Journal of Modern Accounting and Auditing, January 2015, Vol. 11, No. 1, 51-62
doi: 10.17265/1548-6583/2015.01.005

Improved Crude Oil Price Forecasting With Statistical Learning Methods*

Chokri Slim
Manouba University, Manouba, Tunisia

Reliable forecasts of the price of oil are of interest for a wide range of applications. For example, central banks and private sector forecasters view the price of oil as one of the key variables in generating macroeconomic projections and in assessing macroeconomic risks. Of particular interest is the question of the extent to which the price of oil is helpful in predicting recessions. This paper presents a statistical learning method (SLM) based on combined fuzzy system (FS), artificial neural network (ANN), and support vector regression (SVR) to cope with optimum long-term oil price forecasting in noisy, uncertain, and complex environments. A number of quantitative factors were discovered from this model and used as the input. For verification and testing, the West Texas Intermediate (WTI) crude oil spot price is used to test the effectiveness of the proposed learning methodology. Empirical results reveal that the proposed SLM-based forecasting can model the nonlinear relationship between the input variables and price very well. Furthermore, in-sample and out-of-sample prediction performance also demonstrates that the proposed SLM model can produce more accurate prediction results than other nonlinear models.

Keywords: crude oil price, fuzzy system (FS), artificial neural networks (ANNs), support vector regression (SVR)

Introduction

Crude oil has been playing an increasingly important role in the world economy since nearly two thirds of the world's energy demands are met from crude oil (Alvarez-Ramirez, Soriano, Cisneros, & Suarez, 2003). It is said that crude oil is also the world's largest and most actively traded commodity, accounting for over 10% of total world trade (Verleger, 1993). Like most commodities, crude oil price is mainly influenced by international politics, economy, military affairs, diplomacy, and other factors. The frequent change of these factors makes oil price show uncertainty, mutagenicity, and randomness. In addition, crude oil products are one of the world's major commodities with a high volatility level. They are traded in New York Mercantile Exchange (NYMEX) market together with other energy and mineral commodities. Simultaneously, the volatility of this crude oil price depends on demand and supply of the commodity, level of inventories, and economic indicators.

In the past decades, traditional statistical and econometric techniques, such as linear regression (LinR), co-integration analysis, Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models, naive

* **Acknowledgement:** Chokri would like to thank the Higher Institute of Accounting and Business Enterprises and Manouba University for funding support.
Chokri Slim, associate professor, Higher Institute of Accounting and Business Enterprises, Manouba University. Email: chokri.slim@iscae.rnu.tn.

random walk, vector auto-regression (VAR), and error correction models (ECMs), have been widely applied to crude oil price forecasting. Huntington (1994) applied a sophisticated econometric model to predict crude oil prices in the 1980s. Abramson and Finizza (1995) utilized a probabilistic model for predicting oil prices. Morana (2001) suggested a semi-parametric statistical method for short-term oil price forecasting based on the GARCH properties of crude oil price. Similarly, Barone-Adesi, Bourgoin, and Giannopoulos (1986) suggested a semi-parametric approach for oil price forecasting. Gulen (1998) used a co-integration analysis to predict the West Texas Intermediate (WTI) price.

Ye, Zyren, and Shore (2006) presented a simple econometric model of WTI prices, using OECD petroleum inventory levels, relative inventories, and high- and low-inventory variables. Mirmirani and Li (2004) used the VAR model to predict US oil price. Lanza, Manera, and Giovannini (2005) investigated crude oil and oil products' prices using ECM.

Usually, the above models can provide good prediction results when the price series under study is linear or near linear. However, in real-world crude oil price series, there is a great deal of nonlinearity and irregularity. Numerous experiments have demonstrated that the prediction performance might be very poor if one continued using these traditional statistical and econometric models (Weigend & Gershenfeld, 1994). The main reason leading to this phenomenon is that the traditional statistical and econometric models are built on linear assumptions and they cannot capture the nonlinear patterns hidden in the crude oil price series.

Due to the limitations of the traditional statistical and econometric models, some nonlinear and emerging artificial intelligent (AI) models, such as nonlinear regression, artificial neural networks (ANNs), support vector machines (SVMs), and genetic programming (GP), provide powerful solutions to nonlinear crude oil price prediction. For example, Abramson and Finizza (1991) used belief networks, a class of knowledge-based models, to forecast crude oil prices. Kaboudan (2001) employed GP and ANN to forecast crude oil price. Tang and Hammoudeh (2002) proposed a nonlinear regression model to forecast OPEC basket price, and Mirmirani and Li (2004) used the ANN model with genetic algorithm (GA) to predict crude oil price and compared the results with the VAR model. Xi, Poo, and Choo (2007) proposed a support vector regression (SVR) model to predict crude oil price. Similarly, many experiments found that the AI-based models often had some advantages over statistical-based models. However, these AI models also have their own shortcomings and disadvantages. For example, ANN often suffers from local minima and over-fitting, while other AI models, such as SVM and GP including ANN, are sensitive to parameter selection (Chokri, 2007; 2009).

Considering these problems, in Section 2, we give a brief description of the general paradigms of the ANNs and SVR, especially those used for the forecasting purpose. We propose, in Section 3, a new statistical learning method (SLM) based on combined fuzzy system (FS), ANN, and SVR using a number of quantitative factors as the input variables. In Section 4, we first analyzed the variables influencing the oil price and determined the number of lag variables at the same time. Then, we make a nonlinear model by SLM and adjust the variables and SLM parameters until the nonlinear model fits the actual price level very well. The proposed model provides some in-sample and out-of-sample forecasts compared with those derived from other models. The final section summarizes the paper and offers some directions for future work.

Background Knowledge

In this section, we give a brief presentation of some background knowledge of ANNs and SVR for regression. Firstly, we provide the ANNs in this section. We will focus on a particular structure of ANNs, multi-layer feedforward network, which is the most popular and widely used network paradigm in many applications including forecasting. In the second subsection, we describe a novel type of learning machine called SVR, which has been receiving increasing interest in areas ranging from its original application in pattern recognition to other applications such as regression estimation due to its remarkable generalization performance.

ANNs

Numerous research and applications of ANNs in business have proven their advantages in relation to classical linear models. According to Wong, Bodnovich, and Selvi (1995), the most frequent areas of ANN applications in the past 10 years are production/operations (53.5%) and finance (25.4%).

Predicting the future behavior of real-world financial time series using ANNs has been extensively investigated, because neural networks can learn nonlinear relationships between inputs and desired outputs. Integration of knowledge and ANNs has also been extensively investigated, because such integration holds a great promise in solving complicated real-world problems.

Analytically, we could say that there is an extensive literature in financial applications of ANNs (Chokri & Abdelwahed, 2003; Chokri, 2004; 2007; 2009; Trippi & Turban, 1993; Azoff, 1994; Refenes, 1995; Gately, 1996).

ANNs are structures of highly interconnected elementary computational units. Each computational unit has a set of input connections that receive signals from other computational units and a bias adjustment, a set of weights for each input connection and bias adjustment, and a transfer function that transforms the sum of the weighted inputs and bias to decide the value of the output from the computational unit. The sum value for the computational unit is the linear combination of all signals from each connection times. The output is the result of applying a transfer function to the sum value.

An ANN is typically composed of layers of nodes (see Figure 1). In the popular multi-layer models, all the input nodes are in one input layer, all the output nodes are in one output layer, and the hidden nodes are distributed into one or more hidden layers in between. In designing such a model, one must determine the following variables:

(1) The number of input nodes;

(2) The number of hidden layers and hidden nodes;

(3) The number of output nodes.

The selection of these parameters is basically problem-dependent. Although there exist many different approaches such as the pruning algorithm (Sietsma & Dow, 1988; Karnin, 1990; Weigend, Rumelhart, & Huberman, 1991; Reed, 1993; Cottrell, Girard, Girard, Mangeas, & Muller, 1995), the polynomial time algorithm (Roy, Kim, & Mukhopadhyay, 1993), the canonical decomposition technique (Wang, Massimo, Tham, & Morris, 1994), and the network information criterion (Murata, Yoshizawa, & Amari, 1994) for finding the optimal architecture of an ANN, these methods are usually quite complex in nature and are difficult to implement. Furthermore, none of these methods can guarantee the optimal solution for all real forecasting problems. To date, there is no simple clear-cut method for determination of these parameters.

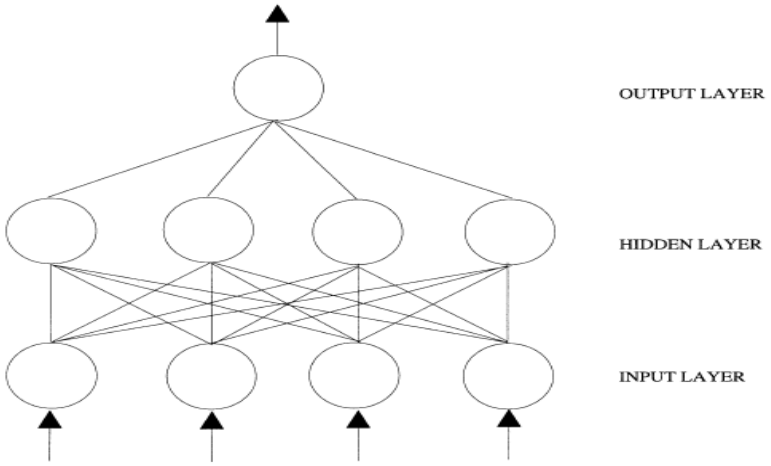

Figure 1. A typical feedforward neural network.

Guidelines are either heuristic or based on simulations derived from limited experiments. Hence, the design of an ANN is more of an art than a science.

The neural network training is an unconstrained nonlinear minimization problem in which weights of a network are iteratively modified to minimize the overall mean or total squared error between the desired and actual output values for all output nodes over all input patterns. The existence of many different optimization methods (Fletcher, 1987) provides various choices for neural network training. There is no algorithm currently available to guarantee the global optimal solution for a general nonlinear optimization problem in a reasonable amount of time. The most popularly used training method is the back propagation algorithm. A back propagation ANN uses a feedforward topology, supervised learning, and the back propagation algorithm (Rumelhart, Hinton, & Williams, 1986).

As we mentioned earlier, training and a test sample are typically required for building an ANN forecaster. The training sample is used for ANN model development and the test sample is adopted for evaluating the forecasting ability of the model. Sometimes, a third one, called the validation sample, is also utilized to avoid the overfilling problem or to determine the stopping point of the training process (Weigend, Huberman, & Rumelhart, 1992). It is common to use one test set for both validation and testing purposes particularly with small data sets.

The first issue here is the division of the data into the training and test sets. Although there is no general solution to this problem, several factors such as the problem characteristics, the data type, and the size of the available data should be considered in making the decision. It is critical to have both the training and test sets representative of the population or underlying mechanism. This is of particular importance for time series forecasting problems. The literature offers little guidance in selecting the training and the test sample. Most authors select them based on the rule of 90% vs. 10%, 80% vs. 20%, 70% vs. 30%, etc..

Another closely related factor is the sample size. No definite rule exists for the requirement of the sample size for a given problem. The amount of data for the network training depends on the network structure, the training method, and the complexity of the particular problem or the amount of noise in the data on hand. In general, as in any statistical approach, the sample size is closely related to the required accuracy of the problem. The larger the sample size is, the more accurate the results will be.

Although there can be many performance measures for an ANN forecaster like the modeling time and training time, the ultimate and the most important measure of performance is the prediction accuracy it can achieve beyond the training data. However, a suitable measure of accuracy for a given problem is not universally accepted by the forecasting academicians and practitioners. An accuracy measure is often defined in terms of the forecasting error which is the difference between the actual (desired) and the predicted values. There are a number of measures of accuracy in the forecasting literature and each has advantages and limitations (Makridakis, Wheelwright, & McGee, 1983). The most frequently used are:

(1) The mean absolute deviation (MAD);

(2) The sum of squared error (SSE);

(3) The mean squared error (MSE);

(4) The root means squared error (RMSE);

(5) The mean absolute percentage error (MAPE).

SVR

SVMs are considered as an alternative to multi-layer neural networks for modeling nonlinear relationships. SVMs are based on the statistical learning theory which Vapnik (1995) pioneered and are new and powerful tools for pattern classification and regression estimation. SVM learning formulation is based on the principle of structural risk minimization. Instead of minimizing an objective function based on training data, an SVM attempts to minimize a bound on the generalization error. Researchers have widely accepted SVMs due to their high generalization ability and robust function in multiple applications, including financial time. When an SVM is applied to regression estimation problems, it is referred to as SVR. The SVR model uses the kernel technique to implicitly perform nonlinear input data mapping to a high-dimensional feature space. SVR employs the margin concept for the regression problem with the help of insensitive loss functions (Smola & Scolkopf, 1998). SVR has been found especially useful in time series predictions (Yang, Chan, & King, 2002).

The following is a brief summary of SVR as described in detail by Yang et al. (2002) and Yang (2003).

Let $\boldsymbol{x}$ be an input vector in the input space $\boldsymbol{X}$. Let y be the output. Let $S = \{(\boldsymbol{x}_1,y_1), (\boldsymbol{x}_2,y_2), \ldots, (\boldsymbol{x}_n,y_n)\}$ be the training set used for supervised prediction. Let us define the inner product of two vectors $\boldsymbol{x}$ and $\boldsymbol{w}$ as:

$$\langle \boldsymbol{x}, \boldsymbol{w} \rangle = \sum_j x_j \times w_j \tag{1}$$

where x_j and w_j are components of the vectors $\boldsymbol{x}$ and $\boldsymbol{w}$, respectively. SVR overcomes the linear restriction by using a mapping Φ of the input space to another feature space of higher dimension. The mapped vectors are then used to define a function $f(\boldsymbol{x})$ as a predicted value of y:

$$f(\boldsymbol{x}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{w}) \rangle + b \tag{2}$$

Usually, a high dimensional transformation is needed in order to obtain a reasonable prediction (Yang et al., 2002; Yang, 2003). Computational overhead can be reduced by not explicitly mapping the data to the feature space, but instead just working out the inner product in that space. In fact, SVMs use a kernel function K corresponding to the inner product in the transformed feature space as:

$$K(\boldsymbol{x}, \boldsymbol{w}) = \langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{w}) \rangle \tag{3}$$

The objective of SVR is to minimize the regression risk:

$$Rf = \frac{1}{2}\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{w}) \rangle + C\sum_i l(f(x_i), y_i) \tag{4}$$

where C is called the cost of error. The first term $Rf = \frac{1}{2}\langle \Phi(\boldsymbol{x}), \Phi(\boldsymbol{w}) \rangle$ can be seen as the margin in SVMs. The similarity between actual y and its prediction is given by the loss function $l(f(\boldsymbol{x}, y)$.

Vapnik (1995) proposed an ε-insensitive loss function:

$$l_\varepsilon(f(\boldsymbol{x}, y) = \max(0, |y - f(\boldsymbol{x})| - \varepsilon \tag{5}$$

As shown in part (a) of Figure 2, the vertical axis denotes the loss. The horizontal axis corresponds to the value of $f(\boldsymbol{x})$. The two axes meet at $f(\boldsymbol{x}) = y$. If the predicted value is within $\pm\varepsilon$ of the actual value, the prediction is considered lossless. Part (b) of Figure 2 shows how the actual values in the margin around the predicted function are considered acceptable or error-free. Increasing the ε-value reduces the number of support vectors. A large enough value of C will lead to a constant regression function. The ε-insensitive loss function is ideally suited for modeling rough values as can be seen by the broken curves around the prediction function in Part (b) of Figure 2. The corresponding SVR is called an ε-SVR. The minimization of Equation (4) is reduced to a quadratic programming problem. The details of the formulation can be found in Yang et al. (2002) and Yang (2003).

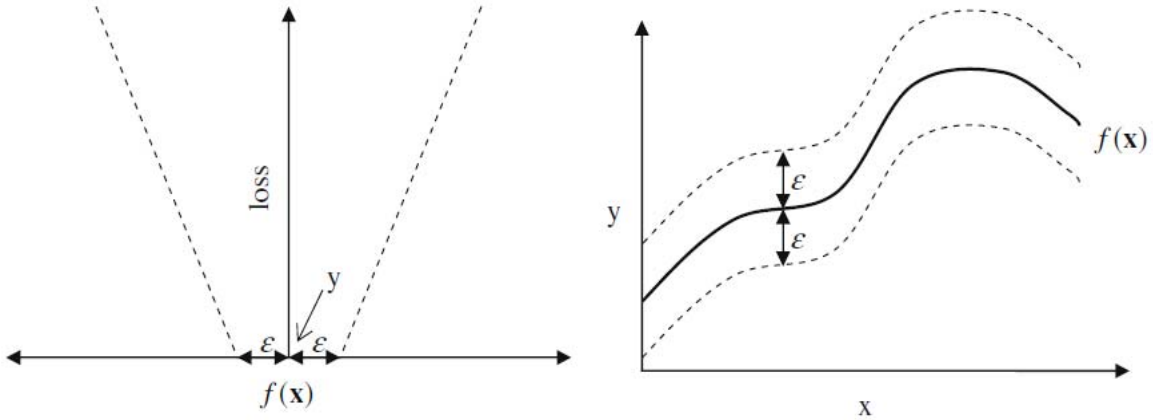

Figure 2. SVR.

Theoretical and Methodological Issues

Structure of the SLM

SLM is a 5-layer network. Each layer is associated with a particular step in the fuzzy inference process. Each node performs a particular node function on the incoming signal, which is characterized by a set of parameters. Some of the nodes are adaptive nodes, whose parameters can be tuned by learning procedure. The functions of each layer are described below. The output from node k in layer r is denoted as f_r^k.

Layer 1 (Input layer): Nodes in Layer 1 are fixed nodes. Nodes in this layer represent the inputs of the systems, which include all the n_{sv} support vectors and the currently being tested input vector. The output of node i is the ith support vector and is expressed as:

$$f_1^i = x_{svi}, \quad i = 1, ..., n_{svi} \tag{6}$$

Each node of the n_{sv} support vectors corresponds to a single fuzzy rule.

Virtual layer: A virtual layer is added to show the mapping function $\Phi(x)$, which maps the data from input space to a higher dimensional feature space. This layer is only for clarity and is not needed in actual computation.

Layer 2 (Kernel layer): Nodes in Layer 2 are adaptive nodes. Each node represents a kernel function, which calculates the dot product of the currently being tested vector and the support vectors:

$$K(x, x_{svi}) = \left\langle \Phi(x), \Phi(x_{svi}) \right\rangle, \ i = 1, ..., n_{sv}$$

The kernels or dot products are equal to the multidimensional Gaussian membership functions in the first-order Takagi-Sugeno fuzzy model (TS fuzzy model). The output of node k is defined as:

$$f_2^i = \exp(-\frac{(x_i - x_{svi})^2}{2\sigma_i^2}) \quad i = 1, ..., n_{svi} \tag{7}$$

Layer 3 (Normalization layer): Nodes in Layer 3 are fixed nodes, which normalize the outputs of Layer 2. The output from this layer is:

$$f_3^i = \overline{w}_i = \frac{w_i}{\sum_{t=1}^{n_{sv}} w_t} = \frac{K(x, x_{svi})}{\sum_{t=1}^{n_{sv}} K(x, x_{svt})}, \ i = 1, ..., n_{sv} \tag{8}$$

Layer 4 (Consequence layer): Nodes in this layer are adaptive nodes. Each node, denoted as y_i, performs the following function for the consequence part of the fuzzy IF-THEN rule in the first-order TS fuzzy model:

$$f_4^i = y_i = \sum_{j=1}^{n} b_j^i \ x_j + b_0^i, \ i = 1, ..., n_{sv} \tag{9}$$

where consequence parameters b_j^i are real values.

Layer 5 (Defuzzification layer): The single node in this layer is a fixed node, which performs the function of overall aggregation of all the fuzzy IF-THEN rules from Layer 4. The output signal of this node is:

$$f_5^i = \hat{y} = \sum_{i=1}^{n_{sv}} \overline{w}_i y_i \tag{10}$$

The Learning Algorithm

Both the network structure and the parameters are learned. The number of fuzzy rules depends on the number of support vectors, which must be obtained during learning. Thus, the learning algorithms include both the network structure identification algorithm, which comes from the SVR algorithm, and the parameter estimation algorithms, which include the least-squares algorithm for adjusting consequence parameters and the gradient descent algorithm for adjusting the Gaussian premise parameter. In addition, we also have the parameters C and ε. The overall learning algorithm for the SLM is summarized in Figure 3.

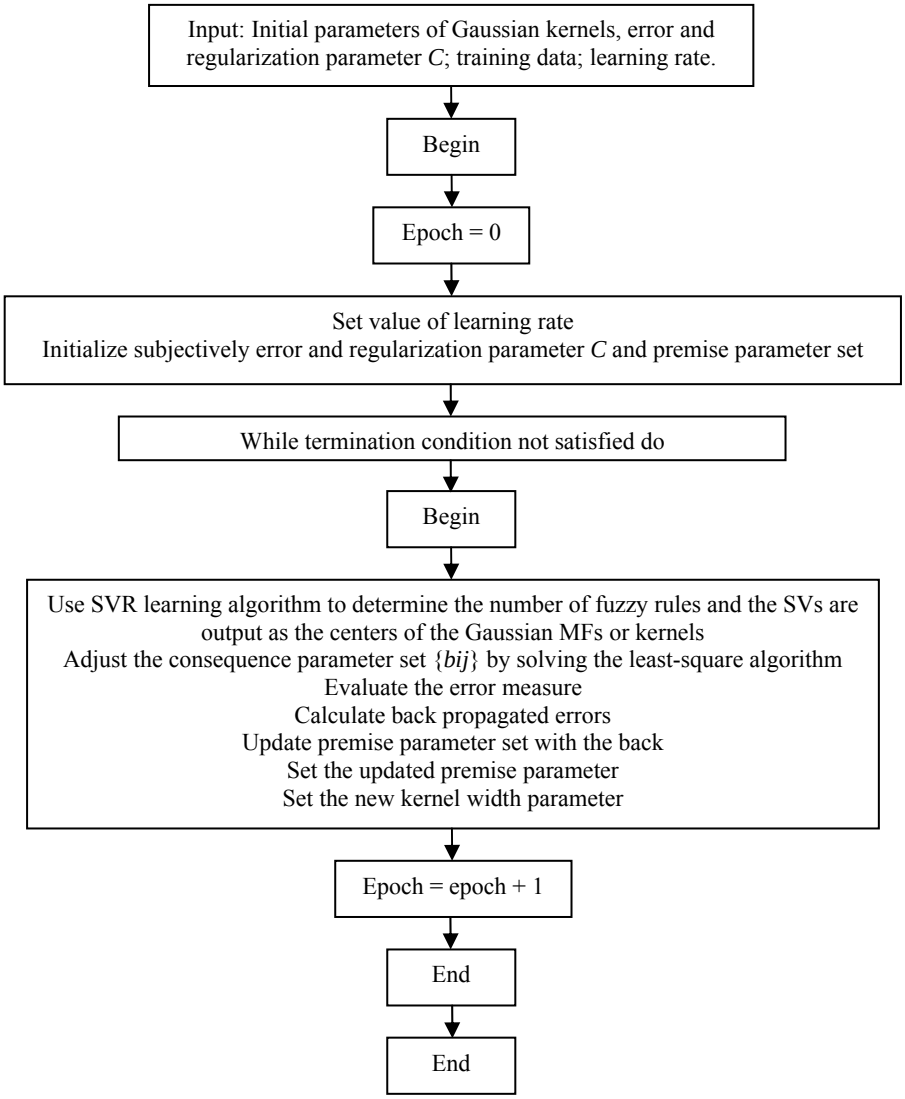

Figure 3. Learning algorithm of the SLM.

Empirical Study

Data Set

In this study, we used monthly average nominal WTI crude oil spot prices, which are considered as world benchmark crude oil prices. All the data come from the Energy Information Administration (EIA).

EIA has published monthly petroleum market inventory data for individual OECD countries since January 1988, so we have used only the total petroleum stock data in the study. In order to compare with other models easily, we limit the analysis from January 1992 to March 2013. This at choice can also ensure a consistent data series that avoids effects of the First Gulf War.

The objective is to predict the monthly average WTI using the given historical data. We divide the data into two sets: one with 70% of the source data, for training the models, the other with 30% of the source data, for testing the models.

Attribute Selection

Our proposed oil price forecasting model is based on the past monthly average WTI (historical data) as one of the candidate input variables. The best input features for our forecasting model are those which have the highest correlation with the output variable and the highest degree of linear independency. Thus, the most effective candidate inputs with minimum redundancy are selected as the model attributes.

The three attributes used in the SLM modeling process are:

The first one is the total OECD inventories between 1992 and 2013. The other two attributes were characterized by two events that significantly impacted crude oil markets. One event was OPEC quota tightening beginning in April 1999, which resulted in the deviation and mean value of the WTI price changing and another event was the September 11, 2001 terrorist attack in the US, which led to market disequilibrium.

For the input variables, the following symbols are used:

(1) Inventory(t) = the total OECD inventories stock data at time t;

(2) OCT911, NOV911, DEC911, JANV911, FEV911, MARCH911, and LAPR99 are dummy variables to account for market disequilibrium and shifting caused by the two events mentioned in the previous paragraph.

Three different input structures are tested separately for WTI price. These are numbered from 1 to 3 and are listed in Table 1.

Table 1

Input Structures

Input	Structure
1	Inventory(t), Inventory(t-1), WTI(t-1), OCT911, NOV911, DEC911, JANV911, FEV911, MARCH911, and LAPR99
2	D(Inventory(t-1)), D(Inventory(t-2)), WTI(t-1), WTI(t-2), and LAPR99
3	D(Inventory(t-1)), WTI(t-1), and LAPR99

Performance Criteria

Although the MSE is a perfectly acceptable measure of performance, in practice, the ultimate goal of any testing strategy is to confirm that the results of models are robust and capable of measuring the profitability of a system. It is important, therefore, to design a test from the outset. This is not always carried out with the level of rigor that it merits, partly because of unfamiliarity with the established methods or practical difficulties intrinsic to nonlinear systems. Consequently, we designed test sets to evaluate the effects of the models. The prediction performance is evaluated using the following statistics: MSE, Normalized Mean Squared Error (NMSE), Mean Absolute Error (MAE), and MAPE.

The performance criteria are defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(a_i - p_i)^2$$

$$NMSE = \frac{1}{\delta^2 n}\sum_{i=1}^{n}(a_i - p_i)^2; \quad \delta^2 = \frac{1}{(n-1)}\sum_{i=1}^{n}(a_i - \overline{a})^2$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|a_i - p_i\right|$$

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{a_i - p_i}{a_i}\right|$$

MSE, NMSE, MAE, and MAPE measure the correctness of a prediction in terms of levels and the deviation between the actual (a_i) and predicted values (p_i) of WTI. The smaller the values, the closer the predicted time series values will be to the actual values.

Results

A comparative study of our proposed model with other machine learning techniques was performed using the train and the test data. Our proposed SLM method was compared with two different prediction techniques: (1) SVR; and (2) Back Propagation Neural Network (BPNN). The three different input structures were tested in all cases. The input structure 3 gives the best results in all the cases. The prediction accuracy of our proposed model compared with the different prediction techniques for predicting WTI of train and test data is shown in Tables 2 and 3.

Table 2

Comparison of the Forecasting Accuracy Using Different Prediction Techniques (Training Set)

Model	MSE	NMSE	MAE	MAPE
BPNN	239.99	0.054	3.210	2.126%
SVR	95.8278	0.0216	2.605	1.394%
SLM	93.1281	0.0209	2.571	1.085%

Table 3

Comparison of the Forecasting Accuracy Using Different Prediction Techniques (Test Set)

Model	MSE	NMSE	MAE	MAPE
BPNN	98.1741	3.1333	1.3457	0.882%
SVR	72.5432	2.6934	1.1582	0.767%
SLM	70.9707	2.6640	1.1519	0.762%

Results obtained in Tables 2 and 3 indicate that the prediction accuracy of the ML-BPNN is not satisfactory. This is due to the problems of local minima and over-fitting associated with ANNs, which tends to decrease the generalization performance for unseen data. Our proposed SLM model proves to be superior in terms of all the performance criteria compared to the SVR and ML-BPNN models. This is due to the presence of the fuzzy component in our model, which fits the clustered training data into the appropriate SVRs based on the Euclidean distance.

Summary and Conclusions

In this paper, the SVR, FS, and ANN are combined by considering two important aspects, the use of support vectors to reduce the structure or the complexity of the network and hence the model, and the use of the ε-insensitive and the regularization parameter C to balance between model complexity and the available data. The combination is accomplished owing to the fact that all the semi-positive definite FSs based on membership functions, which can be used as Mercer kernels, are functional equivalents to SVMs. The two systems are combined in such a way that the resulting SLM system possesses some advantages, such as fuzzy linguistic representation ability, learning ability, and generalization ability of the original approaches.

The proposed SLM technique is applied to the WTI crude oil spot price, which demonstrates the effectiveness and efficiency of the prediction technique in contrast to others.

Results obtained indicated that the proposed SLM model outperforms the other two approaches in terms of all evaluation criteria used in this research. This can be explained by the formulation of the SLM, SVM, and BPNN networks. SLM and SVM methods use a quadratic programming problem which is convex and has a global optimum solution. BPNN networks use the back propagation algorithm to minimize the network error, the problem is non-convex, and it is hard to find the global optimum.

The main problem in using the proposed system is the determination of the numerical values a priori of the parameters such as error, the band, the upper bond C, and the kernel parameter. Further researches are needed to devise better approaches to select more appropriate parameter values.

References

Abramson, B., & Finizza, A. (1991). Using belief networks to forecast oil prices. *International Journal of Forecasting, 7*(3), 299-315.

Abramson, B., & Finizza, A. (1995). Probabilistic forecasts from probabilistic models: A case study in the oil market. *International Journal of Forecasting, 11*(1), 63-72.

Alvarez-Ramirez, J., Soriano, A., Cisneros, M., & Suarez, R. (2003). Symmetry/anti-symmetry phase transitions in crude oil markets. *Physica A: Statistical Mechanics and Its Applications, 322*, 583-596.

Azoff, E. M. (1994). *Neural network time series forecasting of financial markets*. Chichester: John Wiley & Sons.

Barone-Adesi, G., Bourgoin, F., & Giannopoulos, K. (1986). Don't look back. *Risk, 11*, 100-103.

Chokri, S. (2004). Forecasting the volatility of stock index returns: A stochastic neural network approach. In *Lecture notes in computer science* (Vol. 3045, pp. 935-944). Berlin: Springer-Verlag.

Chokri, S. (2007). Fuzzy neural model for bankruptcy prediction. *The Journal of Business Review Cambridge, 8*(2), 117-122.

Chokri, S. (2009). Hybrid approach in neural network design applied to financial time series forecasting. *The Journal of American Academy of Business, Cambridge, 15*(1), 294-300.

Chokri, S., & Abdelwahed, T. (2003). Neural network for modeling nonlinear time series: A new approach. In *Lecture notes in computer science* (Vol. 2659, pp. 159-168). Berlin: Springer-Verlag.

Cottrell, M., Girard, B., Girard, Y., Mangeas, M., & Muller, C. (1995). Neural modeling for time series: A statistical stepwise method for weight elimination. *IEEE Transactions on Neural Networks, 6*(6), 1355-1364.

Fletcher, R. (1987). *Practical methods of optimization* (2nd ed.). Chichester: John Wiley & Sons.

Gately, E. (1996). *Neural networks for financial forecasting*. New York, NY: John Wiley & Sons.

Gulen, S. G. (1998). Efficiency in the crude oil futures market. *Journal of Energy Finance and Development, 3*(1), 13-21.

Huntington, H. G. (1994). Oil price forecasting in the 1980s: What went wrong? *The Energy Journal, 15*(2), 1-22.

Kaboudan, M. A. (2001). Compumetric forecasting of crude oil prices. Proceedings of *IEEE Congress on Evolutionary Computation* (pp. 283-287).

Karnin, E. D. (1990). A simple procedure for pruning back-propagation trained neural networks. *IEEE Transactions on Neural Networks, 1*(2), 239-242.

Lanza, A., Manera, M., & Giovannini, M. (2005). Modeling and forecasting cointegrated relationships among heavy oil and product prices. *Energy Economics, 27*(6), 831-848.

Makridakis, S., Wheelwright, S. C., & McGee, V. E. (1983). *Forecasting: Methods and applications* (2nd ed.). New York, NY: John Wiley & Sons.

Mirmirani, S., & Li, H. C. (2004). A comparison of VAR and neural networks with genetic algorithm in forecasting price of oil. *Advances in Econometrics, 19*, 203-223.

Morana, C. A. (2001). A semiparametric approach to short-term oil price forecasting. *Energy Economics, 23*(3), 325-338.

Murata, N., Yoshizawa, S., & Amari, S. (1994). Network information criterion—Determining the number of hidden units for an artificial neural network model. *IEEE Transactions on Neural Networks, 5*(6), 865-872.

Reed, R. (1993). Pruning algorithms—A survey. *IEEE Transactions on Neural Networks, 4*(5), 740-747.

Refenes, A. N. (1995). *Neural networks in the capital markets*. Chichester: John Wiley & Sons.

Roy, A., Kim, L. S., & Mukhopadhyay, S. (1993). A polynomial time algorithm for the construction and training of a class of multilayer perceptrons. *Neural Networks, 6*(4), 535-545.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6188), 533-536.

Sietsma, J., & Dow, R. (1988). Neural net pruning—Why and how? Proceedings of *the IEEE International Conference on Neural Networks* (Vol. 1, pp. 325-333).

Smola, A., & Scolkopf, B. (1998). *A tutorial on support vector regression*. NeuroCOLT Technical Report Series NC2-TR-1998-030.

Tang, L., & Hammoudeh, S. (2002). An empirical exploration of the world oil price under the target zone model. *Energy Economics, 24*(7), 577-596.

Trippi, R. R., & Turban, E. (1993). *Neural networks in finance and investment: Using artificial intelligence to improve real-world performance*. Chicao: Probus.

Vapnik, V. (1995). *The nature of statistical learning theory*. NY, USA: Springer.

Verleger, P. K. (1993). *Adjusting to volatile energy prices*. Washington DC, USA: Institute for International Economics.

Wang, Z., Massimo, C. D., Tham, M. T., & Morris, A. J. (1994). A procedure for determining the topology of multilayer feedforward neural networks. *Neural Networks, 7*(2), 291-300.

Weigend, A. S., & Gershenfeld, N. A. (1994). *Time series prediction: Forecasting the future and understanding the past*. Reading, MA: Addison-Wesley.

Weigend, A. S., Huberman, B. A., & Rumelhart, D. E. (1992). Predicting sunspots and exchange rates with connectionist networks. In M. Casdagli & S. Eubank (Eds.), *Nonlinear modeling and forecasting* (pp. 395-432). Redwood City, CA: Addison-Wesley.

Weigend, A. S., Rumelhart, D. E., & Huberman, B. A. (1991). Generalization by weight-elimination with application to forecasting. *Advances in Neural Information Processing Systems, 3*, 875-882.

Wong, B. K., Bodnovich, T. A., & Selvi, Y. (1995). A bibliography of neural networks business applications research. *Expert Systems, 12*(3), 253-261.

Xi, X. C., Poo, A. N., & Choo, S. K. (2007). Support vector regression model predictive control on a HVAC plant. *Control Engineering Practice, 15*(8), 897-908.

Yang, H. (2003). Margin variations in support vector regression for the stock market prediction (M. Phil. dissertation, Department of Computer Science and Engineering, The Chinese University of Hong Kong).

Yang, H., Chan, L., & King, I. (2002). Support vector machine regression for volatile stock market prediction. In H. Yin, N. Allinson, R. Freeman, J. Keane, & S. Hubbard (Eds.), *Intelligent data engineering and automated learning*. *Lecture notes in computer science* (Vol. 2412, pp. 391-396). Berlin: Springer-Verlag.

Ye, M., Zyren, J., & Shore, J. (2006). Forecasting short-run crude oil price using high- and low-inventory variables. *Energy Policy, 34*(17), 2736-2743.