Applied Soft Computing Journal 148 (2023) 110867

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Multi-step-ahead stock price prediction using recurrent fuzzy neural network and variational mode decomposition

Hamid Nasiri, Mohammad Mehdi Ebadzadeh *

Department of Computer Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran

ARTICLE INFO

Dataset link: https://github.com/Hamid-Nasiri/VMD-MFRFNN

Keywords:
Particle swarm optimization
Recurrent fuzzy neural network
Stock price prediction
Time series prediction
Variational mode decomposition

ABSTRACT

Financial time series prediction has attracted considerable interest from scholars, and several approaches have been developed. Among them, decomposition-based methods have achieved promising results. Most decomposition-based methods approximate a single function, which is insufficient for obtaining accurate results. Moreover, most existing research has concentrated on one-step-ahead forecasting that prevents market investors from making the best decisions for the future. This study proposes two novel multi-step-ahead stock price prediction methods based on different decomposition techniques, including discrete cosine transform (DCT), i.e., a linear transform, and variational mode decomposition (VMD), i.e., a non-linear transform. DCT-MFRFNN, a method based on DCT and multi-functional recurrent fuzzy neural network (MFRFNN), uses DCT to reduce fluctuations in the time series and simplify its structure and MFRFNN to predict the stock price. VMD-MFRFNN, an approach based on VMD and MFRFNN, brings together their advantages. VMD-MFRFNN consists of two phases. The input signal is decomposed to several intrinsic mode functions (IMFs) using VMD in the decomposition phase. In the prediction phase, each IMF is given to a separate MFRFNN for prediction, and predicted signals are summed to reconstruct the output. DCT-MFRFNN and VMD-MFRFNN use the particle swarm optimization (PSO) algorithm to train MFRFNN. In this research, for the first time, the gradient descent method is used to train MFRFNN. Three financial time series are used to evaluate the proposed methods. Experimental results indicate that VMD-MFRFNN surpasses other state-of-the-art methods. VMD-MFRFNN, on average, shows a decrease of 31.8% in RMSE compared to the MEMD-LSTM method. Also, DCT-MFRFNN outperforms MFRFNN and DCT-LSTM in all experiments, which reveals the favorable effect of DCT on MFRFNN's performance. To assess the effectiveness of PSO in training VMD-MFRFNN, we compared its performance with twelve different metaheuristic approaches. PSO, on average, shows a decrease of 9.4% in MAPE compared to other metaheuristic methods.

1. Introduction

Economic globalization, as one of the three main dimensions of globalization, cause the stock market to be highly fluctuated, polymorphic, and complex. Stock price index prediction is challenging due to the influence of several factors, including economic indicators, political situations, and other related aspects [1]. Moreover, stock price forecasting is vital for investors and financial analysts to build investment strategies and get steady returns [2,3]. Hence, extensive studies [4–6] have been done on improving financial time series prediction approaches as an active research topic. A stock price time series is a series of successive observations collected from the stock market over time. The prediction of these time series can be made in one-step or multi-step ahead forecasting. Obviously, a long-term prediction (i.e., multi-step ahead) is more beneficial for stock market investors to arrive at the best decisions and plans for the future. Nevertheless, most existing stock price prediction researches [7–10] only address one-step ahead forecasting. Therefore, there is a need to investigate multi-step ahead prediction of stock prices [3].

Another issue that requires attention is that financial time series are usually dynamic, non-linear, extremely fluctuated, and complex, which makes their prediction challenging [10–12]. To address this issue, decomposition-based methods have been proposed. The main idea behind decomposition-based methods is to decompose complex time series into a number of components with a simpler structure, i.e., reducing time series complexity. Since these components have simpler structures and more stationary trends, their prediction is easier, and the expected accuracy is higher [13].

Discrete cosine transform (DCT) is a linear orthogonal transformation that transforms a signal from the spatial domain to the frequency

* Corresponding author.
E-mail addresses: h.nasiri@aut.ac.ir (H. Nasiri), ebadzadeh@aut.ac.ir (M.M. Ebadzadeh).

https://doi.org/10.1016/j.asoc.2023.110867
Received 30 May 2023; Received in revised form 8 August 2023; Accepted 15 September 2023
Available online 22 September 2023

domain. The DCT decomposes a signal into a set of cosine functions with different frequencies. Although many researchers have used DCT for time series prediction, it introduces some distortion to the signal and is not a universal transform. Therefore, other decomposition methods, such as empirical mode decomposition (EMD) and variational mode decomposition (VMD), were proposed.

EMD, an adaptive and data-driven decomposition method, was proposed by Huang et al. [14] in 1998. EMD decomposes a time series into some intrinsic mode functions (IMFs) and a monotonic residual signal in an iterative process [15]. So, it can simply isolate highly fluctuating data into lower frequency components [10]. Although many researchers used EMD for financial time series prediction, it has many inherent problems such as mode mixing, lack of mathematical foundations, sensitivity to noise and sampling, and choice of the interpolation method [8,16–18]. To address these shortcomings, VMD [16] was proposed by Dragomiretskiy and Zosso in 2014. VMD is a completely non-recursive decomposition method based on a solid mathematical framework. It decomposes a time series into several eigenmode components with a specific bandwidth in the spectral domain by solving a convex optimization problem. These band-limited components reconstruct the input time series exactly or in the least squares sense. In other words, VMD is capable of separating tones of similar frequencies and avoiding the mode mixing problem [2,11,16].

Although in recent years, many researchers have developed decomposition-based methods using VMD instead of EMD, as far as the authors' knowledge goes, all of them approximate a single function, i.e., produce a specific output determined by the current and previous inputs. As described in [19], in challenging nonlinear problems, different outputs might be produced for a certain input depending on the state of the system. Therefore, to obtain accurate results in financial time series prediction, a method is needed that can identify the system's states and approximate a specific function for each of them, i.e., it should be able to simultaneously learn multiple functions. Multi-functional recurrent fuzzy neural network (MFRFNN) [19] proposed by Nasiri and Ebadzadeh in 2022 can learn multiple functions simultaneously. In light of the issues outlined, two novel methods for financial time series prediction have been proposed in this study: DCT-MFRFNN and VMD-MFRFNN.

DCT-MFRFNN is based on discrete cosine transform (DCT) and MFRFNN. It applies DCT to the time series and computes DCT coefficients. A sequence of high-frequency DCT coefficients is dropped to eliminate high frequencies, reduce fluctuations in the time series, and simplify its structure. Then the time series is reconstructed using only the remaining coefficients. Finally, the reconstructed signal is given to the MFRFNN model for prediction.

VMD-MFRFNN is an approach based on VMD and MFRFNN and brings together their advantages. It consists of two phases: (a) decomposition phase and (b) prediction and reconstruction phase. The input time series is decomposed to several IMFs using VMD in the decomposition phase. In the prediction and reconstruction phase, first, each IMF is given to a separate MFRFNN model for prediction. Then the predicted signals, i.e., the output of each MFRFNN, are summed to reconstruct the final output. The following are the main contributions of this research:

1. DCT-MFRFNN and VMD-MFRFNN, two novel financial time series prediction methods, are proposed. DCT-MFRFNN first applies DCT to the input signal and drops a sequence of high-frequency DCT coefficients. Then the input time series is reconstructed using the remaining coefficients. Finally, the MFRFNN model uses the reconstructed signal for prediction. VMD-MFRFNN first decomposes the input signal into several IMFs and then predicts each by a separate MFRFNN model. Finally, the outputs of MFRFNNs are summed to generate the final prediction.
2. Developing new training algorithms for DCT-MFRFNN and VMD-MFRFNN.
3. For the very first time, the gradient descent method is used to train MFRFNN.
4. The prediction performance of DCT-MFRFNN and VMD-MFRFNN is evaluated using three financial time series, including Hang Seng Index (HSI), Shanghai Stock Exchange (SSE), and Standard & Poor's 500 Index (SPX). The experimental results indicate that VMD-MFRFNN shows promising performance in multi-step-ahead prediction compared with other existing methods. VMD-MFRFNN, on average, shows 35.93%, 24.88%, and 34.59% decreases in RMSE from the second-best model for HSI, SSE, and SPX, respectively. Also, DCT-MFRFNN outperforms DCT-LSTM and MFRFNN in all experiments.
5. The effectiveness of particle swarm optimization (PSO) in training VMD-MFRFNN is assessed by comparing its performance with twelve different metaheuristic algorithms. The results demonstrate the merits of PSO in training VMD-MFRFNN.

The remainder of the paper is organized as follows: Section 2 outlines an overview of the related works. In Section 3, DCT, VMD, and MFRFNN are briefly introduced, with Section 4 presenting the DCT-MFRFNN and VMD-MFRFNN. Section 5 evaluates the effectiveness of the proposed methods and provides results and discussion. Finally, this study is concluded with a summary in Section 6.

2. Literature review

This section outlines an overview of the literature that relates to this study. As a vital problem for investors and financial analysts, financial time series prediction has been extensively studied in recent years [2,3,8–11,13,19–23]. Several researchers have developed decomposition-based methods using empirical mode decomposition (EMD) for time series forecasting. Zhou et al. [20] proposed EMD2FNN, a hybrid method composed of EMD and factorization machine-based neural networks (FMNN) for predicting stock market trends. EMD2FNN brings together the benefits of EMD and FMNN. Yang et al. [21] developed a hybrid method for time series prediction based on recursive EMD and long short-term memory (LSTM). They proposed a novel recursive EMD to solve the mode mixing problem of EMD and used LSTM to predict the IMFs. More recent work by Deng et al. [3] introduced a hybrid modeling framework based on multivariate EMD (MEMD) and LSTM for stock price index forecasting. In this framework, first, MEMD was used to decompose relevant features of the time series. Then, the extracted features were used to train LSTM for the multi-step-ahead prediction task. Moreover, the orthogonal array tuning approach was employed to tune the LSTM's hyperparameters. Deng et al. evaluated their proposed method on three real-world financial time series and obtained promising results.

Based on previous studies [8,17,18], EMD has many inherent problems. To overcome these issues, ensemble empirical mode decomposition (EEMD) [24] was proposed by Wu and Huang in 2009. EEMD is a self-adaptive decomposition method that eliminates the mode mixing problem of EMD by adding white noise [13,22]. Rezaei et al. [10] developed CEEMD-CNN-LSTM, a hybrid method using complete EEMD (CEEMD), convolutional neural network, and LSTM. Their proposed method can extract deep features and time sequences. Although CEEMD-CNN-LSTM achieved encouraging results in the one-step-ahead stock price prediction, it has not been evaluated in multi-step-ahead prediction tasks. In 2021, Lin et al. [9] proposed a hybrid approach combining CEEMD with adaptive noise (CEEMDAN) and LSTM. They assessed their proposed method on SPX and Chinese Securities 300 Index time series. Despite overcoming the problems associated with EMD, EEMD suffers from incompleteness, reconstruction error, high computational complexity, and predicament of parameters selection [9,25].

Variational mode decomposition (VMD) [16] was developed in 2014 to address EMD's shortcomings. Due to its merits, it has gained increasing popularity in recent years, and there have been several attempts to develop decomposition-based methods using VMD. Lahmiri [26] introduced a novel prediction model by integrating VMD and general regression neural network. VMD-LSTM [27], a hybrid technique based on VMD and LSTM, was developed by Niu et al. In another paper, Tang et al. [22] proposed a novel hybrid approach by combining VMD, EEMD, and extreme learning machine (ELM). Their proposed method first applies VMD to the time series and then applies EEMD to the residual signal obtained by VMD. They used ELM to predict the stock price index and differential evolution to optimize its weights. Same as [27], Yujun et al. [11] developed a prediction model based on VMD and LSTM for stock price prediction. A more recent paper by Wang et al. [28] proposed a novel hybrid method based on secondary decomposition (SD), multi-factor analysis, and attention-based LSTM. In this method, SD uses improved VMD and improved CEEMDAN to filter the noise effectively and capture additional nonlinear features. Attention layer was added to LSTM to increase the weights of important data instances and improve LSTM's performance.

VML [2] was developed by Liu et al. in 2022. VML integrates meta-learning and VMD. It uses VMD to decompose time series and applies model-agnostic meta-learning and LSTM to predict each component. Liu et al. used the Shanghai Stock Exchange index and Dow Jones Industrial Average to evaluate the VML. Wan et al. [29] developed a hybrid model combining the gated recurrent unit (GRU) with an autoregressive integrated moving average (ARIMA) model. The proposed model uses VMD to decompose the original time series. It employs GRU to predict the future values of each IMF. Furthermore, the proposed approach utilizes a combination of VMD and ARIMA to correct the error subseries. Guo et al. [8] introduced a hybrid method based on VMD, autoregressive integrated moving average (ARIMA), and Taylor expansion forecasting (TEF). They used VMD to decompose time series, ARIMA to forecast the linear component of each IMF, and TEF to predict the nonlinear component. Zhang and Chen [30] proposed a two-stage forecasting model that uses VMD to decompose time series and three separate methods, i.e., support vector regression, ELM, and deep neural network, to forecast decomposed signal. They used an ELM-based nonlinear ensemble approach to combine initial stock price forecasts in the second stage of the algorithm.

Recent years have seen many researchers shifting towards decomposition-based methods employing VMD rather than EMD. However, to the best of the authors' knowledge, these methods are all designed to approximate a single function, meaning they generate a specific output based on current and past observations. Nevertheless, in complex nonlinear problems, the same input could lead to different outputs depending on the system's state. Consequently, to achieve precise predictions for financial time series, there is a need for a method that can recognize the states of the system and approximate a unique function for each. Considering these concerns, this study develops two forecasting methods for financial time series: DCT-MFRFNN and VMD-MFRFNN.

3. Preliminaries

This section gives a concise overview of discrete cosine transform (DCT), variational mode decomposition (VMD), and multi-functional recurrent fuzzy neural network (MFRFNN) methods to provide further insight into the proposed method.

3.1. Discrete cosine transform

DCT is a time–frequency linear orthogonal transformation that decreases redundant calculations in the discrete Fourier transform [31]. It represents a finite sequence of samples as a sum of cosine functions with different frequencies. The DCT coefficients ($y(k)$) of a discrete signal ($x(n)$) can be calculated as follows [32]:

$$y(k) = \alpha(k) \sum_{n=0}^{N-1} x(n) \cos\left(\frac{\pi(2n+1)k}{2N}\right), k = 0, 1, \dots, N-1 \tag{1}$$

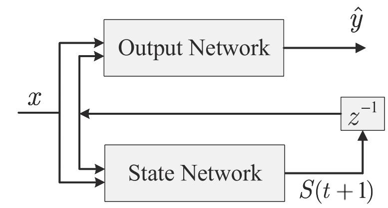


Fig. 1. Overview of MFRFNN [19].

where N is the signal length, and $\alpha(k)$ denotes the scaling factor that is determined by Eq. (2):

$$\alpha(k) = \begin{cases} \sqrt{1/N} & k = 0 \\ \sqrt{2/N} & 1 \le k \le N-1 \end{cases} \tag{2}$$

3.2. Variational mode decomposition

VMD, developed by Dragomiretskiy and Zosso [16], is an adaptive decomposition method based on a solid mathematical framework [33]. It decomposes a complex signal into K IMFs with a specific bandwidth in the spectral domain by solving the variational problem [34,35]. Let f and t denote the input signal and time step, respectively. IMFs ($u_k(t)$) and their corresponding center frequency (ω_k) can be obtained by solving the following constrained optimization problem:

$$\begin{aligned} &\underset{\{u_k\},\{\omega_k\}}{\text{minimize}} \quad \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \\ &\text{subject to} \quad \sum_{k=1}^{K} u_k = f \end{aligned} \tag{3}$$

where $\delta(\cdot)$ represents the Dirac delta function and ∂_t denotes the partial derivative with respect to t [36,37].

To ensure signal reconstruction's accuracy under Gaussian noise, a quadratic penalty term is added to Eq. (3), and to convert it into an unconstrained optimization problem, a Lagrange multiplier is applied as follows [36,38]:

$$\begin{aligned} L\left(\{u_k\},\{\omega_k\},\lambda\right) = \alpha \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \\ + \left\| f(t) - \sum_{k=1}^{K} u_k(t) \right\|_2^2 \\ + \left\langle \lambda(t), f(t) - \sum_{k=1}^{K} u_k(t) \right\rangle \end{aligned} \tag{4}$$

where λ and α denote the Lagrange multiplier and penalty parameter, respectively. Finally, by solving Eq. (4) iteratively using the Alternating Direction Method of Multipliers, the input signal is decomposed into K IMFs [16,36].

3.3. Multi-functional recurrent fuzzy neural network

MFRFNN, proposed by Nasiri and Ebadzadeh [19] in 2022, is a recurrent fuzzy neural network for time series forecasting. It comprises two fuzzy neural networks (FNNs). One of them, i.e., the output network, generates the system's output, whereas the other, i.e., the state network, identifies the state of the system (Fig. 1). MFRFNN uses the least squares algorithm and particle swarm optimization (PSO) technique to learn the weights of output and state networks, respectively. This network can learn and memorize historical data from previous time series samples and approximate several functions simultaneously by incorporating the states in its structure [19].

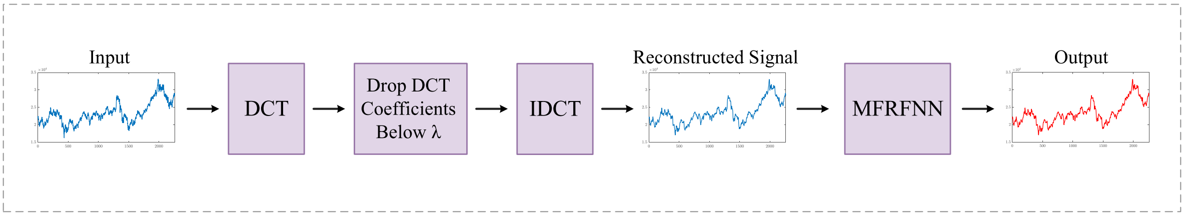


Fig. 2. Structure of DCT-MFRFNN.

4. Proposed methods

This section presents a detailed description of the proposed methods for financial time series prediction.

4.1. DCT-MFRFNN

The proposed DCT-MFRFNN method consists of the following steps: (1) The DCT is applied to the input time series, and DCT coefficients ($\mathbf{c}$) are computed. (2) λ percent of high-frequency DCT coefficients are set to zero. (3) The inverse DCT is applied to the remaining coefficients, and the input signal is reconstructed. (4) The reconstructed signal ($\hat{\mathbf{x}}$) is given to the MFRFNN as input for training. (5) The MFRFNN generates the final prediction ($\hat{\mathbf{y}}$). Fig. 2 shows the structure of DCT-MFRFNN.

In DCT-MFRFNN, a sequence of high-frequency DCT coefficients is dropped to eliminate high frequencies and reduce fluctuations in the time series. As a result, a time series with a simpler structure and more stationary trends is given to MFRFNN for prediction. Intuitively, since this time series has a simpler structure, its prediction is easier for the model. Note that DCT tends to concentrate most of the energy of a time series into a few low-frequency components. This energy compaction property implies that a relatively small number of DCT coefficients can reasonably represent the time series [39]. Moreover, the DCT can transform a time series into its frequency domain representation. Applying the DCT results in obtaining the spectrum of the time series, which reveals the dominant frequencies present in the data. This can help the DCT-MFRFNN in forecasting the time series. Also, the DCT has the property of decorrelating the data, meaning that it transforms the time series into uncorrelated frequency components. By decorrelating the data, the DCT can potentially help capture the underlying patterns more efficiently. Another advantage of using DCT in DCT-MFRFNN is its noise reduction capability. When time series data contains noise or high-frequency components that are irrelevant to the underlying patterns, applying the DCT separates the signal into different frequency components and allows the model to discard or attenuate the high-frequency components associated with noise. This denoising capability can improve prediction accuracy by eliminating irrelevant information. DCT-MFRFNN training algorithm is shown in Algorithm 1.

Algorithm 1: DCT-MFRFNN training algorithm

Input: Input time series $D = \left\{x^{[t]}, x^{[t+h]}\right\}_{t=1}^{p}, \lambda$
Output: Predicted signal ($\hat{\mathbf{y}}$)

Apply DCT to $\mathbf{x}$ to obtain $\mathbf{c} \in \mathbb{R}^{p\times 1}$
$n \leftarrow \left\lfloor p \times \frac{100-\lambda}{100} \right\rfloor + 1$
for $i \leftarrow n$ **to** p **do**
  $c_i \leftarrow 0$
end
Apply IDCT to $\mathbf{c}$ to obtain $\hat{\mathbf{x}}$
Train MFRFNN using $\hat{\mathbf{x}}$ as input
Predict $\hat{\mathbf{y}}$ using MFRFNN model

4.2. VMD-MFRFNN

VMD-MFRFNN consists of two phases: (a) decomposition phase and (b) prediction and reconstruction phase. In the decomposition phase, the input time series is decomposed into several IMFs using the VMD method. In the prediction and reconstruction phase, first, each of IMFs is given to a separate MFRFNN model for training. Then, each model generates its predicted signal, i.e., the output of each MFRFNN. Finally, the predicted signals are summed to reconstruct the final prediction. Fig. 3 illustrates the structure of VMD-MFRFNN. A detailed description of VMD-MFRFNN's phases is given below.

Decomposition Phase: In this phase, the input signal is decomposed into K IMFs using VMD. Let $D = \left\{x^{[t]}, x^{[t+h]}\right\}_{t=1}^{p}$ represent the input time series, where h and p denote the prediction horizon and number of samples, respectively. VMD decomposes $\mathbf{x}$ to K IMFs by taking K as input. Obviously, these components have simpler structures, so the decomposition phase allows the proposed method to predict simpler structures with more stationary trends and achieve higher accuracy. Keep in mind that each IMF represents a different frequency component or temporal scale. This allows the VMD-MFRFNN to separate the time series into its constituent parts, making it easier to analyze and understand the underlying patterns. By isolating different IMFs, the model can focus on specific components more relevant to the forecasting task. Moreover, VMD is well-suited for handling nonlinear and non-stationary time series. It does not impose any assumptions about the linearity or stationarity of the data, making it applicable to a wide range of time series with complex and varying patterns. This flexibility can be valuable in prediction tasks involving financial time series data, where nonlinearities and non-stationarities are often encountered. Also, each IMF obtained from the decomposition represents a distinct oscillatory behavior or trend within the data. These IMFs can capture different characteristics, such as low-frequency trends or high-frequency fluctuations. By capturing these characteristics, VMD-MFRFNN can uncover relevant information to enhance its performance. Another advantage of using VMD in VMD-MFRFNN is its ability in noise reduction. VMD can separate noise or unwanted high-frequency components from the primary time series. This can increase the accuracy of the VMD-MFRFNN in prediction tasks by removing irrelevant information. To clarify the proposed algorithm, in what follows, we denote the ith IMF by $\mathbf{imf}_i = \left[imf_i^{[1]}, imf_i^{[2]}, \ldots, imf_i^{[p]}\right]$. Fig. 4 shows an example of VMD decomposition of the Shanghai Stock Exchange (SSE) time series. As seen, the VMD extracts IMFs in the order from higher to lower frequencies.

Prediction & Reconstruction Phase: Each of the IMFs in this phase is given to a separate MFRFNN as input for training the models. In other words, K MFRFNNs are trained in this phase. The output of each MFRFNN is a component of the predicted signal. By summing these components, the final prediction is reconstructed. Employing the MFRFNN models allows the proposed method to learn and memorize historical data from previous observations, capture the dynamic characteristics of the financial time series, and approximate several functions simultaneously. Formally, let $MFRFNN_i$ represent the ith MFRFNN model, and $\hat{\mathbf{y}}_i = \left[\hat{y}_i^{[1]}, \hat{y}_i^{[2]}, \ldots, \hat{y}_i^{[p]}\right]$ denote the predicted signal by $MFRFNN_i$. The final prediction ($\hat{\mathbf{y}}$) can be computed as follows:

$$\hat{\mathbf{y}} = \sum_{i=1}^{K} \left[\hat{y}_i^{[1]}, \hat{y}_i^{[2]}, \ldots, \hat{y}_i^{[p]}\right] \tag{5}$$

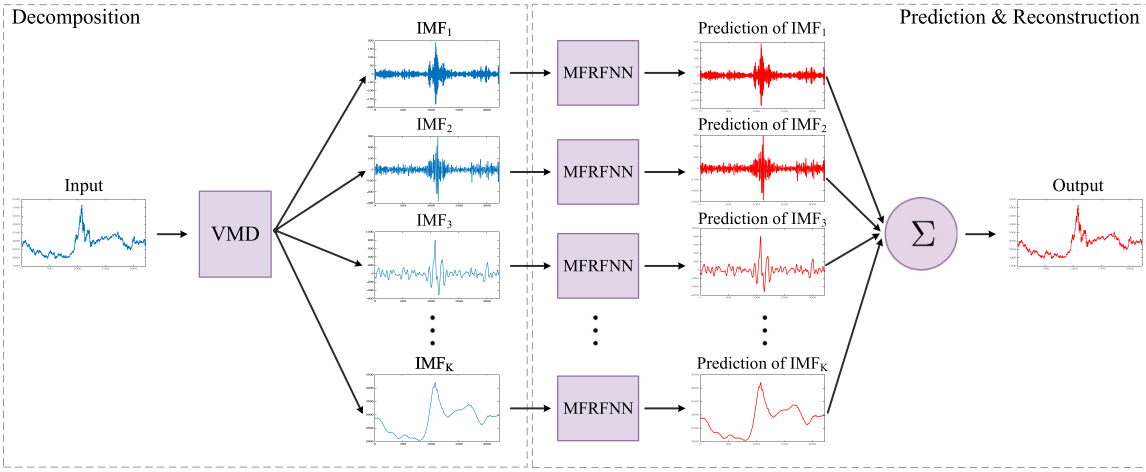


Fig. 3. Structure of VMD-MFRFNN.

VMD-MFRFNN training algorithm is shown in Algorithm 2. Assuming that the number of output and state networks' fuzzy rules are equal in all MFRFNNs, the number of VMD-MFRFNN's trainable parameters is $(K_1 + K_2) \times N \times K$, where K_1 and K_2 represent the number of output and state networks' fuzzy rules, respectively, and N is the number of states of each MFRFNN.

Algorithm 2: VMD-MFRFNN training algorithm

Input: Input time series $D = \{x^{[t]}, x^{[t+h]}\}_{t=1}^{p}, K$
Output: Predicted signal $(\hat{\mathbf{y}})$

Decompose $\mathbf{x}$ to obtain $\{\mathbf{imf}_i\}_{i=1}^{K}$
for $i \leftarrow 1$ **to** K **do**
  Train $MFRFNN_i$ using $\mathbf{imf}_i$ as input
  Predict $\hat{\mathbf{y}}_i$ using $MFRFNN_i$ model
end

$\hat{\mathbf{y}} \leftarrow \sum_{i=1}^{K} \left[\hat{y}_i^{[1]}, \hat{y}_i^{[2]}, \ldots, \hat{y}_i^{[p]}\right]$

5. Results and discussion

In this section, the prediction performance of DCT-MFRFNN and VMD-MFRFNN is evaluated on three financial time series, including Hang Seng Index (HSI), Shanghai Stock Exchange (SSE), and Standard & Poor's 500 Index (SPX). All time series were collected from Yahoo Finance. Table 1 summarizes the statistical description of these time series. The prediction horizons were considered one, three, and five.

Two evaluation metrics were used to assess the performance of the proposed methods and compare them with other methods. Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE). The MAPE measures how accurate the prediction method fits the data and can be computed as follows:

$$\text{MAPE} = \frac{1}{p} \sum_{t=1}^{p} \left| \frac{y_t - \hat{y}_t}{y_t} \right| \tag{6}$$

where y_t, $\hat{y}_t$, and p represent target output, predicted output, and the number of samples, respectively. The RMSE is able to quantify the derivations between the predicted and target values. Overall results are more important to RMSE. It effectively penalizes significant errors and is expressed by (7):

$$\text{RMSE} = \sqrt{\frac{1}{p} \sum_{t=1}^{p} \left(y_t - \hat{y}_t\right)^2} \tag{7}$$

Table 1
The statistical description of financial time series.

Time series	HSI	SSE	SPX
#Samples	2261	2218	2769
Min	16 250.27	1950.01	1022.58
Max	33 154.12	5166.35	3756.07
Mean	23 418.23	2798.88	2077.30
STD	3097.44	550.70	674.72

The parameters of the proposed methods were selected by a process of trial and error using the validation set in each time series. Table 2 lists the chosen parameters for each time series. As mentioned, the VMD extracts IMFs in the order from higher to lower frequencies. Therefore, intuitively, the number of output network's fuzzy rules for the first several MFRFNNs should be higher than the last several ones. We used triangular membership functions for the proposed methods' input layer. For the PSO algorithm, the inertia weight was selected adaptively in the range $[0.1, 1.1]$, and the cognitive and social acceleration coefficients were set to 1.49 in all experiments.

To evaluate the effectiveness of DCT-MFRFNN and VMD-MFRFNN in the time series prediction task, we compared their results with nine other methods, including ARIMA and generalized autoregressive conditional heteroskedastic (GARCH) as two statistical models, feedforward neural network (FFNN) as an example of traditional machine learning models, Prophet [40] as an additive model, LSTM as a deep learning method, EMD-LSTM and DCT-LSTM as two decomposition-based methods, and MEMD-LSTM [3] and MFRFNN [19] as two state-of-the-art models proposed in 2022. In all experiments, we executed twenty independent runs for DCT-LSTM, MFRFNN, DCT-MFRFNN, and VMD-MFRFNN and reported the averages and standard deviations of the results over these runs.

5.1. Hang Seng Index (HSI)

The Hang Seng Index (HSI) is a stock market index in Hong Kong. It is the primary indicator of the performance of the Hong Kong stock market as a whole and is utilized to track daily changes in the top companies listed on the Hong Kong stock exchange. This index is an example of relatively developed markets with active transactions [3]. The HSI data was collected from Yahoo Finance website during a nine-year period from 4-January-2010 to 8-March-2019. The training, validation, and test set in this experiment consist of 1509, 376, and 376 samples, respectively.

Table 3 presents the one, three, and five-step-ahead prediction results of the HSI. The results showed better performance of VMD-MFRFNN compared to other methods in one and three-step-ahead

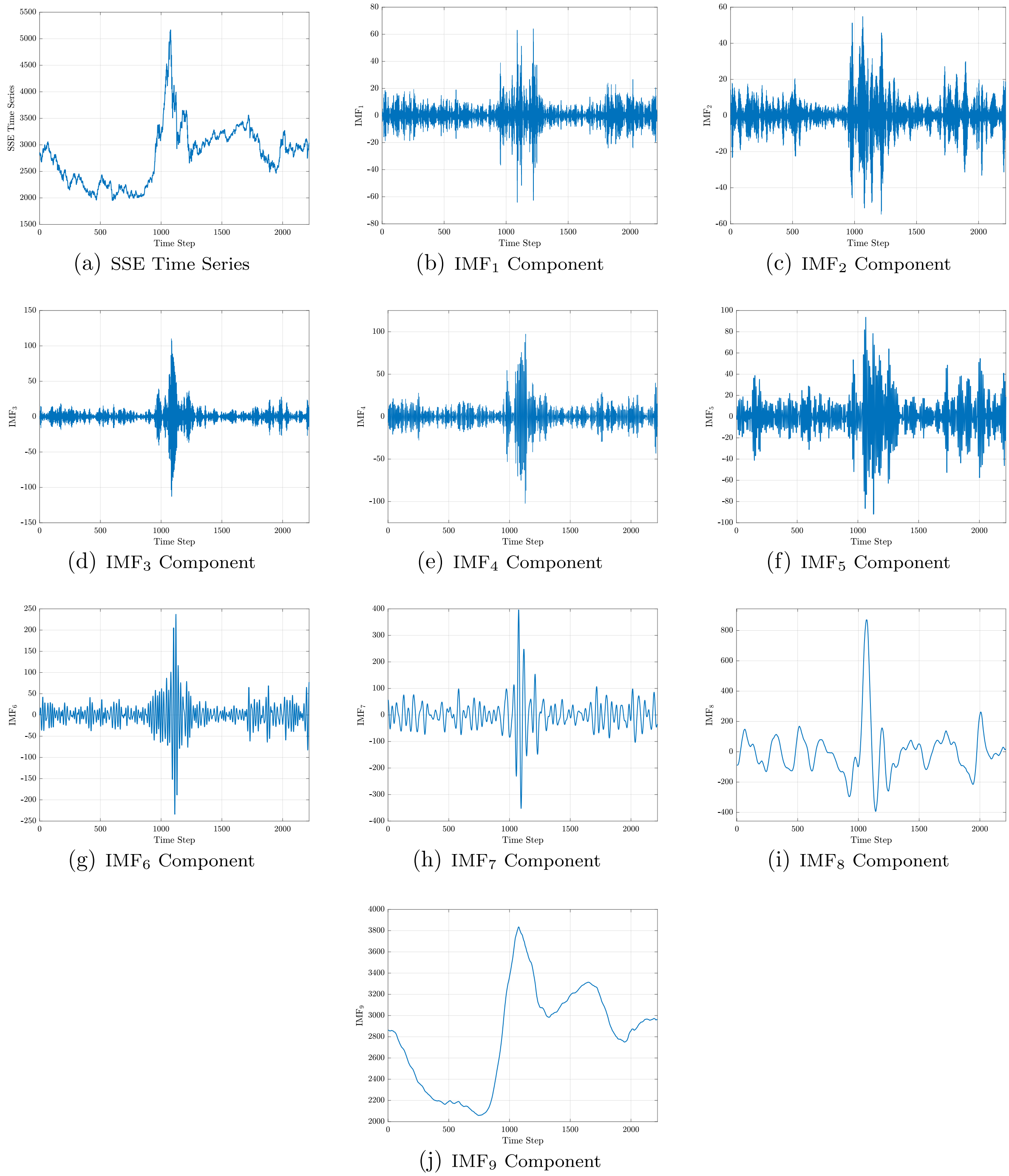


Fig. 4. VMD decomposition of the Shanghai Stock Exchange time series.

Table 2
The parameters of the proposed methods in each financial time series.

Time series	HSI	SSE	SPX
Number of output network's fuzzy rules [$\mathrm{MFRFNN}_1, \mathrm{MFRFNN}_2, \ldots, \mathrm{MFRFNN}_9$]	[6 6 6 5 5 5 4 4 4]	[4 4 4 3 3 3 2 2 2]	[4 4 4 3 3 3 2 2 2]
Number of state network's fuzzy rules (State network)	2	4	2
Number of states	2	2	2
Maximum number of FES (PSO method)	600	800	600
Number of IMFs	9	9	9
λ	80	85	86

Table 3
One, three, and five-step-ahead prediction results of the HSI.

Method	Step	MAPE (%)	p-value	RMSE	p-value
ARIMA	1	3.40E+00	2.72E−38	7.30E+02	8.78E−33
	3	4.24E+00	3.67E−45	7.84E+02	1.72E−36
	5	5.20E+00	4.59E−45	8.16E+02	2.70E−37
GARCH	1	7.63E+00	2.04E−45	3.58E+02	4.76E−25
	3	7.86E+00	8.85E−51	6.01E+02	2.01E−33
	5	8.23E+00	1.83E−49	9.25E+02	5.90E−39
FFNN	1	1.93E+00	5.98E−33	6.21E+02	3.67E−31
	3	2.18E+00	1.77E−38	6.37E+02	4.01E−34
	5	2.61E+00	1.85E−37	6.51E+02	6.71E−34
Prophet	1	5.95E+00	2.90E−43	2.18E+03	8.18E−43
	3	6.09E+00	1.69E−48	2.24E+03	9.21E−47
	5	6.22E+00	8.05E−47	2.31E+03	7.58E−49
LSTM	1	1.20E+00	5.49E−28	4.26E+02	4.14E−27
	3	1.43E+00	1.75E−33	4.37E+02	3.72E−29
	5	1.53E+00	3.12E−29	4.43E+02	1.96E−25
EMD-LSTM	1	6.21E−01	2.98E−19	3.51E+02	8.36E−25
	3	7.93E−01	2.26E−23	3.66E+02	2.70E−26
	5	8.44E−01	1.14E−09	3.79E+02	7.35E−19
MEMD-LSTM	1	5.12E−01	1.12E−15	3.24E+02	8.70E−24
	3	5.87E−01	3.12E−11	3.32E+02	1.80E−24
	5	**6.27E−01**	2.62E−22	3.38E+02	4.04E−07
DCT-LSTM	1	1.42E+00 (1.70E−02)	1.79E−38	6.97E+02 (7.74E+00)	5.77E−45
	3	1.99E+00 (9.43E−02)	1.14E−25	7.12E+02 (2.61E+01)	2.79E−30
	5	2.44E+00 (3.76E−02)	1.50E−46	8.67E+02 (9.25E+00)	5.60E−55
MFRFNN	1	8.59E−01 (1.79E−03)	4.69E−24	3.24E+02 (4.88E−01)	7.95E−24
	3	1.57E+00 (9.94E−04)	8.03E−35	5.65E+02 (4.68E−01)	8.37E−33
	5	2.06E+00 (1.14E−03)	2.13E−34	7.50E+02 (6.88E−01)	9.84E−37
DCT-MFRFNN	1	8.14E−01 (1.65E−03)	2.49E−23	3.18E+02 (1.79E+00)	4.65E−24
	3	1.31E+00 (3.31E−02)	1.18E−38	4.71E+02 (1.26E+01)	3.80E−40
	5	1.96E+00 (2.10E−02)	1.31E−54	6.89E+02 (1.00E+01)	1.44E−46
VMD-MFRFNN	1	**3.16E−01 (3.65E−02)**	–	**1.19E+02 (1.41E+01)**	–
	3	**5.29E−01 (1.91E−02)**	–	**1.95E+02 (8.67E+00)**	–
	5	8.99E−01 (2.24E−02)	–	**3.27E+02 (6.53E+00)**	–

Table 4
One, three, and five-step-ahead prediction results of the SSE.

Method	Step	MAPE (%)	p-value	RMSE	p-value
ARIMA	1	3.68E+00	6.49E−43	4.24E+02	8.34E−46
	3	4.36E+00	1.68E−54	4.51E+02	2.44E−65
	5	5.02E+00	1.31E−39	5.22E+02	1.49E−47
GARCH	1	7.47E+00	4.33E−49	4.03E+01	1.21E−23
	3	7.77E+00	9.02E−60	6.31E+01	3.37E−46
	5	8.18E+00	4.02E−44	9.56E+01	2.42E−31
FFNN	1	2.01E+00	2.40E−37	1.78E+02	2.69E−38
	3	2.23E+00	9.36E−48	2.15E+02	8.94E−59
	5	2.24E+00	3.55E−31	2.28E+02	3.86E−40
Prophet	1	6.06E+00	2.74E−47	2.04E+02	1.69E−39
	3	6.27E+00	7.51E−58	2.11E+02	1.33E−58
	5	6.49E+00	5.14E−42	2.17E+02	1.12E−39
LSTM	1	1.04E+00	1.40E−30	5.49E+01	4.28E−27
	3	1.38E+00	5.88E−42	5.98E+01	1.58E−45
	5	1.40E+00	8.08E−25	6.56E+01	1.10E−26
EMD-LSTM	1	5.79E−01	5.97E−23	4.30E+01	2.12E−24
	3	6.81E−01	6.13E−27	4.64E+01	4.69E−42
	5	7.24E−01	1.57E−04	4.92E+01	2.82E−22
MEMD-LSTM	1	3.81E−01	2.61E−14	2.18E+01	7.34E−15
	3	**4.22E−01**	3.07E−26	2.70E+01	2.34E−30
	5	**4.92E−01**	8.14E−14	2.84E+01	3.39E−05
DCT-LSTM	1	1.24E+00 (2.44E−02)	1.33E−51	5.35E+01 (9.19E−01)	2.55E−34
	3	1.49E+00 (1.93E−01)	5.98E−15	5.56E+01 (6.15E+00)	4.09E−16
	5	1.73E+00 (1.53E−01)	2.93E−19	6.30E+01 (5.39E+00)	5.69E−20
MFRFNN	1	8.86E−01 (7.69E−03)	3.03E−33	3.67E+01 (2.38E−01)	5.91E−23
	3	1.63E+01 (6.45E−04)	7.02E−45	6.32E+01 (9.95E−02)	9.80E−65
	5	2.21E+00 (6.38E−03)	5.21E−32	8.44E+01 (8.37E−02)	5.58E−30
DCT-MFRFNN	1	6.76E−01 (9.72E−03)	5.62E−31	2.98E+01 (1.00E+00)	5.14E−25
	3	1.10E+00 (1.86E−02)	1.37E−34	4.23E+01 (6.64E−01)	4.34E−34
	5	1.46E+00 (2.56E−02)	3.70E−35	5.53E+01 (1.13E+00)	2.98E−33
VMD-MFRFNN	1	**2.77E−01 (2.30E−02)**	–	**1.23E+01 (1.96E+00)**	–
	3	5.46E−01 (6.33E−03)	–	**2.08E+01 (1.92E−01)**	–
	5	6.78E−01 (4.38E−02)	–	**2.61E+01 (1.91E+00)**	–

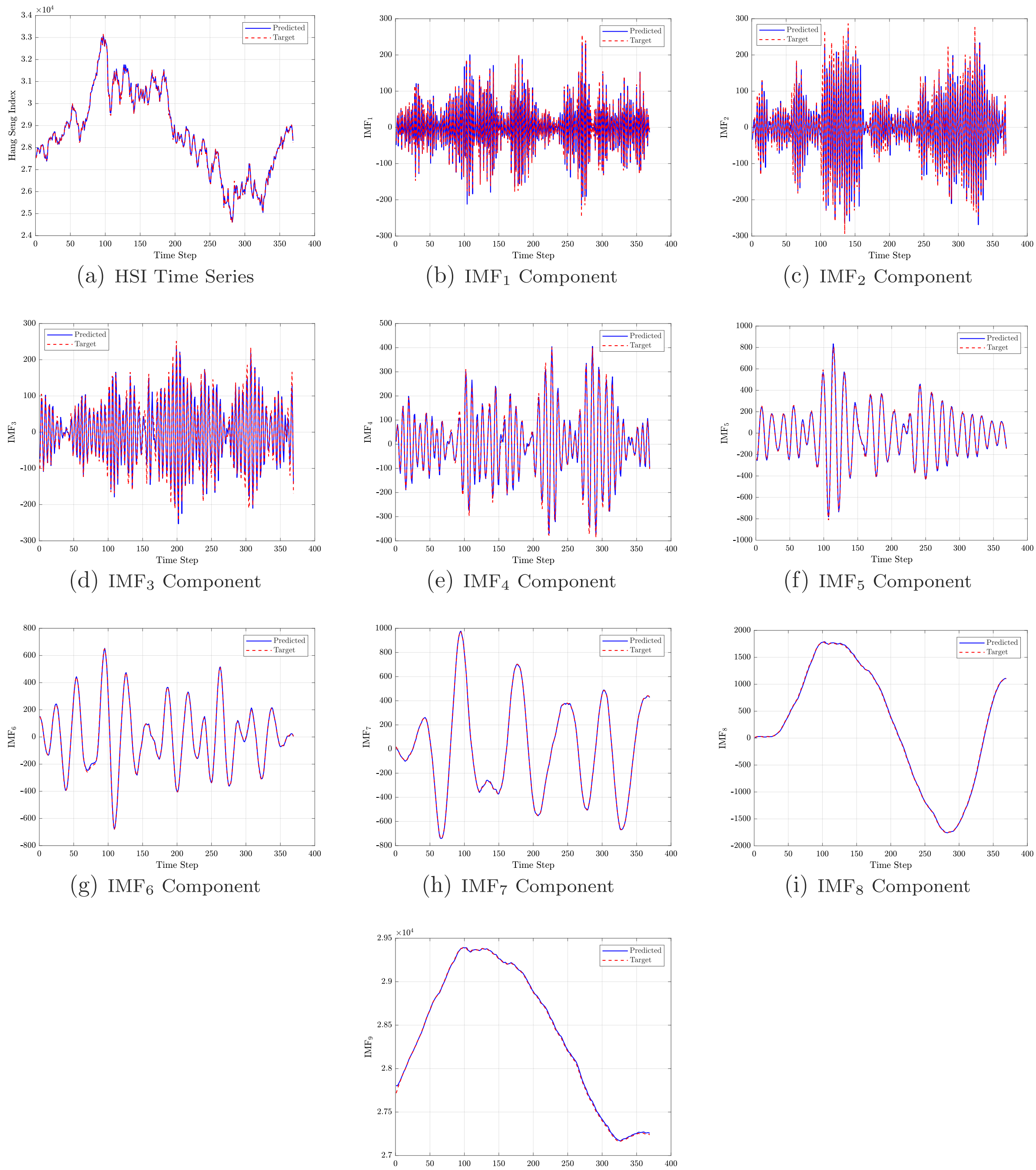
predictions of HSI time series, closely followed by MEMD-LSTM. In five-step-ahead prediction based on MAPE, MEMD-LSTM outperformed other methods, and after that, VMD-MFRFNN had the lowest MAPE. Based on RMSE, VMD-MFRFNN had the highest accuracy in five-step-ahead prediction, closely followed by MEMD-LSTM. Based on this metric, VMD-MFRFNN showed a decrease of 63.27%, 41.27%, and 3.25% from the MEMD-LSTM in one, three, and five-step-ahead forecasting tasks, respectively.

GARCH and Prophet had the lowest accuracy in all experiments based on MAPE and RMSE, respectively. Moreover, the results illustrated in Table 3 indicate that DCT-MFRFNN outperformed DCT-LSTM and MFRFNN in all prediction horizons. Based on RMSE, DCT-MFRFNN showed a decrease of 1.85%, 16.64%, and 8.13% from MFRFNN in prediction horizons of one, three, and five, respectively. The good performance of VMD-MFRFNN is due to its structure. A structure with a feedback loop allows this method to memorize past observations. Moreover, employing multiple states and the VMD decomposition method enables the VMD-MFRFNN to store historical information and capture the dynamic characteristics of the time series over longer prediction horizons.

Comparing the results obtained by LSTM with EMD-LSTM and MFRFNN with VMD-MFRFNN revealed that decomposition approaches have a favorable effect on forecasting results. Decomposition-based methods decompose complex time series into several IMFs with simpler structures and more stationary trends. Reducing the complexity will make it easier for the model to predict the time series. In addition, a two-tailed Welch's t-test with a significance threshold of $\alpha = 0.05$ was used to determine if the superiority of a model was statistically significant or not. The statistical test was performed between VMD-MFRFNN and other methods. Welch's t-test, a nonparametric two-sample statistical test, is beneficial when the samples have unequal variances [41]. As can be seen, in all experiments, the null hypothesis is rejected (i.e., p-value < 0.05), confirming that the results are statistically significant.

Fig. 5 demonstrates one-step-ahead prediction curves of VMD decomposition components for the HSI time series generated by VMD-MFRFNN. Figs. 6 and 7 illustrate the one-step-ahead prediction curves of IMF_2 and IMF_6, respectively.

As can be seen, VMD-MFRFNN predicts low-frequency IMF components with high accuracy and high-frequency IMF components with



(a) HSI Time Series (b) IMF_1 Component (c) IMF_2 Component

(d) IMF_3 Component (e) IMF_4 Component (f) IMF_5 Component

(g) IMF_6 Component (h) IMF_7 Component (i) IMF_8 Component

(j) IMF_9 Component

Fig. 5. One-step-ahead prediction curves of VMD decomposition components for the HSI time series.

low accuracy because the number of fuzzy rules is too small to model high-frequency IMF components accurately. Intuitively, the prediction of high-frequency IMF components is a bias for VMD-MFRFNN that adjust the final output.

Fig. 8 shows prediction curves and histograms of errors for the HSI time series produced by VMD-MFRFNN. The Gaussian distribution of errors indicates that VMD-MFRFNN effectively captured the dynamic properties of the time series.

5.2. Shanghai Stock Exchange (SSE)

The Shanghai Stock Exchange (SSE) is a stock exchange headquartered in Shanghai, China. According to market capitalization, the SSE is the third-largest stock market in the world and is representative of a developing market. The SSE time series data comprised 2218 samples obtained from Yahoo Finance during a nine-year period from 31-December-2010 to 21-February-2020. The first 1478 samples were used for the training, 370 samples for validation, and 370 samples for the test.

The prediction results of the SSE time series are given in Table 4. As can be seen, the results showed better performance of VMD-MFRFNN compared to other methods in one-step-ahead prediction of SSE time series, closely followed by MEMD-LSTM. In three and five-step-ahead predictions based on MAPE, MEMD-LSTM outperformed other methods, and after that, VMD-MFRFNN had the lowest MAPE. Based on

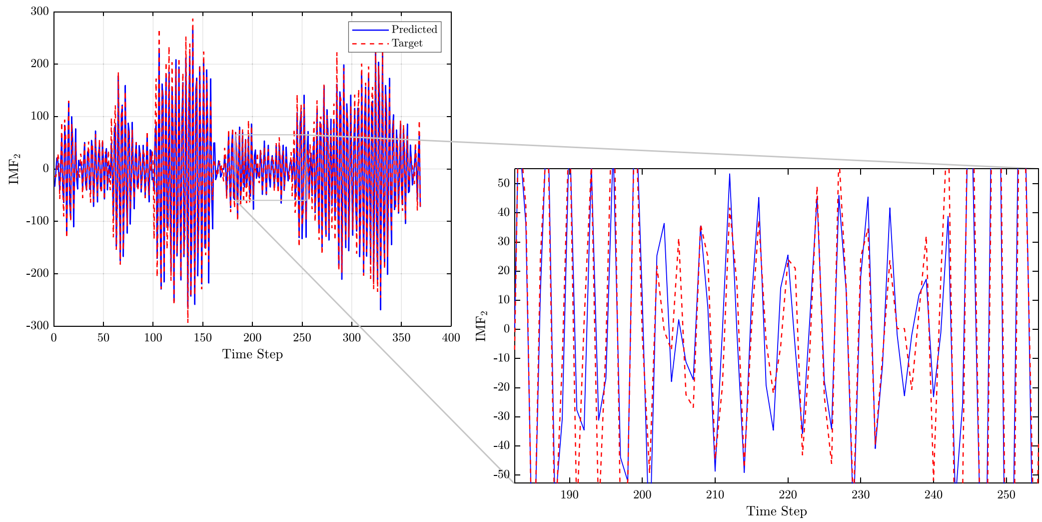


Fig. 6. One-step-ahead prediction curve of IMF_2 for the HSI time series.

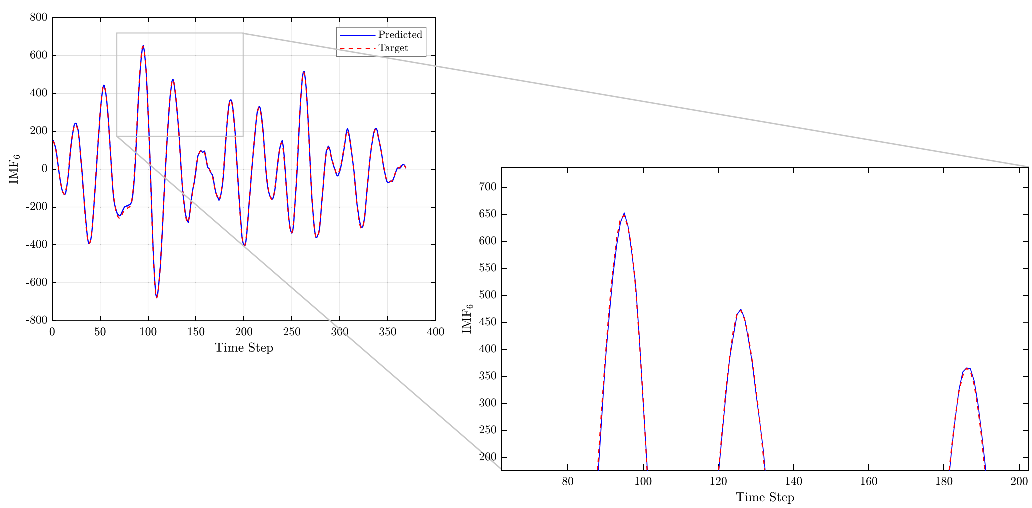


Fig. 7. One-step-ahead prediction curve of IMF_6 for the HSI time series.

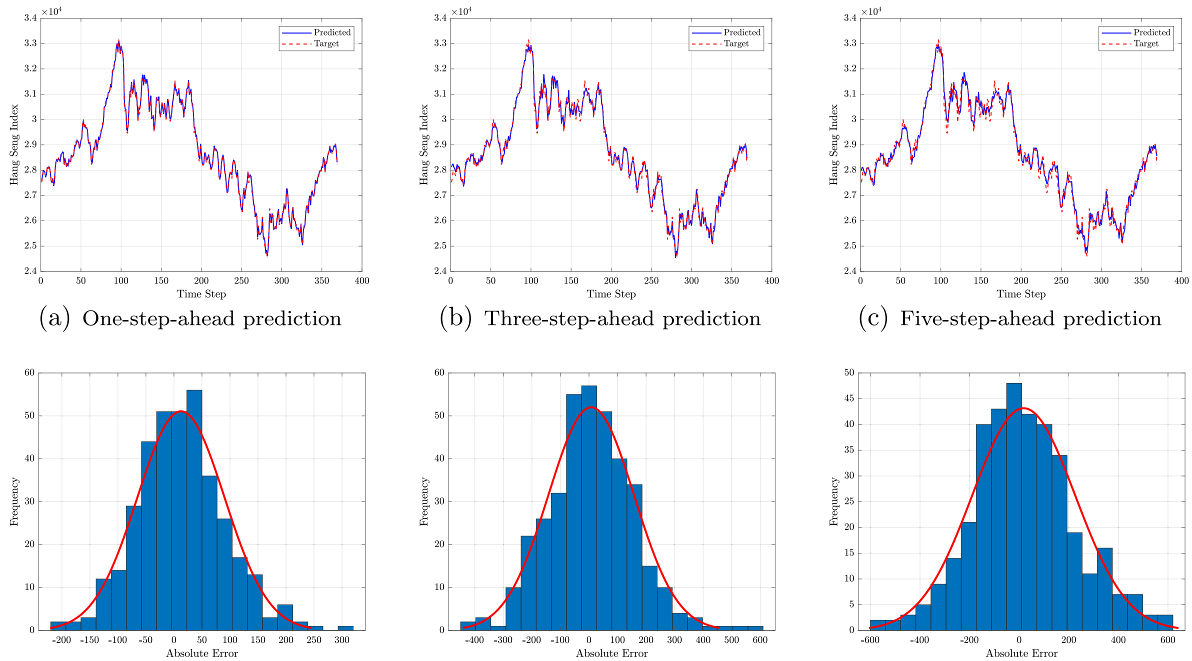
RMSE, VMD-MFRFNN had the highest accuracy in all prediction horizons, closely followed by MEMD-LSTM. Based on this metric, VMD-MFRFNN showed a decrease of 43.58%, 22.96%, and 8.10% from the MEMD-LSTM in one, three, and five-step-ahead prediction tasks, respectively.

From Table 4, it is evident that decomposition-based approaches (i.e., EMD-LSTM, MEMD-LSTM, and VMD-MFRFNN) obtained promising results compared to other methods. GARCH and ARIMA had the worst performance based on MAPE and RMSE, respectively. DCT-MFRFNN outperformed DCT-LSTM and MFRFNN in all prediction horizons. Based on RMSE, DCT-MFRFNN showed a decrease of 18.80%, 33.07%, and 34.48% from MFRFNN in one, three, and five-step-ahead forecasting tasks, respectively. The good performance of VMD-MFRFNN in this experiment can be explained by its potential to learn multiple functions simultaneously, along with employing a decomposition method to make the forecasting problem easier. Using two states allows the VMD-MFRFNN to learn and track dynamic behaviors of the SSE dataset over time. Moreover, VMD-MFRFNN, as an FNN, eliminates the time series noise. Note that, based on the statistical tests, the results of this experiment are regarded as statistically significant since all the reported p-values are less than the significance threshold (i.e., $\alpha = 0.05$). Prediction curves and histograms of errors produced by VMD-MFRFNN for the SSE time series are shown in Fig. 9.

5.3. Standard & Poor's 500 Index (SPX)

The Standard & Poor's 500 Index (SPX) tracks the stock price of 500 publicly traded domestic companies in the United States. SPX is typically recognized as one of the most developed markets in the world due to its mature market operation system [3]. The SPX data was also collected from Yahoo Finance during an eleven-year period from 4-January-2010 to 31-December-2020. From the entire time series data, 1845 samples were considered as the training set, 462 samples as the validation set, and the remaining 462 samples as the test set.

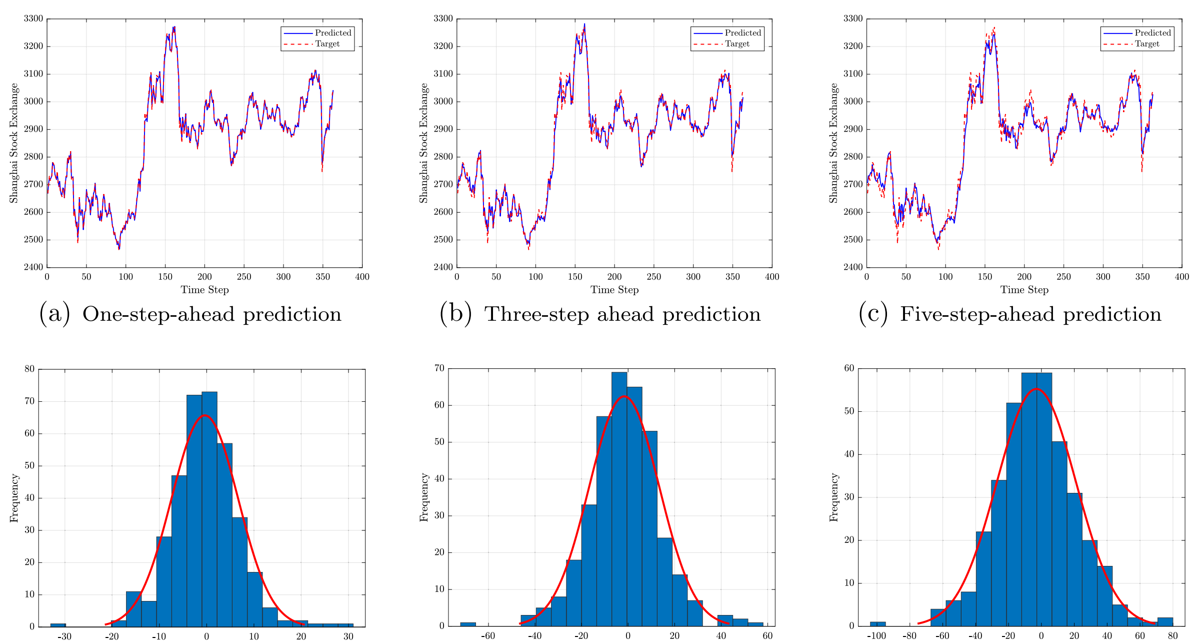
Table 5 compares VMD-MFRFNN's performance in one, three, and five-step-ahead predictions of SPX with other methods. As shown in Table 5, VMD-MFRFNN obtained the smallest MAPE and RMSE in one and three-step-ahead predictions of the SPX time series. The MEMD-LSTM was the second-best method in these experiments. In five-step-ahead prediction based on MAPE, MEMD-LSTM outperformed other methods, and after that, VMD-MFRFNN had the lowest MAPE. The long-term



(a) One-step-ahead prediction (b) Three-step-ahead prediction (c) Five-step-ahead prediction

(d) Histogram of Errors (One-step-ahead) (e) Histogram of Errors (Three-step-ahead) (f) Histogram of Errors (Five-step-ahead)

Fig. 8. Prediction curves and histograms of errors for the HSI time series.



(a) One-step-ahead prediction (b) Three-step ahead prediction (c) Five-step-ahead prediction

(d) Histogram of Errors (One-step-ahead) (e) Histogram of Errors (Three-step-ahead) (f) Histogram of Errors (Five-step-ahead)

Fig. 9. Prediction curves and histograms of errors for the SSE time series.

forecasting capability of MEMD-LSTM can be attributed to the ability to capture important features and estimate the nonlinear dynamic function in the SPX time series.

Based on RMSE, VMD-MFRFNN had the highest accuracy in five-step-ahead prediction, closely followed by MEMD-LSTM. Based on this metric, VMD-MFRFNN showed a decrease of 55.56%, 42.23%, and 6.00% from the MEMD-LSTM in one, three, and five-step-ahead predictions of the SPX dataset, respectively. In this time series, based on RMSE, DCT-MFRFNN showed a decrease of 36.47%, 41.18%, and 41.18% from MFRFNN in prediction horizons of one, three, and five, respectively.

Table 5
One, three, and five-step-ahead prediction results of the SPX.

Method	Step	MAPE (%)	*p*-value	RMSE	*p*-value
ARIMA	1	3.84E+00	1.31E−42	1.28E+02	4.66E−40
	3	4.43E+00	2.20E−54	1.58E+02	1.70E−49
	5	5.59E+00	3.86E−44	1.68E+02	3.60E−46
GARCH	1	1.02E+01	2.83E−51	5.06E+01	2.28E−30
	3	1.04E+01	5.36E−62	7.70E+01	5.32E−42
	5	1.07E+01	3.67E−50	1.14E+02	9.29E−42
FFNN	1	2.21E+00	2.85E−37	1.11E+02	1.06E−38
	3	2.29E+00	6.43E−48	1.54E+02	2.99E−49
	5	2.84E+00	5.15E−37	1.66E+02	4.83E−46
Prophet	1	4.60E+00	2.90E−44	1.91E+02	9.68E−44
	3	4.63E+00	8.57E−55	1.92E+02	2.42E−51
	5	4.66E+00	2.41E−42	1.93E+02	1.29E−47
LSTM	1	1.54E+00	2.08E−33	6.77E+01	1.11E−33
	3	1.61E+00	5.32E−44	7.00E+01	7.18E−41
	5	1.72E+00	2.86E−30	7.47E+01	8.47E−36
EMD-LSTM	1	6.82E−01	1.69E−21	5.65E+01	1.15E−31
	3	7.29E−01	3.13E−31	5.81E+01	1.67E−38
	5	7.99E−01	1.34E−05	6.09E+01	5.82E−32
MEMD-LSTM	1	5.28E−01	2.04E−14	3.60E+01	7.47E−26
	3	6.51E−01	5.88E−28	3.86E+01	5.18E−32
	5	**6.62E−01**	6.84E−17	4.00E+01	2.00E−13
DCT-LSTM	1	1.53E+00 (2.61E−02)	6.76E−53	8.85E+01 (2.30E+00)	6.15E−39
	3	2.10E+00 (3.63E−02)	1.18E−34	1.06E+02 (2.26E+00)	4.48E−33
	5	2.72+00 (5.78E−02)	8.66E−40	1.25E+02 (2.54E+00)	2.15E−33
MFRFNN	1	9.95E−01 (6.38E−03)	9.79E−31	4.47E+01 (1.41E−01)	1.45E−29
	3	1.63E+00 (1.10E−02)	2.50E−59	7.31E+01 (1.88E−01)	2.61E−54
	5	2.13E+00 (1.32E−02)	5.95E−44	9.47E+01 (1.82E−01)	3.61E−45
DCT-MFRFNN	1	5.87E−01 (5.40E−03)	6.87E−19	2.84E+01 (9.02E−01)	7.15E−32
	3	9.43E−01 (1.40E−02)	1.74E−39	4.30E+01 (1.02E+00)	2.91E−32
	5	1.22E+00 (1.09E−02)	6.25E−28	5.57E+01 (7.90E−01)	7.67E−42
VMD-MFRFNN	1	**4.18E−01 (2.40E−02)**	–	**1.60E+01 (1.07E+00)**	–
	3	**4.91E−01 (6.63E−03)**	–	**2.23E+01 (4.13E−01)**	–
	5	8.35E−01 (2.77E−02)	–	**3.76E+01 (5.94E−01)**	–

Comparing the results of VMD-MFRFNN with MFRFNN revealed that VMD-MFRFNN had higher accuracy and standard deviation. Having a high standard deviation is a drawback of the proposed VMD-MFRFNN. Note that the variance of predictions increases as the standard deviation increases. Also, it is worth mentioning that LSTM outperformed FFNN in all experiments due to inherent problems of FFNN, such as overfitting and falling into local minima. The Welch's t-test performed on SPX prediction results confirmed that the results are statistically significant. Fig. 10 shows prediction curves and histograms of errors produced by VMD-MFRFNN for the SPX dataset.

5.4. Comparing different decomposition methods

This section compares various decomposition methods to strengthen this research further. We used different decomposition approaches in the decomposition phase of the proposed method, including VMD [16], EMD [14], EEMD [24], and Robust Empirical Mode Decomposition (REMD) [42]. REMD is an enhanced EMD that employs an adaptive sifting stop criterion to automatically terminate EMD's sifting process.

We set the prediction horizon equal to 5 in all experiments. Results of comparing decomposition methods are presented in Table 6. As seen in Table 6, the VMD method outperformed other methods in all financial time series, closely followed by EEMD. Based on the statistical tests, all the results are statistically significant. So, obviously, using VMD as a decomposition method in the structure of the proposed method is a good choice. Comparing the prediction results of EMD and EEMD demonstrated that EEMD achieved higher accuracy in all experiments. The good performance of EEMD can be attributed to the elimination of the mode mixing problem by this method.

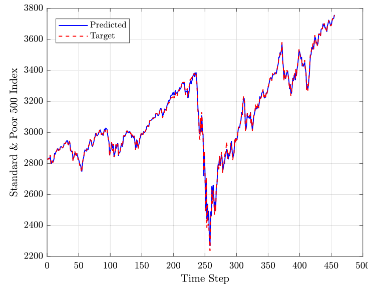
5.5. Comparing different metaheuristic methods

As we discussed earlier, VMD-MFRFNN uses the PSO method to learn the weights of the state network. To evaluate the merits of the PSO algorithm, we compared its performance in training VMD-MFRFNN with various metaheuristic approaches. These approaches include Artificial Bee Colony (ABC) [43], Ant Colony Optimization for Continuous Domains (ACOR) [44], Bees Algorithm (BA) [45], Biogeography-based Optimization (BBO) [46], Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [47], Differential Evolution (DE) [48], Firefly Algorithm (FA) [49], Genetic Algorithm (GA) [50], Harmony Search (HS) [51], Imperialist Competitive Algorithm (ICA) [52], Invasive Weed Optimization (IWO) [53], and Teaching-Learning-based Optimization (TLBO) [54]. Five-step-ahead prediction of the SPX time series was used as the benchmark. In all experiments, we executed twenty independent runs and reported the averages and standard deviations of the results over these runs. The parameters of VMD-MFRFNN were the same as those listed in Table 2.

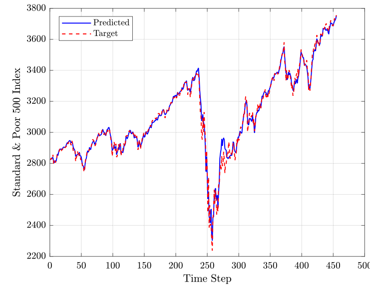
The prediction results of different metaheuristic methods are given in Table 7. As shown in Table 7, PSO outperformed other methods in training VMD-MFRFNN, closely followed by ACOR and TLBO based on MAPE and RMSE, respectively. HS had the worst performance on both metrics. Moreover, the null hypothesis is rejected in all experiments, confirming that the results are statistically significant. Fig. 11 illustrates the box plots of RMSE values for different metaheuristic methods. The PSO obtained the lowest variance of predictions with a minimum RMSE value equal to 36.57 and a maximum RMSE value equal to 38.31, making it the most robust training algorithm for VMD-MFRFNN. After PSO, GA had the lowest variance, but its average RMSE was equal to 39.34, which is not promising. BA had the highest variance of predictions with a minimum RMSE value equal to 36.86 and a maximum RMSE value equal to 51.19. The good performance of PSO can be attributed to its global search capability and ability to avoid being trapped in local optima. PSO can quickly converge to a global optimum, even when the search space is large and complex. Furthermore, PSO is developed to avoid local optima by combining random exploration and social learning.

5.6. Comparing different training methods

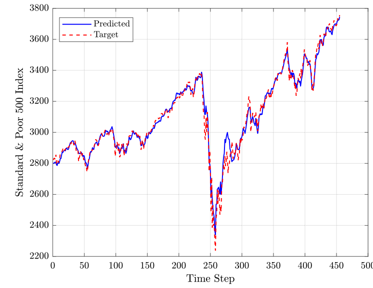
In this section, we employed the Gradient Descent (GD) method instead of PSO to train the state network weights and compared its performance with the PSO method. The prediction horizon was considered 5 in all experiments. For GD, we set the maximum number of iterations and learning rate to 100 and 0.01, respectively. The results of comparing PSO and GD are presented in Table 8. As seen in Table 8, PSO outperformed GD in all benchmarks. Note that, based on the statistical tests, the results are regarded as statistically significant since all the reported p-values are less than 0.05. Fig. 12 shows the box plots of RMSE values for PSO and GD. As shown in Fig. 12, the PSO obtained a lower prediction variance compared to GD in all time series. According to the results, it can be concluded that PSO is more robust than GD for training VMD-MFRFNN. The GD can get stuck in local optima if the objective function has multiple local optima, while PSO is developed to avoid local optima. In PSO, particles update their position and velocities based on their own personal best and the global best, allowing them to explore different areas of the search space and avoid getting stuck in local optima.



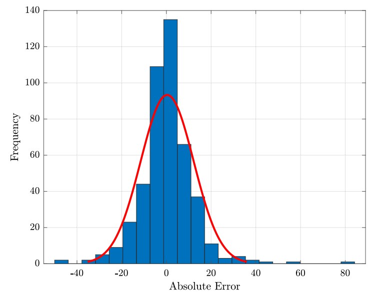
(a) One-step-ahead prediction



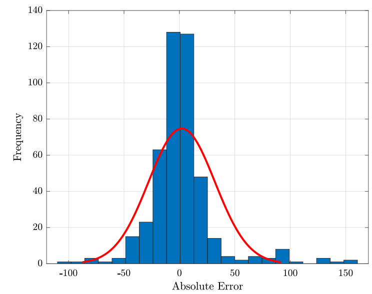
(b) Three-step-ahead prediction



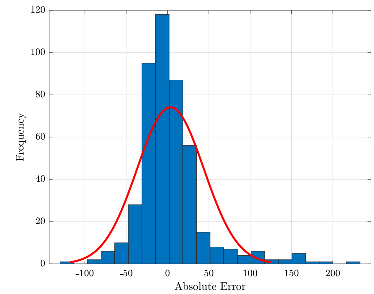
(c) Five-step-ahead prediction



(d) Histogram of Errors (One-step-ahead)



(e) Histogram of Errors (Three-step-ahead)



(f) Histogram of Errors (Five-step-ahead)

Fig. 10. Prediction curves and histograms of errors for the SPX time series.

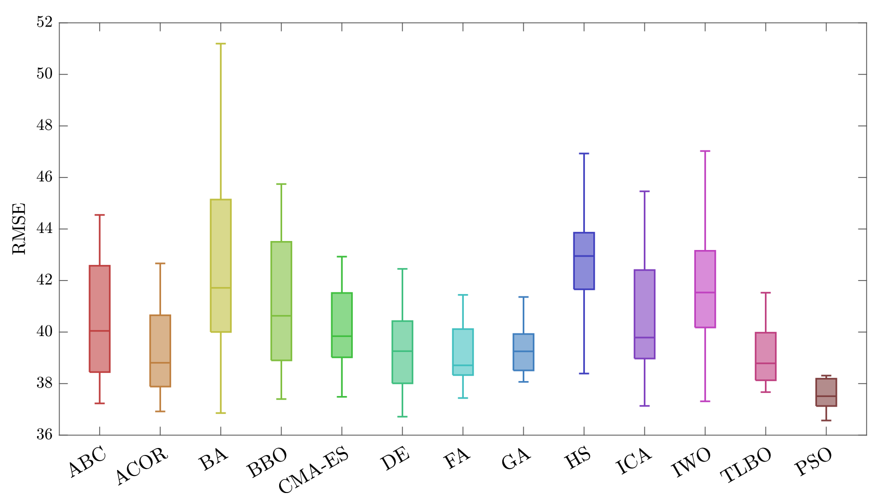
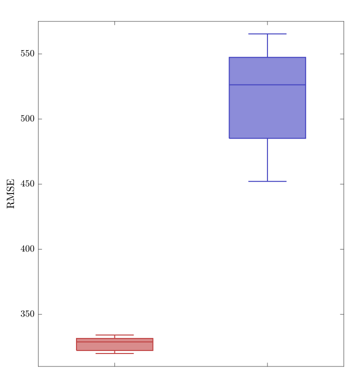
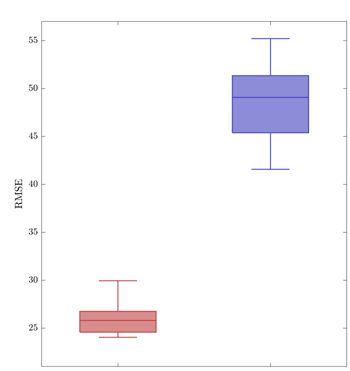


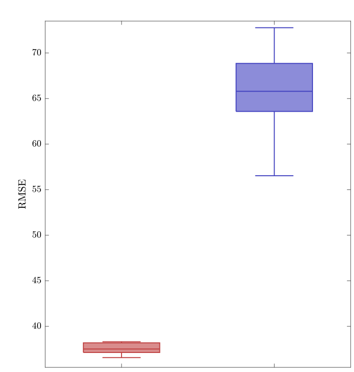
Fig. 11. Box plots of RMSE values for different metaheuristic methods (Five-step-ahead prediction of SPX time series).



(a) HSI Time Series



(b) SSE Time Series



(c) SPX Time Series

Fig. 12. Box plots of RMSE values for different training methods (Five-step-ahead forecasting task).

Table 6
Comparison of different decomposition methods.

Decomposition method		VMD	EMD	REMD	EEMD
HSI	MAPE	**8.99E−01 (2.24E−02)**	1.08E+00 (2.58E−02)	1.08E+00 (2.55E−02)	9.81E−01 (1.33E−01)
	p-value	–	4.57E−24	3.25E−24	1.32E−02
	RMSE	**3.27E+02 (6.53E+00)**	4.03E+02 (7.51E+00)	4.32E+02 (9.50E+00)	3.53E+02 (4.20E+01)
	p-value	–	9.80E−30	3.10E−30	1.28E−02
SSE	MAPE	**6.78E−01 (4.38E−02)**	1.19E+00 (2.59E−02)	1.22E+00 (1.70E−02)	9.28E−01 (1.51E−02)
	p-value	–	1.08E−29	1.37E−26	4.58E−18
	RMSE	**2.61E+01 (1.91E+00)**	4.53E+01 (9.17E−01)	4.87E+01 (6.55E−01)	3.65E+01 (5.11E−01)
	p-value	–	6.11E−26	2.59E−25	6.18E−17
SPX	MAPE	**8.35E−01 (2.77E−02)**	1.31E+00 (5.96E−02)	1.56E+00 (7.25E−02)	9.81E−01 (5.50E−02)
	p-value	–	5.00E−23	3.06E−24	2.54E−11
	RMSE	**3.76E+01 (5.94E−01)**	6.13E+01 (2.60E+00)	8.09E+01 (5.28E+00)	4.60E+01 (2.85E+00)
	p-value	–	3.12E−21	2.15E−19	2.39E−11

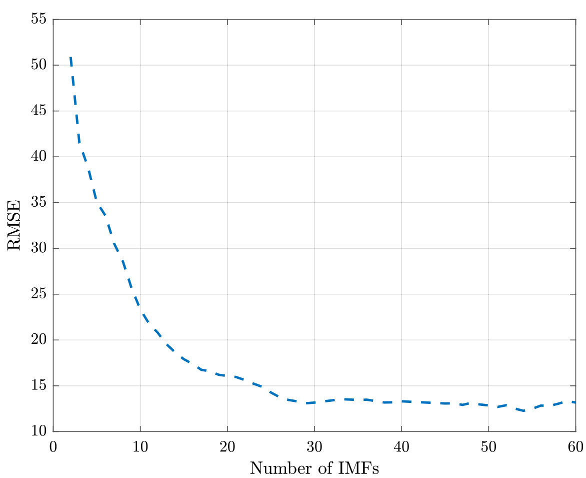


Fig. 13. Relationship between RMSE and the number of IMFs in the five-step-ahead prediction of the SSE.

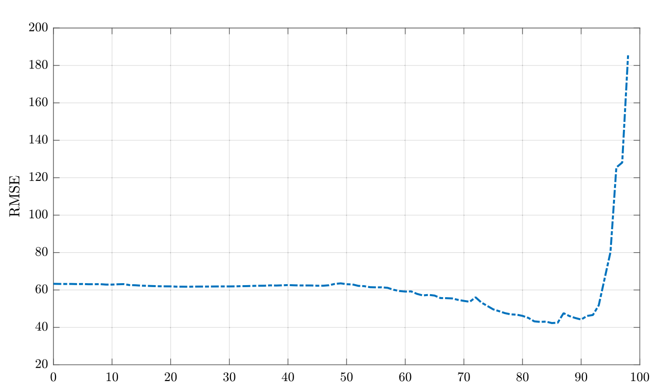


Fig. 14. Relationship between RMSE and λ in the three-step-ahead prediction of the SSE.

Table 7
Comparison of different metaheuristic methods.

Method	MAPE (%)	*p*-value	RMSE	*p*-value
ABC	9.26E−01 (8.02E−02)	7.36E−05	4.05E+01 (2.37E+00)	2.75E−05
ACOR	8.87E−01 (5.40E−02)	6.48E−04	3.93E+01 (1.77E+00)	4.63E−04
BA	9.70E−01 (1.05E−01)	1.46E−05	4.26E+01 (3.54E+00)	4.32E−06
BBO	9.44E−01 (9.30E−02)	4.77E−05	4.10E+01 (2.67E+00)	1.66E−05
CMA-ES	9.09E−01 (5.99E−02)	2.99E−05	4.01E+01 (1.70E+00)	2.21E−06
DE	9.09E−01 (5.88E−02)	2.37E−05	3.93E+01 (1.51E+00)	8.62E−05
FA	8.95E−01 (5.57E−02)	1.82E−04	3.94E+01 (1.71E+00)	1.77E−04
GA	8.91E−01 (4.18E−02)	1.88E−05	3.93E+01 (9.16E−01)	6.25E−08
HS	9.75E−01 (8.10E−02)	1.74E−07	4.30E+01 (2.93E+00)	8.22E−08
ICA	9.25E−01 (6.42E−02)	4.73E−06	4.06E+01 (2.35E+00)	1.60E−05
IWO	9.52E−01 (8.83E−02)	9.80E−06	4.21E+01 (3.30E+00)	6.88E−06
TLBO	8.92E−01 (3.86E−02)	5.54E−06	3.91E+01 (1.22E+00)	3.37E−05
PSO	**8.35E−01 (2.77E−02)**	–	**3.76E+01 (5.94E−01)**	–

5.7. Sensitivity analysis

This section provides sensitivity analyses on the proposed methods to evaluate the sensitivity of VMD-MFRRFNN and DCT-MFRFNN to the number of IMFs and λ, respectively. Five and three-step-ahead prediction of the SSE time series were used as the benchmark for VMD-MFRFNN and DCT-MFRFNN, respectively. The number of IMFs ranged from 2 to 60, and λ ranged from 0 to 99. Other parameters were the same as those provided in Table 2.

The relationship between RMSE and the number of IMFs is shown in Fig. 13. The results indicated that when the number of IMFs increased from 2 to 30, the RMSE was significantly reduced. Increasing the number of IMFs means predicting time series with simpler structures and more stationary trends, which is easier for the model. However,

Table 8
Five-step-ahead prediction comparison of PSO and gradient descent.

Method		PSO	Gradient descent
HSI	MAPE	**8.99E−01 (2.24E−02)**	1.41E+00 (9.09E−02)
	p-value	–	4.67E−17
	RMSE	**3.27E+02 (6.53E+00)**	5.19E+02 (3.62E+01)
	p-value	–	4.17E−16
SSE	MAPE	**6.78E−01 (4.38E−02)**	1.36E+00 (3.13E−01)
	p-value	–	6.48E−09
	RMSE	**2.61E+01 (1.91E+00)**	5.23E+01 (1.74E+01)
	p-value	–	1.88E−06
SPX	MAPE	**8.35E−01 (2.77E−02)**	1.51E+00 (1.75E−01)
	p-value	–	2.35E−13
	RMSE	**3.76E+01 (5.94E−01)**	6.60E+01 (4.15E+00)
	p-value	–	4.79E−18

increasing the number of IMFs from 30 to 60 did not have much impact on RMSE. Note that increasing IMFs also increases the computational complexity of the model. The relationship between RMSE and λ is illustrated in Fig. 14. Results indicated that when λ increased from 50 to 85, the RMSE was reduced. Increasing λ means reducing fluctuations and simplifying the time series structure, which leads to more accurate predictions. On the other hand, increasing λ from 90 to 99 increases the RMSE. When λ increases too much, a lot of meaningful information is discarded from the time series, and as a result, the predictions differ significantly from actual values. Fig. 15 shows reconstructed signals of the SSE time series for different λ values.

6. Conclusion

This study proposed two novel methods for multi-step-ahead stock price prediction. DCT-MFRFNN, a method based on DCT and MFRFNN, employed DCT to eliminate high frequencies, reduce fluctuations in the time series, and simplify its structure. It used MFRFNN to predict financial time series. VMD-MFRFNN, an approach based on VMD and MFRFNN, combined the benefits of VMD and MFRFNN. VMD-MFRFNN consisted of two phases: (a) decomposition phase and (b) prediction and reconstruction phase. The input time series was decomposed to several IMFs using VMD in the decomposition phase. Since these components had simpler structures and more stationary trends, their prediction was easier, and the expected accuracy was higher. In the prediction and reconstruction phase, each of IMFs was given to a separate MFRFNN model for prediction, and the predicted signals were summed to reconstruct the final output. Using the MFRFNN models enabled VMD-MFRFNN to learn and memorize historical data from previous samples, capture the dynamic characteristics of the financial time series, and approximate several functions simultaneously.

The prediction performance of DCT-MFRFNN and VMD-MFRFNN was evaluated using three financial time series, including HSI, SSE, and SPX. The experimental results indicated that VMD-MFRFNN surpassed other state-of-the-art models and showed promising performance in multi-step-ahead prediction tasks. In more detail, based on RMSE, for the HSI time series, VMD-MFRFNN showed a decrease of 63.27%, 41.27%, and 3.25% from the MEMD-LSTM in one, three, and five-step-ahead predictions, respectively. For the SSE time series, it demonstrated an RMSE decrease of 43.58%, 22.96%, and 8.10% from the MEMD-LSTM in prediction horizons of one, three, and five, respectively. For the SPX time series, VMD-MFRFNN showed a decrease of 55.56%, 42.23%, and 6.00% from the MEMD-LSTM in one, three, and five-step-ahead predictions, respectively, in terms of RMSE. Furthermore, DCT-MFRFNN outperformed DCT-LSTM and MFRFNN in all experiments. DCT-MFRFNN, on average, showed 8.87%, 28.78%, and 39.61% decreases in RMSE from MFRFNN for HSI, SSE, and SPX, respectively. The effectiveness of particle swarm optimization (PSO) in training VMD-MFRFNN was assessed by comparing its performance with the gradient descent method and twelve different metaheuristic algorithms. The results highlighted the merits of PSO in training VMD-MFRFNN. The proposed methods in this study are definitely worth investigating further. One interesting area of future research is to apply DCT-MFRFNN and VMD-MFRFNN in other fields, such as meteorology and biomedical engineering.

CRediT authorship contribution statement

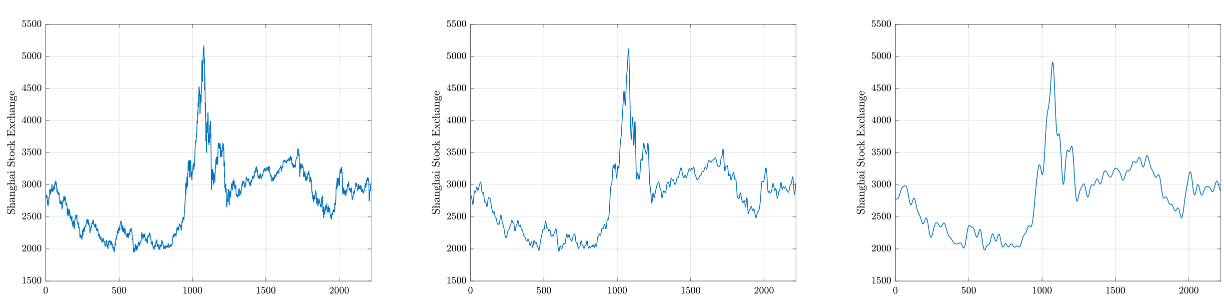
Hamid Nasiri: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Visualization. **Mohammad Mehdi Ebadzadeh:** Conceptualization, Validation, Investigation, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Code availability

The implementation of the proposed method and datasets needed to reproduce the results are provided on GitHub https://github.com/Hamid-Nasiri/VMD-MFRFNN.



(a) $\lambda = 0$ (b) $\lambda = 85$ (c) $\lambda = 95$

Fig. 15. Reconstructed signals of the SSE time series for different λ values.

References

[1] F.S. Nejad, M.M. Ebadzadeh, Stock market forecasting using DRAGAN and feature matching, 2023, http://dx.doi.org/10.48550/arXiv.2301.05693, arXiv preprint arXiv:2301.05693.

[2] T. Liu, X. Ma, S. Li, X. Li, C. Zhang, A stock price prediction method based on meta-learning and variational mode decomposition, Knowl.-Based Syst. 252 (2022) 109324, http://dx.doi.org/10.1016/j.knosys.2022.109324.

[3] C. Deng, Y. Huang, N. Hasan, Y. Bao, Multi-step-ahead stock price index forecasting using long short-term memory model with multivariate empirical mode decomposition, Inform. Sci. 607 (2022) 297–321, http://dx.doi.org/10.1016/j.ins.2022.05.088.

[4] A.Q. Md, S. Kapoor, C.J. A.V., A.K. Sivaraman, K.F. Tee, S. H., J. N., Novel optimization approach for stock price forecasting using multi-layered sequential LSTM, Appl. Soft Comput. 134 (2023) 109830, http://dx.doi.org/10.1016/j.asoc.2022.109830.

[5] A. Thakkar, K. Chaudhari, Information fusion-based genetic algorithm with long short-term memory for stock price and trend prediction, Appl. Soft Comput. 128 (2022) 109428, http://dx.doi.org/10.1016/j.asoc.2022.109428.

[6] Y. Guo, J. Guo, B. Sun, J. Bai, Y. Chen, A new decomposition ensemble model for stock price forecasting based on system clustering and particle swarm optimization, Appl. Soft Comput. 130 (2022) 109726, http://dx.doi.org/10.1016/j.asoc.2022.109726.

[7] J. Cao, J. Wang, Stock price forecasting model based on modified convolution neural network and financial time series analysis, Int. J. Commun. Syst. 32 (12) (2019) e3987, http://dx.doi.org/10.1002/dac.3987.

[8] W. Guo, Q. Liu, Z. Luo, Y. Tse, Forecasts for international financial series with VMD algorithms, J. Asian Econ. 80 (2022) 101458, http://dx.doi.org/10.1016/j.asieco.2022.101458.

[9] Y. Lin, Y. Yan, J. Xu, Y. Liao, F. Ma, Forecasting stock index price using the CEEMDAN-LSTM model, North Am. J. Econ. Finance 57 (2021) 101421, http://dx.doi.org/10.1016/j.najef.2021.101421.

[10] H. Rezaei, H. Faaljou, G. Mansourfar, Stock price prediction using deep learning and frequency decomposition, Expert Syst. Appl. 169 (2021) 114332, http://dx.doi.org/10.1016/j.eswa.2020.114332.

[11] Y. Yujun, Y. Yimei, Z. Wang, Research on a hybrid prediction model for stock price based on long short-term memory and variational mode decomposition, Soft Comput. 25 (21) (2021) 13513–13531, http://dx.doi.org/10.1007/s00500-021-06122-4.

[12] H. Xu, D. Cao, S. Li, A self-regulated generative adversarial network for stock price movement prediction based on the historical price and tweets, Knowl.-Based Syst. 247 (2022) 108712, http://dx.doi.org/10.1016/j.knosys.2022.108712.

[13] H.-P. Nguyen, P. Baraldi, E. Zio, Ensemble empirical mode decomposition and long short-term memory neural network for multi-step predictions of time series signals in nuclear power plants, Appl. Energy 283 (2021) 116346, http://dx.doi.org/10.1016/j.apenergy.2020.116346.

[14] N.E. Huang, Z. Shen, S.R. Long, M.C. Wu, H.H. Shih, Q. Zheng, N.-C. Yen, C.C. Tung, H.H. Liu, The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci. 454 (1971) (1998) 903–995, http://dx.doi.org/10.1098/rspa.1998.0193.

[15] Z. Liu, J. Liu, A robust time series prediction method based on empirical mode decomposition and high-order fuzzy cognitive maps, Knowl.-Based Syst. 203 (2020) 106105, http://dx.doi.org/10.1016/j.knosys.2020.106105.

[16] K. Dragomiretskiy, D. Zosso, Variational Mode Decomposition, IEEE Trans. Signal Process. 62 (3) (2014) 531–544, http://dx.doi.org/10.1109/TSP.2013.2288675.

[17] K. Johny, M.L. Pai, A. S., A multivariate EMD-LSTM model aided with Time Dependent Intrinsic Cross-Correlation for monthly rainfall prediction, Appl. Soft Comput. 123 (2022) 108941, http://dx.doi.org/10.1016/j.asoc.2022.108941.

[18] Y. Liu, C. Yang, K. Huang, W. Gui, Non-ferrous metals price forecasting based on variational mode decomposition and LSTM network, Knowl.-Based Syst. 188 (2020) 105006, http://dx.doi.org/10.1016/j.knosys.2019.105006.

[19] H. Nasiri, M.M. Ebadzadeh, MFRFNN: Multi-Functional Recurrent Fuzzy Neural Network for Chaotic Time Series Prediction, Neurocomputing 507 (2022) 292–310, http://dx.doi.org/10.1016/j.neucom.2022.08.032.

[20] F. Zhou, H.-m. Zhou, Z. Yang, L. Yang, EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction, Expert Syst. Appl. 115 (2019) 136–151, http://dx.doi.org/10.1016/j.eswa.2018.07.065.

[21] Y. Yang, C. Fan, H. Xiong, A novel general-purpose hybrid model for time series forecasting, Appl. Intell. 52 (2) (2022) 2212–2223, http://dx.doi.org/10.1007/s10489-021-02442-y.

[22] Z. Tang, T. Zhang, J. Wu, X. Du, K. Chen, Multistep-Ahead Stock Price Forecasting Based on Secondary Decomposition Technique and Extreme Learning Machine Optimized by the Differential Evolution Algorithm, Math. Probl. Eng. 2020 (2020) 1–13, http://dx.doi.org/10.1155/2020/2604915.

[23] A. Salimi-Badr, M.M. Ebadzadeh, A novel learning algorithm based on computing the rules' desired outputs of a TSK fuzzy neural network with non-separable fuzzy rules, Neurocomputing 470 (2022) 139–153, http://dx.doi.org/10.1016/j.neucom.2021.10.103.

[24] Z. Wu, N.E. Huang, Ensemble Empirical Mode Decomposition: A Noise-Assisted Data Analysis Method, Adv. Adapt. Data Anal. 01 (01) (2009) 1–41, http://dx.doi.org/10.1142/S1793536909000047.

[25] S.Y. Rashida, M. Sabaei, M.M. Ebadzadeh, A.M. Rahmani, An intelligent approach for predicting resource usage by combining decomposition techniques with NFTS network, Cluster Comput. 23 (4) (2020) 3435–3460, http://dx.doi.org/10.1007/s10586-020-03099-x.

[26] S. Lahmiri, A variational mode decompoisition approach for analysis and forecasting of economic and financial time series, Expert Syst. Appl. 55 (2016) 268–273, http://dx.doi.org/10.1016/j.eswa.2016.02.025.

[27] H. Niu, K. Xu, W. Wang, A hybrid stock price index forecasting model based on variational mode decomposition and LSTM network, Appl. Intell. 50 (12) (2020) 4296–4309, http://dx.doi.org/10.1007/s10489-020-01814-0.

[28] J. Wang, Q. Cui, X. Sun, M. He, Asian stock markets closing index forecast based on secondary decomposition, multi-factor analysis and attention-based LSTM model, Eng. Appl. Artif. Intell. 113 (2022) 104908, http://dx.doi.org/10.1016/j.engappai.2022.104908.

[29] W.-D. Wan, Y.-l. Bai, Y.-N. Lu, L. Ding, A hybrid model combining a gated recurrent unit network based on variational mode decomposition with error correction for stock price prediction, Cybern. Syst. (2022) 1–25, http://dx.doi.org/10.1080/01969722.2022.2137634.

[30] J. Zhang, X. Chen, A two-stage model for stock price prediction based on variational mode decomposition and ensemble machine learning method, Soft Comput. (2023) 1–24, http://dx.doi.org/10.1007/s00500-023-08441-0.

[31] A. Vinciguerra, M. Aleardi, A. Hojat, M.H. Loke, E. Stucchi, Discrete cosine transform for parameter space reduction in Bayesian electrical resistivity tomography, Geophys. Prospect. 70 (1) (2022) 193–209, http://dx.doi.org/10.1111/1365-2478.13148.

[32] M. Begum, J. Ferdush, M.S. Uddin, A Hybrid robust watermarking system based on discrete cosine transform, discrete wavelet transform, and singular value decomposition, J. King Saud Univ. - Comput. Inf. Sci. 34 (8) (2022) 5856–5867, http://dx.doi.org/10.1016/j.jksuci.2021.07.012.

[33] H.H. Goh, R. He, D. Zhang, H. Liu, W. Dai, C.S. Lim, T.A. Kurniawan, K.T.K. Teo, K.C. Goh, A multimodal approach to chaotic renewable energy prediction using meteorological and historical information, Appl. Soft Comput. 118 (2022) 108487, http://dx.doi.org/10.1016/j.asoc.2022.108487.

[34] D. He, C. Liu, Z. Jin, R. Ma, Y. Chen, S. Shan, Fault diagnosis of flywheel bearing based on parameter optimization variational mode decomposition energy entropy and deep learning, Energy 239 (2022) 122108, http://dx.doi.org/10.1016/j.energy.2021.122108.

[35] X. Niu, J. Wang, L. Zhang, Carbon price forecasting system based on error correction and divide-conquer strategies, Appl. Soft Comput. 118 (2022) 107935, http://dx.doi.org/10.1016/j.asoc.2021.107935.

[36] J. Lian, Z. Liu, H. Wang, X. Dong, Adaptive variational mode decomposition method for signal processing based on mode characteristic, Mech. Syst. Signal Process. 107 (2018) 53–77, http://dx.doi.org/10.1016/j.ymssp.2018.01.019.

[37] B. Qiao, J. Liu, P. Wu, Y. Teng, Wind power forecasting based on variational mode decomposition and high-order fuzzy cognitive maps, Appl. Soft Comput. 129 (2022) 109586, http://dx.doi.org/10.1016/j.asoc.2022.109586.

[38] H. Yang, Y. Cheng, G. Li, A denoising method for ship radiated noise based on Spearman variational mode decomposition, spatial-dependence recurrence sample entropy, improved wavelet threshold denoising, and Savitzky-Golay filter, Alexandria Eng. J. 60 (3) (2021) 3379–3400, http://dx.doi.org/10.1016/j.aej.2021.01.055.

[39] Y. Tian, S. Xiang, Detection of video-based face spoofing using LBP and multi-scale DCT, in: Digital Forensics and Watermarking: 15th International Workshop, IWDW 2016, Beijing, China, September 17-19, 2016, Revised Selected Papers 15, Springer, 2017, pp. 16–28, http://dx.doi.org/10.1007/978-3-319-53465-7_2.

[40] S.J. Taylor, B. Letham, Forecasting at scale, Amer. Statist. 72 (1) (2018) 37–45, http://dx.doi.org/10.1080/00031305.2017.1380080.

[41] Z. Zojaji, M.M. Ebadzadeh, H. Nasiri, Semantic schema based genetic programming for symbolic regression, Appl. Soft Comput. 122 (2022) 108825, http://dx.doi.org/10.1016/j.asoc.2022.108825.

[42] Z. Liu, D. Peng, M.J. Zuo, J. Xia, Y. Qin, Improved Hilbert–Huang transform with soft sifting stopping criterion and its application to fault diagnosis of wheelset bearings, ISA Trans. 125 (2022) 426–444, http://dx.doi.org/10.1016/j.isatra.2021.07.011.

[43] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471, http://dx.doi.org/10.1007/s10898-007-9149-x.

[44] K. Socha, M. Dorigo, Ant colony optimization for continuous domains, European J. Oper. Res. 185 (3) (2008) 1155–1173, http://dx.doi.org/10.1016/j.ejor.2006.06.046.

[45] D. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, M. Zaidi, The Bees Algorithm — A Novel Tool for Complex Optimisation Problems, in: Intelligent Production Machines and Systems, Elsevier, 2006, pp. 454–459, http://dx.doi.org/10.1016/B978-008045157-2/50081-X.

[46] D. Simon, Biogeography-Based Optimization, IEEE Trans. Evol. Comput. 12 (6) (2008) 702–713, http://dx.doi.org/10.1109/TEVC.2008.919004.

[47] N. Hansen, S.D. Müller, P. Koumoutsakos, Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES), Evol. Comput. 11 (1) (2003) 1–18, http://dx.doi.org/10.1162/106365603321828970.

[48] R. Storn, K. Price, Differential Evolution–A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, J. Global Optim. 11 (4) (1997) 341–359, http://dx.doi.org/10.1023/A:1008202821328.

[49] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, second ed., Luniver Press, 2010.

[50] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73, http://dx.doi.org/10.1038/scientificamerican0792-66.

[51] Z.W. Geem, J.H. Kim, G.V. Loganathan, A New Heuristic Optimization Algorithm: Harmony Search, Simulation 76 (2) (2001) 60–68, http://dx.doi.org/10.1177/003754970107600201.

[52] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007, pp. 4661–4667, http://dx.doi.org/10.1109/CEC.2007.4425083.

[53] A.R. Mehrabian, C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, Ecol. Inform. 1 (4) (2006) 355–366, http://dx.doi.org/10.1016/j.ecoinf.2006.07.003.

[54] R. Rao, V. Savsani, D. Vakharia, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (3) (2011) 303–315, http://dx.doi.org/10.1016/j.cad.2010.12.015.