the scatter plots of both the original and predicted price values to provide a comprehensive visual analysis of the proposed HDFM for stock prices. The distribution of the scatter points aligns closely with the line of slope 1 in all the cases. This finding provides evidence that the proposed model can predict stock index prices with a high level of accuracy and robustness.

V. CONCLUSION

In this study, we propose a hybrid model called HDFM for predicting stock index prices using GRU and decomposition methods. Our model is based on the divide-and-conquer strategy, hierarchically addressing the stock price prediction problem. Specifically, we decompose the stock price time series into several sub-series using the CEEMDAN method. To reduce the computation time, sub-series with similar sample entropy are merged using K-means. The Co-IMF1 with the highest frequency among all Co-IMFs is further decomposed into sub-series for easier prediction. Finally, the prediction of all sub-series are fused to obtain the prediction of the original dataset.

The performance of the proposed model was assessed extensively using three distinct stock market indices. The ablation study demonstrated the efficacy of each element of the model, comprising GRU as the base network, CEEMDAN as the decomposition method, and VMD as the re-decomposition method. Our model outperforms the other methods for all three stock market indices.

Nevertheless, there is scope for the refinement of the proposed model. Primarily, the forecast accuracy of the high-frequency sub-series is inferior to that of the medium- and low-frequency sub-series. Second, the multi-step prediction of stock index prices will be explored in the future.

REFERENCES

[1] C. Kocak, "ARMA(p,q) type high order fuzzy time series forecast method based on fuzzy logic relations," *Appl. Soft Comput.*, vol. 58, pp. 92–103, Sep. 2017.

[2] Y. Wang and Y. Guo, "Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost," *China Commun.*, vol. 17, no. 3, pp. 205–221, Mar. 2020.

[3] L. Wang, F. Ma, J. Liu, and L. Yang, "Forecasting stock price volatility: New evidence from the GARCH-MIDAS model," *Int. J. Forecasting*, vol. 36, no. 2, pp. 684–694, Apr. 2020.

[4] G. Ataman and S. Kahraman, "Stock market prediction in Brics countries using linear regression and artificial neural network hybrid models," *Singap. Econ. Rev.*, vol. 67, no. 2, pp. 635–653, 2022.

[5] Q. Xu, L. Wang, C. Jiang, and Y. Liu, "A novel (U)MIDAS-SVR model with multi-source market sentiment for forecasting stock returns," *Neural Comput. Appl.*, vol. 32, no. 10, pp. 5875–5888, May 2020.

[6] Y. Wang, L. Wang, F. Yang, W. Di, and Q. Chang, "Advantages of direct input-to-output connections in neural networks: The Elman network for stock index forecasting," *Inf. Sci.*, vol. 547, pp. 1066–1079, Feb. 2021.

[7] Y. Trichilli, M. B. Abbes, and A. Masmoudi, "Predicting the effect of googling investor sentiment on Islamic stock market returns: A five-state hidden Markov model," *Int. J. Islamic Middle Eastern Finance Manage.*, vol. 13, no. 2, pp. 165–193, Feb. 2020.

[8] H. J. Park, Y. Kim, and H. Y. Kim, "Stock market forecasting using a multi-task approach integrating long short-term memory and the random forest framework," *Appl. Soft Comput.*, vol. 114, Jan. 2022, Art. no. 108106.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–9.

[10] F. Yang, J. Chen, and Y. Liu, "Improved and optimized recurrent neural network based on PSO and its application in stock price prediction," *Soft Comput.*, vol. 27, no. 6, pp. 3461–3476, Mar. 2023.

[11] Y. Chen, J. Wu, and Z. Wu, "China's commercial bank stock price prediction using a novel K-means-LSTM hybrid approach," *Expert Syst. Appl.*, vol. 202, Sep. 2022, Art. no. 117370.

[12] U. Gupta, V. Bhattacharjee, and P. S. Bishnu, "StockNet—GRU based stock index prediction," *Exp. Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 117986.

[13] W. Xu, Z. Fu, H. Li, J. Huang, W. Xu, and Y. Luo, "A study of the impact of COVID-19 on the Chinese stock market based on a new textual multiple ARMA model," *Stat. Anal. Data Min.*, vol. 16, no. 1, pp. 5–15, 2023.

[14] A. S. Ahmar, P. K. Singh, N. Van Thanh, N. V. Tinh, and V. M. Hieu, "Prediction of BRIC stock price using ARIMA, SutteARIMA, and holt-winters," *Comput., Mater. Continua*, vol. 70, no. 1, pp. 523–534, 2022.

[15] M. L. Challa, V. Malepati, and S. N. R. Kolusu, "S&P BSE sensex and S&P BSE IT return forecasting using ARIMA," *Financial Innov.*, vol. 6, no. 1, p. 47, Dec. 2020.

[16] A. Hossain and M. Nasser, "Comparison of the finite mixture of ARMA-GARCH, back propagation neural networks and support-vector machines in forecasting financial returns," *J. Appl. Statist.*, vol. 38, no. 3, pp. 533–551, Mar. 2011.

[17] Y. Guo, S. Han, C. Shen, Y. Li, X. Yin, and Y. Bai, "An adaptive SVR for high-frequency stock price forecasting," *IEEE Access*, vol. 6, pp. 11397–11404, 2018.

[18] R. Ren, D. D. Wu, and T. Liu, "Forecasting stock market movement direction using sentiment analysis and support vector machine," *IEEE Syst. J.*, vol. 13, no. 1, pp. 760–770, Mar. 2019.

[19] M. Liu, K. Luo, J. Zhang, and S. Chen, "A stock selection algorithm hybridizing grey wolf optimizer and support vector regression," *Exp. Syst. Appl.*, vol. 179, Oct. 2021, Art. no. 115078.

[20] D. Zhang and S. Lou, "The application research of neural network and BP algorithm in stock price pattern classification and prediction," *Future Gener. Comput. Syst.*, vol. 115, pp. 872–879, Feb. 2021.

[21] H.-L. Yang and H.-C. Lin, "An integrated model combined ARIMA, EMD with SVR for stock indices forecasting," *Int. J. Artif. Intell. Tools*, vol. 25, no. 2, Apr. 2016, Art. no. 1650005.

[22] I. Lauriola, A. Lavelli, and F. Aiolli, "An introduction to deep learning in natural language processing: Models, techniques, and tools," *Neurocomputing*, vol. 470, pp. 443–456, Jan. 2022.

[23] P. L. T. Mbouembe, G. Liu, J. Sikati, S. C. Kim, and J. H. Kim, "An efficient tomato-detection method based on improved YOLOv4-tiny model in complex environment," *Frontiers Plant Sci.*, vol. 14, Apr. 2023, Art. no. 1150958.

[24] A. Kanwal, M. F. Lau, S. P. H. Ng, K. Y. Sim, and S. Chandrasekaran, "BiCuDNNLSTM-1dCNN—A hybrid deep learning-based predictive model for stock price prediction," *Exp. Syst. Appl.*, vol. 202, Sep. 2022, Art. no. 117123.

[25] B. Gülmez, "Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm," *Exp. Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120346.

[26] C. Qi, J. Ren, and J. Su, "GRU neural network based on CEEMDAN–wavelet for stock price prediction," *Appl. Sci.*, vol. 13, no. 12, p. 7104, Jun. 2023.

[27] Y. Yao, Z.-Y. Zhang, and Y. Zhao, "Stock index forecasting based on multivariate empirical mode decomposition and temporal convolutional networks," *Appl. Soft Comput.*, vol. 142, Jul. 2023, Art. no. 110356.

[28] C. Cui, P. Wang, Y. Li, and Y. Zhang, "McVCsB: A new hybrid deep learning network for stock index prediction," *Exp. Syst. Appl.*, vol. 232, Dec. 2023, Art. no. 120902.

[29] J. Wang, Q. Cheng, and Y. Dong, "An XGBoost-based multivariate deep learning framework for stock index futures price forecasting," *Kybernetes*, vol. 52, no. 10, pp. 4158–4177, Nov. 2023.

[30] M. Jiang, W. Chen, H. Xu, and Y. Liu, "A novel interval dual convolutional neural network method for interval-valued stock price prediction," *Pattern Recognit.*, vol. 145, Jan. 2024, Art. no. 109920.

[31] Y. Liu, S. Huang, X. Tian, F. Zhang, F. Zhao, and C. Zhang, "A stock series prediction model based on variational mode decomposition and dual-channel attention network," *Exp. Syst. Appl.*, vol. 238, Mar. 2024, Art. no. 121708.

[32] M. Lu and X. Xu, "TRNN: An efficient time-series recurrent neural network for stock price prediction," *Inf. Sci.*, vol. 657, Feb. 2024, Art. no. 119951.

[33] E. T. Sivadasan, N. Mohana Sundaram, and R. Santhosh, "Stock market forecasting using deep learning with long short-term memory and gated recurrent unit," *Soft Comput.*, vol. 28, no. 4, pp. 3267–3282, Feb. 2024.

[34] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, "A complete ensemble empirical mode decomposition with adaptive noise," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 4144–4147.

[35] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: A noise-assisted data analysis method," *Adv. Adapt. Data Anal.*, vol. 1, no. 1, pp. 1–41, 2009.

[36] K. P. Sinaga and M.-S. Yang, "Unsupervised K-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80716–80727, 2020.

[37] J. M. Yentes, N. Hunt, K. K. Schmid, J. P. Kaipust, D. McGrath, and N. Stergiou, "The appropriate use of approximate entropy and sample entropy with short data sets," *Ann. Biomed. Eng.*, vol. 41, no. 2, pp. 349–365, Feb. 2013.

[38] F. Liu and Y. Deng, "Determine the number of unknown targets in open world based on elbow method," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 5, pp. 986–995, May 2021.

[39] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 531–544, Feb. 2014.

[40] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[41] Y. Guo, J. Guo, B. Sun, J. Bai, and Y. Chen, "A new decomposition ensemble model for stock price forecasting based on system clustering and particle swarm optimization," *Appl. Soft Comput.*, vol. 130, Nov. 2022, Art. no. 109726.

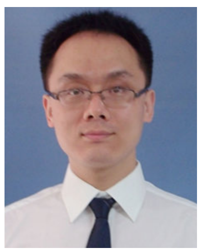
[42] F. Pedregosa, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

YI LI received the bachelor's and master's degrees in finance from Dongbei University of Finance and Economics, China, in 2011 and 2014, respectively, and the Ph.D. degree in business administration from Pusan National University, South Korea, in 2020. She is currently an Assistant Professor with the School of Economics and Management, Weifang University, China. Her current research interests include asset pricing, behavioral finance, and corporate finance.

LEI CHEN received the Ph.D. degree in finance from Shandong University of Finance and Economics, China, in 2018. He is currently an Associate Professor with the School of Economics and Management, Weifang University, China. His research interests include household finance and corporate finance.

CUIPING SUN received the master's degree in technical economy and management from Harbin Institute of Technology, China, in 2010, and the Ph.D. degree in finance from The University of Suwon, South Korea, in 2020. She is currently an Assistant Professor with the School of Economics and Management, Weifang University of Science and Technology, Weifang, China. Her current research interests include corporate finance and household finance.

GUOXU LIU received the M.S. degree in signal processing from Nanyang Technological University, Singapore, in 2014, and the Ph.D. degree in electronics engineering from Pusan National University, South Korea, in 2020. He is currently an Assistant Professor with the School of Computer Engineering, Weifang University, Weifang, China. His research interests include machine learning and deep learning.



CHUNLEI CHEN received the M.Eng. and Ph.D. degrees in pattern recognition and intelligent system from Northwestern Polytechnical University, China, in 2009 and 2014, respectively. He is currently an Associate Professor with the School of Computer Engineering, Weifang University, China. His current research interests include machine learning and parallel computing.

YONGHUI ZHANG received the Ph.D. degree in agricultural informatics from Nanjing Agricultural University, China, in 2013. He is currently an Associate Professor with the School of Computer Engineering, Weifang University, China. His current research interests include data analysis and machine learning.