Contents lists available at ScienceDirect

Expert Systems With Applications

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

Oil price forecasting: A hybrid GRU neural network based on decomposition–reconstruction methods

Shiqi Zhang [a], Jing Luo [a], Shuyuan Wang [b], Feng Liu [b,*]

[a] *School of Mathematics and Statistics, Shandong University, Weihai, 264209, China*
[b] *Business School, Shandong University, Weihai, 264209, China*

ARTICLE INFO

Keywords:
Variational modal decomposition
Sample entropy
Gated recurrent unit
Deep learning
Crude oil price forecasting

ABSTRACT

Significant fluctuations in the price of crude oil in recent years make accurate price estimations of critical importance. A reliable method for crude oil price forecasting is beneficial in guiding production and investment. This study presents two new hybrid predictors for forecasting oil prices based on recurrent neural networks using variational modal decomposition (VMD), sample entropy (SE), and gated recurrent units (GRUs). We consider the West Texas Intermediate daily closing oil prices from August 2, 2010, to December 31, 2019. The hybrid prediction methods first decompose the original time series and then reconstruct the decomposed data using SE. The reconstructed data are used to formulate independent price predictions, which are subsequently combined to produce an ensemble output. With the West Texas Intermediate dataset, the VMD-SE-GRU framework proposed in this paper gives a root mean square error of 0.6735, mean absolute error of 0.4585, mean absolute percentage error of 0.8059, and R^2 value of 0.9272. Overall, the hybrid VMD-SE-GRU framework has several advantages over previous models and produces highly accurate forecasts with a shorter runtime. Thus, the new approach provides an effective tool for predicting oil prices.

1. Introduction

Crude oil is the principal energy source and crucial raw material for the global chemical industry. Thus, forecasting the trends in oil prices is of vital importance. As the futures market handles billions of trades almost instantaneously, it generally behaves as a nonlinear system. Accurate forecasts of oil prices would help reduce the risk of huge losses to investors and allow governments to introduce more effective policies to deal with rises in energy prices. Therefore, it is necessary to establish a stable, accurate, and widely applicable forecasting model for oil prices to help cope with possible market risks.

Numerous price forecasting models have been developed, and considerable efforts have been devoted to finding a suitable model for oil prices. Several such models use traditional analysis methods, e.g., moving average, exponential smoothing, and autoregressive integrated moving average (ARIMA), but these approaches are generally only appropriate for linear systems, and might be unreliable for nonlinear and high-frequency volatile price markets (De Faria et al., 2009; Fong & Yong, 2005; Guo, 2019). Recently, attempts have been made to exploit machine learning and deep learning techniques (Gao et al., 2022; Gumus & Kiran, 2017; Zhao et al., 2017). In particular, support vector machines (SVMs) are superior to traditional approaches when dealing with small sample sizes, nonlinearity, and high dimensionality, and have good generalization ability. Deep learning techniques have also been considered for oil price forecasting (Xie et al., 2006). Most deep learning models mimic the activity of biological systems, such as genetic algorithms (GAs) (Guo et al., 2012), which are based on natural selection and inheritance, and convolutional neural networks (CNNs) (Chen & Huang, 2021), which are based on the structure of the nervous system.

To achieve more efficient and accurate crude oil price forecasting, many composite prediction methods have been proposed, including composite machine learning/deep learning prediction methods and decomposition–reconstr- uction methods using signal decomposition (Kristjanpoller & Minutolo, 2016; Li, Wang, Wei & Zhu, 2021). Because the crude oil index consists of nonlinear complex time series data, we believe that it contains all the information needed for neural network prediction. However, previous studies have found that single neural network models do not fully utilize the information contained within the raw data when making predictions (Selvin et al., 2017). Hence, a decomposition–reconstruction model might be more efficient (Cao et al., 2019; Wang, Cao, Yuan & Cheng, 2021). The

* Corresponding author.
E-mail addresses: zhangshiqi2020@mail.sdu.edu.cn (S. Zhang), luojing0713@mail.sdu.edu.cn (J. Luo), shuyuanwang@mail.sdu.edu.cn (S. Wang), liufeng@sdu.edu.cn (F. Liu).

https://doi.org/10.1016/j.eswa.2023.119617
Received 19 October 2022; Received in revised form 31 December 2022; Accepted 26 January 2023
Available online 30 January 2023

decomposition–reconstruction method decomposes the raw time series into several independent components, forecasts these components separately and then integrates the results. This approach has been widely used in areas such as stock markets, complex flood control systems, carbon pricing, and performance prediction for combined cooling/heating/power systems and wind power systems (Meng et al., 2021; Wang, Sun, Cheng & Cui, 2021; Wei et al., 2021; Xu et al., 2015). Overall, this approach exhibits good generalization and prediction ability. This paper describes a hybrid method for forecasting oil prices based on the decomposition–reconstruction approach. Different deep learning methods and signal decomposition methods are combined to produce optimal predictions. The oil price is considered as a common wave signal, and empirical mode decomposition (EMD) (Huang et al., 1998), complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) (Torres et al., 2011), and variational modal decomposition (VMD) (Dragomiretskiy & Zosso, 2013) are used to reduce the degree of error and remove redundant information from the signal. The deep learning method is based on long short-term memory (LSTM) units (Hochreiter & Schmidhuber, 1997) and gated recurrent units (GRUs) (Cho et al., 2014), and is thus not subject to gradient explosion and gradient disappearance problems. These techniques can predict events in the case of long intervals and temporal lags, making them highly suitable for oil price time series. To ensure satisfactory signal decomposition–reconstruction, the sample entropy (SE) method (Richman & Moorman, 2000) is also adopted.

Compared with previous studies, we believe that the proposed method is highly original and superior to existing price prediction methods.

(1) The originality of this study lies in the proposal of two hybrid forecasting frameworks: VMD-GRU and VMD-SE-GRU. VMD-GRU achieves strong forecasting performance, solving the problem of incomplete utilization of information in previous methods. The VMD-GRU framework is further optimized by SE to obtain the VMD-SE-GRU decomposition–reconstruction forecasting framework. VMD-SE-GRU provides a good explanation of the meaning of the combined signals. We apply the proposed models to the West Texas Intermediate (WTI) oil price data to test the performance of the hybrid framework. The experimental results show that the VMD-SE-GRU framework possesses better performance than previous models in terms of accuracy and runtime.

(2) The superiority of the VMD-SE-GRU framework is demonstrated through five experiments, including comparisons between different base forecasting methods, different hybrid forecasting methods, different algorithms, different signal reconstruction methods, and forecasts of other oil market prices. The experimental results show that the VMD-SE-GRU framework outperforms other methods in terms of four model evaluation metrics.

(3) In the signal reconstruction stage, multiple ways of reconstructing the signal are employed to verify the rationality of introducing SE. In the related simulation experiments, we examine signal reconstruction without SE, signal reconstruction with SE, and random signal reconstruction. The experimental results show that the introduction of SE significantly reduces the runtime of the hybrid framework and achieves similar accuracy to that of the VMD-GRU hybrid model.

(4) The VMD-SE-GRU hybrid forecasting framework performs well on the WTI and Brent crude oil price data. For the former, the proposed model gives a root mean square error (RMSE) of 0.6735, mean absolute error (MAE) of 0.4584, mean absolute percentage error (MAPE) of 0.8059, and R^2 value of 0.9272; for the latter, RMSE = 0.2265, MAE = 0.1849, MAPE = 0.3006, and R^2 = 0.9732. Overall, the VMD-SE-GRU hybrid prediction framework effectively improves the prediction accuracy and exhibits good generalization ability.

The remainder of this paper is structured as follows. Section 2 summarizes previous attempts at forecasting oil prices. Section 3 presents the rationale for the core approach used in this paper, including EMD, CEEMDAN, VMD, SE, LSTM, GRU, and the model evaluation metrics. Section 4 analyzes the original data and introduces two new framework structures for oil price forecasting, namely VMD-GRU and VMD-SE-GRU. Section 5 compares the performance of different decomposition–reconstruction forecasting frameworks and various data processing methods under the same framework. Section 6 presents a summary of this study and discusses the shortcomings of the methodology, with the aim of providing some guidance for future research.

2. Existing literature

Price volatility in the oil market is a concern for a wide range of investors. Knowing and predicting the trend of oil prices can greatly improve the success rate of trading, and the reasonable prediction of oil price changes would allow the risks involved in oil import and export transactions to be identified (Chen et al., 2021). However, oil prices are highly volatile, and improving the accuracy of their forecasting is a challenging problem (Kristjanpoller & Minutolo, 2016). Previous studies have developed many models for predicting oil prices. From a methodological point of view, these methods can be divided into traditional statistical models, classical machine learning models, and deep learning models. From a model perspective, they can be divided into single and hybrid forecasting models (Cheng et al., 2022; Safari & Davallou, 2018).

A single forecasting model uses only one type of model to predict oil prices. This may be either a traditional statistical model or a deep learning model. Among the studies to have used traditional statistical approaches, Zhao and Wang (2014) applied the ARIMA technique to oil price forecasting and found that the ARIMA$(4;3;0)$ model gave reasonable predictions of the world average annual oil price, with a MAPE of 4.059. Zhang et al. (2019) showed that the generalized autoregressive conditional heteroskedasticity (GARCH) model performs well in oil price forecasting. The GARCH(1, 1) model was found to be optimal for forecasting the day-after oil price, with an MAE of 1.6459 and an MSE of 3.5990. Although the above models achieve good short-term forecasts, their long-term results are often unsatisfactory, and they require a huge amount of raw data.

To achieve highly accurate oil price predictions, various machine learning models have been developed. Khashman and Nwulu (2011) used an SVM to predict oil prices and obtained a prediction accuracy of around 81%, superior to statistical methods such as ARIMA. Guo et al. (2012) proposed an improved GA-based SVM model that deals with the overfitting and underfitting of traditional SVMs and obtained better prediction accuracy, with an RMSE of 0.858. Herrera et al. (2019) used a random forest model to predict the long-term oil price, and achieved an RMSE of 7.8334 and a MAPE of 8.1483, they also found that machine learning has the potential to predict data turning points.

While deep learning networks help with long-term predictions, Yahşi et al. (2019) have argued that predictions using artificial neural networks are ineffective. With deep learning models, the predictions are strongly sample-dependent, and the generalization ability of these models is closely related to the sample data used for learning. If the samples are weakly representative, the prediction results are unlikely to reach the expected quality.

Extensive experiments have shown that hybrid prediction models may produce more accurate time series predictions. There are several variants of such models. For example, Wang, Song and Li (2018) developed nonlinear gray models based on psychological concepts to create a nonlinear metabolic gray model (NMGM), and combined this with an ARIMA model to achieve an MAE of 0.163, MAPE of 1.307, and RMSE of 0.224. Wang, Zhao et al. (2018) integrated a data fluctuation network (DFN) and an artificial intelligence (AI) algorithm, and reported a MAPE of 0.069 and an RMSE of 4.170. The results from these models indicate that, compared with a single model, hybrid models achieve better prediction results. However, such models consume more resources. As hybrid models have several advantages over single

Table 1
Advantages and disadvantages of different forecasting models.

Category	Model	Advantages	Disadvantages	Related work
Traditional predictive models	ARIMA; GARCH	• Broad application. • Simple to construct models.	• Require extensive data support. • Lack of long-term forecasting capability.	Zhao and Wang (2014) Zhang et al. (2019)

Machine learning models	SVM;
	GA-SVM;
	Random forest
		• Have the ability to capture inflection points.
		• Reduced risk of over-fitting and underfitting.
			• Dependent on the dataset.
			• Weak ability to handle non-linear data.
				Khashman and Nwulu (2011)
				Guo et al. (2012)
				Herrera et al. (2019)

Deep learning models	NMGM-ARIMA;
	DFN-AI;
	CNN-LSTM;
	CNN-GRU
		• High model accuracy.
		• Deal with nonlinear datasets.
		• Capture the long-term nature of the data.
			• Insufficient model interpretability.
			• Require significant computing resources.
				Wang, Song and Li (2018)
				Wang, Zhao et al. (2018)
				Zang et al. (2020)
				Luo et al. (2022)

Hybrid decomposition–reconstruction models	EEMD-EELM;
	VMD-ICA-ARIMA;
	EEMD-LSTM
		• Generally better performance than single forecasting models.
		• Low dataset dependence.
			• Require significant computing resources
			• Incomplete utilization of decomposed signals.
				Yu et al. (2016)
				Jianwei et al. (2017)
				Wu et al. (2019)

Proposed framework	VMD-SE-GRU	• Hybrid decomposition–reconstruction prediction framework. • SE guides signal reconstruction. • Reduced computational runtime.	• Parameter settings of VMD need to be optimized.	

models, various deep learning-based hybrids have been developed for time series prediction.

Considering the long-term nature of the data that can be captured by deep learning methods, CNNs and their variants have been extensively used in hybrid prediction models. Zang et al. (2020) proposed a spatiotemporal correlation model based on a hybrid CNN-LSTM to predict solar irradiance. This achieved an MAE of 41.88, RMSE of 78.17, and R-squared of 0.9667. Luo et al. (2022) proposed a price prediction model based on a multi-round CNN-GRU approach, combining the advantages of CNNs and GRUs in capturing long-term memory. This model produced an MAE of 4.136, RMSE of 6.456, and MAPE of 0.006. However, few studies have used hybrid frameworks to predict oil prices, particularly hybrid deep learning frameworks.

One widely used hybrid forecasting model is the decomposition–reconstru- ction approach, which is based on the assumption that the raw time series data contain all the information needed for forecasting. Decomposition–reconstruction methods have been widely used for forecasting oil prices. Because oil prices are nonlinear and volatile, signal decomposition methods are required to decompose the raw data into multiple simple signals, from which the characteristics of the data can be extracted. Methods for time series decomposition include the wavelet transform, VMD, EMD, and CEEMDAN (Zhou et al., 2022).

Yu et al. (2016) used a hybrid ensemble empirical mode decomposition (EEMD)–extended extreme learning machine (EELM) approach to predict crude oil spot prices. This performed better than single learning models and similar ensemble models, with a MAPE of 0.0001 and RMSE of 0.0094. Jianwei et al. (2017) combined VMD, independent component analysis (ICA), and ARIMA for crude oil prices, and reported an MAE of 0.020, MAPE of 16.131, and RMSE of 0.035, better than the single ARIMA model and the EEMD-ICA-ARIMA model. Wu et al. (2019) combined deep learning with signal decomposition in an EEMD-LSTM hybrid, which achieved an RMSE of 0.953 and a MAPE of 0.010. These findings show that the introduction of deep learning improves the results compared with traditional methods. All of the above studies demonstrate the feasibility of using decomposition–reconstruction models for oil price forecasting.

Based on previous studies, the advantages and disadvantages of each forecasting approach are summarized in Table 1. In the present study, we combine different signal decomposition methods with deep learning models and use SE to improve the signal reconstruction process. Specifically, we propose a decomposition–partial reconstruction–prediction–reconstruction framework. The prediction performance of this framework is comparable to that of decomp- osition–prediction–reconstruction, but requires less time for implementation.

3. Methods

In this section, we describe the principles of the main methods used in the proposed approach, namely EMD, CEEMDAN, VMD, GRUs, and LSTM. We also present the method and metrics used for model evaluation.

3.1. EMD and CEEMDAN

Huang et al. (1999) proposed the EMD technique for the decomposition of nonlinear and fluctuating time series data using the Hilbert–Huang transform (HHT). The principle of EMD is to decompose the signal of interest into single-component signals, each of which contains only one mode of oscillation. These decomposed components are called intrinsic mode functions (IMFs), and must each satisfy the following conditions: (1) the number of extreme values and the number of zero-crossing points must be equal throughout the dataset, or differ by at most one; (2) at any point, the average value of the envelope defined by the local maximum and the local minimum is zero. When these conditions are satisfied, most of the IMFs can be defined, but sometimes the decomposition of the original data generates IMF components with confounding modes, resulting in a loss of meaning of some of the IMFs (Huang et al., 1999). In this case, the raw signal $x(t)$ $(t = 1, 2, \dots, n)$ can be decomposed using the following steps:

1. Find all the local extrema of $x(t)$, including local maxima and local minima.
2. Connect all local maxima and minima using cubic splines to form the upper envelope $x_{\mathrm{up}}(t)$ and lower envelope $x_{\mathrm{low}}(t)$, respectively, and then compute the mean $m(t)$ point-by-point as

$$m(t) = [x_{\mathrm{up}}(t) - x_{\mathrm{low}}(t)]/2. \tag{1}$$

3. Define the components

$$h(t) = x(t) - m(t). \tag{2}$$

4. Check the properties of $h(t)$:

   (a) if $h(t)$ satisfies the above two conditions, obtain the IMF. Then, replace the raw signal by the residual $r(t)$:

   $$r(t) = x(t) - h(t). \tag{3}$$

   (b) If $h(t)$ does not satisfy the above two conditions, replace $x(t)$ with $h(t)$.

5. Repeat steps 1–4 until $r(t)$ becomes a monotonic function and no more IMFs can be partitioned. From the above formulas, we then have

$$x(t) = \sum_{j=1}^{n} c_j(t) + r_n(t). \tag{4}$$

As the IMF components obtained from EMD suffer from modal mixing, we also use the CEEMDAN method (Torres et al., 2011), which is based on EMD. Compared with EMD, CEEMDAN eliminates mode mixing in a more effective way, exhibits little reconstruction error, and significantly reduces the computational cost (Cao et al., 2019).

Let $x(t)$ be the raw signal. Let $EMD_j(\cdot)$ be the operator that obtains the jth mode through EMD, set $w_i(t)$ as white noise that follows the normal distribution $N(0,1)$, and denote the noise coefficient by ε_0. The CEEMDAN procedure is then performed as follows:

1. Decompose each $x_i(t) = x(t) + \varepsilon_0 w_i(t)$ $(i = 1, 2, \ldots, n)$ by EMD to obtain the first IMF, and then define the first mode as

$$\overline{IMF_1} = \frac{1}{n} \sum_{i=1}^{n} IMF_{i1}. \tag{5}$$

2. Calculate the first residue

$$r_1(t) = x(t) - \overline{IMF_1}. \tag{6}$$

3. Decompose the first residue as

$$r_1(t) + \varepsilon_1 EMD_1(\omega_i(t)), \tag{7}$$

and then obtain the second mode as

$$\overline{IMF_2} = \frac{1}{n} \sum_{i=1}^{n} IMF_1(r_1(t) + \varepsilon_1 EMD_1(\omega_i(t))). \tag{8}$$

4. For the remaining IMFs, repeat steps 1–3 until the final residue $R_M(t)$ is obtained:

$$R_M(t) = x(t) - \sum_{j=1}^{M} \overline{IMF_j}, \tag{9}$$

where M is the total number of IMFs.

3.2. VMD

VMD is a time–frequency analysis method in which time series data are decomposed into several series (Dragomiretskiy & Zosso, 2013). To obtain each component, VMD uses an iterative search to find the optimal center frequency and finite bandwidth. This allows us to effectively separate the signal in a one-time, adaptive manner, avoiding the endpoint effects and false component problems that can occur during iterative processes. Through this operation, VMD achieves reliable signal decomposition.

The steps of VMD are as follows:

1. Let f be the raw signal and denote by $u_k(t)$ the kth IMF. Apply the HHT to the function $u_k(t)$ and calculate the unilateral spectrum. The unilateral spectrum is then transformed to the base band corresponding to the estimated frequency:

$$\left[\left(\delta(t) + \frac{j}{\pi t}\right) u_k(t)\right] e^{-j\omega_k t}, \tag{10}$$

where $\delta(t)$ is the impulse function, $j = \sqrt{-1}$ is the imaginary unit, and ω_k is the corresponding center frequency.

2. Calculate the L^2 norm of the demodulated signal gradient and estimate the bandwidth of each modal signal:

$$\begin{cases} \min\limits_{\{u_k,\omega_k\}} \left\{ \sum\limits_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t}\right) u_k(t)\right] e^{-j\omega_k t} \right\| \right\} \\ \text{s.t.} \ \sum\limits_{k=1}^{K} u_k = f \end{cases} . \tag{11}$$

3. To transform the constrained problem into an unconstrained variational problem, we introduce a Lagrange multiplier operator λ and a second penalty factor α. This allows us to apply variational techniques to solve the problem. Within the limits of the desired accuracy, the Lagrange operator maintains a tight constraint. The following augmented Lagrange expression is thereby obtained:

$$L(\{u_k\}, \{\omega_k\}, \lambda) = \alpha \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t}\right) u_k(t)\right] e^{-j\omega_k t} \right\| + \left\| f(t) - \sum_{k=1}^{K} u_k(t) \right\|^2 + \left\langle \lambda(t), f(t) - \sum_{k=1}^{K} u_k(t) \right\rangle \tag{12}$$

Finally, through repeated iterations, we obtain the mode function $u_k(t)$ and the corresponding center frequency ω_k.

3.3. LSTM and GRUs

LSTM units and GRUs are neural network structures developed from recurrent neural networks (RNNs). Conventional RNNs suffer from directional loops during information transfer, which can easily lead to gradient explosion and gradient disappearance problems (Rather et al., 2015). To deal with this, Hochreiter and Schmidhuber (1997) proposed an LSTM neural network. The logic of LSTM and GRU operations is shown in Fig. 1.

3.3.1. LSTM

In contrast to RNNs, LSTMs have a gate mechanism to control the amount and rate of information accumulation, thereby allowing long-term dependencies to be identified. The LSTM network introduces a forgetting unit and a memory unit to the hidden layer, enabling the selective retention of important information and the forgetting of incidental information (Hochreiter & Schmidhuber, 1997). The three gate structures used in LSTM units are as follows:

1. Forgetting gates f_t: primarily designed to determine what information will be discarded from the cell state; in other words, what important information is kept and what unimportant information is forgotten.
2. Input gates i_t: sometimes referred to as the update gates, these determine how much new information can be added to the cell state.
3. Output gates o_t: used to determine what value is to be output. This output is based on a filtered cell state.

The formulas that control the LSTM unit are as follows:

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ \tilde{C}_t &= \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \\ C_t &= f_c * C_{t-1} + i_t * \tilde{C}_t, \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ h_t &= o_t * \tanh(C_t). \end{aligned} \tag{13}$$

In Eq. (13), upper-case variables represent matrices and lower-case variables represent vectors. x_t is the input vector for the LSTM network at time t. h_t represents the state of the hidden layer of the LSTM network at the current time, while h_{t-1} represents the state at the previous time. f_t, i_t, and o_t denote forgetting gates, input gates, and output gates, respectively. The sigmoid function $\sigma(x) = 1/(1+e^{-x})$ and hyperbolic tangent function $\tanh(x) = (e^x - e^{-x})/(e^x + e^{-x})$ are known as SoftMax functions. W_f, W_i, W_c, and W_o are the weights for each layer of the network, while b_f, b_i, b_c, and b_o are the biases for different gates and cell states. The symbol $*$ represents the Hadamard product.

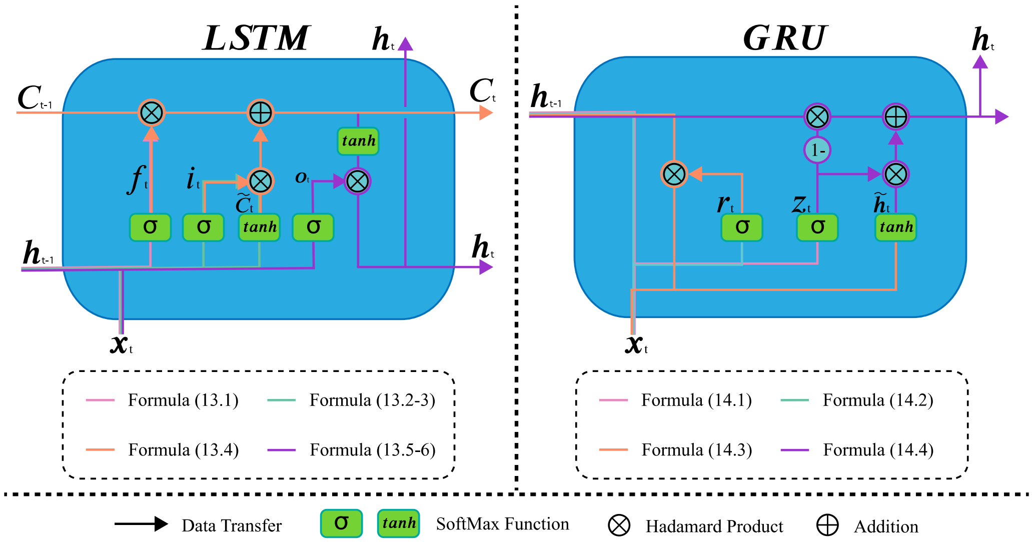


Fig. 1. LSTM and GRU structures.

In this study, we introduce a dropout mechanism to avoid overfitting in the experiments. We also use the Adam optimization algorithm, which is simple and efficient, and can adaptively adjust the size of the training weights, making it ideal for prediction problems with nonlinear complex time series.

3.3.2. GRUs

GRU neural networks were first proposed to deal with problems related to machine translation (Cho et al., 2014a), after which they were applied in finance (Sethia & Raut, 2019). A GRU network can solve the gradient explosion and gradient disappearance problems that arise in RNNs, and simplifies the complex LSTM computations. The formulas for controlling GRUs are as follows:

$$\begin{aligned} z_t &= \sigma(W_z \cdot x_t + U_z \cdot h_{h-1} + b_z), \\ r_t &= \sigma(W_r \cdot x_t + U_r \cdot h_{h-1} + b_r), \\ \tilde{h}_t &= \tanh[W_h \cdot x_t + U_h(r_t * h_{t-1})], \\ h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t. \end{aligned} \tag{14}$$

In Eq. (14), upper- and lower-case variables represent matrices and vectors, respectively. x_t is the input vector for the GRU network at time t. h_t and h_{t-1} represent the hidden layer states of the GRU network at the current and previous times, respectively. z_t and r_t are the update and reset gates, respectively. The sigmoid function $\sigma(x)$ and hyperbolic tangent function $\tanh(x)$ are SoftMax functions. $W_{\cdot}$ and $U_{\cdot}$ are the weights for each layer of the network, while $b_{\cdot}$ is the bias vector. $\tilde{h}_t$ and h_t are the candidate vector and output vector, respectively, of the GRU.

GRUs have a different gate structure from LSTM units:

1. The update gate z_t controls the amount of past information that can be known in the future, helping to prevent gradient disappearance.
2. The reset gate r_t determines which past information should be forgotten.

The most important aspect of GRU neural networks is that the forgetting and remembering steps are performed simultaneously. In the equation for h_t, the z term that controls the information to be forgotten is associated with the $(1 - z)$ term of the selected memory, and the forgetting and memory weights remain constant. This design makes the GRU structure simpler than that of LSTM units, with no reduction in computational accuracy.

3.4. SE

SE is a measure of the complexity of a time series. This is quantified by determining the probability of generating a new pattern in the signal based on approximate entropy (Richman & Moorman, 2000). SE has two advantages over approximate entropy: it is not influenced by the length of the data and it has improved consistency. A lower SE value indicates a higher level of self-similarity in a sequence, while a higher SE value indicates a more complex sequence of samples.

The SE is calculated as follows:

1. The time series $x(n)$ consisting of N data forms a vector of dimension m according to the ordinal number.
2. The distance $d[X_m(i), X_m(j)]$ between $X_m(i)$ and $X_m(j)$ is defined as the absolute value of the maximum difference between corresponding elements:

$$d[X_m(i), X_m(j)] = \max[|x(i+k) - x(j+k)|]. \tag{15}$$

3. Given a threshold r, for each i, determine the ratio $B_i^m(r)$ of the number of $d[X_m(i), X_m(j)]$ that are less than r to the total number of distances $N - m$:

$$B_i^m(r) = \frac{1}{N-m-1}\{d[X_m(i), X_m(j) < r]\}, \\ i = 1, 2, \dots, N-m+1;\ i \neq j, \tag{16}$$

4.

$$B^m(r) = \frac{1}{N-m+1}\sum_{i=1}^{N-m+1} B_i^m(r). \tag{17}$$

5. Increase the number of dimensions to $m+1$ and calculate the value of Q, i.e., the number of $X_{m+1}(i)$ and $X_{m+1}(j)$ that are less than or equal to r, denoted by A_i:

$$A_i^m(r) = \frac{1}{N-m-1}A_i, \tag{18}$$

$$A^m(r) = \frac{1}{N-m}\sum_{i=1}^{N-m} A_i^m(r). \tag{19}$$

6. The SE is defined as

$$S_n(m,r) = \lim_{N\to\infty}\left\{-\ln\left[\frac{A^m(r)}{B^m(r)}\right]\right\}. \tag{20}$$

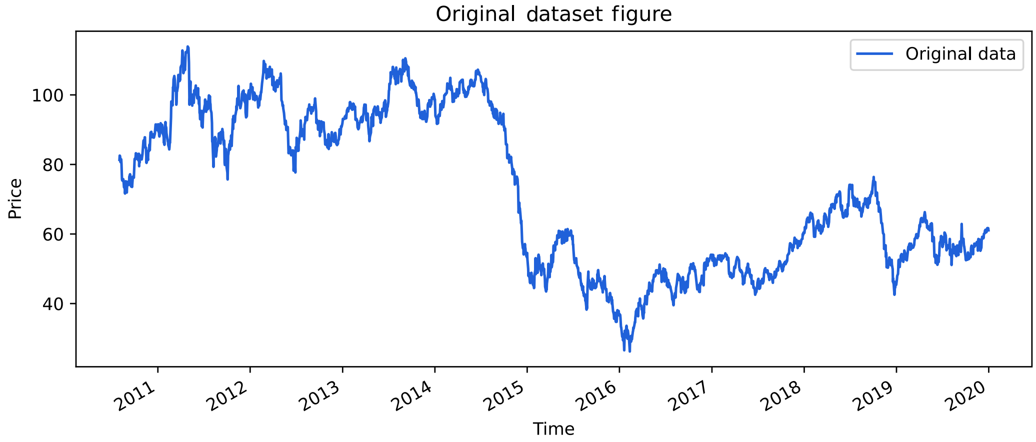


Fig. 2. Original dataset.

3.5. Normalization and evaluation

3.5.1. Min–max normalization

To enhance the efficiency of the model and minimize the influence of the magnitude of various indicators on the prediction results, normalization is applied to restrict the data to the range $[0,1]$. This helps to eliminate the negative effects of irregular sample data on model performance. The normalization formula is as follows:

$$x = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \tag{21}$$

where $x_{\min}$ and $x_{\max}$ are the maximum and minimum values in the raw time series data.

3.5.2. Model evaluation

To measure the prediction accuracy of our proposed model, we use four evaluation metrics: RMSE, MAE, MAPE, and R^2. They are defined as follows:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(f(x_i) - y_i)^2}, \tag{22}$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|f(x_i) - y_i|, \tag{23}$$

$$MAPE = \frac{100}{m}\sum_{i=1}^{m}\left|\frac{y_i - f(x_i)}{y_i}\right|, \tag{24}$$

$$R^2 = 1 - \frac{RSS}{TSS},$$
$$RSS = \sum_{i=1}^{m}[f(x_i) - y_i]^2, \tag{25}$$
$$TSS = \sum_{i=1}^{m}[f(x_i) - \bar{y}]^2.$$

Here, $f(x_i)$ and y_i are the test value and the predicted value, respectively, and m is the size of the raw dataset. RMSE and MSE measure the accuracy of the training set. For these metrics, lower values indicate a better fit. MAPE is the average absolute difference between the actual and predicted values, expressed as a percentage. A lower MAPE value indicates a smaller prediction error. The coefficient of determination R^2 measures how well the regression equation fits the observations. A value closer to 1 indicates better model performance.

4. Fundamental frameworks and predictor establishment

Machine learning and deep learning have been widely used in time series forecasting. Experiments typically involve data collection, data preprocessing, construction of training models, evaluation of experimental data, and model analysis. We introduce the experimental data in Section 4.1, the data preprocessing method in Section 4.2, and the main forecasting framework in Section 4.3.

4.1. Data

WTI crude oil is one of the three major pricing benchmarks in the global oil market and is widely traded by investors. Data on the daily WTI crude oil prices were collected from the US Energy Information Administration (http://www.eia.doe.gov/). The sample data include 2446 WTI closing prices from August 2, 2010, to December 31, 2019, as shown in Fig. 2. A total of 2346 WTI closing price data (from August 2, 2010, to August 21, 2019) were used as the training set, and the remaining 100 WTI closing price data (from August 22, 2019 to December 31, 2019) were used as the test set. These data are the daily closing prices of the WTI; we assume that the WTI prices are continuous and correlated with the previous 30 days.

The smoothness, autocorrelation, and normality of the dataset were verified using the "statsmodels" and "SciPy" modules in Python. First, the augmented Dickey–Fuller test (ADF) was used to test the stability of the data. The zeroth-order ADF results show a p-value of 0.594, which indicates that the original hypothesis can be accepted. The zeroth-order ADF test value is -1.375, which is greater than the critical value of -2.863 at the 5% significance level. This indicates that the WTI price is an unsteady time series in terms of its zeroth-order difference. The first-order ADF results show a p-value of 0.000 and an ADF test value of -51.838, indicating significance at this level; the time series is smooth at this point. Subsequently, we determine the autocorrelation of price using the autocorrelation function (ACF) and partial autocorrelation function (PACF). Fig. 3 indicate a trailing ACF and a two-section truncated PACF, with autocorrelation between prices. Finally, the Shapiro–Wilk test was applied to verify whether the sample data obey a normal distribution. The results show that the p-value is 0.000, the skewness is 0.105, and the kurtosis is -1.44, indicating that the data do not obey a normal distribution.

Data processing shows that the WTI price time series data are smooth, autocorrelated, and do not obey a normal distribution. The data contain a lot of internal noise. Thus, it would be difficult for conventional forecasting models to predict the WTI prices with high accuracy if the raw data were not noise-reduced using correlation methods. The results for the relevant experimental data are presented in Table 2.

4.2. Decomposition–reconstruction principle

Based on the mathematical techniques described in Sections 3.1 and 3.2, the "vmdpy" module in Python was applied together with the "PyEMD" module to decompose the WTI price data. Fig. 4 shows

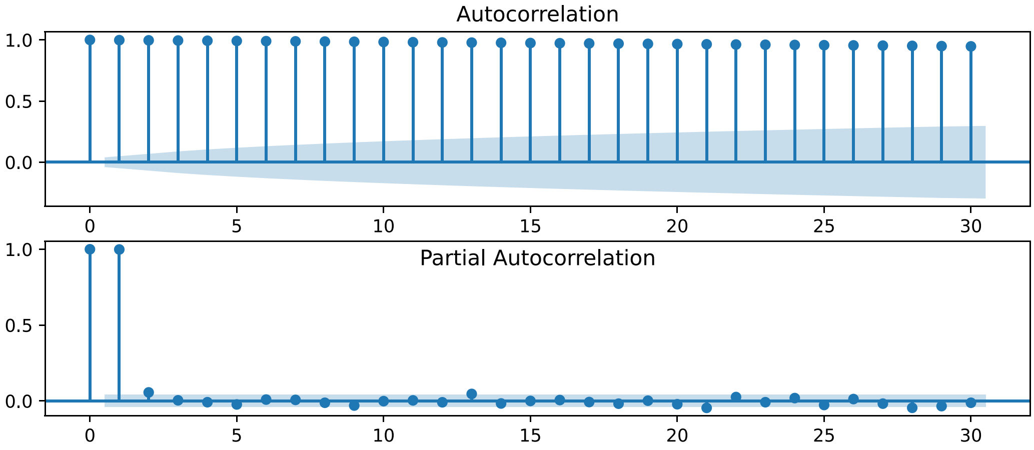


Fig. 3. ACF and PACF of WTI price.

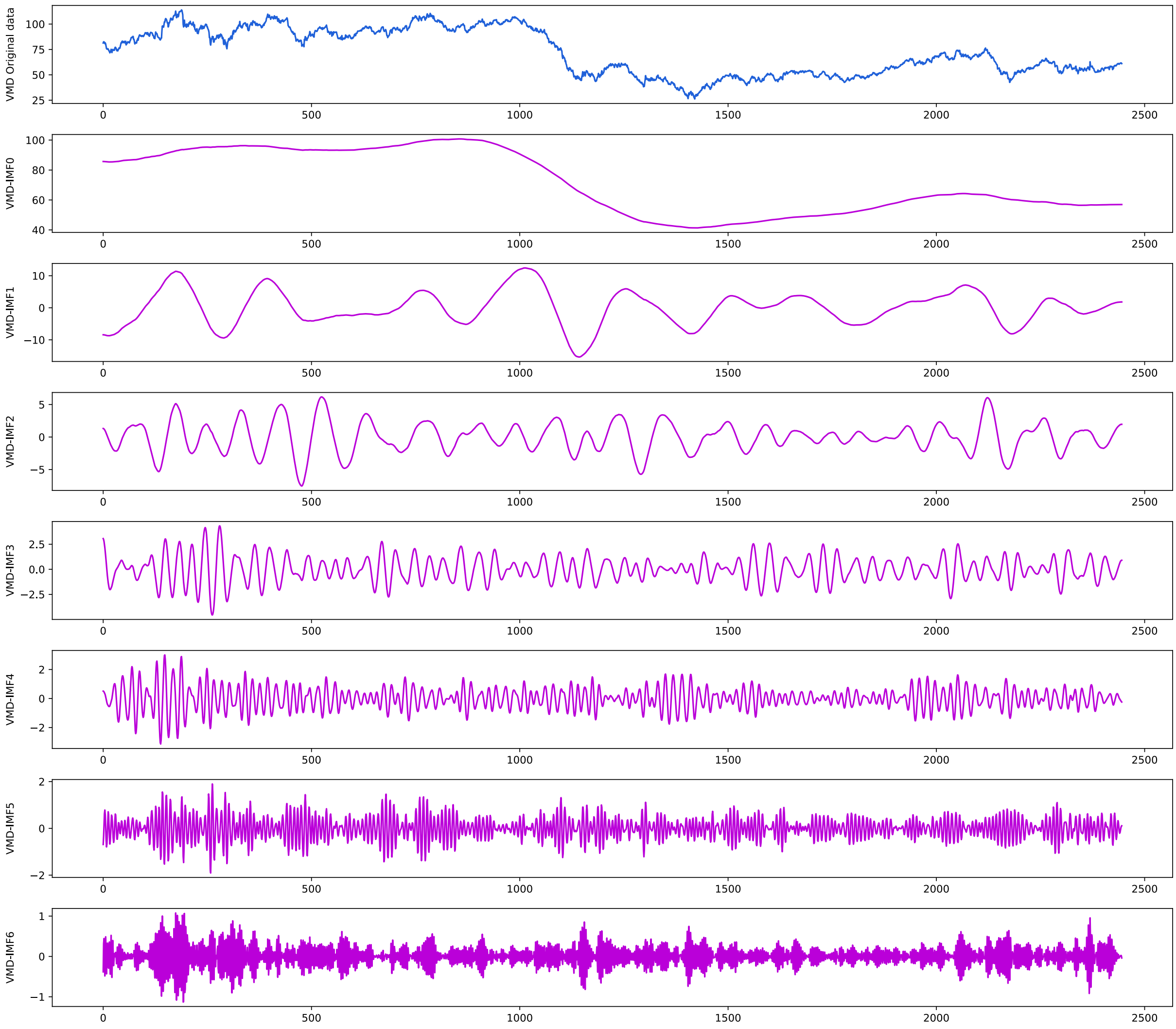


Fig. 4. VMD-IMF signal.

the decomposition results from applying VMD to the original data, illustrating the seven signal components VMD-IMF0, …, VMD-IMF6. Analyzing the decomposition results, we find that the periods of VMD-IMF0 to VMD-IMF2 are significantly longer than those of the other decomposed signals. Thus, we utilize these three components to capture the standing trend of oil prices with a period of 2–3 years, recorded as low-frequency modules (Li, Hu, Heng & Chen, 2021; Zhu et al., 2019). VMD-IMF3 and VMD-IMF4 are medium-frequency modules with a period of 2–4 months, possibly recording oil price fluctuations under the impact of international political and economic events. VMD-IMF5 and VMD-IMF6 have periods of only 3–21 days, potentially reflecting investor speculation with short-term fluctuations in the market.

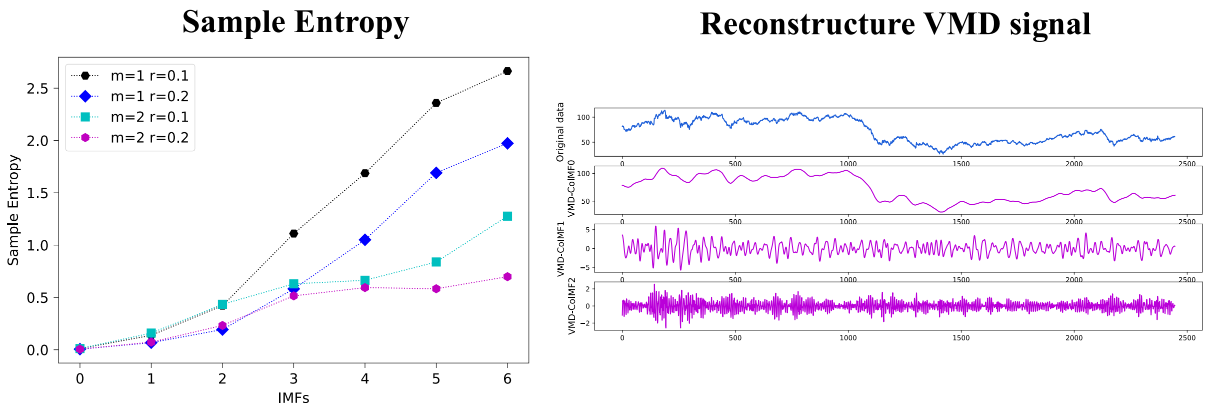


Fig. 5. Sample entropy and VMD-CoIMFs.

Table 2
Statistical description of WTI price data.

Method	*t*-value	*p*-value	5% critical value	Status
ADF: zeroth-order difference	−1.375	0.594	−2.863	N/A
ADF: first-order difference	−51.838	0.000	−2.863	Smooth
ACF	N/A	N/A	N/A	Autocorrelation
PACF	N/A	N/A	N/A	Autocorrelation
Shapiro–Wilk	N/A	0.000	N/A	Non-normality

Based on previous studies, the integration of decomposed signals using SE similarity can reduce computational costs, improve the computational speed of neural networks, and avoid the problem of overfitting (Zhang et al., 2021; Zhou et al., 2022). In fact, a recombination classification based on SE provides a reasonable and reliable basis for the high-, medium-, and low-frequency classification scheme mentioned above. Fig. 5 shows the respective SEs of the VMD subsignals, with signals that have similar entropies grouped to obtain the same classification scheme. The reconstructed signals clearly show the variation of oil prices at different periods, indicating that the neural network can extract signal features for learning.

From the reconstructed signals in Fig. 5, the following can be seen. (1) From 2010 onwards, for the first 1000 trading days, the WTI oil price maintains a high level of oscillation about $80. Both high- and medium-frequency signals exhibit violent fluctuations, indicating that the oil market was dramatically affected by worldwide political and economic factors over this period, including wars and economic sanctions. (2) During the period from trading day 1000–1800, oil prices fell rapidly to approximately $50, dipped below $30, and eventually stabilized at around $50 with some fluctuations. This was due to the development of shale oil extraction technology in the USA, breaking the monopoly of other organizations in the oil world. From the medium- and high-frequency signals, the volatility decreased at this time and supply in the oil market exceeded demand. (3) The period from trading day 1800–2500, the world oil market re-entered a period of equilibrium and oil prices oscillated upwards. The medium- and high-frequency signals suggest gradually increasing volatility, with prices eventually rising back to $60. (4) From the low-frequency signals, the oil market is in a process of slow recovery after a decline, and a longer time is needed for the market to return to smoothness with the development of new technologies.

4.3. Main forecasting framework

4.3.1. VMD-GRU framework

The VMD-GRU framework is a complex structure in which each signal is subjected to VMD decomposition and then the independent components are placed into the GRU neural network for training. The results of the independent training are subsequently merged to obtain the predicted values. We assume that the closing price of the WTI on the current day is related to the closing prices over the past 30 days, and we input these data into the GRU neural network. Fig. 7 shows the prediction results for signals VMD-IMF(·) after training, with the blue line representing the original price and the red line representing the predicted price. The experiments in the current study were mainly conducted using the Python language on the Google Colab platform.

The specific steps of the VMD-GRU framework are described below:

1. Decompose the original time series into several VMD-IMF(·) signals using VMD.
2. Normalize each VMD-IMF(·) and formulate the training set and test set.
3. Input VMD-IMF(·) into the GRU model for independent prediction.
4. Combine the predicted VMD-IMF(·) to recover the final results.
5. Perform inverse normalization of the final results and evaluate the model using the relevant indicators.

On the basis of previous studies and several experiments, the GRU structure shown in Fig. 6 was applied, as this exhibits good performance for WTI oil data, although it is not optimal. Based on the assumptions in Section 4.1, a three-layer GRU structure was chosen, with 256, 128, and 64 cells in the respective layers. Each GRU structure contained a Dropout layer with a factor of 0.2. The structure finally applied a Dense layer to produce the output, with the “tanh” activation function, Adam optimizer, an input data size of $[(\mathrm{None}, 30, 1)]$, and the MSE loss function.

To ensure that the data converged in a reasonable range and to avoid underfitting and overfitting, we applied an adaptive learning rate and early stopping mechanism. Based on the performance of Google Colab, the batch size of the input data was set to 25, the number of epochs was set to 500, and 10% of the data were extracted to form the validation set. The adaptive learning rate was set as follows: if the training loss did not decrease over 50 epochs, the learning rate was reduced to one-tenth of its original value. The early stopping mechanism was set to 300 epochs.

The GRU structure contains a Dropout layer, which leads to some randomness in the training results. Thus, Fig. 8 only shows the results of a particular experiment. Table 3 presents the VMD-GRU results. The RMSE, MAE, MAPE, and R^2 values are 0.6547, 0.4295, 0.7569, and 0.9312, respectively. The total training time is 968.7 s, with good prediction results and no significant lagging problems. However, it is remarkable that the model is unable to make good predictions of sudden rises and falls.

4.3.2. VMD-SE-GRU framework

Compared with the VMD-GRU framework, the VMD-SE-GRU framework is much easier to compute. With the introduction of SE, two data

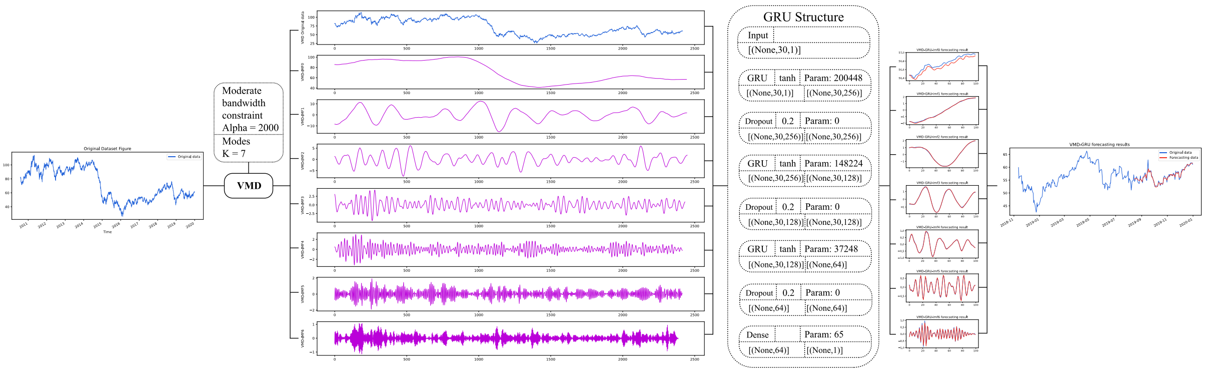

Original Dataset Figure
Original data
Time
Moderate bandwidth constraint Alpha = 2000
Modes K = 7
VMD
VMD Original data
VMD-IMF0
VMD-IMF1
VMD-IMF2
VMD-IMF3
VMD-IMF4
VMD-IMF5
VMD-IMF6
GRU Structure
Input
[(None,30,1)]
GRU | tanh | Param: 200448
[(None,30,1)] | [(None,30,256)]
Dropout | 0.2 | Param: 0
[(None,30,256)] | [(None,30,256)]
GRU | tanh | Param: 148224
[(None,30,256)] | [(None,30,128)]
Dropout | 0.2 | Param: 0
[(None,30,128)] | [(None,30,128)]
GRU | tanh | Param: 37248
[(None,30,128)] | [(None,64)]
Dropout | 0.2 | Param: 0
[(None,64)] | [(None,64)]
Dense | Param: 65
[(None,64)] | [(None,1)]
VMD-GRU forecasting results
Original data
Forecasting data

Fig. 6. VMD-GRU framework structure.

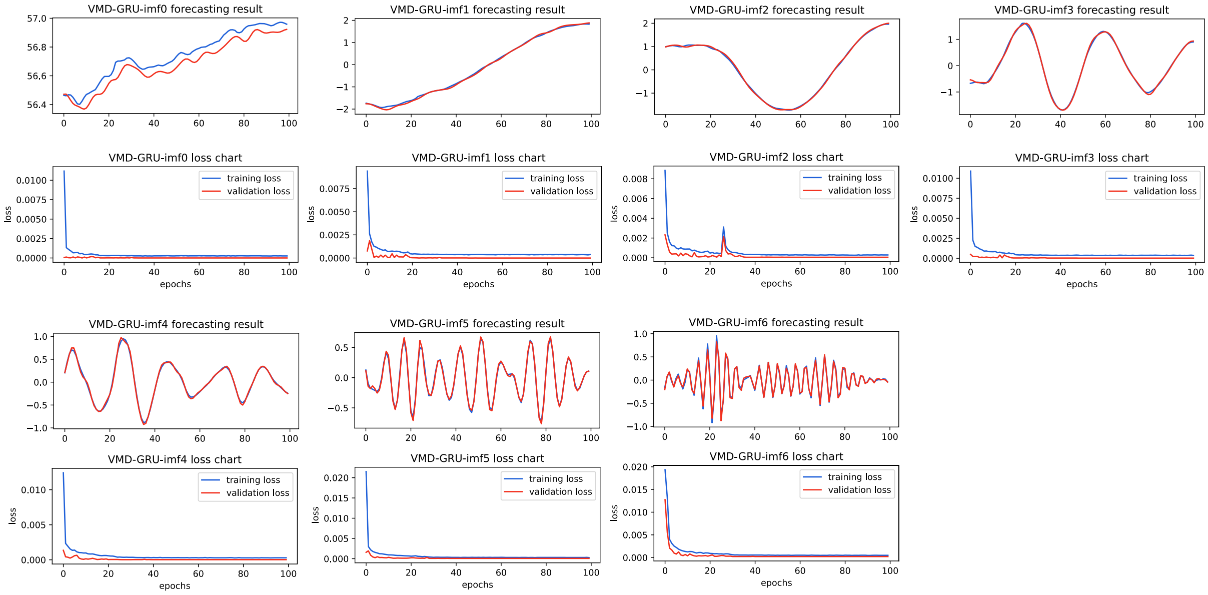


Fig. 7. Forecasting result and loss chart of VMD-IMFs.

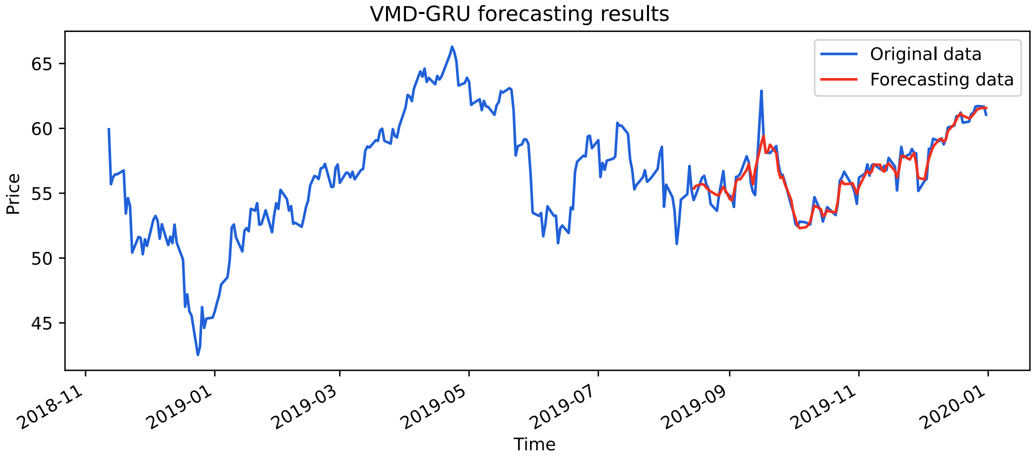
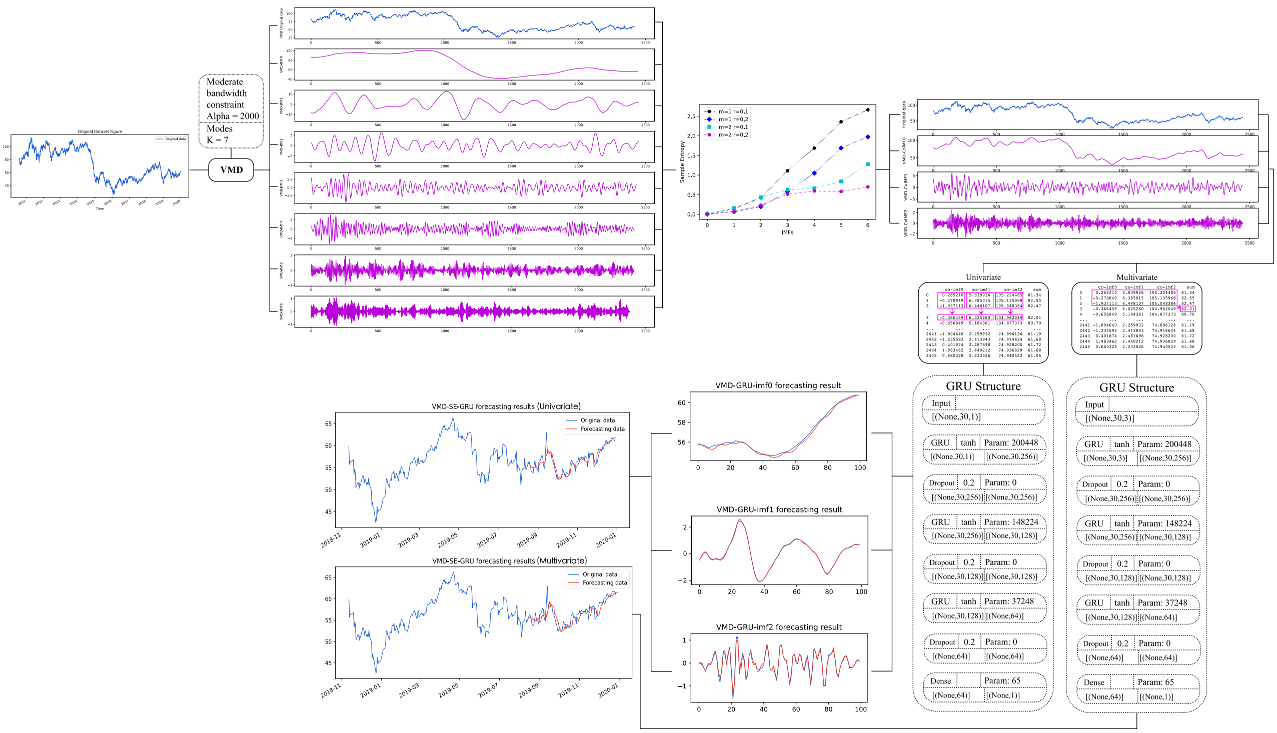


Fig. 8. VMD-GRU forecasting results.

prediction methods are used when the reconstituted data are added to the GRU for prediction: (1) the resynthesized VMD-CoIMF(⋅) signals are imported separately into the GRU for prediction, similar to univariate time series prediction, and the results are eventually combined; (2) the resynthesized VMD-CoIMF(⋅) signals are used, as in a multivariate forecasting model, with each VMD-CoIMF(⋅) considered as a variable factor. These factors are imported together into the GRU to forecast the closing price, giving a direct forecasting result.

The VMD-SE-GRU experimental steps are shown in Fig. 9. As the steps of the univariate VMD-SE-GRU framework are similar to those of the VMD-GRU framework, they are not described in detail here.


Original Dataset Figure
Original data
Time
Moderate bandwidth constraint Alpha = 2000
Modes K = 7
VMD
VMD Original data
VMD-IMF0
VMD-IMF1
VMD-IMF2
VMD-IMF3
VMD-IMF4
VMD-IMF5
VMD-IMF6
m=1 r=0.1
m=1 r=0.2
m=2 r=0.1
m=2 r=0.2
Sample Entropy
IMFs
Original data
VMD-CoIMF0
VMD-CoIMF1
VMD-CoIMF2
Univariate
Multivariate
GRU Structure
Input
[(None,30,1)]
GRU | tanh | Param: 200448
[(None,30,1)] | [(None,30,256)]
Dropout | 0.2 | Param: 0
[(None,30,256)] | [(None,30,256)]
GRU | tanh | Param: 148224
[(None,30,256)] | [(None,30,128)]
Dropout | 0.2 | Param: 0
[(None,30,128)] | [(None,30,128)]
GRU | tanh | Param: 37248
[(None,30,128)] | [(None,64)]
Dropout | 0.2 | Param: 0
[(None,64)] | [(None,64)]
Dense | Param: 65
[(None,64)] | [(None,1)]
GRU Structure
Input
[(None,30,3)]
GRU | tanh | Param: 200448
[(None,30,3)] | [(None,30,256)]
Dropout | 0.2 | Param: 0
[(None,30,256)] | [(None,30,256)]
GRU | tanh | Param: 148224
[(None,30,256)] | [(None,30,128)]
Dropout | 0.2 | Param: 0
[(None,30,128)] | [(None,30,128)]
GRU | tanh | Param: 37248
[(None,30,128)] | [(None,64)]
Dropout | 0.2 | Param: 0
[(None,64)] | [(None,64)]
Dense | Param: 65
[(None,64)] | [(None,1)]
VMD-GRU-imf0 forecasting result
VMD-GRU-imf1 forecasting result
VMD-GRU-imf2 forecasting result
VMD-SE-GRU forecasting results (Univariate)
VMD-SE-GRU forecasting results (Multivariate)
Original data
Forecasting data

Univariate

```
        co-imf0   co-imf1     co-imf2    sum
0      0.265210  5.839926  105.224485  81.34
1     -0.278849  6.385015  105.135968  82.55
2     -1.937113  6.448107  105.048384  82.47

3     -0.366459  6.025260  104.962049  82.01
4     -0.656849  5.184361  104.877373  80.70
...         ...       ...         ...    ...
2441  -1.804640  2.259932   74.896126  61.19
2442  -1.235592  2.413843   74.914624  61.68
2443   0.401874  2.487698   74.928200  61.72
2444   1.983462  2.440212   74.936829  61.68
2445   0.660328  2.233026   74.940522  61.06
```

Multivariate

```
       co-imf0   co-imf1     co-imf2    sum
0     0.265210  5.839926  105.224485  81.34
1    -0.278849  6.385015  105.135968  82.55
2    -1.937113  6.448107  105.048384  82.47
3    -0.366459  6.025260  104.962049  82.01
4    -0.656849  5.184361  104.877373  80.70
...        ...       ...         ...    ...
2441 -1.804640  2.259932   74.896126  61.19
2442 -1.235592  2.413843   74.914624  61.68
2443  0.401874  2.487698   74.928200  61.72
2444  1.983462  2.440212   74.936829  61.68
2445  0.660328  2.233026   74.940522  61.06
```

Fig. 9. VMD-SE-GRU structure.

Table 3
Evaluation of different IMF signals.

Signal	RMSE	MAE	MAPE	R^2	Time (s)
VMD-IMF0	0.0008	0.0007	0.2797	0.9164	N/A
VMD-IMF1	0.0012	0.0009	0.1785	0.9993	N/A
VMD-IMF2	0.0028	0.0024	0.4401	0.9989	N/A
VMD-IMF3	0.0042	0.0033	0.6620	0.9982	N/A
VMD-IMF4	0.0038	0.0030	0.5975	0.9965	N/A
VMD-IMF5	0.0079	0.0061	1.2736	0.9919	N/A
VMD-IMF6	0.0162	0.0116	2.8712	0.9855	N/A
VMD-IMF-SUM	0.6547	0.4295	0.7569	0.9312	968.7

Table 4
Evaluation of different CoIMF signals.

Signal	RMSE	MAE	MAPE	R^2	Time (s)
Univariate					
VMD-CoIMF0	0.0020	0.0016	0.5040	0.9932	N/A
VMD-CoIMF1	0.0040	0.0030	0.6097	0.9979	N/A
VMD-CoIMF2	0.0184	0.0136	2.9495	0.9572	N/A
VMD-CoIMF-SUM	0.6735	0.4585	0.8059	0.9272	429
Multivariate					
VMD-CoIMF-SUM	0.7968	0.5329	0.9423	0.9011	134

The steps of the multivariate VMD-SE-GRU framework are as follows:

1. Apply VMD to the time series data to obtain VMD-IMF(⋅).
2. Import VMD-IMF(⋅) into the SE function for calculation.
3. Integrate VMD-IMF(⋅) into high- and low-frequency signals VMD-CoIMF(⋅) using their respective SEs.
4. Normalize the data and form the training set and test set.
5. Import VMD-CoIMF(⋅) into the GRU in the form of a matrix for multivariate time series prediction.
6. Perform inverse normalization of the final results and evaluate the model using the relevant indicators.

Fig. 9 shows the difference between the univariate and multivariate VMD-SE-GRU frameworks for predictive data processing. This is discussed in more detail in Section 5.3. Owing to the presence of the Dropout layer, the results of each run will be different; only the results of a particular run are given in Table 4. The model parameters used in the experiments are consistent with the VMD-GRU model, and the results of the univariate VMD-SE-GRU model runs are as follows: the RMSE, MAE, MAPE, and R^2 values are 0.6735, 0.4585, 0.8059, and 0.9272, with the run taking 429 s. For the multivariate VMD-SE-GRU model, the corresponding values are 0.7968, 0.5329, 0.9423, and 0.9011, respectively, and the runtime is 134 s. The prediction results of the univariate VMD-SE-GRU model are shown in Fig. 10

Interestingly, the multivariate VMD-SE-GRU model is the least effective—it cannot fit the high-volatility segment accurately and only provides a general trend, but has the shortest runtime. The univariate VMD-SE-GRU model is slightly less effective than the VMD-GRU framework, but captures the lows of the data more clearly, and has only half the runtime of the VMD-GRU framework.

5. Argumentation framework

5.1. Comparison with a single model

To explore whether the advantages of the basic neural network framework and the hybrid neural network framework are consistent, the WTI price index was imported into a GRU, LSTM, deep neural network (DNN), and a backpropagation neural network (BPNN) for training. We used the TensorFlow module in Python to construct the relevant models, thus ensuring that the structure remained largely consistent between the frameworks. Because the neural network structure contains forgetting layers, which can lead to randomness in the predicted results, Table 5 presents the average metrics over 10 runs of each different neural network. The GRU framework performs best with a single original model, achieving an RMSE of 1.268, MAE of

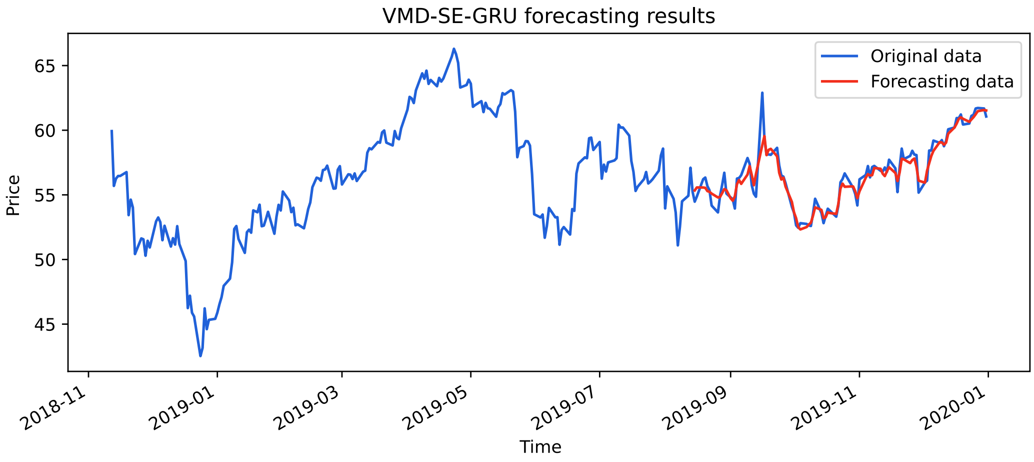


Fig. 10. VMD-SE-GRU forecasting results.

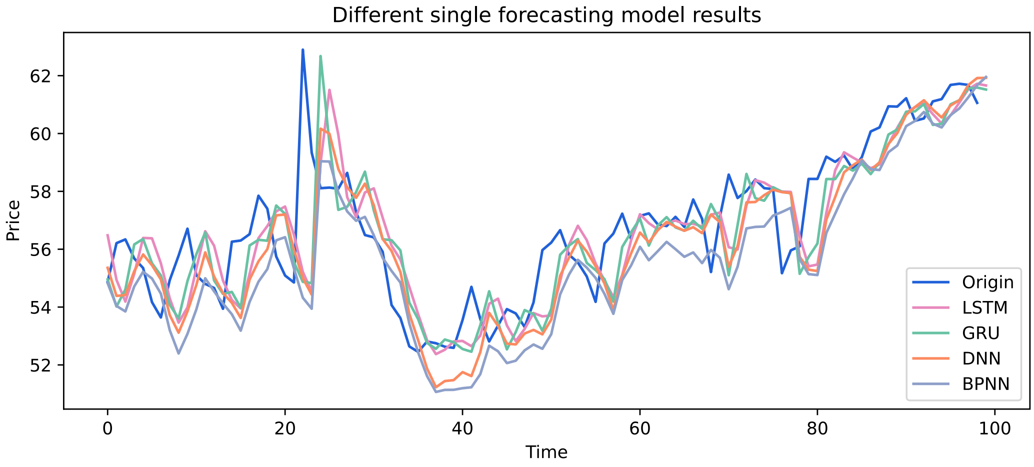


Fig. 11. Different single forecasting model results.

Table 5
Contrast with the basic framework.

Predictor	RMSE	MAE	MAPE	R^2	Time (s)
LSTM	1.402	0.935	1.646	0.685	170.9
GRU	1.268	0.816	1.430	0.742	133.8
DNN	1.481	0.962	1.723	0.626	49.9
BPNN	1.666	1.213	2.131	0.554	82.9

0.816, MAPE of 1.430, and R^2 of 0.742. The LSTM framework also performs well, with an RMSE of 1.402, MAE of 0.935, MAPE of 1.645, and R^2 of 0.685. However, the DNN and BPNN frameworks perform poorly, with R^2 values of 0.626 and 0.554, respectively. As seen in Fig. 11, the single GRU and single LSTM models capture the highest and lowest points of the data well, but there is some temporal lag in their predictions. The DNN and BPNN frameworks also exhibit this drawback, which is likely to lead to missed opportunities when the model is used for price decisions. The predicted values trained by a single DNN and single BPNN often show large deviations from the true values and weak prediction performance for extreme value points.

The final column of Table 5 presents the training time for each individual framework. Both DNN and BPNN take less than 90 s, while GRU and LSTM take 133 s and 170 s, respectively. The training speed of DNN and BPNN means they are suitable when only the future price trend of WTI is of interest. However, when studying the future target price of WTI, the GRU and LSTM network structures should be used.

5.2. *Comparison of different decomposition methods*

From the discussion of single prediction models in Section 5.1, we can conclude that the GRU and LSTM frameworks there are superior to the other frameworks. This subsection discusses the different decomposition models that can be applied. We used the “EMD-signal” and “vmdpy” modules in Python to implement the EMD, CEEMDAN, and VMD methods described in Sections 3.1 and 3.2, and compared their effects on the different prediction models. After the WTI data had been decomposed, each IMF signal was input to the predictor, and after all forecasting operations were complete, the results of each IMF signal were unified and combined to generate the WTI forecasting results. As the predictor must operate on each IMF signal, we set the number of epochs to 100. The specific experimental data are presented in Table 6.

The different decomposition methods can lead to inconsistencies in the number of IMF signals, with eight IMF signals for EMD, eight or nine for CEEMDAN, and seven for VMD. This can cause differences in the computation time of the predictor. From Table 6, it can be seen that VMD-GRU gives the best forecasting results, with an RMSE of 0.654, MAE of 0.429, MAPE of 0.756, and R^2 of 0.931. Thus, this framework is most suitable for forecasting WTI prices. The R^2 values of the EMD and CEEMDAN models are less than 0.900, but their forecasting performance is superior to the single forecasting models. The forecasting models using the VMD method not only achieve R^2 values of 0.931 and 0.925, respectively, but also have RMSEs of 0.654 and 0.681. This suggests that the introduction of VMD helps to improve the accuracy of forecasting WTI prices.

Table 6
Comparison of different decomposition methods and prediction frameworks.

Predictor	RMSE	MAE	MAPE	R^2	Time (s)
EMD-GRU	0.848	0.595	1.051	0.884	863.4
CEEMDAN-GRU	0.884	0.601	1.060	0.874	914.1
VMD-GRU	0.654	0.429	0.756	0.931	968.7
EMD-LSTM	0.868	0.618	1.092	0.878	966.8
CEEMDAN-LSTM	0.834	0.559	0.987	0.888	1008.6
VMD-LSTM	0.681	0.462	0.815	0.925	1020.0

Table 7
Comparison of different prediction models.

Predictor	RMSE	MAE	MAPE	R^2	Time (s)
LSTM	1.402	0.935	1.646	0.685	170.9
GRU	1.268	0.816	1.430	0.742	133.8
VMD-LSTM	0.681	0.462	0.815	0.925	1020
VMD-GRU	0.654	0.429	0.756	0.931	968.7
VMD-SE-LSTM (univariate)	0.817	0.556	0.982	0.892	437.9
VMD-SE-LSTM (multivariate)	0.796	0.532	0.942	0.898	144.2
VMD-SE-GRU (univariate)	0.669	0.451	0.793	0.928	429.3
VMD-SE-GRU (multivariate)	0.783	0.523	0.924	0.901	134

5.3. Comparison of different forecasting models

Section 5.2 has shown that the VMD method provides the greatest improvement in predictor performance. In this subsection, the introduction of SE based on the VMD method is discussed to determine whether the model training time can be shortened while maintaining the prediction accuracy.

After obtaining the VMD-CoIMF(·) signal based on SE, the experimental flow of the univariate VMD-SE-GRU model and the multivariate VMD-SE-GRU model are as shown in Fig. 12. The LSTM model also possesses this experimental flow.

Four decomposition integration models were considered, namely VMD-SE-LSTM (univariate), VMD-SE-LSTM (multivariate), VMD-SE-GRU (univariate), and VMD-SE-GRU (multivariate). These are integrated into a single table to facilitate comparison with the models from Sections 4.3.1 and 4.3.2. Table 7 presents the experimental results. After the introduction of SE, we grouped VMD-IMF0-2, VMD-IMF3-4, VMD-IMF5-6 to synthesize VMD-CoIMF0-2. When the VMD-SE- GRU (univariate) model is used, the average training time for each VMD-CoIMF signal is 134.1 s, and the total runtime is 429.3 s. The RMSE of the VMD-SE-GRU (univariate) model is 0.669, the MAE is 0.451, the MAPE is 0.793, and R^2 is 0.928. Although the prediction accuracy of this model is slightly lower than that of the VMD-GRU model in Section 4.3.1, the prediction time required to achieve satisfactory results is greatly reduced. It is worth noting the following points: (1) all the decomposition–integration–prediction models have worse prediction performance than the decomposition–prediction model; (2) the LSTM performs worse than the GRU and has a longer runtime when using the WTI dataset; (3) when the integrated (multivariate) prediction method is used, the LSTM and GRU predictors perform equally well, taking 144.2 s and 134 s, respectively, but the prediction accuracy is still not sufficient for practical application.

5.4. Comparison between different reconstruction models

Fig. 12 indicates that, in the VMD-SE environment, the low-frequency sequence VMD-CoIMF0 exhibits some lag in prediction, the medium-frequency sequence VMD-CoIMF1 performs well, and the high-frequency sequence VMD-CoIMF2 has some fluctuations. Overall, the results are encouraging, with $R^2 = 0.927$.

To simulate the recombination strategy without SE, we used the following approaches to integrate the IMF signals: (1) with four groups: IMF0, IMF1, IMF2-3, and IMF4-6; (2) with four groups: IMF0-1, IMF2-3, IMF4-5, and IMF6; (3) with four groups: IMF0-2, IMF3-4, IMF5, and IMF6; (4) with three groups: IMF0, IMF1-3, and IMF4-6; (5) with three groups: IMF0-1, IMF2-3, and IMF4-6; (6) with three groups: IMF0-2, IMF3-4, and IMF5-6; (7) with two groups: IMF0-2 and IMF3-6; and (8) a comparison group without any combinations, i.e., the individual signals.

Table 8
Comparison between different recombination strategies.

Predictor	Strategy	RMSE	MAE	MAPE	R^2	Time (s)
VMD-None SE-GRU	1123	0.654	0.437	0.770	0.931	657.2
	2221	0.640	0.426	0.750	0.934	653.9
	3211	0.642	0.430	0.756	0.933	586.4
	133	0.681	0.462	0.814	0.925	432.2
	223	0.654	0.437	0.769	0.931	421.7
	322	0.669	0.451	0.793	0.928	429.5
	25	0.677	0.462	0.848	0.916	280.3
	34	0.663	0.457	0.814	0.921	277.8
	1111111	0.654	0.429	0.756	0.931	968.6

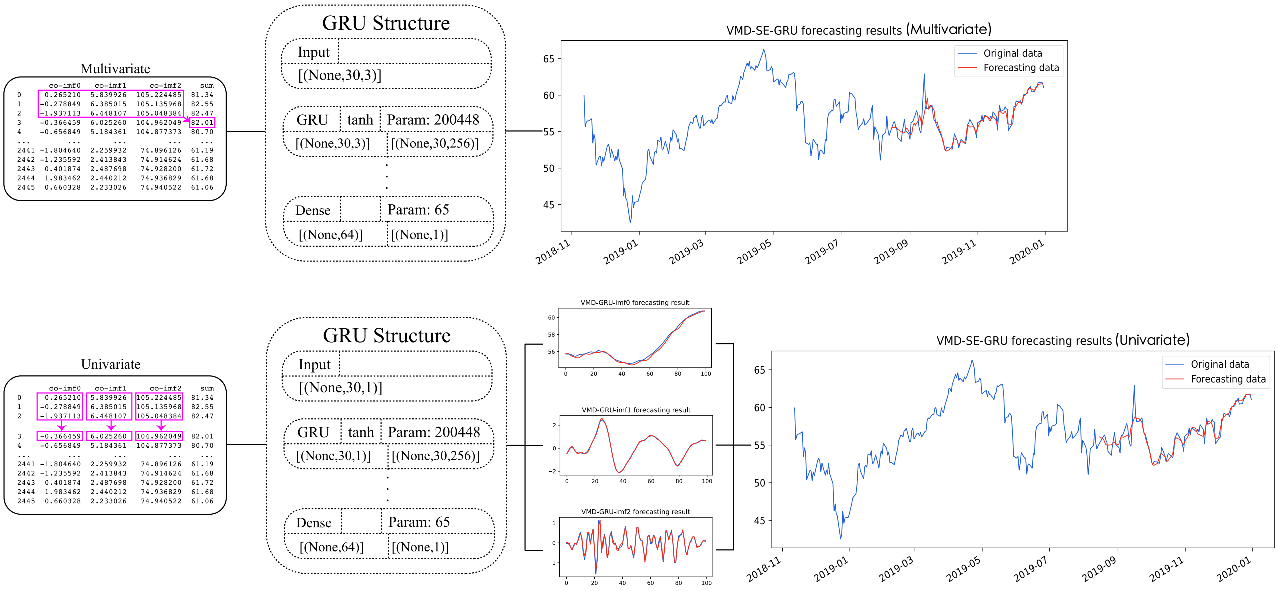
Each VMD-CoIMF in a combination takes an average of 140 s to train after 100 epochs of the GRU, and so the comparison group takes about 970 s to train seven IMF signals and combination (3) takes about 580 s. Table 8 lists the performance of the different combinations: for example, strategy 3211 represents the first VMD-CoIMF consisting of the first three VMD-IMFs (0, 1, and 2) and the second VMD-CoIMF consisting of the last two VMD-IMFs (3 and 4). The individual digits of strategy 3211 in the table must add up to 7, indicating that there are seven VMD-IMF signals.

The results in Table 8 show that the greatest impact on the predictor performance is the number of CoIMFs in each strategy and the VMD-IMF signal high- and low-frequency classification strategy. For the same number of CoIMF signals, when the low-frequency signals are reasonably reconstructed, the predictor performance is enhanced. Regarding the trend-prediction signals IMF0 and IMF1, containing low-frequency information, it is better to combine them in the signal reconstitution process while ensuring that no other signals are added, because the trend sequence is very important for the overall prediction. As for the high-frequency signals, their fluctuations are too strong and combining them arbitrarily degrades the predictor performance. Therefore, for the signals input to the GRU, we should reconstruct the low-frequency trend signals as much as possible and keep the high-frequency fluctuating signals independent.

In the case of limited equipment performance or long prediction times, it is still possible to use the SE method for the coarse grouping of signals. Additionally, because there are certain subjective factors involved in the grouping, Table 8 suggests that in the group with three VMD-CoIMF signals, the results of SE group 322 are suboptimal, although the prediction performance is not too different from that of the optimal group. Finally, Fig. 13 compares the different combination methods and the original data.

5.5. Comparison based on WTI and brent oil markets

To further demonstrate the performance of the VMD-GRU and VMD-SE-GRU frameworks over different periods with different oil price markets, four different datasets were selected to test the proposed frameworks. We considered the WTI oil prices from August 2, 2010, to December 31, 2019 (a total of 2446 trading days); WTI oil prices from January 2, 2015, to December 31, 2018 (a total of 1054 trading days); Brent oil prices from August 2, 2010, to December 31, 2019 (a total of 2427 trading days); and Brent oil prices from January 2, 2015, to December 31, 2018 (a total of 1031 trading days). Fig. 14 shows the trading data for the WTI and Brent oil markets from August 2, 2010, to December 31, 2019. We can observe that the markets are very similar in terms of their general trend, but the Brent oil prices are generally higher than the WTI oil prices. Additionally, the Brent oil prices exhibit a certain lag compared with the WTI oil prices.



Multivariate

```
      co-imf0   co-imf1    co-imf2   sum
0     0.265210  5.839926  105.224485  81.34
1    -0.278849  6.385015  105.135968  82.55
2    -1.937113  6.448107  105.048384  82.47
3    -0.366459  6.025260  104.962049  82.01
4    -0.656849  5.184361  104.877373  80.70
...        ...       ...         ...    ...
2441 -1.804640  2.259932   74.896126  61.19
2442 -1.235592  2.413843   74.914624  61.68
2443  0.401874  2.487698   74.928200  61.72
2444  1.983462  2.440212   74.936829  61.68
2445  0.660328  2.233026   74.940522  61.06
```

Univariate

```
      co-imf0   co-imf1    co-imf2   sum
0     0.265210  5.839926  105.224485  81.34
1    -0.278849  6.385015  105.135968  82.55
2    -1.937113  6.448107  105.048384  82.47

3    -0.366459  6.025260  104.962049  82.01
4    -0.656849  5.184361  104.877373  80.70
...        ...       ...         ...    ...
2441 -1.804640  2.259932   74.896126  61.19
2442 -1.235592  2.413843   74.914624  61.68
2443  0.401874  2.487698   74.928200  61.72
2444  1.983462  2.440212   74.936829  61.68
2445  0.660328  2.233026   74.940522  61.06
```

Fig. 12. Univariate and multivariate VMD-SE-GRU models.

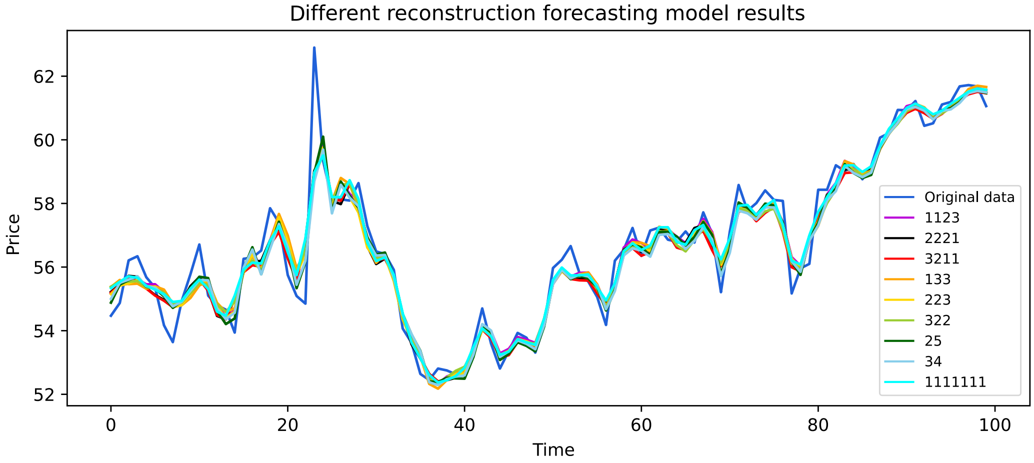


Fig. 13. Results for different reconstruction forecasting models.

Table 9 compares the model performance between different oil markets over time. The R^2 value of the VMD-GRU model is better than that of the VMD-SE-GRU model, but the difference is not significant. The VMD-GRU model has approximately twice the runtime of VMD-SE-GRU. Moreover, the VMD-SE-GRU model achieves good performance with different datasets and has a shorter runtime. For the different markets, the Brent oil prices are predicted with greater accuracy than the WTI prices, with an R^2 of 0.973 for the Brent market and 0.927 for the WTI market under the VMD-SE-GRU framework across 2400 trading days. We speculate that this may be due to the Brent oil market having a smoother trading price curve, suggesting that the VMD-SE-GRU framework may be more suitable for datasets with smooth time series. To test whether the VMD-SE-GRU framework is equally effective with small data volumes, we reduced the WTI and Brent oil price trading datasets from about 2450 data to about 1050 data. The experimental results reveal that the prediction accuracy of both VMD-GRU and VMD-SE-GRU decreases with the smaller dataset. The R^2 value of the VMD-SE-GRU framework decays more rapidly than that of VMD-GRU: in the case of the Brent market, VMD-GRU gives R^2 values of 0.990 (large dataset) and 0.919 (small dataset), a difference of 0.071, whereas VMD-SE-GRU gives R^2 values of 0.973 (large dataset) and 0.896 (small dataset), a difference of 0.077. This suggests that the VMD-GRU framework may be more suitable for forecasting from small datasets.

In summary, the VMD-SE-GRU framework is suitable for prediction using large datasets and yields more accurate results in a shorter time. The VMD-GRU framework is more suitable for computing on small-sample datasets. The VMD-GRU and VMD-SE-GRU models both exhibit excellent performance with strong generalization ability in different oil markets.

6. Conclusions

An effective and stable oil price forecasting model provides guidance for national infrastructure, industrial development, and financial investment. In this study, using Python and its various internal modules together with the Google Colab platform, two basic frameworks have been constructed: (1) a VMD-GRU oil price framework and (2) a hybrid VMD-SE-GRU forecasting framework using VMD, SE, and a GRU. Comparison experiments have been conducted to examine the performance and effectiveness of the two frameworks. The following conclusions can be drawn:

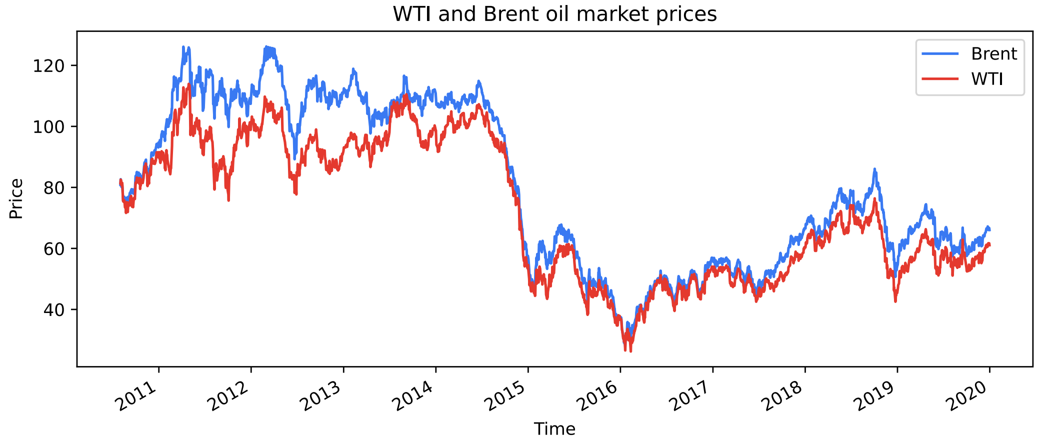


Fig. 14. WTI and Brent oil market prices.

Table 9
Comparison among different oil markets.

Predictor	Market	Trading days	RMSE	MAE	MAPE	R^2	Time (s)
VMD-GRU	WTI	2446	0.655	0.430	0.757	0.931	968
VMD-GRU	WTI	1054	0.925	0.723	1.333	0.912	447
VMD-SE-GRU	WTI	2446	0.674	0.459	0.806	0.927	429
VMD-SE-GRU	WTI	1054	1.436	1.182	1.776	0.889	207
VMD-GRU	Brent	2427	0.096	0.076	0.125	0.990	944
VMD-GRU	Brent	1031	0.899	0.737	1.108	0.919	494
VMD-SE-GRU	Brent	2427	0.226	0.184	0.300	0.973	393
VMD-SE-GRU	Brent	1031	2.795	2.245	2.921	0.896	163

1. The use of a signal decomposition method to decompose the original oil price data and subsequently produce forecasts is an effective means of improving the accuracy of a single predictor. A combination of the VMD and GRU frameworks gives the highest prediction accuracy for oil prices, without increasing the computation time.
2. Recombination of the decomposed signals can reduce the time required for forecasting while maintaining accuracy. The introduction of SE provides a satisfactory solution for signal recombination.
3. For predicting oil prices, GRU models exhibit excellent performance, outperforming both LSTM and DNN models in terms of accuracy and runtime.
4. Among the decomposition methods, VMD outperforms EMD and CEEMDAN and is more suitable for oil price prediction.
5. The VMD-SE-GRU framework is an efficient oil forecasting framework that achieves excellent forecasting performance, has a short computation time, and is robust to different oil indices.

With the introduction of adaptive learning and an early stopping mechanism, GRUs have a shorter training time and good overall accuracy. The use of adaptive learning and an early stopping mechanism in the VMD-GRU framework gives prediction results with an RMSE of 0.654, MAE of 0.429, MAPE of 0.756, and R^2 of 0.931, with a total runtime of 968 s. The VMD-SE-GRU framework also gives excellent results and has only half the runtime of the VMD-GRU framework, with an RMSE of 0.669, MAE of 0.451, MAPE of 0.793, and R^2 of 0.928, with a total runtime of 429 s.

This study has successfully forecast changes in oil prices using two hybrid forecasting frameworks. However, the proposed approach has some limitations. Although this paper has described several innovations in the hybrid framework and improved the accuracy compared with previous studies, VMD-GRU and VMD-SE-GRU are still difficult to put into practical use because they cannot accurately predict the highs generated after sudden price fluctuations. This may be because of insufficient decomposition of the low-frequency part of the IMF signal (e.g., GRU-IMF0). In addition, this study has only focused on predictions of the oil market, and the validity of our frameworks in other trading markets is yet to be verified. Finally, the VMD parameters may be related to the characteristics of the dataset itself., When the parameter settings are unreasonable, the original data may be incorrectly decomposed, resulting in confusing information in the IMF signals. Therefore, in future research, we will introduce more oil price indicators to the VMD-GRU and VMD-SE-GRU predictors, find more reasonable methods to optimize the parameters of VMD, or further decompose the low-frequency signal.

CRediT authorship contribution statement

Shiqi Zhang: Conceptualization, Methodology, Formal analysis, Data curation, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Jing Luo:** Methodology, Formal analysis, Writing – review & editing. **Shuyuan Wang:** Methodology, Formal analysis, Writing – review & editing. **Feng Liu:** Conceptualization, Methodology, Formal analysis, Writing – original draft, Writing – review & editing, Resources, Supervision, Project administration.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the Humanities and Social Sciences Foundation of the Ministry of Education of China [Grant No. 21YJC630076]. We gratefully acknowledge insightful suggestions from the Editor-in-Chief Prof. Binshan Lin, Associate Editor Prof. Erfan Babaee Tirkolaee, and the anonymous reviewers, which substantively improved this article. We would also like to thank Dr. Mingjie Fang, Dr. Caixia Liu, and the members of Star-lights Machine Learning Research Team for their comments on earlier versions of the manuscript.

References

Cao, J., Li, Z., & Li, J. (2019). Financial time series forecasting model based on CEEMDAN and LSTM. *Physica A: Statistical Mechanics and its Applications*, *519*, 127–139.

Chen, Y. -C., & Huang, W. -C. (2021). Constructing a stock-price forecast CNN model with gold and crude oil indicators. *Applied Soft Computing*, *112*, Article 107760.

Chen, S., Song, Y., Ding, Y., Zhang, M., & Nie, R. (2021). Using long short-term memory model to study risk assessment and prediction of China's oil import from the perspective of resilience theory. *Energy*, *215*, Article 119152.

Cheng, Y., Yi, J., Yang, X., Lai, K. K., & Seco, L. (2022). A CEEMD-ARIMA-SVM model with structural breaks to forecast the crude oil prices linked with extreme events. *Soft Computing*, 1–15.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., & Schwenk, H. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

De Faria, E., Albuquerque, M. P., Gonzalez, J., Cavalcante, J., & Albuquerque, M. P. (2009). Predicting the Brazilian stock market through neural networks and adaptive exponential smoothing methods. *Expert Systems with Applications*, *36*(10), 12506–12509.

Dragomiretskiy, K., & Zosso, D. (2013). Variational mode decomposition. *IEEE Transactions on Signal Processing*, *62*(3), 531–544.

Fong, W. M., & Yong, L. H. M. (2005). Chasing trends: Recursive moving average trading rules and internet stocks. *Journal of Empirical Finance*, *12*(1), 43–76.

Gao, R., Liu, J., Zhou, Q., Duru, O., & Yuen, K. F. (2022). Newbuilding ship price forecasting by parsimonious intelligent model search engine. *Expert Systems with Applications*, *201*, Article 117119.

Gumus, M., & Kiran, M. S. (2017). Crude oil price forecasting using xgboost. In *2017 international conference on computer science and engineering* (pp. 1100–1103). IEEE.

Guo, J. (2019). Oil price forecast using deep learning and ARIMA. In *2019international conference on machine learning, big data and business intelligence* (pp. 241–247). IEEE.

Guo, X., Li, D., & Zhang, A. (2012). Improved support vector machine oil price forecast model based on genetic algorithm optimization parameters. *Aasri Procedia*, *1*, 525–530.

Herrera, G. P., Constantino, M., Tabak, B. M., Pistori, H., Su, J. -J., & Naranpanawa, A. (2019). Long-term forecast of energy commodities price using machine learning. *Energy*, *179*, 214–221.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Huang, N. E., Shen, Z., & Long, S. R. (1999). A new view of nonlinear water waves: The Hilbert spectrum. *Annual Review of Fluid Mechanics*, *31*, 417–457.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., & Zheng, Q. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, *454*(1971), 903–995.

Jianwei, E., Bao, Y., & Ye, J. (2017). Crude oil price analysis and forecasting based on variational mode decomposition and independent component analysis. *Physica A: Statistical Mechanics and its Applications*, *484*, 412–427.

Khashman, A., & Nwulu, N. I. (2011). Intelligent prediction of crude oil price using support vector machines. In *2011 IEEE 9th international symposium on applied machine intelligence and informatics* (pp. 165–169). IEEE.

Kristjanpoller, W., & Minutolo, M. C. (2016). Forecasting volatility of oil price using an artificial neural network-GARCH model. *Expert Systems with Applications*, *65*, 233–241.

Li, R., Hu, Y., Heng, J., & Chen, X. (2021). A novel multiscale forecasting model for crude oil price time series. *Technological Forecasting and Social Change*, *173*, Article 121181.

Li, Y., Wang, S., Wei, Y., & Zhu, Q. (2021). A new hybrid VMD-ICSS-BiGRU approach for gold futures price forecasting and algorithmic trading. *IEEE Transactions on Computational Social Systems*, *8*(6), 1357–1368.

Luo, S., Ni, Z., Zhu, X., Xia, P., & Wu, H. (2022). A novel methanol futures price prediction method based on multicycle CNN-GRU and attention mechanism. *Arabian Journal for Science and Engineering*, 1–15.

Meng, E., Huang, S., Huang, Q., Fang, W., Wang, H., & Leng, G. (2021). A hybrid VMD-SVM model for practical streamflow prediction using an innovative input selection framework. *Water Resources Management*, *35*(4), 1321–1337.

Rather, A. M., Agarwal, A., & Sastry, V. N. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications*, *42*(6), 3234–3241.

Richman, J. S., & Moorman, J. R. (2000). Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, *278*(6), H2039–H2049.

Safari, A., & Davallou, M. (2018). Oil price forecasting using a hybrid model. *Energy*, *148*, 49–58.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E., Menon, V. K., & Soman, K. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017international conference on advances in computing, communications and informatics* (pp. 1643–1647). IEEE.

Sethia, A., & Raut, P. (2019). Application of LSTM, GRU and ICA for stock price prediction. In *Information and communication technology for intelligent systems* (pp. 479–487). Springer.

Torres, M. E., Colominas, M. A., Schlotthauer, G., & Flandrin, P. (2011). A complete ensemble empirical mode decomposition with adaptive noise. In *2011 IEEE international conference on acoustics, speech and signal processing* (pp. 4144–4147). IEEE.

Wang, J., Cao, J., Yuan, S., & Cheng, M. (2021). Short-term forecasting of natural gas prices by using a novel hybrid method based on a combination of the CEEMDAN-SE-and the PSO-ALS-optimized GRU network. *Energy*, *233*, Article 121082.

Wang, Q., Song, X., & Li, R. (2018). A novel hybridization of nonlinear grey model and linear ARIMA residual correction for forecasting US shale oil production. *Energy*, *165*, 1320–1331.

Wang, J., Sun, X., Cheng, Q., & Cui, Q. (2021). An innovative random forest-based nonlinear ensemble paradigm of improved feature extraction and deep learning for carbon price forecasting. *Science of the Total Environment*, *762*, Article 143099.

Wang, M., Zhao, L., Du, R., Wang, C., Chen, L., & Tian, L. (2018). A novel hybrid method of forecasting crude oil prices using complex network science and artificial intelligence algorithms. *Applied Energy*, *220*, 480–495.

Wei, D., Wang, J., Niu, X., & Li, Z. (2021). Wind speed forecasting system based on gated recurrent units and convolutional spiking neural networks. *Applied Energy*, *292*, Article 116842.

Wu, Y. -X., Wu, Q. -B., & Zhu, J. -Q. (2019). Improved EEMD-based crude oil price forecasting using LSTM networks. *Physica A: Statistical Mechanics and its Applications*, *516*, 114–124.

Xie, W., Yu, L., Xu, S., & Wang, S. (2006). A new method for crude oil price forecasting based on support vector machines. In *International conference on computational science* (pp. 444–451). Springer.

Xu, D., Liu, Q., Lei, J., & Jin, H. (2015). Performance of a combined cooling heating and power system with mid-and-low temperature solar thermal energy and methanol decomposition integration. *Energy Conversion and Management*, *102*, 17–25.

Yahşi, M., Çanakoğlu, E., & Ağralı, S. (2019). Carbon price forecasting models based on big data analytics. *Carbon Management*, *10*(2), 175–187.

Yu, L., Dai, W., & Tang, L. (2016). A novel decomposition ensemble model with extended extreme learning machine for crude oil price forecasting. *Engineering Applications of Artificial Intelligence*, *47*, 110–121.

Zang, H., Liu, L., Sun, L., Cheng, L., Wei, Z., & Sun, G. (2020). Short-term global horizontal irradiance forecasting based on a hybrid CNN-LSTM model with spatiotemporal correlations. *Renewable Energy*, *160*, 26–41.

Zhang, Q., Ding, J., Kong, W., Liu, Y., Wang, Q., & Jiang, T. (2021). Epilepsy prediction through optimized multidimensional sample entropy and bi-LSTM. *Biomedical Signal Processing and Control*, *64*, Article 102293.

Zhang, Y. -J., Yao, T., He, L. -Y., & Ripple, R. (2019). Volatility forecasting of crude oil market: Can the regime switching GARCH model beat the single-regime GARCH models? *International Review of Economics & Finance*, *59*, 302–317.

Zhao, Y., Li, J., & Yu, L. (2017). A deep learning ensemble approach for crude oil price forecasting. *Energy Economics*, *66*, 9–16.

Zhao, C. -l., & Wang, B. (2014). Forecasting crude oil price with an autoregressive integrated moving average (ARIMA) model. In *Fuzzy information & engineering and operations research & management* (pp. 275–286). Springer.

Zhou, F., Huang, Z., & Zhang, C. (2022). Carbon price forecasting based on CEEMDAN and LSTM. *Applied Energy*, *311*, Article 118601.

Zhu, Q., Zhang, F., Liu, S., Wu, Y., & Wang, L. (2019). A hybrid VMD–BiGRU model for rubber futures time series forecasting. *Applied Soft Computing*, *84*, Article 105739.