The Prediction of Gold Futures Prices at the Shanghai Futures Exchange Based on the MEEMD-CS-Elman Model

Xiaowen Wang¹, Ying Ma¹,², and Wen Li²

Abstract
The Gold futures market is a complex nonlinear system with the prediction of the futures prices of gold, one of the core issues faced by investors. Compared with more traditional approaches, empirical mode decomposition (EMD) and artificial neural network are the more powerful tools with which to deal with nonlinear and nonstationary price problems. By introducing mirroring extension (ME), EMD, Cuckoo Search (CS) algorithm, and Elman neural network, this article constructs the mirroring extension empirical mode decomposition (MEEMD)-CS-Elman model to forecast the price of gold futures using gold future AU0 price data from August 29, 2013, to October 18, 2018, at the Shanghai Futures Exchange (SFE) in China. Empirical results show that Elman combined with EMD is superior to single Elman in performance. Moreover, there exists an obvious endpoint effect by applying EMD to the price of AU0. By introducing the ME method, the endpoint effect can be dealt with better. Furthermore, by introducing the CS algorithm to optimize the initial weights and biases for Elman, the constructed MEEMD-CS-Elman model achieves far more accurate prediction results compared with either the EMD-Elman or the MEEMD-Elman model in terms of performance criterion: mean absolute difference (MAD), mean absolute percentage error (MAPE), root-mean-square error (RMSE), and directional symmetry (DS). In particular, the DS indicator, which reflects rising and falling prices, tends to be more attractive for investors. The value of the DS indicator in the MEEMD-CS-Elman model reaches 0.75207, meaning that the proposed model predicts the directions of increasing and falling prices quite precisely. Hence, by applying the proposed model, investors can make more scientific and accurate decisions and better reduce their investment risks.

Keywords
gold futures, mirroring extension method, empirical mode decomposition, Cuckoo Search algorithm, Elman neural network

Introduction
Gold, like stocks and bonds, has the functions of hedging and investment in its financial nature. Therefore, the gold price forecasting is significant in influencing the decision making of investors and helping support the national economic policy and the national economic development plan. Because futures price has the characteristics of nonlinearity, chaos, and a long history (Wu & Duan, 2017), many previous studies have focused on comparing the accuracy of various models in price forecasting. Generally, time-series models and their combined approaches are widely used in the analysis and prediction of prices (Batten & Lucey, 2010; Xu, 2017, 2018, 2019a, 2019b, 2020). Compared with traditional methods, however, intelligent computing models, which are better and more powerful tools to deal with such problems, have been proposed to predict the price. This article approaches the prediction of gold futures prices by using intelligent computing models to maximize the reduction of investors’ investment risk.

For price forecasting, several individual and combined models are highly applied. Autoregression moving average (ARMA) models, a mixture of autoregression (AR) and moving average (MA), are quite common. They have the characteristics of wide application and small prediction error, despite they might ignore other relevant information that is important to predict (Gülerce & Ünal, 2017; Xu, 2017, 2018, 2020). The vector autoregression (VAR) and vector error
correction models (VECMs) cover these kinds of information (Baumeister & Kilian, 2012), but the choices of modeling strategies are vague (Allen & Fildes, 2005). Autoregressive conditional heteroskedasticity (ARCH) model is great at forecasting the volatility by reflecting the variance changes intensively (Engle, 1982). However, it is not convenient with the strict restrictions on parameters. To deal with such problems, Bollerslev (1986) developed generalized autoregressive conditional heteroskedasticity (GARCH) model in 1986. The issue with that is the failure to catch the information on returns that might influence the degree of fluctuation. Therefore, GARCH-class models have been investigated (Y. Lin et al., 2020; Marchese et al., 2020). Comparatively, the combination models are superior to the individual models. They describe the change rule of time-series data better and diversify against the uncertainty (Xu, 2020). Combined models contain the previous best forecast, equal-weighted average, inverse mean-squared error, least-squares estimates of combination weights, bias-adjusted mean, shrinkage, and odds matrix approach (Capistrán & Timmermann, 2009). In practice, Xu has used the above seven models plus three variations of the least-squares estimates of combination weights method and two of the shrinkage to forecast the corn cash prices (Granger & Ramanathan, 1984; Xu, 2020).

Even though time-series models are widely used in prediction, intelligent computing models are better than them. Blaschke proved this by comparing the autoregressive integrated moving average (ARIMA) model with neural network model, and it turned out that the effectiveness of neural network was better (Blaschke et al., 2018). Adebìyi et al. (2014) tested the prediction effect of traditional ARIMA model and artificial neural network (ANN) model through the stock data of New York Stock Exchange, and they found ANN has superior performance. The reason why intelligent computing models work better is that they are handling a data series with noisy well, which is difficult for traditional models (Hamid & Iqbal, 2004). Aside from that, time-series models put forward a problem in the first step, which is difficult to identify the suitable model and the appropriate order specification. Intelligent computing models do not depend on the assumptions regarding the data, but adapt to the data (Hamid & Iqbal, 2004).

Intelligent computing models are often based on ANNs. ANNs make it easier to deal with incomplete, uncertain, or irregular data, which imply that the capability of function approximation is extremely effective. The effectiveness of different ANNs has been studied (Guresen et al., 2011); for example, single Elman neural networks (ENN) have been applied in China to forecast the price of gold futures at the Shanghai Futures Exchange (SFE) and achieve better results (Wu & Duan, 2017). To achieve better performance, some scholars have also combined ANN with other methods to deal with asset price prediction problems. For example, ANN, combined with the GARCH model, has been developed where the performance of the ANN-GARCH model is better than the GARCH model alone in predicting fluctuations in both gold spot price and futures prices (Kristjanpoller & Minutolo, 2015). ANN, integrated with text mining, also improved predictions. This was because a number of non-quantifiable factors, such as terrorist attacks, weather, and war, could be dealt with in this model (Chen et al., 2016).

Those combined models, which obtain better forecasting results, provide us with a way of improving the performance of single models in analyzing and predicting financial time series. Fortunately, this approach has been proven by Krogh and Vedelsby (1995). Their research suggested that when each single model is both accurate and diversified enough in a combined model, the combined model is better at forecasting than the single model itself. In general, the structure of a combined model for forecasting is divided into two parts: One is an analysis of time series and the extraction of characteristics; the other is the establishment of a prediction and calculation model based on these characteristics, where these characteristics are regarded as the input and the future trend the output. Looking at the problem of signal analysis in a general time series, a Hilbert–Huang transform (HHT) was proposed by Huang et al. (1971) at NASA in 1998. This is an empirical, nonparameterized, and complete analysis method for nonlinear and nonstationary data, which takes into account both time and frequency domains. The empirical mode decomposition (EMD) algorithm and the Hilbert transformation are the main processes. With EMD, any complex signal can be decomposed into a limited number of intrinsic mode functions (IMFs). These IMFs perform the Hilbert transformation well, reflecting each local oscillation structure for the original data such that the Hilbert spectrum has better time–frequency characteristics.

Since the EMD method was put forward, it was initially used to predict financial time series, such as in crude oil and foreign exchange (C.-S. Lin et al., 2012; Yu et al., 2008). Hence, compared with using single ANN to analyze and forecast, the use of “EMD + ANN” is more effective. Indeed, this kind of combined model has recently been applied to solve problems in several fields. The most widely used field is in finance. C. Zhang and Pan (2015) proposed the hybrid model to test the historical data of stock indexes in the United States and the United Kingdom, and got the better prediction results. Dhanya compared EMD-ANN model with support vector regression (SVR) and ANN models, and found that EMD-ANN model was more accurate for short-term prediction of U.S. stock index (Jothimani et al., 2016). Yu Lean proved the effectiveness of EMD-based neural network ensemble learning methodology by forecasting the spot price of world crude oil (Yu et al., 2008). In general, the superiority of this kind of paradigm in the financial communities has been confirmed by more and more scholars (Pan et al., 2017; Shi & Zhuang, 2019; Wei, 2016). Meanwhile, the model is applied to the power domain. Wang used this method to predict the wind speed of power generation efficiently (Wang et al., 2016).
Wang and Luo mentioned that the introduction of EMD-based model into the field of electricity loading has improved the accuracy of prediction (Jun et al., 2018). In addition, the validity of this hybrid model has been proved in other fields, such as meteorological science and fault diagnosis (Li et al., 2018; Shrivastava & Singh, 2018; J. Zhang et al., 2018; X. Zhang et al., 2018). However, there is little literature associated with the forecasting of gold futures prices using this improved method. Wu and Duan (2017) used to apply single Elman to predict the price of gold futures AU0, but without EMD, the accuracy of prediction results is lower.

However, almost all of the literature has not noticed that the EMD method itself has a congenital defect, that is, the endpoint effect in which decomposition results, corresponding to the endpoints of the time series, are unstable. Therefore, the majority of the published papers related to those financial forecasting models that are based on the EMD method imply that the models become unreliable due to this congenital defect. In addition, for ANN, the initial weights and biases are randomly given according to a specified distribution. However, these random initial weights and biases are not optimal in most cases so that may reduce the prediction accuracy of ANN. The work of this article is to attempt to deal with these problems.

The contributions of this article are mainly divided into the following aspects. First, since the gold futures market at the SFE was only established in China in 2008, it is still an emerging market when compared with the stock market. Therefore, our research contributes to enriching the existing research on the price of gold futures at the SFE. Second, both the EMD and the ENN methods were introduced for forecasting the trend of gold futures prices using gold futures price data from August 29, 2013, to October 18, 2018, at the SFE in China. Third, we examine that, in the gold futures market, whether there is an obvious endpoint effect to applying the EMD such that the predictive capability is quite poor in a small area of endpoints. Fourth, to deal with the endpoint effect problem, the mirroring extension (ME) method is introduced; this improves the predictive accuracy. Fifth, by introducing the Cuckoo Search (CS) algorithm to optimize the initial weights and biases of the ENN, the predictive accuracy is improved. The rest of this article is organized as follows. The “Model Formulation” section formulates the model. The “Empirical Analysis” section presents the empirical analysis. The “Discussion” section discusses the validity of the model, and the “Conclusion” section concludes the article.

**Model Formulation**

In this section, we first present the EMD and the ME methods which are appropriate to deal with the endpoint effect of EMD. Second, the ENN and the CS algorithm, with which the ENN can be optimized, are presented. Finally, based on those methods, the mirroring extension empirical mode decomposition (MEEMD)-CS-Elman model is constructed.

**The Analysis of EMD and ME Method**

**EMD method analysis.** EMD developed by Huang et al. (1998) is an effective tool to deal with nonlinear and nonstationary time series, a fundamental part of the Hilbert–Huang transformation. The key to the EMD method is to determine the interpolation method. In general, the cubic spline interpolation is preferred as the interpolation method. The main advantage of EMD is to enable the complex signals to be decomposed into a limited number of independent and orthogonal stationary IMFs, which approximate a normal distribution. The decomposed IMF contains the eigenvalues of the signals with different frequency fluctuations in the original signal. The IMF must satisfy the following conditions:

1. The number of both extreme points and zero-crossing points are the same or the maximum difference is one.
2. The upper envelope and lower envelope are locally symmetric about the time axis.

Given an original signal \( s(t) \), the screening process is as follows:

1. To identify all the locally maximum and minimum points for original signal \( s(t) \), denoted by \( \{ s_{1 \text{max}}^{1 \text{max}} (j_{1 \text{max}}) \} \) \( (1 \leq j_{1 \text{max}}^{1 \text{max}} \leq n_{1 \text{max}}^{1 \text{max}}) \) and \( \{ s_{1 \text{min}}^{1 \text{min}} (j_{1 \text{min}}) \} \) \( (1 \leq j_{1 \text{min}}^{1 \text{min}} \leq n_{1 \text{min}}^{1 \text{min}}) \), where \( n_{1 \text{max}}^{1 \text{max}} \) and \( n_{1 \text{min}}^{1 \text{min}} \) represent the number of maximum and minimum values for \( s(t) \).
2. By applying cubic spline interpolation algorithm, all the locally maximum and minimum points are fitted to the upper envelope \( U_1(t) \) and lower envelope \( L_1(t) \), respectively.
3. To calculate the mean value denoted by \( m_1(t) \) of both upper envelope \( U_1(t) \) and lower envelope \( L_1(t) \):

\[
m_1(t) = \frac{U_1(t) + L_1(t)}{2}.
\]

4. To obtain the candidate \( h_1(t) \) by subtracting the mean value \( m_1(t) \) from original signal \( s(t) \):

\[
h_1(t) = s(t) - m_1(t).
\]

5. To examine whether \( h_1(t) \) satisfies the two conditions mentioned above that IMF must satisfy. If satisfied, \( h_1(t) \) can be identified as an IMF denoted by \( c_1(t) \) by screening from the original signal \( s(t) \) or repeating Steps 1 to 4 until \( h_1(t) \) satisfies conditions. To terminate the decomposition, some conditions are necessary.

6. To obtain the residue \( r_1(t) \), by subtracting the IMF \( c_1(t) \) from original signal \( s(t) \), that is,

\[
r_1(t) = s(t) - c_1(t).
\]
7. The residue \( r_1(t) \) is regarded as the input. By repeating Steps 1 to 6, the second IMF denoted by \( c_2(t) \) is obtained. By repeating Steps 1 to 6 \( n \) times, we can obtain \( n \) IMFs. Denoting the IMF series by \( \{c_i(t)\}_{1 \leq i \leq n} \), we have

\[
\begin{align*}
    r_1(t) &= s(t) - c_1(t) \\
    r_2(t) &= r_1(t) - c_2(t) \\
    & \vdots \\
    r_n(t) &= r_{n-1}(t) - c_n(t)
\end{align*}
\]

(4)

8. When \( c_n(t) \) or \( r_n(t) \) satisfies terminating condition in which the current number of IMFs denoted by \( \text{comp} \) is equal to \( M \), where \( M \) denotes the number of IMFs needed, the loop is completed. By Equation 4, we have

\[
s(t) = r_n(t) + \sum_{i=1}^{n} c_i(t),
\]

(5)

where \( n \) denotes the number of IMFs and \( r_n(t) \) is the residual function representing the averaging trend for original signal \( s(t) \). Figure 1 shows the screening process for EMD algorithm.

**ME method.** In the EMD method, to obtain the instantaneous average envelope of the signal, the local maximum and minimum values of the original signal need to be fitted by the cubic spline interpolation algorithm; then the local mean value of the signal can be calculated. Because both ends of the signal cannot be at the maximum and minimum points at the same time, the upper and lower envelopes inevitably deviate at both ends of the data signal, and this deviation phenomenon will gradually spread to the signal interior with the progress of EMD, which seriously affects the quality of signal decomposition and the phenomenon is called endpoint effect (Zeng & He, 2004).

The essence of ME method is to symmetrically extend the original signal into a closed ring signal. This ring signal has no endpoints, and its upper and lower envelopes are determined entirely by the inside of the signal itself, thus avoiding the endpoint effect fundamentally (Flandrin et al., 2004).

The endpoint effect can be remarkably improved by applying ME method and the EMD being improved by ME method is called MEEMD in this article.

**An Improved ENN Based on the CS Algorithm**

This section first presents the ENN. Then, to identify the optimal initial weights and biases of the ENN, the CS algorithm is introduced to search for the optimal initial weights and biases.

**The analysis of ENN.** ENN, developed by Elman (1990), is a typical feed-forward neural network that is widely applied. It is also a typical dynamic recurrent neural network. It adds a
context layer to the hidden layer on the basis of the basic structure of back propagation neural network. As a one-step delay operator, the ENN achieves the purpose of memory, and thus the system has the ability to adapt to time-varying characteristics; it enhances the global stability of the network. There is more effective computational ability than feed-forward neural network and it can also be used to solve the problem of fast optimization.

In general, the structure of the ENN is divided into four layers: input layer, hidden layer, context layer, and output layer. The connection between the input layer, hidden layer, and output layer is similar to that of a feedback neural network. The input layer only acts as a signal transmission unit, and the output layer plays a weighting role. The hidden layer elements have two kinds of activation function: linear and nonlinear. In general, nonlinear sigmoid function is taken as the activation function. The context layer is used to memorize the output value of the hidden layer unit at the previous moment and can be considered as a delay operator with one-step delay. The output of the hidden layer is connected to the input of the hidden layer through the delay and storage of the connection layer, making it sensitive to historical data. The addition of the internal feedback neural network enhances the ability of the network itself to process dynamic information to achieve the purpose of dynamic modeling.

Let the weight matrix of input, context, and output be \( W^{hi} \), \( W^{hc} \), and \( W^{oh} \), respectively. Let input vector and output vector be \( x(t) = [x_1(t), x_2(t), \ldots, x_k(t)]^T \) and \( y(t) = [y_1(t), y_2(t), \ldots, y_n(t)]^T \), respectively. Assume that there is \( m \) neurons in hidden layer. Therefore, \( W^{hi} \in \mathbb{R}^{m \times k} \), \( W^{hc} \in \mathbb{R}^{m \times m} \), and \( W^{oh} \in \mathbb{R}^{n \times m} \). Let context vector be \( c(t-1) = [c_1(t-1), c_2(t-2), \ldots, c_m(t-1)]^T \), which is the output vector of the hidden layer regarded as the input vector along with \( x(t) \). Denote \( c(t-1) \) by \( x^2(t) \). Hence, \( x^2(t) = [x_1^2(t), x_2^2(t), \ldots, x_k^2(t)]^T = c(t-1) \), and the input vector can be completely written as

\[
x(t) = \left[ x_1^2(t), x_2^2(t), \ldots, x_k^2(t), x^2_1(t), x^2_2(t), \ldots, x^2_k(t) \right]^T
\]

where \( k = m + n \). The structure of ENN is shown in Figure 2.

The output vector is obtained according to the following equations:

\[
y_i(t) = f \left( a_i^0(t) \right) = \frac{1}{1 + \exp \left( -a_i^0(t) \right)}, \quad i = 1, 2, \ldots, n, \tag{7}
\]

\[
a_i^0(t) = \sum_{j=1}^{m} W^{oj}_{ji}(t) \times h_j(t), \quad i = 1, 2, \ldots, n. \tag{8}
\]

Therefore, the complete input weight matrix can be defined as

\[
W^{h} = \begin{bmatrix} W^{hi} & W^{hc} \end{bmatrix} \in \mathbb{R}^{m \times k}. \tag{9}
\]
So, the output corresponding to complete input \( x(t) \) in Equation 6 is as follows:

\[
h_j(t) = f\left(a^h_j(t)\right) = \frac{1}{1 + \exp\left(-a^h_j(t)\right)}, \quad j = 1,2,\ldots,m, \quad (10)
\]

\[
a^h_j(t) = \sum_{j=1}^{k} W^h_j(t) \times x_j(t), \quad j = 1,2,\ldots,m. \quad (11)
\]

The error function, called the loss function for the ENN, is defined as

\[
E(t) = \frac{1}{2} \left\| e(t) \right\|^2, \quad (12)
\]

\[
e(t) = d(t) - y(t),
\]

where \( d(t) \) is the desired output.

Then, the standard learning process of the ENN is shown below:

\[
W^{a,h}(t+1) = W^{a,h}(t) - \mu \frac{\partial E(t)}{\partial W^{a,h}(t)} \quad (13)
\]

\[
= W^{a,h}(t) + \mu y'(t)e(t)h'(t),
\]

\[
W^h(t+1) = W^h(t) - \mu \frac{\partial E(t)}{\partial W^h(t)} \quad (14)
\]

\[
= W^h(t) + \mu h'(t)\left[W^{a,h}(t)\right]^T y'(t)e(t)x^T(t),
\]

where

\[
y' = \text{diag}\begin{bmatrix} f'(a^0_1(t)) & f'(a^0_2(t)) & \cdots & f'(a^0_m(t)) \end{bmatrix} \in R^{m \times m}, \quad (15)
\]

\[
h' = \text{diag}\begin{bmatrix} f'(a^h_1(t)) & f'(a^h_2(t)) & \cdots & f'(a^h_m(t)) \end{bmatrix} \in R^{m \times m},
\]

and \( \mu \) is the learning rate.

Next, the training and test process for the ENN is presented. Assume that there is a financial time series denoted by \( \{p(t)\}_{t \in [N_1, N_2]} \), where \( N_1 + N_2 \) is the number of the entire data. Also suppose that the \( p(t) \) is determined by \( p(t-1), p(t-2),\ldots,p(t-T) \), where \( T \) is the previous \( T \)th price. Then, the mapping relation denoted by \( g(t) \) between \( p(t) \) and \( p(t-1), p(t-2),\ldots,p(t-T) \) can be defined as follows:

\[
p(t) = g\left(p(t-1), p(t-2),\ldots,p(t-T)\right). \quad (17)
\]

To train and test the ENN, the entire data set should be segmented into a training data set and a test data set. The training data set is used for the training model. Fundamentally, the process of the training network is used to approximate the functional relation in Equation 17. Denote training data set by \( \{p(t)\}_{1 \leq t \leq N_1} \) where \( N_1 \) denotes the number of training data. When ENN is trained well, the next step is to examine the performance using the test data set. Denote test data set by \( \{p(t)\}_{N_1+1 \leq t \leq N_2} \). Then, the training matrix and the test matrix for the ENN can be written as follows:

\[
D_{T, \text{training}} = \begin{bmatrix} p(1) & p(2) & \cdots & p(N_1-T) \\ p(2) & p(3) & \cdots & p(N_1-T+1) \\ \vdots & \vdots & \cdots & \vdots \\ p(T) & p(T+1) & \cdots & p(N_1) \end{bmatrix}, \quad (18)
\]

\[
D_{T, \text{test}} = \begin{bmatrix} p(N_1+1) & p(N_1+2) & \cdots & p(N_1+N_2-T) \\ p(N_1+2) & p(N_1+3) & \cdots & p(N_1+N_2-T+1) \\ \vdots & \vdots & \cdots & \vdots \\ p(N_1+T-1) & p(N_1+T) & \cdots & p(N_1+N_2-1) \\ p(N_1+T) & p(N_1+T+1) & \cdots & p(N_1+N_2) \end{bmatrix}. \quad (19)
\]

By using matrix \( D_{T, \text{training}} \) to train the ENN, the trained network is obtained. After that, taking the first row in matrix \( D_{T, \text{test}} \) as the input, the prediction values \( \begin{bmatrix} \hat{p}(N_1+T), \hat{p}(N_1+T+1),\ldots, \hat{p}(N_1+N_2) \end{bmatrix} \) can be obtained. Then by examining the difference between prediction values and actual values, the performance of the model can be judged.

The analysis of the CS algorithm. It is necessary to point out that when you implement the ENN, you first should initialize the weights and biases. However, these weights and biases are given randomly according to a certain distribution, meaning that these are not optimal in most cases. Therefore, the predictive capability of the ENN could be affected. Therefore, to achieve better accuracy, the initial weights and biases must be optimized. Taking into account of the optimization capability of the CS algorithm, this article applies it to optimize the initial weights and biases.

The CS algorithm is a new heuristic algorithm based on swarm intelligence technology and is an efficient algorithm used to solve the optimization problem by simulating the parasitic brood parasitism of some cuckoo species (Yang &
Deb, 2009, 2010, 2014). In addition, the CS algorithm can be enhanced by the so-called Lévy flight, rather than a simple isotropic random walk. Existing research shows that the algorithm is probably more effective than the genetic algorithm, the particle swarm optimization (PSO) algorithm, and other algorithms.

To describe the CS algorithm simply, we introduce the following three idealized rules.

1. Each cuckoo randomly selects one nest and lay one egg each time.
2. The nest with the highest quality egg is kept for the next generation.
3. The number of host nests \( n \) is fixed, and the probability that a host bird discovers the egg laid by a cuckoo is \( P_a \in [0, 1] \). In this case, the host bird either throws out the cuckoo egg or abandons the existing nest.

In the CS algorithm, each egg in the nest represents a solution and the cuckoo’s egg represents a new solution. The goal is to replace the bad solution in the nest with a new solution or a potential better solution.

1. To set the number of nests at \( n \) and the dimension of search space \( d \), initializing the location of nests \( p_0 = [x_0^{(0)}, x_1^{(0)}, \ldots, x_n^{(0)}]^T \), and identifying the optimal nest location \( x_0^{(0)} \) and optimal solution \( f_{\text{min}} \).
2. To keep the last optimal nest location \( x_{n-1}^{(t)} \) and update the other birds’ nest positions, according to the following formula:
   \[
   x_{i}^{(t+1)} = x_{i}^{(t)} + \alpha \odot L(\lambda),
   \]
   where \( \alpha \) denotes the step size, \( \odot \) is the dot product, and \( L(\lambda) \) denotes the Lévy randomly search route and its walk step randomly follows the Lévy distribution, that is,
   \[
   L(\lambda) \sim \mu = t^{-\lambda} \quad (1 \leq \lambda \leq 3).
   \]

After the location is updated, a number \( r \in (0, 1) \) is generated randomly. Then, if \( r > P_a \), the nest position \( x_{i}^{(t+1)} \) is changed, otherwise it is not. Local stochastic processes can be described as follows:

\[
\begin{align*}
\lambda_{i}^{(t+1)} &= \lambda_{i}^{(t)} + \alpha \odot H(p_a - \varepsilon) \odot (x_j - x_k),
\end{align*}
\]

where \( \lambda_{i}^{(t+1)} \) and \( \lambda_{i}^{(t)} \) are two different random sequences, \( H(\cdot) \) denotes the Heaviside function, \( \varepsilon \) is a random number taken from the random distribution, and \( s \) denotes the step size.

Finally, a better set of bird nest positions \( \lambda_{i}^{(t+1)} \) will be retained for the next generation and still denoted by \( \lambda_{i}^{(t)} \).

In this way, a group of new nest locations are obtained. The nest locations of this group are then tested, replacing those of the previous generation \( p_{t+1} = [\lambda_{i}^{(t-1)}, \lambda_{i}^{(t-2)}, \ldots, \lambda_{i}^{(t-1)}]^T \), characterized by a group of nests with poor fitness, so that a group of better nest locations \( g_t = [\lambda_{i}^{(t)}, \lambda_{i}^{(t)}, \ldots, \lambda_{i}^{(t)}]^T \) can be substituted. In so doing, a new set of better nest positions \( p_t = [\lambda_{i}^{(t)}, \lambda_{i}^{(t)}, \ldots, \lambda_{i}^{(t)}]^T \) is obtained.

3. To find out the optimal bird nest position \( \lambda_{i}^{(t)} \) and the best value \( f_{\text{min}} \) in the last \( p_t \) obtained in Step 2. If the iteration stop condition is achieved, the global optimal location \( \lambda_{i}^{(t)} \) are output. If not, go back to Step 2 to continue iterating. Note that the iteration termination condition is to achieve the required number of iterations or the precision required.

By using the CS algorithm to optimize the initial weights and bias of the ENN, this model is called CS-Elman model.

**The Construction of MEEMD-CS-Elman Model**

In this section, by using the MEEMD and CS-Elman methods, an MEEMD-CS-Elman prediction model is constructed that is able to deal with the prediction issue of gold futures prices. Based on how the market performs, the model needs to constantly learn to evolve. In general, we take long enough historical data and determine the time frame of the data. In this article, we focus on the closing price of each trading day. Any time point \( t \) corresponds to a trading day in the daily data. Let \( X(t) \) be the closing price at time \( t \). Assume that there is enough data denoted by \( DX(N_1 + N_2) \), where \( N_1 + N_2 \) denotes the length of entire data set. Then, \( DX(t, N_1 + N_2) \) can be given by

\[
DX(N_1 + N_2) = \begin{cases} X(1), X(2), \ldots, X(N_1), X(N_1 + 1), \\ X(N_2 + 1), \ldots, X(N_1 + N_2) \end{cases}.
\]

Let the training data set and the test data set be \( DX_{\text{tra}}(N_1) \) and \( DX_{\text{tes}}(N_2) \), respectively. The data before time \( N_1 \) are used as the training data set and after as test data set. Therefore, we have

\[
DX_{\text{tra}}(N_1) = \{X(1), X(2), \ldots, X(N_1)\},
\]

\[
DX_{\text{tes}}(N_2) = \{X(N_1 + 1), X(N_2 + 2), \ldots, X(N_1 + N_2)\}.
\]

By applying the MEEMD approach for each element in \( DX_{\text{tra}}(N_1) \) and \( DX_{\text{tes}}(N_2) \), respectively, we have

\[
D_{\text{IMF}}^{DX} = \begin{bmatrix} \text{IMF}_1(1) \quad \text{IMF}_1(2) \quad \ldots \quad \text{IMF}_1(N_1) \\
\text{IMF}_2(1) \quad \text{IMF}_2(2) \quad \ldots \quad \text{IMF}_2(N_1) \\
\vdots \quad \vdots \quad \ldots \quad \vdots \\
\text{IMF}_k(1) \quad \text{IMF}_k(2) \quad \ldots \quad \text{IMF}_k(N_1) \\
\text{Res}_1(1) \quad \text{Res}_1(2) \quad \ldots \quad \text{Res}_1(N_1) \end{bmatrix}.
\]
where \( k \) denotes the number of IMFs and \( \text{Res}(\cdot) \) is the residue.

Next, we first apply CS-Elman to model the \( \text{IMF}_1 \) and then denote the training and test data sets of \( \text{IMF}_1 \) by \( \text{IMF}_{1,\text{tra}} \) and \( \text{IMF}_{1,\text{tes}} \) separately. Therefore, we have

\[
\text{IMF}_{1,\text{tra}} = \{ \text{IMF}_1(1), \text{IMF}_1(2), \ldots, \text{IMF}_1(N_1) \},
\]

(28)

\[
\text{IMF}_{1,\text{tes}} = \{ \text{IMF}_1(N_1 + 1), \text{IMF}_1(N_1 + 2), \ldots, \text{IMF}_1(N_1 + N_2) \}.
\]

(29)

By using Equations 18 and 19, the prediction values \( \hat{\text{IMF}}_1(N_1 + T), \hat{\text{IMF}}_1(N_1 + T + 1), \ldots, \hat{\text{IMF}}_1(N_1 + N_2) \) can be obtained. Similarly, the prediction values of \( \text{IMF}_{2,\text{tra}}, \ldots, \text{IMF}_{k,\text{tra}} \), \( \hat{\text{Res}}(N_1 + 1) \), \( \hat{\text{Res}}(N_1 + 2) \), \ldots, \( \hat{\text{Res}}(N_1 + N_2) \) can be obtained. Then we have

\[
D_{\text{pre}}^\text{IMF} = \left\{ \begin{array}{c}
\hat{\text{IMF}}_1(N_1 + T) & \hat{\text{IMF}}_1(N_1 + T + 1) & \ldots & \hat{\text{IMF}}_1(N_1 + N_2) \\
\hat{\text{IMF}}_2(N_1 + T) & \hat{\text{IMF}}_2(N_1 + T + 1) & \ldots & \hat{\text{IMF}}_2(N_1 + N_2) \\
\vdots & \vdots & \ddots & \vdots \\
\hat{\text{IMF}}_k(N_1 + T) & \hat{\text{IMF}}_k(N_1 + T + 1) & \ldots & \hat{\text{IMF}}_k(N_1 + N_2) \\
\hat{\text{Res}}(N_1 + 1) & \hat{\text{Res}}(N_1 + T + 1) & \ldots & \hat{\text{Res}}(N_1 + N_2)
\end{array} \right\},
\]

(30)

Accumulating matrices in the direction of columns, the prediction values \( \hat{X} \) can be obtained as follows:

\[
D_{\text{pre}} = \{ \hat{X}(N_1 + T), \hat{X}(N_1 + T + 1), \ldots, \hat{X}(N_1 + N_2) \}.
\]

(31)

Then, by comparing with \( X(N_1 + T), X(N_1 + T + 1), \ldots, X(N_1 + N_2) \), the accuracy can be presented.

The steps for the proposed model are shown in Figure 3.

**Empirical Analysis**

**Data Descriptive Statistical Analysis and Parameter Setting**

The data range is from August 29, 2013, to October 18, 2018, at the SFE in China. Training data are from August 29, 2013, to October 10, 2017, amounting to 1,000 pieces of data. Test data are from October 11, 2017, to October 18, 2018, amounting to 250 pieces of data. The closing price of gold futures AU0 in each trading day is used in the empirical analysis. The statistical analysis of the data is shown in Table 1. Basic parameters are set as shown in Table 2. The trend of closing prices is shown in Figure 4.

To verify the validity of the proposed model, several performance functions are introduced. These are the mean absolute difference (MAD), the mean absolute percentage error (MAPE), the root-mean-square error (RMSE), and the directional symmetry (DS), respectively. The calculation formulas of these performance functions are listed in Table 3. Among them, MAD, MAPE, and RMSE are used to evaluate prediction errors. The smaller the three performance indicators are, the closer the prediction value is to the actual value, meaning that the error between actual value and predictive value is smaller. The performance indicator DS gives the correct rate for forecasting price directions. It is necessary to point out that investors are more concerned about whether the price of gold future is rising or falling, and less about the specific forecast value. Therefore, DS seems to be more significant.

**The Comparison of Prediction Capability Among Elman, SVR, and EMD-Elman**

This section presents a comparison of prediction capabilities of the Elman, the SVR, and the EMD-Elman methods. Because investors are more concerned about price rises and falls, the DS indicator is more favored by investors. This section is therefore focused on a comparison of DS indicators using the Elman, the SVR, and the EMD-Elman models. Table 4 shows the results of predictive DS for Elman, SVR, and EMD-Elman for a different prediction step \( r \). It is obvious that the DS indicator increases significantly through the application of the EMD method. For Steps 1 to 9, the prediction capability of the EMD-Elman model is more
Figure 3. Flowchart of MEEMD-CS-Elman model.

Note. MEEMD = mirroring extension empirical mode decomposition; CS = Cuckoo Search.

Table 1. Descriptive Statistics of Data Set.

| Sample data       | Observations | $M$   | $SD$  | Minimum | Maximum | Skewness | Kurtosis |
|-------------------|--------------|-------|-------|---------|---------|----------|----------|
| All data          | 1,250        | 261.25| 17.864| 218.15  | 296.7   | −0.45571 | 2.161    |
| Training data     | 1,000        | 258.23| 18.69 | 218.15  | 296.7   | −0.097918| 1.96     |
| Test data         | 250          | 273.34| 3.982 | 263.4   | 282.05  | 0.098036 | 2.6741   |

Table 2. Parameter Setting.

| Model | Parameters’ description | Benchmark values |
|-------|--------------------------|-------------------|
| ME    | The number of boundary extension points | 2 |
|       | The number of nests | 10 |
|       | Probability of being discovered by the host bird | 0.25 |
|       | Lower bound of optimization parameters | 0 |
|       | Upper bound of optimization parameters | 1 |
|       | The error tolerance of iteration | $1 \times 10^{-5}$ |
| CS    | The number of neurons in hidden layer | 18 |
|       | The maximum epochs | 2,000 |

Note. ME = mirroring extension; CS = Cuckoo Search; ENN = Elman neural network.
accurate than that of SVR and Elman models. In particular, when \( \tau = 7 \), the value of the DS indicator for EMD-Elman is 0.71901. This is the maximum value for Steps 1 to 9, meaning that the predictive capability of EMD-Elman model is best at \( \tau = 7 \). Therefore, in the analysis of the following section to improve the accuracy of the model, we mainly consider the case of \( \tau = 7 \). In addition, it is necessary to point out that the endpoint effect of the EMD-Elman model in this section is not taken into account, so, in the next section, we focus on the endpoint effect of EMD-Elman.

**The Analysis of Endpoint Effect**

This section presents a comparison of the predictive capability and relative error between both the MEEMD-Elman model dealing with the endpoint effect and the EMD-Elman model not dealing with the endpoint effect. A comparison of performance indicators between EMD-Elman and MEEMD-Elman models is shown in Table 5. From the perspective of MAD, MAPE, and RMSE, the values in MEEMD-Elman are less than that in EMD-Elman. From the perspective of DS which is of more concern to investors, the DS indicator of MEEMD-Elman is 0.73554; this is greater than for EMD-Elman. From the analysis above, it is obvious that the accuracy of the MEEMD-Elman model is higher than that of the EMD-Elman model, meaning that the accuracy of the EMD-Elman model can be improved by dealing with the endpoint effect and applying the ME method. In Figure 5, intuitively, the prediction capability of EMD-Elman is rather poor being in a small area at the endpoints. However, the prediction capability of MEEMD-Elman is better than that of EMD-Elman found in a small area at the endpoint as shown in Figure 6. The relative error in Figures 7 and 8 also shows this improvement. From the above analysis, we find that when using the EMD method to analyze the price of gold futures AU0, there is obviously an endpoint effect as shown in Figures 5 and 7. This also provides direct evidence for the existence of an endpoint effect in the analysis of gold futures prices when using EMD. In addition, by using the ME method, the endpoint effect can be improved.

![Figure 4. Price trend of gold future AU0 at the SFE.](image)

*Note. SFE = Shanghai Futures Exchange.*

| Table 3. Performance Criterion Function. |
|----------------------------------------|
| Performance criterion | Calculation formulas |
|------------------------|----------------------|
| **MAD**                | \( \text{MAD} = \frac{1}{n} \sum_{i=1}^{n} |T_i - A_i| \) |
| **MAPE**               | \( \text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{T_i - A_i}{T_i} \right| \) |
| **RMSE**               | \( \text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (T_i - A_i)^2} \) |
| **DS**                 | \( \text{DS} = \frac{1}{n} \sum_{i=1}^{n} d_i \), \( d_i = \begin{cases} 1, & (T_i - T_{i-1}) \times (A_i - A_{i-1}) \geq 0; \\ 0, & \text{otherwise.} \end{cases} \) |

*Note. Here, \( A \) and \( T \) denote the actual and predictive values, respectively. \( n \) is the total number of sample points. MAD = mean absolute difference; MAPE = mean absolute percentage error; RMSE = root-mean-square error; DS = directional symmetry.*

| Table 4. Prediction Results: Comparison of DS Indicator Among Elman, SVR, and EMD-Elman. |
|----------------------------------------|
| Model       | \( \tau \)  |
|-------------|----------------|
| Elman       | 1 2 3 4 5 6 7 8 9 |
| SVR         | 0.50403 0.49393 0.48374 0.49796 0.50410 0.48148 0.47934 0.48548 0.48333 |
| EMD-Elman   | 0.69355 0.70850 0.69106 0.66939 0.70082 0.67901 0.71901 0.71369 0.71250 |

*Note. DS = directional symmetry; EMD = empirical mode decomposition; SVR = support vector regression.*
The Comparison of Prediction Capability Between MEEMD-Elman and MEEMD-CS-Elman

This section presents the comparison of prediction capability between MEEMD-Elman and MEEMD-CS-Elman. From the results of the last section, we know that the MEEMD-Elman model is an improved model in forecasting of the price of gold futures AU0. However, for the ENN, initialized weights and biases which are randomly given are not optimal in most cases. Therefore, a CS algorithm is introduced to search the optimal initial weights and biases to achieve better accuracy. The comparison of performance indicators between MEEMD-Elman and MEEMD-CS-Elman model is shown in Table 6. From the perspective of MAD, MAPE, and RMSE, the values using MEEMD-CS-Elman are less than those for MEEMD-Elman. From the perspective of DS, which is more of a concern for investors, the DS indicator of MEEMD-CS-Elman is 0.75207, which is greater than 0.73554 for MEEMD-Elman. From the analysis above, it is obvious that the accuracy of the MEEMD-CS-Elman model is greater than that for the MEEMD-Elman model, meaning that the accuracy of the MEEMD-Elman model can be improved by incorporating the CS algorithm. Figures 9 and 10 intuitively give the price trend and relative error.

Compared with the research of Wu and Duan (2017) which just applied single Elman to predict the price of gold futures AU0, our model decomposes the time series by EMD. The decomposed function contains the local characteristics of the original signal at different time scales, which dramatically reduces the difficulty of modeling and effectively improves the prediction accuracy. Furthermore, our model
optimizes the initial weights and biases of Elman by CS algorithm, and the prediction ability of ENN is improved. In general, MEEMD-CS-Elman model achieves significantly better prediction accuracy. That implies that for investors who are willing to invest in gold futures AU0, our model can reduce better investment risk.

### Diebold-Mariano (DM) Type Test

This section presents a comparison of DM test in different forecasting models. Table 7 shows the result of DM test about Elman, SVR, MEEMD-Elman, and MEEMD-CS-Elman models, which they all reject the original hypothesis at 1% confidence level. It means the price forecasting capabilities of these models are well. Furthermore, according to the above analysis, the validity of the proposed models can be verified by performance function. The smaller the performance indexes of MAD, MAPE, and RMSE are, the closer the predicted value is to the actual value. Therefore, the accuracy of MEEMD-CS-Elman model is better.

### Discussion

With the rapid development of big data and artificial intelligence, the application of machine learning method has become more and more popular. Meanwhile, getting across many empirical evidences, scholars have found that the accuracy of the combined models is better than the single model (Chen et al., 2016; Kristjanpoller & Minutolo, 2015; Krogh & Vedelsby, 1995). Moreover, compared with the traditional statistical analysis method, machine learning method significantly improves the efficiency of prediction (Adebiyi et al., 2014; Blaschke et al., 2018; Hamid & Iqbal, 2004), because it overcomes the defect that the research hypothesis in statistical method is too strict. Based on the advantages of more precise and more efficient, intelligent computing model has been successively used in finance, power domain, and other fields for forecasting, and has gradually become the main trend (Jothimani et al., 2016; Jun et al., 2018; Li et al., 2018; C.-S. Lin et al., 2012; Pan et al., 2017; Shi & Zhuang, 2019; Shrivastava & Singh, 2018; Wang et al., 2016; Wei, 2016; Yu et al., 2008; J. Zhang et al., 2018; X. Zhang et al., 2018; C. Zhang & Pan, 2015). This article constructs an improved hybrid model to predict the futures price of gold, and the result is better than Elman, SVR, and MEEMD-Elman. Our conclusions do match the achievements of existed research, which means the effectiveness of this model has been verified (Adebiyi et al., 2014; Blaschke et al., 2018; Chen et al., 2016; Hamid & Iqbal, 2004; Jothimani et al., 2016; Jun et al., 2018; Kristjanpoller & Minutolo, 2015; Krogh & Vedelsby, 1995; Li et al., 2018; C.-S. Lin et al., 2012; Pan et al., 2017; Shi & Zhuang, 2019; Shrivastava & Singh, 2018; Wang et al., 2016; Wei, 2016; Yu et al., 2008; J. Zhang et al., 2018; X. Zhang et al., 2018; C. Zhang & Pan, 2015).

### Conclusion

In this article, we build on the MEEMD-CS-Elman model to forecast the price trends of gold futures AU0 at the SFE in China. Up to now, a large body of literature has shown that the futures prices tend to be nonlinear and nonstationary.
Therefore, it is quite effective to deal with futures prices by applying EMD and the ENN. The demonstration of empirical results can be divided into three aspects.

1. In terms of DS indicator, we find that the prediction capability of EMD-Elman is significantly superior to that of single Elman and SVR. Even though SVR is one of the most popular methods, the proposed MEEMD-CS-Elman model is now proved to be better in this article.

2. By introducing the ME method into EMD, the endpoint effect of EMD for the price of AU0 is dealt with better.

3. By introducing the CS algorithm to search for the optimal initial weights and biases, the MEEMD-CS-Elman achieves the most accurate predictions when compared with the EMD-Elman and MEEMD-Elman models in terms of performance criterion: MAD, MAPE, RMSE, and DS. In particular, the DS indicator tends to be more attractive for investors. The value of the DS indicator in MEEMD-CS-Elman model reaches 0.75207, meaning that the proposed model predicts the directions of increasing and falling prices quite precisely. Hence, neural network combined with EMD in physical method and CS algorithm in statistics is a constructive perspective, which endows with stronger predictability in gold futures AU0.

4. By DM test, it is clearly confirmed that the prediction accuracy of MEEMD-CS-Elman model is higher than others.

Overall, by applying the proposed model, investors can make more scientific and accurate decisions and reduce their investment risks better. Deep learning has shown that increasing the number of hidden layers can achieve a better prediction result. However, there is only one hidden layer in the ENN. Hence, how to add more hidden layers into the ENN and optimize the ENN using CS algorithm to obtain the more accurate predictions will be the focus of further research in the future.

Acknowledgment

The authors are very grateful to the editor and anonymous referees for their insightful and valuable suggestions that have led to an improved version of this paper.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was funded by the Department of Philosophy and Social Science of Gansu Province (No. 19YB006).

ORCID iD

Ying Ma https://orcid.org/0000-0001-9171-2908

References

Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of ARIMA and artificial neural networks models for stock price prediction. Journal of Applied Mathematics, 2014, 614342.

Allen, P. G., & Fildes, R. (2005). Levels, differences and ECMs—Principles for improved econometric forecasting. Oxford Bulletin of Economics and Statistics, 67, 881–904.

Batten, A. J., & Lucey, B. M. (2010). Volatility in the gold futures market. Applied Economics Letters, 17(2), 187–190.

Baumeister, C., & Kilian, L. (2012). Real-time forecasts of the real price of oil. Journal of Business & Economic Statistics, 30, 326–336.

Blaschke, B., Neubauer, A., & Scherzer, O. (2018). On convergence rates for the iteratively regularized Gauss–Newton method. IMA Journal of Numerical Analysis, 17, 421–436.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31, 307–327.

Capistrán, C., & Timmermann, A. (2009). Forecast combination with entry and exit of experts. Journal of Business & Economic Statistics, 27, 428–440.

Chen, H.-H., Chen, M., & Chiu, C.-C. (2016). The integration of artificial neural networks and text mining to forecast gold futures prices. Communications in Statistics-Simulation and Computation, 45(4), 1213–1225.

Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14(2), 179–211.

Engle, R. F. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of UK inflation. Econometrica, 50, 987–1008.

Flandrin, P., Rilling, G., & Goncalves, P. (2004). Empirical mode decomposition as a filter bank. IEEE Signal Processing Letters, 11(2), 112–114.
Granger, C. W. J., & Ramanathan, R. (1984). Improved methods of combining forecasts. *Journal of Forecasting*, 3, 197–204.

Gülerce, M., & Unal, G. (2017). Forecasting of oil and agricultural commodity prices: Varma versus ARMA. *Annals of Financial Economics*, 12, 1750012.

Guresen, E., Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert Systems With Applications*, 38(8), 10389–10397.

Hamid, S. A., & Iqbal, Z. (2004). Using neural networks for forecasting volatility of S&P 500 Index futures prices. *Journal of Business Research*, 57, 1116–1125.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., . . . Liu, H. H. (1997). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 454(1971), 903–995.

Jothimani, D., Shankar, R., & Yadav, S. S. (2016). A hybrid EMD-ANN model for stock price prediction. In B. Panigrahi, P. Suganthan, S. Das, & S. Satapathy (Eds.), *International conference on swarm, evolutionary, and memetic computing* (pp. 60–70). Springer.

Jun, W., Yuyan, L., Lingyu, T., & Peng, G. (2018). A new weighted CEEMDAN-based prediction model: An experimental investigation of decomposition and non-decomposition approaches. *Knowledge-Based Systems*, 160, 188–199.

Kristjanpoller, W., & Minutolo, M. C. (2015). Gold price volatility: A forecasting approach using the artificial neural network-GARCH model. *Expert Systems With Applications*, 42(20), 7245–7251.

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. In *Advances in neural information processing systems*. https://papers.nips.cc/paper/1994/file/b8c37e33defde51cf91e1e03e51657da-Paper.pdf

Li, H., Zhao, J., Ni, X., & Zhang, X. (2018). Fault diagnosis for machinery based on feature extraction and general regression neural network. *International Journal of System Assurance Engineering and Management*, 9(5), 1034–1046.

Lin, C.-S., Chiu, S.-H., & Lin, T.-Y. (2012). Empirical mode decomposition-based least squares support vector regression for foreign exchange rate forecasting. *Economic Modelling*, 29(6), 2583–2590.

Lin, Y., Xiao, Y., & Li, F. (2020). Forecasting crude oil price volatility via a HM-EGARCH model. *Energy Economics*, 87, 104693.

Marchese, M., Kyriakou, I., Tamvakis, M., & Di Iorio, F. (2020). Forecasting crude oil and refined products volatilities and correlations: New evidence from fractionally integrated multivariate GARCH models. *Energy Economics*, 88, 104757.

Pan, H., Ma, Y., & Zhang, C. (2017, December 11–14). FEPA—An integrated computational intelligence model for predicting financial time series [Conference session]. IEEE/SICE International Symposium on System Integration, Taipei.

Shi, C., & Zhuang, X. (2019). A study concerning soft computing approaches for stock price forecasting. *Axioms*, 8(4), 116.

Shrivastava, Y., & Singh, B. (2018). Possible way to diminish the effect of chatter in CNC turning based on EMD and ANN approaches. *Arabian Journal for Science and Engineering*, 43(9), 4571–4591.

Wang, S., Zhang, N., Wu, L., & Wang, Y. (2016). Wind speed forecasting based on the hybrid ensemble empirical mode decomposition and GA-BP neural network method. *Renewable Energy*, 94, 629–636.

Wei, L. (2016). A hybrid ANFIS model based on empirical mode decomposition for stock time series forecasting. *Applied Soft Computing*, 42, 368–376.

Wu, B., & Duan, T. (2017). The fractal feature and price trend in the gold future market at the Shanghai Futures Exchange (SFE). *Physica A: Statistical Mechanics and Its Applications*, 474, 99–106.

Xu, X. (2017). Short-run price forecast performance of individual and composite models for 496 corn cash markets. *Journal of Applied Statistics*, 44, 2953–2960.

Xu, X. (2018). Using local information to improve short-run corn price forecasts. *Journal of Agricultural & Food Industrial Organization*, 16, Article 20170018.

Xu, X. (2019a). Contemporaneous and granger causality among US corn cash and futures prices. *European Review of Agricultural Economics*, 46, 663–695.

Xu, X. (2019b). Price dynamics in corn cash and futures markets: Cointegration, causality, and forecasting through a rolling window approach. *Financial Markets and Portfolio Management*, 33, 155–181.

Xu, X. (2020). Corn cash price forecasting. *American Journal of Agricultural Economics*, 102(4), 1297–1320.

Yang, X., & Deb, S. (2009, December 9–11). Cuckoo Search via Lévy flights [Conference session]. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India.

Yang, X., & Deb, S. (2010). Engineering optimisation by Cuckoo Search. *arXiv Preprint arXiv*, 1005.2908. https://arxiv.org/pdf/1005.2908.pdf

Yang, X., & Deb, S. (2014). Cuckoo Search: Recent advances and applications. *Neural Computing and Applications*, 24(1), 169–174.

Yu, L., Wang, S., & Lai, K. K. (2008). Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. *Energy Economics*, 30(5), 2623–2635.

Zeng, K., & He, M. (2004, September 20–24). A simple boundary process technique for empirical mode decomposition [Conference session]. 2004 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2004), Anchorage, AK, United States.

Zhang, C., & Pan, H. (2015, December 18–20). A novel hybrid model based on EMD-BPNN for forecasting US and UK stock indices [Conference session]. IEEE International Conference on Progress in Informatics & Computing, Nanjing, China.

Zhang, J., Li, D., Hao, Y., & Tan, Z. (2018). A hybrid model using signal processing technology, econometric models and neural network for carbon spot price forecasting. *Journal of Cleaner Production*, 204, 958–964.

Zhang, X., Zhang, Q., Zhang, G., Nie, Z., Gui, Z., & Que, H. (2018). A novel hybrid data-driven model for daily land surface temperature forecasting using long short-term memory neural network based on ensemble empirical mode decomposition. *International Journal of Environmental Research and Public Health*, 15(5), 1032.