Statistics and Its Interface Volume 10 (2017) 119–130

A hybrid transfer learning model for crude oil price forecasting

Jin Xiao, Yi Hu, Yi Xiao*, Lixiang Xu, and Shouyang Wang

Most of the existing models for oil price forecasting only use the data in the forecasted time series. This study proposes a hybrid transfer learning model (HTLM) for crude oil price forecasting. We first selectively transfer some related time series in the source domain to assist in modeling the target time series by using a transfer learning technique, and then construct the forecasting model using the analog complexing (AC) method. Further, we introduce a genetic algorithm to find the optimal match between two important parameters in HTLM. Finally, we use two main crude oil price time series—the West Texas Intermediate (WTI) and the Brent crude oil spot prices—for empirical analysis. Our results show the effectiveness and superiority of the proposed model compared with existing models.

Keywords and phrases: Hybrid transfer learning model, Analog complexing, Genetic algorithm, Crude oil price forecasting, Transfer learning technique.

1. INTRODUCTION

Oil is the most widely used energy source in the world, accounting for 36.4% of primary energy consumption [1] and 94.5% of global energy used for transportation [2]. Similar to most commodities, oil price is basically determined by its supply and demand [3, 4], and also strongly influenced by many irregular events such as weather, stock levels, GDP growth, political aspects, and even people's psychological expectations. These factors lead to strong fluctuations in the oil market. What's more, the sharp movements of oil price may disturb aggregate economic activity, which brings dramatic uncertainty for the global economy. In particular, since January 2004, global oil price has been rising rapidly. It is said that a 10% increase in oil price is equivalent to 0.6–2.5% GDP growth decrease for the US [5, 6]. Therefore, oil price forecasting has become an important research area in the last decades.

Currently, there are abundant studies on the analysis and forecasting of crude oil price. The approaches can be classified into two categories: 1) econometric forecasting models, such as autoregressive integrated moving average (ARIMA), autoregressive conditional heteroscedasticity (ARCH), and vector autoregression (VAR) models, which assume that the time series is just a random sequence with linear correlation [7, 8, 9, 10, 11]; and 2) non-linear dynamics forecasting models. In general, the econometric forecasting models can provide near-accurate prediction results if the oil price series is linear or near linear. However, the influencing factors of crude oil price are very complicated. Moreover, it has been demonstrated that oil price fluctuations appear highly non-linearly dependent and even chaotic [12, 13]. Therefore, the performance may be extremely poor only by using these econometric models [14]. In recent years, non-linear dynamics forecasting models have been introduced to forecast crude oil price, including artificial neural network (ANN), support vector regression (SVR), and belief network (BN) [15, 16, 17, 18, 19, 20, 21]. However, these non-linear models also have disadvantages, e.g., ANN is easy to sink into local minima and suffers from over-fitting, while other models, such as SVR, are sensitive to parameter selection [19].

The limitations of the econometric and non-linear dynamics models motivate the demand to look for a new method for crude oil price forecasting. In fact, as early as 1994, Peters [22] pointed out that most financial time series were long-memory processes, and the future developments of the series depended on not some parts, but all of the history data. In this case, prediction approaches based on historical pattern matching should be chosen, which can overcome the disadvantages of short-memory process forecasting methods. In addition, the Fractal Market Hypothesis offers sturdy support for the feasibility of historical pattern matching. Both Peters [22] and Wen et al. [23] presented that the financial markets appeared to have a fractal structure. Peters [22] further found that the markets did not change completely at random; on the contrary, there were some non-periodical cycles difficult to be captured by neural networks and traditional econometric forecasting methods. He claimed that financial markets were predictable and long-memory based methods were necessary to obtain accurate forecasting results. Further, Farmer and Sidorowich [24] found that for the chaotic time series, the local approximation method, which divides the time series into several smaller segments first and then analyzes them respectively, is superior to the global approximation method. Recently, Alvarez-Ramirez et al. [25, 26], Robinson and Yajima [27], and Bernabe et al. [28] have researched oil price and the characteristics of oil market, and presented that the time series of oil price is a non-linear long-memory process. However, Fan et al. [29]

*Corresponding author.

have applied the pattern-matching technique to crude oil price forecasting and proposed a new approach: generalized pattern matching based on genetic algorithm (GPMGA).

The above-mentioned studies have made important contributions to crude oil price forecasting based on the pattern-matching method. However, after careful analysis, it can be found that all of them only use the forecasted time series, i.e., target domain data. In fact, there are plenty of data available outside the target domain and are often obtained from other time series, i.e., source domains. For example, if we want to forecast the West Texas Intermediate (WTI) crude oil spot price, which is called target domain, then the Brent crude oil spot price series may be utilized as a part of the source domain, because they are very similar in trend. Therefore, making full use of the data in the related time series from source domains is expected to improve the forecasting accuracy of the WTI crude oil spot price. In this regard, the transfer learning technique [30] developed from machine learning technologies seems ideal. The underlying objective is to utilize the knowledge acquired from the related tasks to assist individuals in learning the target task for superior performance. In the last decades, transfer learning has received increasing attention, and has largely been applied to many areas successfully [30], such as text mining and image recognition, but scarcely in the field of economic time series forecasting.

This study introduces the transfer learning technique to crude oil price forecasting, combines it with a pattern-matching forecasting method called analog complexing (AC) and a genetic algorithm, and constructs a hybrid transfer learning model (HTLM). The empirical analysis results show that the forecasting performance of HTLM is superior to that of some existing models.

The structure of this study is organized as follows. Section 2 describes the formulation process of the proposed HTLM model in detail. Utilizing two main crude oil price series—the West Texas Intermediate (WTI) and the Brent crude oil spot prices—Section 3 demonstrates the effectiveness of the proposed methodology, and reports the corresponding results. Finally, Section 4 concludes the study.

2. METHODOLOGY FORMULATION

This section presents the overall process of formulating the HTLM methodology. First, the analog complexing technique and transfer learning method are briefly reviewed. Second, the HTLM methodology is proposed. Finally, the overall steps of the methodology are summarized.

2.1 The analog complexing method

The analog complexing (AC) method was developed by Lorenz and first applied to meteorological forecasting [31]. Further, Lemke and Mueller [32] enhanced the AC algorithm by using an inductive self-organizing approach and an advanced selection procedure to make it applicable to evolutionary processes as well. AC can be considered as a sequential pattern recognition method for predicting, clustering, and classifying complex systems. This method is based on the assumption that typical situations of a time process will repeat in some forms, that is, each actual period of development state of a given multi-dimensional time process may have one or more analogous periods in history. In this way, the forecast of the present state can be obtained by transforming and combining the development states of analogous periods in history.

In general, the AC model follows a four-step procedure [33]: 1) generation of candidate patterns; 2) transformation of analogues; 3) selection of the most similar patterns; and 4) combination forecasting. These processes are discussed in detail.

2.1.1 Generation of candidate patterns

Given an m-dimensional real value series with N observations $x_t = \{x_{1t}, x_{1t}, ..., x_{mt}\}$, $t = 1, 2, ..., N$, a pattern is defined as a table $P_k(i)$ with k rows (observations) beginning from the i-th line (period), where k is the pattern length ($i = 1, 2, ..., N - k + 1$):

(1)

$$P_k(i) = \begin{pmatrix} x_{1i} & \cdots & x_{li} & \cdots & x_{mi} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{1,i+j} & \cdots & x_{l,i+j} & \cdots & x_{m,i+j} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{1,i+k-1} & \cdots & x_{l,i+k-1} & \cdots & x_{m,i+k-1} \end{pmatrix}_{k \times m}.$$

In general, the last pattern, $P^R = P_k(N-k+1)$, just before the forecast origin is selected as the reference pattern. Next, all the remaining candidate patterns $P_k(i)(i = 1, 2, ..., N-k)$ are compared with the reference pattern to find one or more similar patterns with the reference pattern and accordingly forecast the behavior of the system.

Example: For a three-dimensional series with five observations $\begin{pmatrix} 1 & 2 & 3 \\ 5 & 5 & 6 \\ 7 & 9 & 9 \\ 10 & 11 & 13 \\ 15 & 16 & 16 \end{pmatrix}$, suppose the pattern length $k = 3$, then there are two candidate patterns: $P_3(1) = \begin{pmatrix} 1 & 2 & 3 \\ 5 & 5 & 6 \\ 7 & 9 & 9 \end{pmatrix}$, $P_3(2) = \begin{pmatrix} 5 & 5 & 6 \\ 7 & 9 & 9 \\ 10 & 11 & 13 \end{pmatrix}$, and one reference pattern $P^R = \begin{pmatrix} 7 & 9 & 9 \\ 10 & 11 & 13 \\ 15 & 16 & 16 \end{pmatrix}$.

2.1.2 Transformation of analogues

For a reference pattern with a length of k observations, there may be one or more similar patterns in history. However, because the system is dynamic, patterns with similar shapes may have different means and standard deviations. Thus, to compute the similarity between patterns, we must

look for a transformation from candidate patterns to the reference pattern to describe these differences. It is advisable to define the transformed pattern $T[P_k(i)]$ as a linear function of the pattern $P_k(i)$:

(2)
$$T[P_k(i)] = \begin{pmatrix} x'_{1i} & \dots & x'_{li} & \dots & x'_{mi} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_{1,i+j} & \dots & x'_{l,i+j} & \dots & x'_{m,i+j} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x'_{1,i+k-1} & \dots & x'_{l,i+k-1} & \dots & x'_{m,i+k-1} \end{pmatrix}.$$

where $x'_{l,i+j} = a^i_{0l} + a^i_{1l}x_{l,i+j}, i = 1,2,...,N-k; j = 0,1,...,k-1; l = 1,2,...,m$. Considering the data $x_{lz}(l = 1,2,...,m; z = N-k+1, N-k+2,...,N)$ in reference pattern P^R as the datum value (i.e., the dependent variable value), and those in the transformed candidate pattern $T[P_k(i)]$ as the independent variable value, the unknown weights a^i_{0l} and a^i_{1l} for $T[P_k(i)]$ can be estimated by the least squares (LS) method. The parameter a^i_{0l} can be interpreted as the difference of state between the reference pattern and the candidate pattern, and the parameters a^i_{1l} are considered as some uncertainties. It is worth noting that the pattern length $k \geq 3$, because if $k = 2$, the transformed candidate patterns will be the same as the reference pattern.

Taking the candidate pattern $P_3(1)$ in Subsection 2.1.1 as an example, obviously the dimension $m = 3$. We need to estimate the weights a^1_{0l} and a^1_{1l} $(l = 1, 2, 3)$ in each dimension. In the first dimension, the column vectors in P^R and $P_3(1)$ are $\begin{pmatrix} 7 \\ 10 \\ 15 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 5 \\ 7 \end{pmatrix}$. Thus we can obtain $a^1_{01} = 5.25$ and $a^1_{11} = 1.25$ by the LS method. Further the transformed value of the first column in $P_3(1)$ is $\begin{pmatrix} 6.5 \\ 11.5 \\ 14 \end{pmatrix}$ $(=5.25+1.25*\begin{pmatrix} 1 \\ 5 \\ 7 \end{pmatrix})$. Similarly, the weights in the second and third dimensions can be computed as follows: $a^1_{02} = 6.60$, $a^1_{12} = 1.01$, $a^1_{03} = 5.67$, $a^1_{13} = 1.17$. Finally, the transformed pattern $T[P_3(1)] = \begin{pmatrix} 6.5 & 8.62 & 9.18 \\ 11.5 & 11.65 & 12.69 \\ 14 & 15.69 & 16.2 \end{pmatrix}$.

2.1.3 Selection of the most similar patterns

To measure the similarity between the candidate pattern $P_k(i)$ transformed in the subsection above and the reference pattern P^R, we need to compute the distance between the two patterns. In the AC method, the distance between the i-th $(i = 1,2,...,N-k)$ candidate pattern and reference pattern is defined as:

(3)
$$d_i = \frac{1}{k+1}\sum_{j=0}^{k-1}\sqrt{\sum_{r=1}^{m}(x_{r,i+j} - x_{r,N-k+j+1})^2}.$$

Then, the pattern similarity between the i-th candidate pattern and the reference pattern is defined as:

(4)
$$s_i = \frac{1}{d_i}.$$

Obviously, the larger the distance value, the smaller the pattern similarity.

2.1.4 Combination forecasting

Let us assume that F most similar candidate patterns are selected. Then the continuation of each selected pattern is transformed to the reference pattern for forecasting. Suppose the candidate pattern $P_3(1)$ in the above example is selected and the forecast interval length is 1. Then, the continuation of $P_3(1)$ is $\begin{pmatrix} 10 & 11 & 13 \end{pmatrix}$. According to the estimated weights in Subsection 2.1.2, the three dimensions of the continuation of $P_3(1)$ are transformed as follows: $5.25 + 10 * 1.25 = 17.75$, $6.60 + 11 * 1.01 = 17.71$, and $5.67 + 13 * 1.17 = 20.88$. Finally, $\begin{pmatrix} 17.75 & 17.71 & 20.88 \end{pmatrix}$ are just regarded as the one-step forecasting results of the selected pattern $P_3(1)$. Let $R_1, R_2, .., R_F$ be the forecasting results of F patterns, and the combination forecasting results are obtained by the following formula:

(5)
$$R^* = \sum_{i=1}^{F} w_i R_i,$$

where $w_i (i = 1,2,...,F)$ are the weights. In this study, the weights are computed according to pattern similarity.

In the past decades, the AC model has been successfully applied to many areas, including stock price prediction [33] and marketing data analysis [34]. In this study, we apply it to crude oil price forecasting.

2.2 The transfer learning technique

The concept of transfer learning originates from Psychology [30]. It means the ability of individuals to utilize their experience and knowledge learned in the related areas in learning a new task. People can learn new knowledge directly and can also utilize the old knowledge to assist in learning new knowledge. Since its emergence, machine learning has always attempted to simulate the learning of people. Learning new knowledge directly is the traditional machine learning paradigm that we are familiar with. Such methods often suppose that each learning task is independent of the other, so the past learning experience and knowledge will be discarded in learning a new task. Since the 1990s, transfer learning has gained increasing attention worldwide. The underlying objective of transfer learning is to utilize the data or information of the related source tasks to assist in modeling the target task [30, 35] (see Figure 1).

2.3 The HTLM model

For crude oil price time series forecasting, most of the existing models, such as ARIMA, ANN, and GPMGA, only utilize the data information of the forecasted time series,

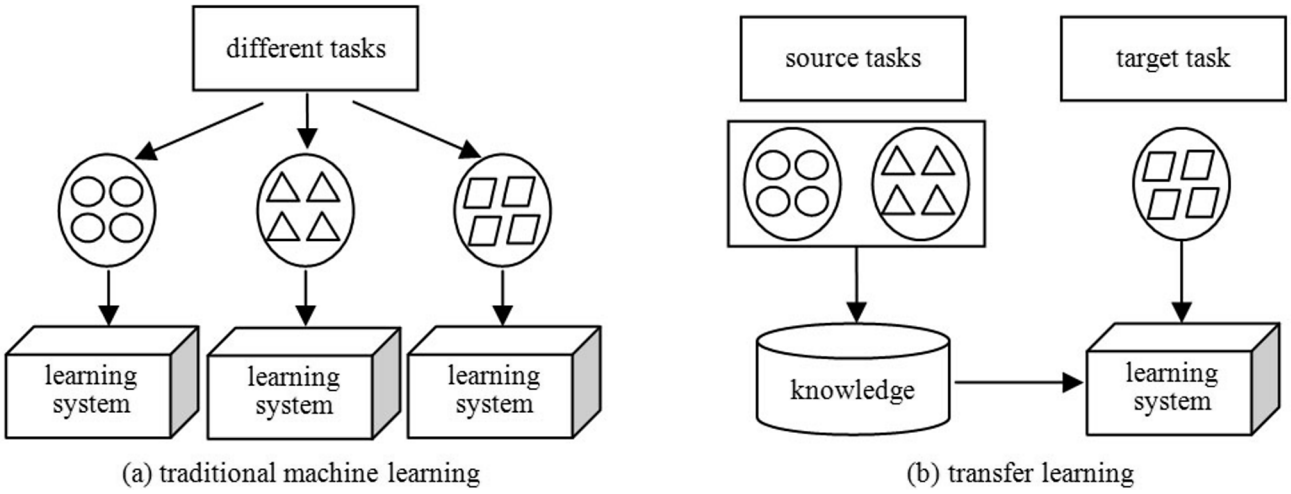

Figure 1. A comparative analysis of traditional machine learning and transfer learning.

i.e., target domain data to model. This may ignore a potentially rich source of information contained in the related time series. Therefore, in this study, the time series data in the related source domain are selectively transferred to the target domain for assisting the target time series modeling and forecasting using the transfer learning technology.

The HTLM model proposed in this study contains two phases: 1) Selection of the related time series in the source domain. Usually, there are numerous source domain time series that may be related to the target domain time series and these data often include much noise. The forecasting performance may be damaged if the data are transferred improperly, that is, negative transfer. Thus, how to transfer the most useful data is very important. 2) Genetic algorithm-based parameter optimization. There are two important parameters when applying analog complexing (AC) to a model: pattern length k and the number of the most similar patterns F. The conventional AC model usually requires fixing the pattern length at a certain level first, generating the reference pattern and candidate patterns, selecting F most similar candidate patterns with the reference pattern to combine, and obtaining the combination forecasting result; subsequently, it changes the value of the pattern length repeatedly. Thus, each pattern length value corresponds to a combination forecasting result, and finally the optimal forecasting result is selected from all combination forecasting results. However, in the conventional AC model, it is difficult to find the optimal match between the pattern length k and the number of the most similar candidate patterns F. To overcome the deficiency of the conventional AC model, this study introduces genetic algorithm to optimize the parameters in the HTLM model.

2.3.1 Selection of the related time series in the source domain

For any economic time series, we can always find many related time series from the source domain. To describe the relevance between these time series and the target time series, we calculate their Pearson correlation coefficient (PCC).

However, because the number of data points in different time series is often different, the PCC cannot be calculated directly. Therefore, in this study, assuming that the target time series and source time series have the same frequency (e.g., all time series data are either daily or monthly), we segment the time series using the following method. If the time series data are daily, we segment each time series into a series of sub-time series with a length of 255 (excluding the Saturdays and Sundays of one year, $360 * 5/7 \approx 255$) starting from the latest time point, and discard the sub-time series of the last part which has less than 255 data points directly. If the time series data are monthly, each original series is divided into a series of sub-series with the length of 12 (i.e., the number of months in one year), and the sub-time series of the last part which has less than 12 data points is discarded.

Suppose that Y is the target time series of a forecasting issue, there are g related time series $S_i(i = 1, 2, ..., g)$ in the source domain, the target time series Y is divided into u sub-time series: $y_1, y_2, ..., y_u$, and the source domain time series S_i is divided into v sub-time series: $s_1, s_2, ..., s_v$. Then the PCC between the sub-time series $y_j(j = 1, 2, ..., u)$ and $s_k(k = 1, 2, ..., v)$ is calculated as follows:

$$r_{j,k} = \frac{\sum_{t=1}^{n}(y_{j,t} - \bar{y}_j)(s_{k,t} - \bar{s}_k)}{\sqrt{\sum_{t=1}^{n}(y_{j,t} - \bar{y}_j)^2}\sqrt{\sum_{t=1}^{n}(s_{k,t} - \bar{s}_k)^2}}, \tag{6}$$

where n denotes the number of data points in each sub-time series ($n = 255$ or 12), $\bar{y}_j$ and $\bar{s}_k$ are the mean values of sub-time series y_j and s_k, respectively. Further, the average similarity between the target time series Y and source domain time series $S_i(i = 1, 2, ..., g)$ is defined as follows:

$$\bar{r} = \frac{\sum_{j=1}^{u}\sum_{k=1}^{v} r_{j,k}}{u * v}. \tag{7}$$

It is easy to observe that $\bar{r} \in [-1, 1]$, and the closer to 1 the value of $\bar{r}$, the stronger the positive correlation between the two time series. In this study, the development trend of the source domain time series, which has stronger positive relevance with the target domain time series, is closer to that of the target domain time series, and thus, it is more helpful to forecast the target domain time series. Therefore, in the HTLM model, we can select some source domain time series that have the largest average similarity with the target time series into the target domain and assist in modeling.

2.3.2 Genetic algorithm-based parameter optimization

The fundamental principle of genetic algorithm (GA) was first introduced by Holland [36]. GA manipulates concepts derived from biology and is philosophically based on Darwin's theory of survival of the fittest [37]. It encodes a possible solution to a specific issue on simple chromosome string-like data structures, applies specified operators to these structures so as to preserve important information, and produces a new population with the purpose of generating strings that map to high function values. The key feature of GA is the manipulation of a population whose individuals are characterized by possessing a chromosome. The latter can be coded as a string of characters which are called bits. Each string represents a feasible solution to the optimization issue. The most important advantage of GA is their ability to use accumulated information about the initially unknown search space in order to bias subsequent searches into useful subspaces [38].

In short, GA is characterized by bit-string representations of potential solutions for a given issue, and these bit-string representations alter and improve these coded solutions by using GA operations. The main operations of GA–selection, crossover, and mutation of genetic information–affect the binary string characteristic in natural evolution. Each generation of GA consists of a new population produced from the previous generation. The binary representation and the main operations of GA in this study can be summarized as follows.

Variable encoding. Before a genetic algorithm can be put to work on any issue, a method is needed to encode potential solutions to that issue in a form that a computer can process. One common approach is to encode solutions as binary strings: sequences of 1 and 0, where the digit at each position represents the value of some aspect of the solution. In the HTLM model, there are two important parameters: pattern length k and the number of most similar patterns F. These parameters are used to combine and obtain the final forecasting results. In this study, we let $1 \leq F \leq 8$ (when $F = 1$, this implies the selection of the most familiar candidate pattern to forecast), and $3 \leq k \leq 13$. Therefore, we let the length of each potential solution (each particle) be 14 binary series, where the front 3 bits denote the value of F and the latter 11 bits denote if the candidate patterns are considered when $k = 3, 4, ..., 13$ in turn (Figure 2).

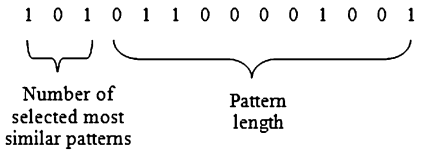

Figure 2. The encoding of a potential solution.

The value of $F = (1 * 2^2 + 1 * 2^0) + 1 = 6$, and the candidate patterns with the lengths of 4, 5, 10, and 13 will be considered simultaneously.

Selection operator. Selection operator biases the search process in favor of the more fitting members based on their fitness value. The fitness of the i-th member in the population can participate in this operation on the basis of probabilities. This probability of the i-th member in the population is calculated as below:

$$\alpha_i = \frac{f_i}{\sum_{c=1}^{q} f_c}, \tag{8}$$

where q is the population size, $f_i (i = 1, 2, ..., q)$ is the fitness of the i-th member. In the selection operation, the members of the population with better fitness can participate several times, while those with worse fitness may be deleted, which leads to an increase in average fitness.

Crossover operator. The crossover operator allows for an exchange of design characteristics among the mating members. The operation is executed by selecting two mating parents, randomly choosing two sites on each of the chromosomal strings and swapping strings between the sites among the pair. An illustration of the crossover operation is as follows:

Parent 1 = 1 0 1 $\hat{0}$ 1 1 0 $\hat{1}$ 1 0 1 $\hat{0}$ 1 1
Parent 2 = 1 0 0 $\underline{1}$ 0 1 0 $\underline{0}$ 1 0 0 $\underline{1}$ 0 1
Child 1 = 1 0 1 $\underline{1}$ 0 1 0 $\underline{0}$ 1 0 0 $\underline{1}$ 0 1
Child 2 = 1 0 0 $\hat{0}$ 1 1 0 $\hat{1}$ 1 0 1 $\hat{0}$ 1 1

The crossover operation is applied with a probability of β which takes the probabilistic values from 0.2 to 0.8 [39].

Mutation operator. Mutation is another essential operator in the GA process and it acts on each chromosome after the crossover operator in the following way. A random number is produced for each bite of a chromosome. If this number is smaller than γ, mutation will occur in that bite; otherwise, it will not happen. If mutation is not applied, after the crossover, the offspring will enter the new generation. According to the research, γ produces most favorable outcomes while varying between 1% and 5%. The mutation operation prevents losing unexpected valuable genetic information in the population during selection and crossover operations. This operator acts at a random place of a chromosome with a low probability of γ [40].

The fitness function. In the HTLM model, for each chromosome, we can determine the values of F and k according to its structure, e.g., in the chromosome shown in

Figure 2, $F = 6, k = 4$, 5, 10, and 13. Then, the reference pattern and all candidate patterns with the length of 4, 5, 10, and 13 are generated, and the average similarity between the reference pattern and each candidate pattern is calculated according to Eq. (7). Next, it selects F (=6) candidate patterns with the highest similarity, and combines their forecasting results to obtain the final forecasting results of the chromosome. To evaluate the forecasting results of each chromosome, we define the following fitness function:

$$Min\Delta = \frac{1}{\lambda}\sum_{t=1}^{\lambda}(\rho_t - \hat{\rho}_t)^2, \tag{9}$$

where ρ_t and $\hat{\rho}_t$ are the observed values and predicted values, and λ is the number of observations in the test set. As the function is minimization, individuals with less amount of fitness are chosen for each generation.

Let $S_1, S_2, ..., S_g$ be g source domain time series, Y the target domain time series, T_{rain} the training set for modeling in target domain, T_{est} the test set used to verify the performance of each chromosome in target domain ($Y = T_{rain} \cup T_{est}$), δ the generation number of genetic algorithm, η the population size, β the probability of crossover, and γ be the probability of mutation. Figure 3 shows the modeling process of the HTLM model.

3. EMPIRICAL ANALYSIS

3.1 Research data

There are numerous crude oil price series. The West Texas Intermediate (WTI) crude oil spot price and Brent crude oil spot price are two of the most famous benchmark prices, and are widely used as the basis of many crude oil price formulae. Therefore, in this study, we select the two oil price series as the experimental samples. The two crude oil price data used in this study are monthly data, and are downloaded from the energy information administration (EIA) website of the Department of Energy (DOE) of USA (http://www.eia.doe.gov/).

For the WTI crude oil spot price, we take the monthly data from January 1986 to December 2014, and there are 348 observations. For convenience of modeling, the data from January 1986 to December 2010 are used for the training set (300 observations), and the remainder are used as the test set (48 observations) for evaluating its prediction performance. For the Brent crude oil spot price, the sampling data are from January 1988 to December 2014, which includes 324 observations. Similarly, we take the data from January 1988 to December 2010 as the training set (276 observations), and the data from January 2011 to December 2014 as the test set (48 observations).

Further, we regard four oil price time series as the source time series, including the New York Harbor conventional gasoline regular spot price FOB (NYCGRS), U.S. Gulf Coast conventional gasoline regular spot price FOB (GCCGRS), New York Harbor No. 2 heating oil spot price FOB (NY-2-HOS), and U.S. Gulf Coast kerosene-type jet fuel spot price FOB (GCKJFS). The first three series are from January 1987 to December 2010, and the last one is from January 1991 to December 2010. It is worth noting that the Brent crude oil spot price is also regarded as the source time series when forecasting the WTI crude oil spot price, and vice versa.

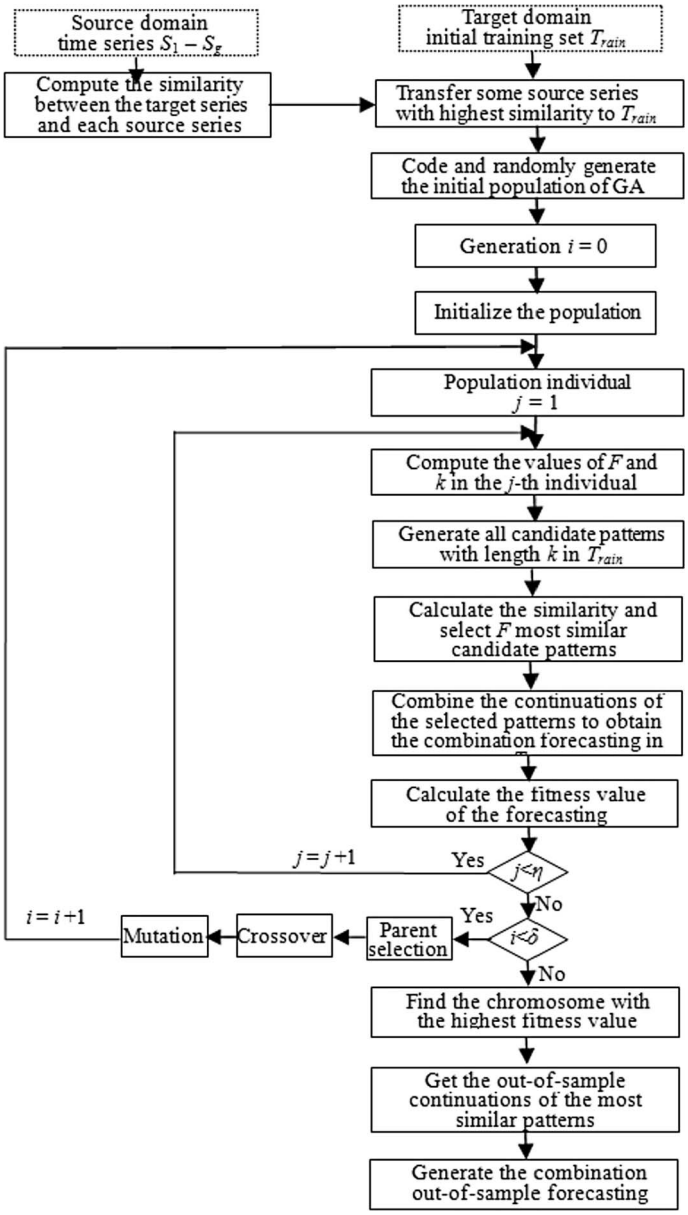

Figure 3. Block diagram of the HTLM model.

3.2 Experimental setting and evaluation criteria

In this study, we compare the HTLM model with the traditional AC, ARIMA, GPMGA, and Elman network models.

Table 1. The average similarity between the target and source time series

	WTI	Brent	NYCGRS	GCCGRS	NY-2-HOS	GCKJFS
WTI	-	-0.0547	**0.0822**	**0.0662**	0.0310	0.0642
Brent	-0.0547	-	**0.0839**	0.0641	0.0402	**0.0738**

The ARIMA model is implemented via the Eviews software package, and the other models are implemented in the platform Matlab 7.0.

In the ARIMA model, the future value of a series is assumed to be a linear function of several past observations and random errors [41]. Suppose that z_t is the observation of a time series, and its mean is μ. A mixed autoregressive moving average (ARMA) model is expressed as:

$$\phi(B)\bar{z}_t = \theta(B)a_t, \tag{10}$$

where $\bar{z}_t = z_t - \mu$, a_t and B are the random error of the time series at time t and the backward shift operator, respectively. $\phi(B) = 1 - \sum_{i=1}^{p} \phi_i B^i$ and $\theta(B) = 1 - \sum_{j=1}^{q} \theta_j B^j$ are polynomials of degree p and q respectively, $\phi_i (i = 1, 2, ..., p)$ and $\theta_j (j = 1, 2, ..., q)$ are model parameters, and p and q are integers and often referred as the lag orders of the model. Further, the random errors a_t are assumed to be independently and identically distributed with a mean of zero and a constant variance of σ^2. If the d-th difference of $\{z_t\}$ is an ARMA process of order p and q, then z_t is called an ARIMA $(p - d - q)$ process.

The architecture of Elman neural network (ENN) introduced by Elman [42] is very similar to the standard feedforward architecture with layers of input units, hidden units, and output units. However, it is augmented at the input layer by additional units, called context units. The number of context units is equal to that of hidden units. The augmented input units, including both the input units and the context units, activate the hidden units. The input value of a context unit at time step $t + 1$ is exactly the same as the output value of the hidden unit at time step t. Thus, the current context units, which transfer the previous state of the hidden units to the input layer, are recognized as a one-step time delay [43]. In this study, a three-layer ENN $(I - H - O)$ is selected, where I, H, and O denote the number of input neurons, hidden nodes, and output neurons, respectively.

The GPMGA model [29] is developed on the basis of pattern modeling and recognition system (PMRS). In traditional PMRS, the reference pattern and the past candidate pattern are matched without any transformation. However, the complexity of oil price movements often makes this kind of direct matching inaccurate. Thus, GPMGA takes the differences between the reference pattern and candidate pattern in addition to their similarities into consideration, and the candidate pattern is scaled both in the x- and y-axes directions to match the reference pattern indirectly. Finally, the parameters are optimized by genetic algorithm.

As for ARIMA $(p - d - q)$ and ENN $(I - H - O)$, the parameters can be determined by trial-and-error. For WTI crude oil spot price series, we find that ARIMA (2–1–1) and ENN (10–12–1) can achieve the best performance, while for Brent crude oil spot price series, ARIMA (2–1–1) and ENN (12–15–1) are selected.

For the GPMGA model, the parameter setting is the same as in [29, 44]. Specifically, the lower and upper bounds of the pattern transform factors (α, β) are set as $\underline{\alpha} = 0.5$, $\bar{\alpha} = 2$, $\underline{\beta} = 0.5$, $\bar{\beta} = 2$ and pattern length $2 \leq k \leq 25$. Finally, we find that when the pattern length $k = 14$ for WTI data and $k = 12$ for Brent data, the GPMGA model can achieve the best forecasting performance through comparison.

In AC and HTLM models, the pattern length k and the number of the most similar patterns F are two important parameters. As for the AC model, we let $k = 12$ and $F = 2$ for WTI crude oil time series, and $k = 11$ and $F = 6$ for Brent crude oil time series, after repeated experiments. While for the HTLM model, we first compute the average similarity between the target time series and the source time series (see Table 1). The bold face in this table denotes the two maximum values of each row. We transfer two source time series with the highest average similarity to the target domain. According to Table 1, we transfer the NYCGRS and GCCGRS series to assist in modeling for the WTI crude oil spot price, and NYCGRS and GCKJFS for the Brent crude oil spot price. Further, the parameters k and F of the HTLM model are optimized by genetic algorithm with the following user-specified parameters: $\delta = 50$, $\eta = 100$, $\beta = 0.9$, and $\gamma = 0.05$. Finally, it is found that the optimal values of k are 3, 8, and 11, and the optimal value of F is 3 for WTI data. Meanwhile, the optimal values of k are 3 and 6, and the optimal value of F is 5 for Brent data.

In order to evaluate the forecasting performance, it is necessary to introduce some evaluation criteria. In this study, three commonly used evaluation criteria, root mean square error $(RMSE)$, mean absolute percentage error $(MAPE)$, and direction statistics (D_{stat}) [19, 44], are introduced.

Given λ pairs of the observed values ρ_t and predicted values $\hat{\rho}_t$, the $RMSE$, which describes the estimates' deviation from the real values, is calculated as:

$$RMSE = \sqrt{\frac{1}{\lambda}\sum_{t=1}^{\lambda}(\rho_t - \hat{\rho}_t)^2}. \tag{11}$$

The second is the $MAPE$ that can be written as

$$MAPE = \frac{1}{\lambda}\sum_{t=1}^{\lambda}\left|\frac{\rho_t - \hat{\rho}_t}{\rho_t}\right|. \tag{12}$$

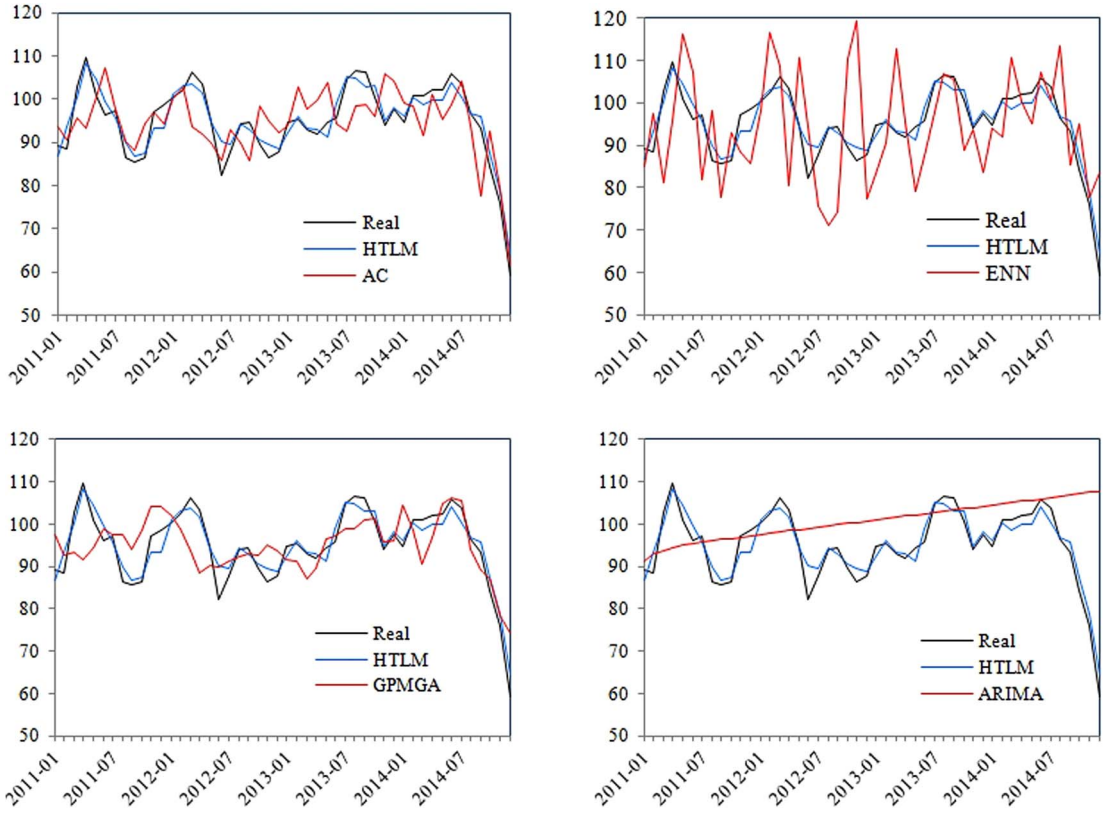

Figure 4. Forecasting results of the five models for WTI crude oil spot price series on the test set.

The smaller the values of $RMSE$ and $MAPE$, the smaller the error between the actual values and the forecasting values and the better the performance of the model.

However, $RMSE$ and $MAPE$ cannot provide direct suggestions to decision makers. Many decision makers, such as investors, are more interested in the direction of the change. Therefore, we introduced directional change statistics D_{stat}. Typically, the D_{stat} can be defined as:

$$D_{stat} = \frac{1}{\tau}\sum_{t=1}^{\tau}\pi_t, \tag{13}$$

where $\pi_t = 1$, if $(\rho_t - \rho_{t-1})(\hat{\rho}_t - \hat{\rho}_{t-1}) > 0$, and $\pi_t = 0$ otherwise. The larger the value of D_{stat}, the smaller the difference between the actual directional change of crude oil price time series and the forecasted directional change, and the better the performance of the model.

3.3 Experimental results

The forecasting results of the five models in the test set of WTI crude oil spot price series are shown in Figure 4. As demonstrated, the forecasting results of two traditional models, ARIMA and ENN, are largely different from the actual values. Especially, the forecasting results of the ARIMA model approximately follow a straight line, which cannot forecast the fluctuations in the WTI crude oil spot price series. Thus, their performances are the poorest. The forecasting results of the AC model are distinctly superior to those of ARIMA and ENN models. Its forecasting values on the wave crest and trough distributions are closer to the actual values, although there is still some difference. Whereas, the forecasting results of the GPMGA model are similar to those of the AC model. Finally, the forecasting results of the HTLM model proposed in this study are very close to the actual values in the whole test set. In particular, its forecasting results on wave crest and trough distributions are also better than those of the other four models. Therefore, the forecasting performance of the HTLM model is the best among the five models.

Further, Table 2 shows the comparison of the forecasting performance of the five models in the test set of WTI crude oil spot price series. With regard to the RMSE and MAPE evaluation criteria, the smaller their values, the better the performance of the model. Therefore, we rank the five models in ascending order. While for the Dstat criterion, the larger their values, the better the performance of the model. Therefore, we rank the five models in descending order. Focusing on the RMSE and MAPE indicators, the two evaluation criteria which measure the goodness-of-fit of the forecasting result, the HTLM model exhibits the best performance, followed by GPMGA, AC, ARIMA, and ENN. Further, as for the mea-

Table 2. Forecasting performance comparison of the five models for the WTI crude oil spot price

Methods	$RMSE$	Rank	$MAPE$	Rank	D_{stat}	Rank
HTLM	2.6682	1	0.0238	1	0.8125	1
AC	7.1248	3	0.0611	3	0.5625	3
ARIMA	11.7099	4	0.0970	4	0.5000	4
ENN	13.1316	5	0.1177	5	0.4375	5
GPMGA	6.8955	2	0.0594	2	0.5833	2

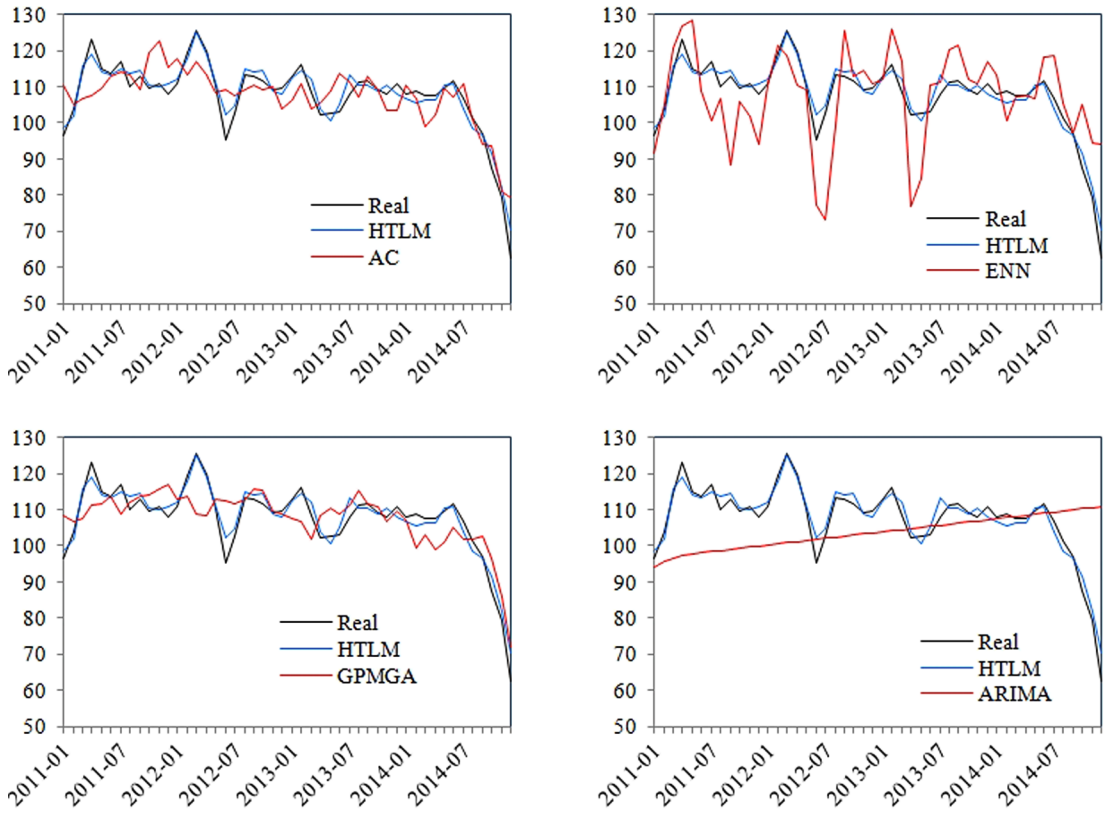

Figure 5. Forecasting results of the five models for the Brent crude oil spot price series on the test set.

surement Dstat, which describes the forecasting ability of directional change and is more important for business practitioners, we find that the HTLM model outperforms the others.

Figure 5 displays the forecasting results of the five models in the test set of Brent crude oil spot price series. As shown in the figure, the forecasting results of the ENN model sharply fluctuate around the actual values in the whole test set, and those of the ARIMA model are too smooth to exhibit the violent fluctuation of the actual Brent crude oil spot price series. The forecasting results of the AC and GPMGA models are closer to the actual values than those of the ARIMA and ENN models. Last, but not the least, the forecasting results of the HTLM model are still superior, and they are closest to the actual values of the Brent crude oil spot price series.

In addition, the forecasting performance comparison of the five models in the test set of Brent crude oil spot price series is shown in Table 3. We observe that the RMSE and MAPE criteria values of the HTLM model are the smallest and its Dstat criterion value is the largest, which demonstrates that the whole forecasting performance of the HTLM model is the best. The performance of the two traditional models, ARIMA and ENN, is still the poorest. They are ranked fifth and fourth respectively according to three evaluation criteria. It is worth noting that the AC model outperforms the GPMGA model according to RMSE and MAPE criteria, while the GPMGA model performs better than the AC model as to Dstat, which demonstrates that the low RMSE and MAPE criteria values do not necessarily imply that there is a high hit rate in forecasting the crude oil price movement direction.

3.4 Further discussion

According to the experiments presented in this study, we can conclude the following.

Table 3. Forecasting performance comparison of the five models for the Brent crude oil spot price

Methods	$RMSE$	Rank	$MAPE$	Rank	D_{stat}	Rank
HTLM	2.5914	1	0.0203	1	0.7292	1
AC	6.7708	2	0.0528	2	0.5833	3
ARIMA	13.6319	5	0.0988	5	0.5000	5
ENN	11.5926	4	0.0860	4	0.5208	4
GPMGA	6.9416	3	0.0532	3	0.6458	2

(1) The forecasting performance of HTLM, GPMGA, and AC models is better than that of the linear forecasting model ARIMA and non-linear forecasting model ENN. Compared with the actual crude oil spot price series, the forecasting results of the ARIMA model are too smooth. Thus, it cannot forecast the wave crest and trough distributions of crude oil spot price time series accurately. However, the forecasting results of HTLM, GPMGA, and AC models are closer to the actual values. The HTLM, GPMGA, and AC models all belong to pattern matching methods. The experimental results in this study also demonstrate that the forecasting performance of pattern matching methods is better than linear and non-linear models for non-linear long memory processes such as crude oil price time series, which is basically consistent with the conclusion of Peters [22].

(2) The forecasting performance of the HTLM model proposed in this study is superior to that of the AC model according to three evaluation criteria when forecasting WTI and Brent crude oil spot price series. This may be caused by two reasons: 1) The HTLM model introduces the transfer learning technique. It can transfer part of source domain series most related to target time series to the target domain, enrich the data for pattern matching, and make it be possible to search for more similar candidate patterns with the reference pattern to improve the forecasting performance. 2) Genetic algorithm is adopted to optimize the parameters of the HTLM model, which can overcome the disadvantages of the AC model (i.e., hard to find the optimal match among the parameters) to some extent.

(3) The HTLM model outperforms the GPMGA model. Both GPMGA and HTLM models belong to pattern-matching methods; however, their principles are largely different. First, as for the pattern transformation mechanism, in the GPMGA model, the candidate pattern is scaled both in the x- and y-axes directions to match the reference pattern; while in the HTLM model, each candidate pattern is translated to the same position with the reference pattern through a linear transformation (the transformation coefficients are obtained through the LS method), and then the similarity between the candidate pattern and the reference pattern is calculated. Second, the GPMGA model only selects the candidate pattern that is most similar to the reference pattern to forecast. In reality, the forecasting results obtained from selecting the most similar single pattern may be poorer than those obtained from combining some most similar patterns, that is to say, forecasting ensemble. While, the HTLM model just overcomes this deficiency of the GPMGA model, and it selects the most one or more similar candidate patterns for forecasting adaptively.

4. CONCLUSION

This study combines the transfer learning technique with a pattern-matching forecasting method called analog complexing (AC) and a genetic algorithm, and constructs a hybrid transfer learning-based analog complexing model (HTLM) for crude oil price forecasting. The empirical analysis results for the West Texas Intermediate (WTI) and the Brent crude oil spot prices show that the proposed HTLM model outperforms the other four models in terms of different criteria. In all testing cases, the $RMSE$ and $MAPE$ are the lowest and the D_{stat} is the highest, indicating that the HTLM forecasting model can be used as a very promising methodology for world crude oil price prediction.

ACKNOWLEDGEMENTS

The authors thank the anonymous referees and the editor for their helpful comments. Their comments have played an important role in improving this paper. This research is partly supported by the Natural Science Foundation of China under Grant Nos. 71471124 and 71301160, National Social Science Foundation of China under Grant No. 14BGL175, Youth Foundation of Sichuan Province under Grant No. 2015RZ0056, Excellent Youth fund of Sichuan University under Grant No. skqx201607, MOE Youth Project of Humanities and Social Sciences under Grant No. 15YJC860034, and Natural Science Foundation of Anhui Higher Education Institutions under Grant No. KJ2016A604, Youth backbone visiting research key project (gxfxZD2016219), CSC(201506500007).

Received 17 May 2015

REFERENCES

[1] EIA (2006). World consumption of primary energy by energy type and selected country groups, 1980–2006. *International Energy Annual 2006*.

[2] OECD/IEA (2008). *World Energy Outlook 2008*. Paris.

[3] Hagen, R. (1994). How is the international price of a particular crude determined? *OPEC Review* **18** 145–158.

[4] Stevens, P. (1995). The determination of oil prices 1945–1995. *Energy Policy* **23** 861–870.

[5] Rotemberg, J. J. and Woodford, M. (1996). Imperfect competition and the effects of energy price increases on economic activity. *Journal of Money, Credit and Banking* **28** 550–577.
[6] Sauter, R. and Awerbuch, S. (2003). Oil price volatility and economic activity: a survey and literature review. *IEA Research Paper* **28** 550–577.
[7] Kang, S. H., Kang, S. M., and Yoon, S. M. (2009). Forecasting volatility of crude oil markets. *Energy Economics* **31** 119–125.
[8] Zagaglia, P. (2010). Macroeconomic factors and oil futures prices: a data-rich model. *Energy Economics* **32** 409–417.
[9] Mohammadi, H. and Su, L. (2010). International evidence on crude oil price dynamics: applications of ARIMA–GARCH models. *Energy Economics* **32** 1001–1008.
[10] Wen, F., He, Z., Dai, Z., and Yang, X. (2014). Characteristics of investors' risk preference for stock markets. *Economic Computation and Economic Cybernetics Studies and Research* **3**(48) 235–254.
[11] Emery, M., Ryan, L., and Whiting, B. (2012). Multi model forecasts of the west Texas intermediate crude oil spot price. *Available at SSRN:* http://ssrn.com/abstract=2079341.
[12] Panas, E. and Ninni, V. (2000). Are oil markets chaotic? A nonlinear dynamic analysis. *Energy Economies* **22**(5) 549–568.
[13] Adrangi, B., Chatrath, A., Dhanda, K. K., and Raffiee, K. (2001). Chaos in oil prices? Evidence from futures markets. *Energy Economies* **23** 405–425.
[14] Weigend, A. S. and Gershenfeld, N. A. (1994). *Time Series Prediction: Forecasting the Future and Understanding the Past.* Addison–Wesley, MA.
[15] Abramson, B. and Finizza, A. (1991). Using belief networks to forecast oil prices. *International Journal of Forecasting* **7** 299–315.
[16] Xiong, T., Bao, Y., and Hu, Z. (2013). Beyond one-step-ahead forecasting: evaluation of alternative multi-step-ahead forecasting models for crude oil prices. *Energy Economics* **40**(7) 405–415.
[17] Wang, S. Y., Yu, L., and Lai, K. K. (2004). A novel hybrid AI system framework for crude oil price forecasting. *Lecture Notes in Computer Science* **3327** 233–242.
[18] Xie, W., Yu, L., Xu, S. Y., and Wang, S. Y. (2006). A new method for crude oil price forecasting based on support vector machines. *Lecture Notes in Computer Science* **3994** 441–451.
[19] Yu, L., Wang, S. Y., and Lai, K. K. (2008). Forecasting crude oil price with an EMD–based neural network ensemble learning paradigm. *Energy Economics* **30** 2623–2635.
[20] Moshiri, S.(2004). Testing for deterministic chsaos in futures crude oil price: does neural network lead to better forecast? *University of Allameh Tabatabaie Working Paper.*
[21] Yu, L., Wang, S. Y., and Lai, K. K. (2006). *Forecasting Foreign Exchange Rates and International Crude Oil Price Volatility–TEI@I Methodology.* Hunan University Press, Changsha.
[22] Peters, E. (1994). *Fractal Market Hypothesis: Applying Chaos Theory to Investment and Economics.* New York, Wiley.
[23] Wen, F., Li, Z., Xie, C., and Shaw, D. (2012). Study on the fractal and chaotic features of the Shanghai composite index. *Fractals-Complex Geometry Patterns and Scaling in Nature and Society* **20**(2) 133–140.
[24] Farmer, J. D. and Sidorowich, J. J. (1988). Predicting chaotic dynamics. In: *Kelso, J. A. S., Mandell, A. J., and Shlesinger, M. F. (Eds.), Dynamic Patterns in Complex Systems.* World Scientific, Singapore, 265–292. MR1037036
[25] Alvarez-Ramirez, J., Cisneros, M., and Iban-Valdez, C. (2002). Muhifractal hurst analysis of crude oil prices. *Physica A* **313**(3–4) 651–670.
[26] Alvarez-Ramirez, J., Soriano, A., Cisneros, M., and Suarez, R. (2003). Symmetry/anti-symmetry phase transitions in crude oil markets. *Physica A* **322**(5) 583–596. MR1980954
[27] Robinson, P. M. and Yajima, Y. (2002). Determination of cointegrating rank in fractional systems. *Journal of Econometrics* **106** 217–241. MR1884249
[28] Bernabe, A., Martina, E., Alvarez-Ramirez, J., and Ibarra-Valdez, C. (2004). A multi-model approach for describing crude oil price dynamics. *Physica A* **338** 567–584.
[29] Fan, Y., Liang, Q., and Wei Y. M. (2008). A generalized pattern matching approach for multi-step prediction of crude oil price. *Energy Economics* **30** 889–904.
[30] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* **22** 1345–1359.
[31] Lorenz, E. N. (1969). Atmospheric predictability as revealed by naturally occurring analogues. *Journal of the Atmospheric Sciences* **26** 636–646.
[32] Lemke, F. and Mueller, J. A. (1997). Self-organizing data mining for a portfolio trading system. *Journal of Computational Intelligence in Finance* **5** 12–26.
[33] Mueller, J. A. and Lemke, F. (2000). *Self-Organising Data Mining: An Intelligent Approach to Extract Knowledge from Data.* Hamburg, Libri.
[34] Ivakhnenko, G. A. (2008). Application of forecasting GMDH algorithms for marketing problems solution. *Journal of Automation and Information Sciences* **40** 70–82.
[35] Taylor, M. E. and Stone, P. (2009). Transfer learning for reinforcement learning domains: a survey. *Journal of Automation and Information Sciences* **40** 70–82. MR2534874
[36] Holland, J. H. (1975). *Adaptations in Natural Artificial Systems.* University of Michigan Press, Michigan. MR0441393
[37] Dunn, L. C. and Dobhansky, T. (1950). *Principles of Genetics.* McGraw-Hill, New York.
[38] Ceylan, H. and Ozturk, H. K. (2004). Estimating energy demand of Turkey based on economic indicators using genetic algorithm approach. *Energy Conversion and Management* **45** 2525–2537.
[39] Canyurt, O. E. and Ozturk, H. K. (2008). Application of genetic algorithm (GA) technique on demand estimation of fossil fuels in Turkey. *Energy Policy* **36** 2562–2569.
[40] Azadeh, A. and Tarverdian, S. (2007). Integration of genetic algorithm, computer simulation and design of experiments for forecasting electrical energy consumption. *Energy Policy* **35** 5229–5241.
[41] Box, G. E. P. and Jenkins, G. (1970). *Time Series Analysis: Forecasting and Control.* Holden–Day, San Francisco, CA. MR0272138
[42] Elman, J. L. (1990). Finding structure in time. *Cognitive Science* **14**(2) 179–211.
[43] Kremer, S. C. (1995). On the computational power of Elman–style recurrent networks. *IEEE Transactions on Neural Networks* **6** 1000–1004.
[44] Yu, L., Zhao, Y., and Tang, L. (2014). A compressed sensing based AI learning paradigm for crude oil. *Energy Economics* **46**(11) 236–245.

Jin Xiao
Business School
Sichuan University
Chengdu 610064
China

Department of Mathematics and Computer Science
University of Muenster
Muenster, 48149
Germany
E-mail address: xiaojin@scu.edu.cn

Yi Hu
School of Economics and Management
University of Chinese Academy of Sciences
Beijing 100190
China
E-mail address: huyi@ucas.ac.cn

Yi Xiao
School of Information Management
Central China Normal University
Wuhan 430079
China
E-mail address: xybill@amss.ac.cn

Lixiang Xu
Department of Mathmatics and Physics
Hefei University
Hefei 230601
China

Department of Mathematics and Computer Science
University of Muenster
Muenster, 48149
Germany
E-mail address: xulixianghf@163.com

Shouyang Wang
Academy of Mathematics and Systems Science
Chinese Academy of Sciences
Beijing 100190
China
E-mail address: sywang@amss.ac.cn