Research Article

Forecasting Energy Commodity Prices Using Neural Networks

Massimo Panella,[1] Francesco Barcellona,[2] and Rita L. D'Ecclesia[2]

[1] *Department of Information Engineering, Electronics and Telecommunications (DIET),*
 Sapienza University of Rome, Via Eudossiana 18, 00184 Rome, Italy
[2] *Department of Social Science (DISS), Sapienza University of Rome, P.le Aldo Moro 5,*
 00185 Rome, Italy

Correspondence should be addressed to Massimo Panella, massimo.panella@uniroma1.it

A new machine learning approach for price modeling is proposed. The use of neural networks as an advanced signal processing tool may be successfully used to model and forecast energy commodity prices, such as crude oil, coal, natural gas, and electricity prices. Energy commodities have shown explosive growth in the last decade. They have become a new asset class used also for investment purposes. This creates a huge demand for better modeling as what occurred in the stock markets in the 1970s. Their price behavior presents unique features causing complex dynamics whose prediction is regarded as a challenging task. The use of a Mixture of Gaussian neural network may provide significant improvements with respect to other well-known models. We propose a computationally efficient learning of this neural network using the maximum likelihood estimation approach to calibrate the parameters. The optimal model is identified using a hierarchical constructive procedure that progressively increases the model complexity. Extensive computer simulations validate the proposed approach and provide an accurate description of commodities prices dynamics.

1. Introduction

Energy is a principal factor of production in all aspects of every economy. Energy price dynamics are affected by complex risk factors, such as political events, extreme weather conditions, and financial market behavior. Crude oil is a key and highly transportable component for the economic development and growth of industrialized and developing countries, where it is refined into the many petroleum products we consume. Over the last 10 years, the global demand for crude oil and gas has increased largely due to the rapidly increasing demands of non-OECD countries, especially China [1]. Local gas and coal are mainly used in the

electricity generation process and recently their supply and demand experienced a profound transformation. The economic exploitation at higher prices of non-conventional forms of oil and gas, such as shale gas and shale oil, is modifying the demand for the three fossil fuels. The production of shale gas in the US will shortly bring the US to be less dependent on imported oil and, in addition, it means a large part of the electricity generation process has been switched from coal to gas.

The deregulation of gas and electricity markets makes the prices of these commodities to be formed in competitive markets. Crude oil and natural gas in the last decade have been largely traded on spot, derivative, and forward markets by producers, consumers, and investors. Crude oil and gas are largely traded on over-the-counter (OTC) derivative markets making them a readily usable tool for both investor flows as well as hedging and risk management purposes. Finally the role of geopolitical aspects represents an additional source of price volatility mainly for the crude oil as the Middle East is still the major exporter. If we look at the recent price dynamics of crude oil prices and natural gas, an exceptionally high oil price volatility has been observed since the beginning of the 2008 financial crisis and liquidity problems. Oil prices skyrocketed to almost 150 USD/bbl in July 2008 before retreating to 60 USD/bbl in the subsequent four months. Since then, prices have continued to be extremely volatile and in the first quarter of 2011, we saw year-on-year gains of some fifty per cent. This high volatility of oil prices exacerbated uncertainty in other energy commodity prices, leading to increased investor flows from a wide variety of sources, both traditional and new.

It is critical to be able to forecast the price direction of these commodities in order to try to reduce the negative impact of high price fluctuations on investment results and on risk management strategies. Energy derivatives represent the main tool to manage risk so derivative pricing is affected by an accurate estimation of the underlying spot price. Commodity prices forecasting on a daily basis cannot be easily obtained using standard structural models, given the lack of daily data on supply and demand, normally available monthly and a quarter in arrears. Reduced form models are commonly used to price energy commodities; that is, two state variable stochastic models provide an accurate description of oil and gas price dynamics [2] allowing to account for different sources of randomness, while Markov regime switching models seem to work better for electricity prices [3–5]. A review of the main features of the energy price models can be found in [6]. In most cases, the implementation of numerical procedures has to be set to solve complex stochastic differential equations. Neural networks have been successfully applied to describe stock market dynamics and their volatilities in [7–9]. Recently, they have also been applied to provide short-term forecasts of oil and electricity prices [10–17].

Neural networks can be used as nonlinear regression models, generalizing the stationary and univariate models used in econometrics, they provide an effective tool to capture the main features of price returns, that is, fat tails, volatility clustering or persistence, and leverage effects [18–25]. Some applications focus on the principal processes generating the observed time series and make use of neural networks as nonlinear models that are more suited to identify the chaotic behavior of specific commodity prices with respect to common Autoregressive Integrated Moving Average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models. On the other hand, rule-based neurofuzzy systems based on the integration of neural networks and high-level linguistic information, for instance extracted by a Web mining process, have been proposed.

Actually neural networks enable a "black-box" approach that is intended to exploit the noisy information contained in the input data and to avoid some critical assumptions often

necessary to setup the analytical models widely used so far. For example, using the historical prices is useful (as most of the studies have done); however, it is by no means enough. In the case of electricity prices, neural networks have been applied to provide a short-term forecast of the System Marginal Price. Szkuta used the three layered Multilayer Perceptron (MLP) model with backpropagation [10]. The training and testing was performed on deregulated Victorian prices given that in 1997 the market turnover was very large. In addition short-term forecasts drive the analysts decisions and the reduction/increase of generation or demand.

In this paper, we propose an alternative approach to describe the dynamics of energy commodity prices based on machine learning and signal processing, adopting neural network techniques already proven successful in a wide range of forecasting problems [26–31]. More precisely, we apply the Mixture of Gaussian (MoG) neural networks to estimate, to a given degree of approximation, any probability density. This is possible given that MoG use the Gaussian Mixture Models (GMM), Probabilistic Principal Component Analysis (PPCA), and latent variables for the singular value decomposition (SVD) of signal subspaces [32–38], within a general data-driven estimation of probability density functions for clustering, classification, regression, and the related problems [39–46]. The MoG paradigm adopts a learning process, as for any other neural network model. For instance, in the case of regression, the model parameters are obtained using a training set of input-output samples of the function to be approximated [37, 47–49]. To obtain a good generalization capability of the resulting neural network, a good approximation is mandatory on the samples never involved in the training process. A suitable number of mixture components has to be defined; however, the determination of the optimal number is a very critical problem to be solved, especially when dealing with risk sensitive applications such as medical or financial analysis, since the neural network might be easily overfitted in the case of noisy or ill-conditioned data. For this purpose, a hierarchical constructive procedure for the automatic determination of the mixture components is used; it regularizes the network architecture by progressively increasing the model complexity and controlling the computational cost [50, 51].

In our knowledge, neural network techniques have been applied only to forecast crude oil and electricity prices. In this paper, we want to study the dynamics of the energy price complex, which has shown large unexpected volatility in the last decade. In order to understand the whole picture the entire complex should be studied. In this context, a powerful tool providing accurate price forecasting is needed. Natural gas, coal, and electricity prices, unlike crude oil, present seasonality features that are usually measured using deterministic techniques. In this paper, we aim to forecast short-term price dynamics in order to be able to adequately measure the existing correlations between the various commodities. To this extent the seasonality component of the gas and coal prices will not affect the results. We apply the MoG approach to forecast crude oil, natural gas, electricity, and coal prices using data for both the European and the US market. The proposed system is trained using daily data collected for the last decade.

The paper is organized as follows. In Section 2, the general framework for time series modeling and forecasting is briefly summarized, while in Section 3, the use of neural networks is proposed as nonlinear regression models suited in this regard. In Section 4, the application of MoG neural networks is illustrated and a hierarchical constructive procedure to train MoG networks is proposed. Extensive computer simulations prove the validity of the proposed approach and provide accurate forecasts of the chosen time series. In Section 5, the numerical results obtained on both reference benchmarks and actual time series are reported and, finally, we draw some conclusions in Section 6.

2. Basic Concepts for Time Series Modeling and Prediction

Prices related to financial assets are commonly collected as time series uniformly sampled at hourly, daily or monthly rate. Let S_t be a generic time series of prices, where t denotes the time index. In order to work with stable data sets for modeling, the financial analysis is usually carried out on the return series y_t defined as

$$y_t = \ln \frac{S_t}{S_{t-1}}. \tag{2.1}$$

Given the randomness of prices, a return series is conveniently modeled as a discrete time stochastic process, for which we wish to know the conditional probability density function (pdf) denoted as $f_{y_t|\mathcal{I}_{t-1}}(y_t \mid \mathcal{I}_{t-1}; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the vector of pdf parameters and $\mathcal{I}_{t-1}$ is the conditioning set associated with all the available information available *prior to* time t (i.e., past observations and estimated models). Although any pdf depends in general on a set of numerical parameters, to simplify the notation, we will explicit in the following this dependence only when necessary.

Almost all the prediction techniques aim to estimate the conditional moments of this pdf, which imply explicit dependence on past observations. In fact, the unconditional moments are relevant to the unconditional distribution $f_{y_t}(y_t)$ of the return process and they represent the long-term behavior of the time series, assuming no explicit knowledge of the past. We will assume in the following that all the necessary estimations can be realized in a time smaller than the interval between two consecutive observations, in such a way, we can limit our analysis to a "one-step-ahead" prediction, for which the information $\mathcal{I}_{t-1}$ is available for the analysis prior to time t. Otherwise, the prediction should start earlier, by using the information $\mathcal{I}_{t-s}$, $s > 1$, to perform a prediction at distance s of the sample at time t.

We consider in this work the reference background of econometrics, where an additive model is used for time series:

$$y_t = \mu_t + \epsilon_t, \tag{2.2}$$

where μ_t is a deterministic component, representing the forecast, and ϵ_t is a random variable, which takes into account the uncertainty of prediction. In fact, $\epsilon_t = y_t - \mu_t$ can be considered as the forecast error, or innovation, and it is in itself a random process. Another hypothesis is that $f_{y_t|\mathcal{I}_{t-1}}(y_t \mid \mathcal{I}_{t-1})$ is a normal distribution

$$f_{y_t|\mathcal{I}_{t-1}}(y_t \mid \mathcal{I}_{t-1}; \boldsymbol{\theta}) = \mathcal{N}\left(y_t; \mu_t, \sigma_t^2\right), \tag{2.3}$$

where the parameters $\boldsymbol{\theta} = \{\mu_t, \sigma_t^2\}$ are determined on the basis of the conditioning information $\mathcal{I}_{t-1}$, being μ_t and σ_t^2 mean and variance of the univariate normal distribution. Thus, the conditional mean of y_t is $\mathbb{E}_{y_t|\mathcal{I}_{t-1}}[y_t] = \mu_t$ and the innovation process has a zero-mean normal conditional distribution for which $\mathbb{E}_{\epsilon_t|\mathcal{I}_{t-1}}[\epsilon_t] = 0$ and $\text{Var}_{\epsilon_t|\mathcal{I}_{t-1}}(\epsilon_t) = \mathbb{E}_{\epsilon_t|\mathcal{I}_{t-1}}[\epsilon_t^2] = \text{Var}_{y_t|\mathcal{I}_{t-1}}(y_t) = \sigma_t^2$.

The conditional variance σ_t^2 can be considered as the volatility of the return series at time t and its accurate estimation, with respect to the implied volatility of the actual process, is mandatory to any financial tool. In general, σ_t^2 changes over time and this phenomenon is called heteroscedasticity. So, many models tend to capture this behavior as, in particular,

the GARCH model considered in this paper, which is widely used in both theoretical and practical applications. A GARCH(P, Q) model is based on the following estimation of σ_t^2 by using past observations [52]:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{Q} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{P} \beta_j \sigma_{t-j}^2, \quad P \geq 0, \ Q > 0, \ \alpha_0 > 0, \ \alpha_i \geq 0,$$

$$\beta_j \geq 0, \ i = 1 \cdots Q, \ j = 1 \cdots P. \tag{2.4}$$

A GARCH(P, Q) model can be considered as an ARMA process in ϵ_t^2 and it is generally followed by the "stationary condition":

$$\sum_{i=1}^{Q} \alpha_i + \sum_{j=1}^{P} \beta_j < 1, \tag{2.5}$$

by which some assumptions on the unconditional innovation process ϵ_t are also possible. Namely, (2.5) is a necessary and sufficient condition for which ϵ_t is a wide-sense stationary (WSS) process with $\mathbb{E}_{\epsilon_t}[\epsilon_t] = 0$, $\mathrm{Cov}_{\epsilon_k, \epsilon_s}(\epsilon_k, \epsilon_s) = \mathbb{E}_{\epsilon_k, \epsilon_s}[\epsilon_k, \epsilon_s] = 0$ for any $k \neq s$, and the unconditional variance is finite and converging to

$$\mathrm{Var}_{\epsilon_t}(\epsilon_t) = \mathbb{E}_{\epsilon_t}\left[\epsilon_t^2\right] = \frac{\alpha_0}{1 - \sum_{i=1}^{Q} \alpha_i - \sum_{j=1}^{P} \beta_j}. \tag{2.6}$$

This is a common assumption when modeling financial time series, where the forecast errors are zero-mean random disturbances that are serially uncorrelated from one period to the next although not independent, evidently.

A GARCH(P, Q) model is a generalization of the early ARCH model proposed by Engle in [53] and hence, an ARCH(Q) model coincides with a GARCH$(0, Q)$ model. However, specifying the order P, Q of a GARCH model is not easy and it is still an open problem. Consequently, only low orders are usually adopted in most applications. Nevertheless, several extensions of the original GARCH model have been proposed in the past, by specifying different parameterizations to capture serial dependence on the conditional variance. For instance, some of them are integrated GARCH (IGARCH), exponential GARCH (EGARCH), threshold GARCH (TGARCH or GJR), GARCH-in-mean (GARCH-M), and so forth. All of them are able to observe some common characteristics of returns series related to energy commodity prices, in particular volatility clustering, leverage effect, and heavier/fat tails, although they remain weak in capturing wild market fluctuations and unanticipated events.

It is well known that volatility is subject to clustering: large shocks, that is, prediction errors increase the volatility and hence large returns in the next steps are more likely. Volatility clustering is a type of heteroscedasticity accounting for some of the excess kurtosis typically observed in financial data. However, the excess kurtosis can also result from non-normal pdf that happen to have fat tails. In this regard, there are possible other choices for the conditional pdf $f_{y_t | \mathcal{I}_{t-1}}$, the most popular is the Student's t-distribution introduced by Bollerslev in [54]. Certain classes of asymmetric GARCH models can also capture the leverage effect, which results in observed returns negatively correlated with changes in volatility. Namely, volatility rises when returns are lower than expected and vice versa.

The time series model should be completed by a suited hypothesis of the conditional mean μ_t as well. A general choice can be based on the linear regression model

$$\mu_t = a_0 + \sum_{i=1}^{R} a_i y_{t-i} + \sum_{j=1}^{M} b_j \epsilon_{t-j}. \tag{2.7}$$

Bearing in mind (2.2), y_t follows a general ARMA process where usual conditions are given on the eigenvalues associated with the characteristic AR and MA polynomials, in order to ensure stationarity and invertibility, respectively. Generally speaking, ARMA processes make the assumption of white Gaussian noise for ϵ_t; however, if an ARMA(R, M) process is coupled with a WSS GARCH(P, Q) process satisfying (2.5), then ϵ_t meets the condition of weak white noise only. By the way, the condition of strong white noise for ϵ_t is obtained in the particular case of a GARCH$(0, 0)$ or ARCH(0) process, by which σ_t^2 is constant over time and hence ϵ_t would be an independent and identically distributed (i.i.d.) Gaussian process. The ARMA model gets in this case a classical homoskedastic process. Energy commodity returns are typically modeled as WSS processes, with constant unconditional mean and constant unconditional variance but nonconstant conditional variance. In the following, we will consider a default association of an ARMA(R, M) process coupled with a GARCH(P, Q) WSS process, denoted as ARMA(R, M)-GARCH(P, Q), for which the unconditional variance $\mathrm{Var}_{y_t}(y_t)$ is a computable function of $\mathrm{Var}_{\epsilon_t}(\epsilon_t)$ obtained in (2.6).

Some generalizations of the ARMA model are also possible for modeling the conditional mean μ_t; for example, the ARIMA model, which is stationarized by adding lags of the differenced series and/or lags of the forecast errors, and ARMA with eXogenous inputs (ARMAX), where some "exogenous" or independent variables are added as an explanatory regression data. There are also nonlinear variants to such models as the nonlinear ARMA (NARMA) and nonlinear ARMAX (NARMAX). Finally, we outline that the previous ones are all univariate models. A multivariate generalization is possible by using Vector Autoregression (VAR) models, which are intended to capture the evolution and the interdependencies between multiple time series. As evidenced in the following, the extension to the multivariate analysis of the proposed GMM approach is straightforward.

3. Generalized Regression Approach by Neural Networks

The general approach to time series modeling and prediction described in the previous section evidences how both conditional mean μ_t and volatility σ_t^2 can be estimated through a suited regression problem, which can be compactly defined by the following equation for μ_t:

$$\mu_t = h_\mu\left(\mathbf{x}_t^{(\mu)}; \boldsymbol{\omega}_t^{(\mu)}\right),$$

$$\mathbf{x}_t^{(\mu)} = \begin{bmatrix} y_{t-1} & y_{t-2} & \cdots & y_{t-R} & \epsilon_{t-1} & \epsilon_{t-2} & \cdots & \epsilon_{t-M} \end{bmatrix}, \tag{3.1}$$

and by the following equation for σ_t^2:

$$\sigma_t^2 = h_\sigma\left(\mathbf{x}_t^{(\sigma)}; \boldsymbol{\omega}_t^{(\sigma)}\right),$$

$$\mathbf{x}_t^{(\sigma)} = \begin{bmatrix} \sigma_{t-1}^2 & \sigma_{t-2}^2 & \cdots & \sigma_{t-P}^2 & \epsilon_{t-1}^2 & \epsilon_{t-2}^2 & \cdots & \epsilon_{t-Q}^2 \end{bmatrix}, \tag{3.2}$$

where the orders R, M, P, and Q are fixed in advance (they are analogous to ARMA and GARCH models); $\boldsymbol{\omega}_t^{(\mu)}$ and $\boldsymbol{\omega}_t^{(\sigma)}$ are the parameter vectors of the regression functions h_μ and h_σ, respectively, which obviously change over time.

We propose in this paper a new approach to solve the regression problem, and it is characterized by three main features as follows.

  (i) A pure data-driven modeling is adopted, which is suited for neural network learning.

  (ii) The parameters of regression functions are determined simultaneously, since (3.1) and (3.2) are coupled through the values of ϵ_t determined in the previous time.

  (iii) Nonlinear functions are used for h_μ and h_σ, in particular by means of neural network models.

By following this approach, we are intended to the direct modeling of the time series dynamics and of its volatility, as actually pursued by GARCH models. However, both linear and nonlinear ARMA-GARCH models are global regression methods that do not involve a parametric basis function expansion of the regression models, similarly to spline functions or the MLP neural network. For this reason, they can be affected by the "curse of dimensionality" problem, since their performances dramatically decrease with the increment of the model order because of the increasing sparseness of data in the input space.

We propose the use of clustering for partitioning the data space, so that clusters of points in significant regions of the data space can be linked directly to the basis functions of a nonlinear regression expansion. All the used neural models allow a form of clustering for nonlinear basis function expansion, this is particularly true for the MoG neural networks based on GMM illustrated in Section 4. Unlike classical neural networks applied to financial time series prediction (see, e.g., [55, 56]), MoG neural networks are very suited to clustering for basis expansion and, more in general, for time series particularly noisy or leading to a nonconvex or multivalued regression problems. Classical neural networks, being standard regression models, may fail in giving an accurate description of the observed data and providing their statistical distribution features. MoG networks, instead, succeed in modeling the conditional distribution of the return series, in line to what has been previously proposed in [57, 58].

From a practical point of view, the problem to be solved is the prediction prior to time t of the sample y_t of the return series and the prediction of the related volatility. In the paper, these problems are being also referred to, indifferently, as the estimation of the conditional mean μ_t and the conditional variance σ_t^2, respectively. The main data base for the financial analysis is the collection in the past of the price series S_k, $k = 0 \cdots (t - 1)$, where price S_0 is the first sample ever collected in the past. By using (2.1), the data base of returns y_k, $k = 1 \cdots (t - 1)$, is obtained. Usually, the analysis at any time t makes use of a training set $\mathfrak{D}_{t-1} \subseteq \mathcal{O}_{t-1}$, which consists of a limited number N_T of previous observations. As explained in next section, $\mathfrak{D}_{t-1}$ is determined by means of previous models and predictions, using the samples y_k, ϵ_k, σ_k^2, $k = (t - N_T) \cdots (t - 1)$.

A prediction process is usually applied for a given number N_S of time intervals starting at time $t = T_S$, that is, for $t = T_S \cdots (T_S + N_S - 1)$ and $1 \leq N_T \leq T_S - 1$, where N_S also represents the horizon of prediction. Two alternatives are possible in this regard: the prediction models are estimated only once, prior to time $t = T_S$; the prediction models are estimated at every time step, by changing consequently the training set at any time. We will consider in the following the second approach, since the former is a generalization of a

multi-step-ahead prediction for which suited techniques have been proposed in the literature [59, 60].

The proposed algorithm for the data-driven regression and forecasting can be summarized in the iteration of the following steps.

Initialization

Let $t = T_S$ and find the initial conditions for ϵ_k and σ_k^2, $k = (T_S - N_T) \cdots (T_S - 1)$. These values can be inferred by using any adequate technique. We used in this regard an ARMA-GARCH model applied to the samples y_k from $T_S - N_T$ to $T_S - 1$.

Step 1. With the current value of t, determine the training set to be used for the model learning. It consists of two matrices $\mathbf{D}_t^{(\mu)}$ and $\mathbf{D}_t^{(\sigma)}$, where

(i) $\mathbf{D}_t^{(\mu)}$ is a $N_T \times (R + M + 1)$ matrix whose the ith row $\mathbf{d}_{t,i}^{(\mu)}$, $i = 1 \cdots N_T$, is

$$\mathbf{d}_{t,i}^{(\mu)} = \begin{bmatrix} \mathbf{x}_{t-N_T+i-1}^{(\mu)} & y_{t-N_T+i-1} \end{bmatrix}; \tag{3.3}$$

(ii) $\mathbf{D}_t^{(\sigma)}$ is a $N_T \times (P + Q + 1)$ matrix whose the ith row $\mathbf{d}_{t,i}^{(\sigma)}$, $i = 1 \cdots N_T$, is

$$\mathbf{d}_{t,i}^{(\sigma)} = \begin{bmatrix} \mathbf{x}_{t-N_T+i-1}^{(\sigma)} & \sigma_{t-N_T+i-1}^2 \end{bmatrix}. \tag{3.4}$$

Each row of these matrices is an input-output pattern that can be used for learning. In fact, the first $M + N$ columns of $\mathbf{D}_t^{(\mu)}$ and the first $P + Q$ columns of $\mathbf{D}_t^{(\sigma)}$ represent the inputs to h_μ and h_σ, respectively, for every sample of the training set. The last column of both matrices is the expected value to be estimated in correspondence with every pattern. The last row of matrices holds the most recent observation.

Step 2. Determine, at the current time t, the parameters $\boldsymbol{\omega}_t^{(\mu)}$ of the regression function h_μ by using the training matrix $\mathbf{D}_t^{(\mu)}$ and an appropriate learning algorithm according to the chosen regression model. Similarly, learn the parameters $\boldsymbol{\omega}_t^{(\sigma)}$ of h_σ by using $\mathbf{D}_t^{(\sigma)}$. For example, if an ARMA-GARCH model is used, the parameters can be estimated by maximum Gaussian likelihood [52]. The specific procedure for MoG neural networks is illustrated in the next section.

Step 3. By means of the parameters $\boldsymbol{\omega}_t^{(\mu)}$ and $\boldsymbol{\omega}_t^{(\sigma)}$ determined in the previous Step 2, apply (3.1) and (3.2) to forecast the conditional mean μ_t and the volatility σ_t^2, respectively. Then, let $\epsilon_t = y_t - \mu_t$, $t \leftarrow t + 1$, and go back to Step 1 if $t \leq T_S$.

Once the iteration is stopped, we have N_S samples of conditional mean (forecast), innovation and volatility pertaining to the time interval where prediction is carried out. The performance of prediction can be evaluated by means of suited benchmarks and error measures applied to the obtained results. A useful collection of such measures will be illustrated in Section 5.

4. Mixture of Gaussian Neural Network Applied to Time Series Analysis

We introduce in the following the architecture of the MoG neural network to be used for regression in (3.1) and (3.2). It is based on a GMM that is suited to approximate any pdf as well as any generic function. The model parameters are estimated through a maximum likelihood approach and a constructive procedure is adopted in order to find a suitable number of Gaussian components.

4.1. The GMM Model for Regression

Let us consider the estimation of the conditional mean first; we should determine the parameters $\boldsymbol{\omega}_t^{(\mu)}$ of the regression function $h_\mu : \Re^{R+M} \to \Re$. To simplify the notation, let $\mathbf{x} \in \Re^{R+M}$ be a generic column vector in the input space of h_μ and let $\mathbf{y} \in \Re$ be a generic column vector in the output space of h_μ. Although the output is a scalar, it is convenient to keep the vector notation also for $\mathbf{y}$. In fact, the regression approach adopted by MoG neural networks can be immediately applied to a multivariate time series analysis and all the successive considerations are independent of the dimension of the input and output spaces where MoG regression is applied.

The peculiarity of the MoG approach is the estimation of the joint pdf of data $f_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y})$, with no distinction between input and output variables. The joint density is successively conditioned, so that the resulting $f_{\mathbf{y}|\mathbf{x}}(\mathbf{y} \mid \mathbf{x})$ can be used for approximating any function [61, 62]. The joint pdf $f_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y})$ is based on a GMM of C multivariate Gaussian components in the joint input-output space:

$$f_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{C} \pi^{(j)} \mathcal{N}\left(\mathbf{z}; \mathbf{m}^{(j)}, \mathbf{K}^{(j)}\right), \tag{4.1}$$

where $\mathbf{m}^{(j)}$ and $\mathbf{K}^{(j)}$ are, respectively, mean and covariance matrix of the jth multivariate normal distribution, $j = 1 \cdots C$, $\pi^{(j)}$ is the prior probability of the component, and

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}; \qquad \mathbf{m}^{(j)} = \begin{bmatrix} \mathbf{m}_{\mathbf{x}}^{(j)} \\ \mathbf{m}_{\mathbf{y}}^{(j)} \end{bmatrix}; \qquad \mathbf{K}^{(j)} = \begin{bmatrix} \mathbf{K}_{\mathbf{xx}}^{(j)} & \mathbf{K}_{\mathbf{xy}}^{(j)} \\ \mathbf{K}_{\mathbf{yx}}^{(j)} & \mathbf{K}_{\mathbf{yy}}^{(j)} \end{bmatrix}. \tag{4.2}$$

The conditional pdf $f_{\mathbf{y}|\mathbf{x}}(\mathbf{y} \mid \mathbf{x})$ can be readily determined from (4.1), that is

$$f_{\mathbf{y}|\mathbf{x}}(\mathbf{y} \mid \mathbf{x}) = \frac{f_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y})}{f_{\mathbf{x}}(\mathbf{x})} = \sum_{j=1}^{C} g_j(\mathbf{x}) \mathcal{N}\left(\mathbf{y}; \mathbf{m}_{\mathbf{y}|\mathbf{x}}^{(j)}, \mathbf{K}_{\mathbf{y}|\mathbf{x}}^{(j)}\right), \tag{4.3}$$

where

$$\mathbf{m}_{\mathbf{y}|\mathbf{x}}^{(j)} = \mathbf{m}_{\mathbf{y}}^{(j)} + \mathbf{K}_{\mathbf{yx}}^{(j)} \mathbf{K}_{\mathbf{xx}}^{(j)^{-1}} \left(\mathbf{x} - \mathbf{m}_{\mathbf{x}}^{(j)}\right),$$

$$\mathbf{K}_{\mathbf{y}|\mathbf{x}}^{(j)} = \mathbf{K}_{\mathbf{yy}}^{(j)} - \mathbf{K}_{\mathbf{yx}}^{(j)} \mathbf{K}_{\mathbf{xx}}^{(j)^{-1}} \mathbf{K}_{\mathbf{xy}}^{(j)}, \tag{4.4}$$

and $g_j(\mathbf{x})$ is the weighted projection into the input space of the jth component, obtained through the marginal densities:

$$g_j(\mathbf{x}) = \frac{\pi^{(j)} \mathcal{N}\left(\mathbf{x}; \mathbf{m}_{\mathbf{x}}^{(j)}, \mathbf{K}_{\mathbf{xx}}^{(j)}\right)}{\sum_{n=1}^{C} \pi^{(n)} \mathcal{N}\left(\mathbf{x}; \mathbf{m}_{\mathbf{x}}^{(n)}, \mathbf{K}_{\mathbf{xx}}^{(n)}\right)}. \tag{4.5}$$

The conditional mean and the conditional variance of $f_{\mathbf{y}|\mathbf{x}}(\mathbf{y} \mid \mathbf{x})$ can be calculated easily from (4.3) and (4.4); when $\mathbf{y}$ is a scalar, $f_{\mathbf{y}|\mathbf{x}}(\mathbf{y} \mid \mathbf{x})$ will be a univariate conditional pdf and hence

$$\mathbb{E}_{\mathbf{y}|\mathbf{x}}[\mathbf{y} \mid \mathbf{x}] = \sum_{j=1}^{C} g_j(\mathbf{x}) \mathbf{m}_{\mathbf{y}|\mathbf{x}}^{(j)}; \tag{4.6}$$

$$\mathrm{Var}_{\mathbf{y}|\mathbf{x}}(\mathbf{y} \mid \mathbf{x}) = \sum_{j=1}^{C} g_j(\mathbf{x}) \left[\mathbf{K}_{\mathbf{y}|\mathbf{x}}^{(j)} + \left(\mathbf{m}_{\mathbf{y}|\mathbf{x}}^{(j)} - \mathbb{E}_{\mathbf{y}|\mathbf{x}}[\mathbf{y} \mid \mathbf{x}]\right)^2\right]. \tag{4.7}$$

The mean obtained using (4.6) corresponds to the least-squares estimation of any output $\mathbf{y}$ associated with an input pattern $\mathbf{x}$. The said equation defines distinctly the regression model of a MoG neural network, which has some evident similarities with respect to other well-known neural models such as Constrained Topological Mapping (CTM), Radial Basis Function (RBF), Adaptive Neurofuzzy Inference system (ANFIS), and so on. It is a piecewise linear regression model, which is evidently based on a suitable clustering procedure yielding several regions of the input space, defined by $g_j(\mathbf{x})$, $j = 1 \cdots C$, where the input-output mapping can be locally approximated by the linear functions $\mathbf{m}_{\mathbf{y}|\mathbf{x}}^{(j)}$ obtained in (4.4). Moreover, by analyzing (4.5), we notice that

$$\sum_{j=1}^{C} g_j(\mathbf{x}) = 1, \quad \text{for any } \mathbf{x}. \tag{4.8}$$

As a consequence of this constraint, if the basis functions in the input space are well separated and nonoverlapping then, for any $\mathbf{x}$ there exists a Gaussian component q such that $g_q(\mathbf{x}) \cong 1$ and $g_j(\mathbf{x}) \cong 0$ for $j \neq q$. The architecture of the MoG neural network resulting from the determination of the previous regression model, that is, equations from (4.1) to (4.6), is reported in Figure 1.

Resuming the original problem of time series forecasting, the MoG model can be used to estimate the conditional mean μ_t; namely, it corresponds to a specific implementation of h_μ. Looking in particular at (3.1), (4.4), and (4.6), when $\mathbf{x} = \mathbf{x}_t^{(\mu)^T}$, we have

$$\mu_t = \sum_{j=1}^{C} g_j\left(\mathbf{x}_t^{(\mu)^T}\right) \left[\mathbf{m}_{\mathbf{y}}^{(j)} + \mathbf{K}_{\mathbf{yx}}^{(j)} \mathbf{K}_{\mathbf{xx}}^{(j)-1}\left(\mathbf{x}_t^{(\mu)^T} - \mathbf{m}_{\mathbf{x}}^{(j)}\right)\right]. \tag{4.9}$$

Nevertheless, we remark that a conditional pdf is obtained in (4.3), which is just an instance of the generic pdf $f_{y_t|\mathcal{I}_{t-1}}(y_t \mid \mathcal{I}_{t-1}; \boldsymbol{\theta})$. As a consequence of this consideration, the estimation of MoG parameters can also be considered as a direct estimation of the parameters $\boldsymbol{\theta}$ in a GMM conditional pdf of the time series, which yields a nonlinear model (4.9) for the conditional
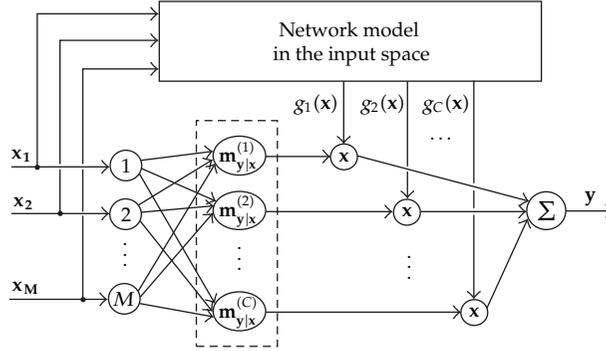
Figure 1: Regression model obtained by an MoG neural network. The number of scalar inputs, that is, the dimension of the input vector $\mathbf{x}$, is limited to M for illustration.

mean and also a nonlinear heteroskedastic model (4.7) for the volatility. By the way, when $C = 1$, there will be only one Gaussian component in the MoG mixture and hence, $f_{\mathbf{y}|\mathbf{x}}(\mathbf{y} \mid \mathbf{x})$ will be a simple normal distribution and the MoG regression will be identical to a linear ARMA model with homoskedastic (constant) volatility $\sigma_t^2 = \mathbf{K}_{\mathbf{y}|\mathbf{x}}^{(1)}$ given by (4.4).

In spite of the last considerations, MoG neural networks can also be used for the direct estimation of the volatility in order to complete the heteroskedastic model of time series introduced in (3.2). Following the computational scheme illustrated in the previous section, we should determine the parameters $\boldsymbol{\omega}_t^{(\sigma)}$ of the function $h_\sigma : \mathfrak{R}^{P+Q} \to \mathfrak{R}$. All the previous equations remain valid; in this case, $\mathbf{x} \in \mathfrak{R}^{P+Q}$ will be a generic column vector in the input space of h_σ and $\mathbf{y} \in \mathfrak{R}$ will be associated with the output space of h_σ. For instance, the forecast of volatility can be obtained similarly to (4.9)

$$\sigma_t^2 = \sum_{j=1}^{C} g_j \left(\mathbf{x}_t^{(\sigma)^T} \right) \left[\mathbf{m}_{\mathbf{y}}^{(j)} + \mathbf{K}_{\mathbf{yx}}^{(j)} \mathbf{K}_{\mathbf{xx}}^{(j)^{-1}} \left(\mathbf{x}_t^{(\sigma)^T} - \mathbf{m}_{\mathbf{x}}^{(j)} \right) \right]. \tag{4.10}$$

The use of clustering in MoG network for kernel-based regression is also suited to capture volatility clustering in energy commodity prices. GARCH models are able to model volatility clustering mainly because the model parameters are estimated repeatedly over time. This is also obtained by our proposed approach. However, the standard GARCH model in (2.4) is only a linear one, which unlikely can capture the clusters present inside the training set. Consequently, this makes very critical the choice of the number of past observations to be used for prediction. Nonlinear GARCH models can alleviate this problem, although using a global nonlinear regression. The training algorithms of MoG neural networks and other similar neural models are intended to find such clusters as a primary goal in their optimization routines. The capability of MoG to find volatility clustering and nonlinear phenomena in the time series analysis will be clearly proved in Section 5.

It is also important to mention that an MoG neural network can generalize the GMM mixture model by using latent variables, which allow a parsimonious representation of the mixture model by projecting data in a suitable subspace; consequently, they are widely used in factor analysis, principal component analysis, data coding, and other similar applications [61]. A full equivalence, under basic conditions, has been proved in [63] between the PPCA clustering performed by the MoG network and the reconstruction of signal subspaces based

on the SVD [32]. Therefore, this equivalence allows the adaptive determination of clusters identified on data, in order to maximize the quality of the reconstructed signal (i.e., the predicted time series). Furthermore, using known results in the SVD framework, the training algorithm may be performed with the automatic estimation of the latent variable subspace or, equivalently, of the signal subspace.

4.2. Training of MoG Neural Network

Without loss of generality, let us consider in the following the regression for conditional mean only. A useful way for determining the parameters of the whole Gaussian mixture $f_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})$, that is, $\pi^{(j)}$, $\mathbf{m}^{(j)}$ and $\mathbf{K}^{(j)}$, $j = 1 \cdots C$, is based on the maximum likelihood (ML) approach. It concerns with the maximization of the log-likelihood function:

$$\mathcal{L} = \sum_{i=1}^{N_T} \log \left[f_{\mathbf{x},\mathbf{y}} \left(\mathbf{d}_{t,i}^{(\mu)^T} \right) \right] = \sum_{i=1}^{N_T} \log \left[\sum_{j=1}^{C} \pi^{(j)} \mathcal{N} \left(\mathbf{d}_{t,i}^{(\mu)^T}; \mathbf{m}^{(j)}, \mathbf{K}^{(j)} \right) \right]. \tag{4.11}$$

For a fixed number C of mixture components, the ML estimation can be pursued by using different optimization schemes as, for example, the expectation-maximization (EM) algorithm [47]. It consists in the iteration of two steps, that is, the E-step and the M-step. The essential operations in the case of the GMM (4.1) are summarized in the following.

 (i) *E-step.* With the current estimation of the mixture parameters, the posterior probabilities R_{ij}, $i = 1 \cdots N_T$, $j = 1 \cdots C$, are updated

$$R_{ij} = \frac{\pi^{(j)} \mathcal{N} \left(\mathbf{d}_{t,i}^{(\mu)^T}; \mathbf{m}^{(j)}, \mathbf{K}^{(j)} \right)}{\sum_{n=1}^{C} \pi^{(n)} \mathcal{N} \left(\mathbf{d}_{t,i}^{(\mu)^T}; \mathbf{m}^{(n)}, \mathbf{K}^{(n)} \right)}. \tag{4.12}$$

 (ii) *M-step.* With the current posterior probabilities, the parameters of each Gaussian component, $j = 1 \cdots C$, are updated

$$\pi^{(j)} = \frac{1}{N_T} \sum_{i=1}^{N_T} R_{ij},$$

$$\mathbf{m}^{(j)} = \frac{\sum_{i=1}^{N_T} R_{ij} \mathbf{d}_{t,i}^{(\mu)^T}}{\sum_{i=1}^{N_T} R_{ij}}, \tag{4.13}$$

$$\mathbf{K}^{(j)} = \frac{\sum_{i=1}^{N_T} R_{ij} \left(\mathbf{d}_{t,i}^{(\mu)^T} - \mathbf{m}^{(j)} \right) \left(\mathbf{d}_{t,i}^{(\mu)} - \mathbf{m}^{(j)^T} \right)}{\sum_{i=1}^{N_T} R_{ij}},$$

where $\mathbf{K}^{(j)}$ is updated by using the new values of $\mathbf{m}^{(j)}$.

The algorithm starts with an initial determination of the mixture parameters. It stops when the absolute difference of the log-likelihood values (4.11) calculated in two successive E-steps is less than a given threshold.

When training an MoG network, the main problem is related to the local convergence of the EM algorithm and to the correct determination of the number C of Gaussian components. The former problem mainly depends on a good (usually random) initialization of the mixture parameters; the latter is a well-known problem, which is directly related to the generalization capability of the neural network. In fact, its performance could be inadequate if the training set is overfitted by an excessive number of Gaussian components. A plain solution to these problems could be based on the use of the EM algorithm with different values of C and with different initializations for each value of C. Once a set of different MoG neural networks has been generated, the choice of the best mixture model can be accomplished by relying on the supervised nature of the prediction problem, that is, by using a cost function measuring the overall generalization capability of the network in terms of its prediction error.

The plain optimization approach still suffers from some serious drawbacks, which mainly depend on the number of different initializations performed for each value of C. The lower is the number of initializations, the lower is the probability to obtain a satisfactory mixture after the EM algorithm. Conversely, the higher is the number of initializations, the higher is the computational cost of the optimization procedure. In order to overcome also these problems, we propose the use of a constructive procedure, where C is increased progressively and only one execution of the EM algorithm is needed for every value of C. This procedure eliminates any random initializations of EM algorithm and, consequently, the necessity to optimize different EM solutions for a fixed C. Thus, the computational cost of the whole training procedure is heavily reduced with respect to a plain optimization approach. The algorithm will stop when C reaches a value representing the maximum complexity allowed to the network. In this way, the training algorithm is structurally constrained, so that overfitting due to the presence of outliers in the training set can be prevented when C is low, ensuring robustness with respect to noise in the time series to be predicted.

The constructive procedure is intended to find a component in the GMM performing a poor local approximation and to substitute it by a pair of new components. The underlying idea is to prevent the initialization of the new components in underpopulated zones of the training set, since this is the typical situation where the EM algorithm will converge to a poor local maximum of (4.11). Several heuristics are possible to select the component to be split; we use in the following the one clearly illustrated in [39] that is based on the supervised nature of the present regression problem. More details about the constructive procedure and the demonstration that it yields better mixture weightings than a random initialization approach can also be found, for instance, in [30].

5. Performance Evaluation

The validity of the proposed approach has been validated by extensive computer simulations. Some illustrative examples are summarized in this section, firstly simulated data as reference benchmarks are considered, then actual return series related to energy commodity prices are used. The numerical results are obtained by using well-known neural and neurofuzzy models, which are compared with respect to the commonly used combination of ARMA-GARCH models estimated by maximum Gaussian likelihood.

The training procedure of neural regression models follows the scheme illustrated in Section 3. Bearing in mind the introduced notation, let y_t, $t = T_S \cdots (T_S + N_S - 1)$, be the set of actual samples of the time series to be predicted. For each sample, the neural networks are trained by using the previous N_T observations of the time series, that is, from $t - N_T$ to $t - 1$,

together with the related innovations and conditional variances. A one-step-ahead prediction is therefore applied and this procedure is iterated N_S times. After every prediction, the sequences of innovations, conditional mean and conditional variance forecast are updated.

In addition to the suggested MoG neural network, trained by the constructive procedure previously illustrated, we further consider two well-known neural architectures: RBF and ANFIS. The former is a feed-forward neural network trained by a constructive procedure that iteratively creates a radial basis network one neuron at a time. Neurons are added to the network until an error goal or a maximum number of neurons is reached [64, 65]. The ANFIS is a neurofuzzy network, combining the data-driven learning of neural networks with the robustness of fuzzy logic and linguistic fuzzy rules. ANFIS networks have been trained by a subtractive clustering method for rule extraction [66], while the rule parameters are obtained by means of a standard least-squares method coupled with the back-propagation optimization [67]. All these training procedures also aim to optimize the structural complexity (i.e., number of kernels, hidden nodes, fuzzy rules, etc.) of the resulting neural network. For reasons of conciseness, we will not provide details about complexity in the following, since it is optimal as long as the neural network exhibits a good generalization capability, which is evaluated as described in the following by means of the network's performance on test sets not used during training.

We take particular care to the criteria used to evaluate the performance of the algorithms. There are many error measures adopted in the literature, in particular for measuring the prediction error. They are differently used according to different fields as statistics, engineering, econometrics, and so forth. Let $\mu_t, t = T_S \cdots (T_S + N_S - 1)$, be the set of conditional means representing the prediction (obtained by using any model) of the corresponding values y_t. The error measures used in this paper are the following ones:

(i) Mean Squared Error (MSE):

$$\text{MSE} = \frac{1}{N_S} \sum_t (y_t - \mu_t)^2; \tag{5.1}$$

(ii) Normalized Mean Squared Error (NMSE):

$$\text{NMSE} = \frac{\sum_t (y_t - \mu_t)^2}{\sum_t (y_t - \overline{y})^2}; \tag{5.2}$$

(iii) Noise-to-Signal Ratio (NSR):

$$\text{NSR}_{dB} = 10 \log_{10} \frac{\sum_t (y_t - \mu_t)^2}{\sum_t y_t^2}; \tag{5.3}$$

where $\overline{y}$ is the average of the actual N_S samples of y_t. In general, a prediction error can be defined on the return series only by using the estimated conditional means. However, when dealing with simulated data, we can also determine the true sequence of conditional variances. So, the same measures can be applied to the sequence of predicted conditional variances, considering σ_t^2 in place of μ_t and the actual conditional variance in place of y_t. For this reason, we will distinguish the errors where necessary.

Best approximation is linked to small values of (5.1), (5.2), and (5.3) and it is interesting to notice that the MSE is the energy of the prediction error, while in NMSE and NSR, this energy is normalized to the variance and to the energy of the observed sequence, respectively. Each measure aims to quantify how much the prediction has been corrupted by errors and should allow an objective evaluation of the prediction accuracy, independently of the used time series. So, we should be able to set a sound threshold of the NSR that assures a good performance especially for what concerns the statistical distributions. In this circumstance, we will prove that good performances are usually obtained when the NSR is lower than approximately 1 dB.

Energy prices are critical to producers, consumers, and investors. Derivative instruments are an efficient tool for hedging and managing the price and volatility risks. To properly price energy derivatives is critically dependent on accurately modeling the underlying spot price. A successful model describing spot prices dynamics must capture the statistical features of the analyzed data in the simulated series. To this aim, the unconditional moments from the first up to the fourth order are estimated (as time averages) and considered for both actual and predicted sequences. A given model is suited to forecast and model energy commodity prices when the first four moments of the predicted sequences result as close as possible to the moments estimated on the market data. Being able to reproduce the probability distribution of the observed series together with an accurate prediction of the daily prices will allow investors and risk managers to estimate the profit/loss scenarios to set up the adequate risk management strategies.

5.1. Synthetic Benchmarks

For the sake of comparison, we consider the following artificial data sets with Gaussian innovations investigated in [55, 68], also useful to assess the proposed approach in case of high degree of nonlinearity:

(1) heteroskedastic model with sinusoidal mean

$$y_t = y_{t-1}\sin(y_{t-1}) + \epsilon_t; \tag{5.4}$$

and GARCH$(1,1)$ model for volatility

$$\sigma_t^2 = 0.1 + 0.85\sigma_{t-1}^2 + 0.1\epsilon_t^2; \tag{5.5}$$

(2) heteroskedastic zero-mean model

$$y_t = \sigma_t\epsilon_t; \tag{5.6}$$

and highly nonlinear volatility

$$\sigma_t^2 = \left(0.4y_{t-1}^2 + 0.5\sigma_{t-1}^2\right)^{3/4} \\ + \left[0.8\left(0.1 + 0.2|y_{t-1}| + 0.9y_{t-1}^2\right)e^{(-1.5|y_{t-1}|\sigma_{t-1}^2)}\right]. \tag{5.7}$$

A realization of 1200 samples is generated by using each model; $N_T = 1000$ samples are used to train the prediction model for each sample; $N_S = 200$ samples are used for test,
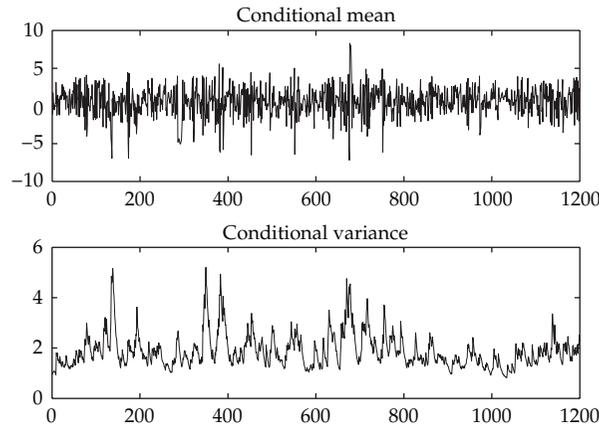
Figure 2: Realization of mean and variance for the sinusoidal mean benchmark defined by (5.4) and (5.5).
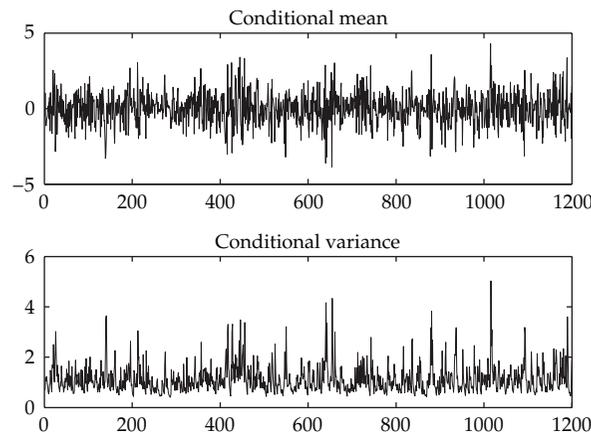
Figure 3: Realization of mean and variance for the zero-mean benchmark defined by (5.6) and (5.7).

starting at sample $T_S = 1001$; hence, the index of predicted samples ranges from $t = 1001$ to $t = 1200$. We generated 50 different realizations for both models and we report in the following the mean results over the different realizations. An illustrative realization of mean and variance obtained by the first model is shown in Figure 2, while a realization of the second model is reported in Figure 3. As default model parameters we adopted $R = 1$, $M = 0$, $P = 1$, and $Q = 1$; therefore, the standard reference model is an ARMA(1,0)-GARCH(1,1), which will be briefly denoted in this context as "GARCH."

The numerical results for the two models are summarized in Tables 1 and 2, respectively. As explained, we can apply the error measures to the predicted time series and to the predicted variance (or volatility) as well. In all cases, when the original time series are highly nonlinear, we find that GARCH models are not able to capture the real nature of the data. In the first benchmark, which is a classic heteroskedastic process, the nonlinear models of neural networks, especially RBF and ANFIS, are able to well represent the sinusoidal mean (5.4). Although GARCH and MoG obtain comparable results on the mean, MoG is much more accurate to model the volatility since GARCH tends to shape the variability due to mean

Table 1: Numerical results for the sinusoidal mean benchmark of (5.4) and (5.5).

Model	Mean			Variance		
	MSE	NMSE	NSR	MSE	NMSE	NSR
GARCH	3.32	1.16	−0.22	10.84	55.73	5.51
RBF	1.92	0.83	−1.86	3.87	20.19	1.04
ANFIS	2.37	1.02	−0.96	5.67	29.55	2.70
MoG	2.72	1.16	−0.39	3.55	18.54	0.67

Table 2: Numerical results for the zero-mean benchmark of (5.6) and (5.7).

Model	Mean			Variance		
	MSE	NMSE	NSR	MSE	NMSE	NSR
GARCH	2.38	1.76	2.45	1.41	4.02	−0.44
RBF	1.22	1.17	0.68	0.21	0.82	−8.44
ANFIS	1.17	1.12	0.49	0.14	0.54	−10.21
MoG	1.15	1.10	0.40	0.05	0.23	−14.56

as an increased volatility [55]. Poor results are obtained also when GARCH is used to estimate the variance of the time series in the second benchmark. The nonlinear regression models provided by neural networks perform better and, in particular, MoG network achieves the best prediction of the variance as illustrated in Figure 4.

5.2. Energy Commodity Prices

We consider the time series obtained from the daily prices of specific energy commodities. We carried out several tests to assess the validity of the proposed approach. Coal, natural gas, crude oil, and electricity prices over the period 2001–2010 were collected for both the European and the US markets. The results reported in this paper refer to the US market only, similar results have been obtained for the European data set not showing significant differences in the forecasting ability. The studied commodities and the related indexes are as follows: coal (DJUSCL, in $/ton), Henry Hub natural gas (HH, in $/MMBtu), crude oil (WTI, in $/barrel), and electricity (PJM, in $/MWh). For electricity prices, we chose the peak load contract referred to h. 12:00. For each commodity price log-returns are estimated using (2.1). We chose a well representative time window across the "very critical" year 2008, that is, from the beginning of 2006 to the end of 2009. So, taking into account that we have approximately 250 prices and 249 returns per year in the case of coal, natural gas, and crude oil series, each return series consists of about 1000 samples. In the case of electricity prices we have a series of 365 data, given that electricity is traded every day of the year; in this application, for comparison purposes, we adjust the series to 250 trading days only. Each model is trained on the previous $N_T = 500$ samples (almost two years) and $N_S = 500$ samples are predicted, that is, the last two years starting from $t = 501$ up to the last sample of 2009.

The prediction errors of the conditional mean are evaluated; in addition, the four unconditional moments are estimated for both the predicted sequences and the related original series. Prior investigations can be made in order to find the best combination of the model orders and the size of the training set as well. The goal of this paper is to propose a new machine learning approach to forecast the time series related to energy commodities, so we only demonstrate how it favorably compares with respect to other choices. Fine tuning for the optimal estimate of any models can be addressed in future research works. However,
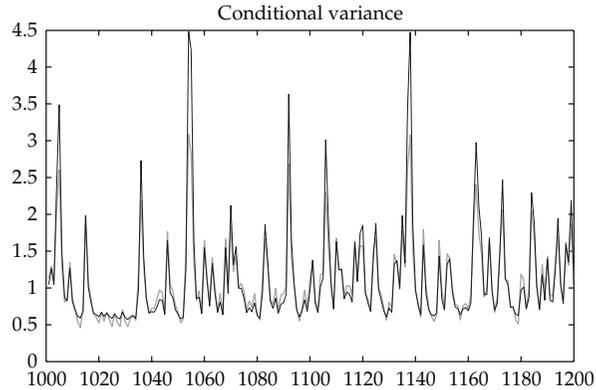
Figure 4: Prediction of conditional variance of the zero-mean benchmark obtained by the MoG neural network: forecast (black); actual (gray).

Table 3: Numerical results for DJUSCL return series.

Model	Errors			Unconditional moments			
	MSE	NMSE	NSR	Mean	Variance	Skewness	Kurtosis
DJUSCL				−0.31	0.74	−0.41	5.60
GARCH	0.80	1.08	0.33	0.34	0.33	0.07	9.36
RBF	0.76	1.03	0.13	−0.39	0.88	−0.18	5.28
ANFIS	0.77	1.04	0.17	−0.27	0.91	−0.41	5.34
MoG	0.78	1.04	0.18	0.77	1.01	−0.32	5.20

MSE, mean, and variance are scaled by 10^{-3}.

in order to obtain an accurate comparison of the performances obtained by the proposed neural networks with respect to standard models, we carried out a preliminary optimization of the main model parameters, that is, R, M, P, and Q, so as to obtain the best performance of the reference "GARCH" model for a given time series. Then, every model will use the same parameters when applied to the same time series.

The coal DJUSCL daily prices and returns are shown in Figure 5 and the numerical results are reported in Table 3. The optimal parameters are $R = 1$, $M = 1$, $P = 5$, $Q = 1$; hence, a GARCH reference model ARMA(1, 5)-GARCH(1, 1) is fitted. All the neural models score a prediction error better than GARCH: RBF obtains the best NSR but the skewness is not properly matched as in the case of ANFIS. MoG neural network achieves a good NSR performance of 0.18 dB, and its results are plotted in Figure 6. Although not the best model, MoG neural network is able to follow the dynamic of changes of the actual conditional mean and to forecast the increasing volatility as proved by the behavior of the estimated conditional variance.

The HH daily prices and returns of natural gas are shown in Figure 7 and the numerical results are reported in Table 4. The optimal parameters are in this case $R = 2$, $M = 2$, $P = 2$, $Q = 1$, so a GARCH reference model ARMA(2, 2)-GARCH(2, 1) is fitted. The MoG neural network has the best NSR performance of 0.21 dB and the related moments adequately fit with those of the original time series. A sufficient accuracy is also obtained by RBF and ANFIS neural networks. GARCH is not suitable for the prediction of HH returns, since the moments are estimated very poorly, especially the kurtosis. The numerical results

Table 4: Numerical results for HH return series.

Model	Errors			Unconditional moments			
	MSE	NMSE	NSR	Mean	Variance	Skewness	Kurtosis
HH				−0.46	1.72	1.46	10.62
GARCH	1.97	1.14	0.57	1.65	0.12	2.49	67.15
RBF	1.87	1.09	0.37	−0.44	1.92	0.27	4.54
ANFIS	2.05	1.19	0.75	−0.47	1.87	0.71	7.35
MoG	1.80	1.05	0.21	−0.34	1.90	1.13	9.94

MSE, mean, and variance are scaled by 10^{-3}.
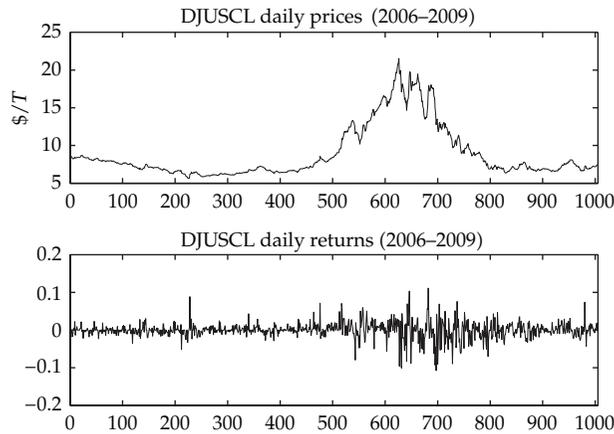
Figure 5: Daily prices and returns of the DJUSCL coal index.

of the MoG neural network are qualitatively confirmed by the accurate prediction of the conditional mean and volatility reported in the plots of Figure 8.

The behaviors of crude oil WTI daily prices and returns are shown in Figure 9. The large volatility at the end of 2008 is the feature that requires an accurate forecasting technique. A more complex model is therefore necessary, using $R = 4$, $M = 2$, $P = 2$, $Q = 3$. The GARCH reference model ARMA(4, 2)-GARCH(2, 3) is evidently outperformed by the neural networks, as evidenced by the results summarized in Table 5. The best NSR is now obtained by ANFIS although the predicted sequence does not fit the skewness of the original one. Once again MoG is able to fit the original moments, also maintaining a good prediction accuracy and following the changes of volatility in the underlying process as illustrated in Figure 10.

Finally, the prices and returns of PJM electricity index are shown in Figure 11 and the numerical results are reported in Table 6. The model parameters are $R = 3$, $M = 2$, $P = 1$, $Q = 1$; the GARCH reference model is ARMA(3, 2)-GARCH(1, 1). Similarly to the previous cases, MoG neural network performs better than GARCH, RBF, and ANFIS. Globally neural networks improve the NSR performance of more than 2 dB with respect to GARCH despite a biasing that shifts the estimate of the mean to negative values.

PJM returns are not characterized by the same amount of volatility than WTI or coal. However, some power markets usually show some extreme behaviors, in particular spikes that enhance the kurtosis to very high values. Actually, 2008 is deeply affected by these events and it has been analyzed, though compounded with 2009. Anyway, in order to test the proposed approach also in such situations, we focus the last quarter (Q4) of 2008 where the kurtosis increases to about 9. Leaving the same the training process, as previously illustrated,
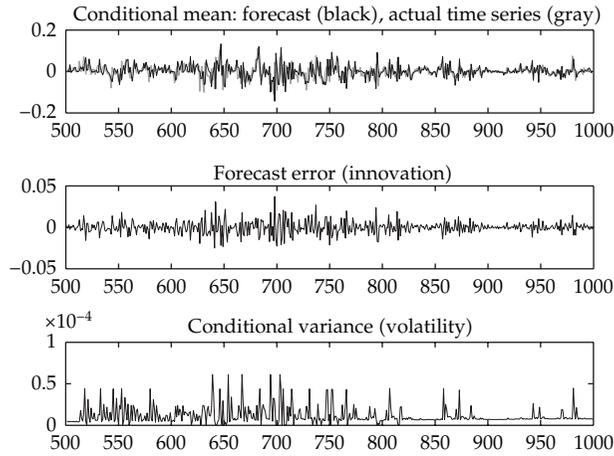
Figure 6: Prediction of coal returns using the MoG neural network.
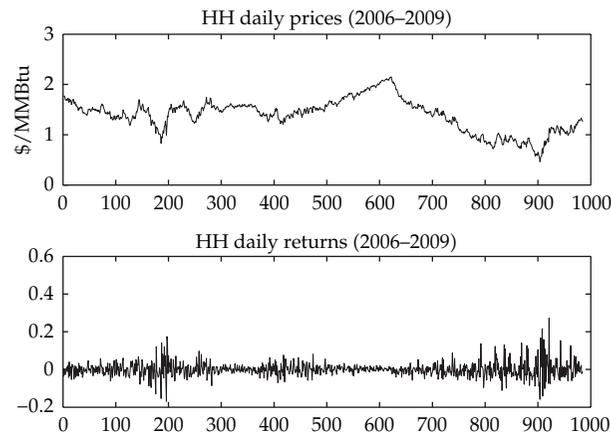
Figure 7: Daily prices and returns of the HH natural gas index.
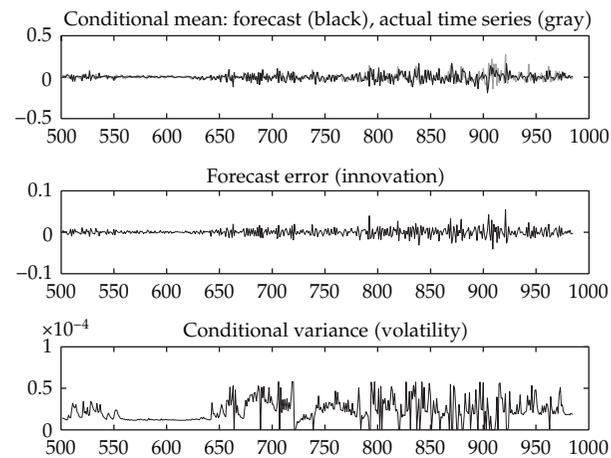
Figure 8: Prediction of natural gas returns using the MoG neural network.

Table 5: Numerical results for WTI return series.

Model	Errors			Unconditional moments			
	MSE	NMSE	NSR	Mean	Variance	Skewness	Kurtosis
WTI				−0.10	0.11	0.34	6.51
GARCH	0.12	1.00	0.01	0.43	0.01	0.48	9.38
RBF	0.11	0.94	−0.27	−0.01	0.12	0.16	4.06
ANFIS	0.09	0.89	−0.50	−0.05	0.12	−0.25	7.22
MoG	0.11	0.92	−0.36	−0.10	0.10	0.44	5.73

MSE, mean, and variance are scaled by 10^{-3}.

Table 6: Numerical results for PJM return series.

Model	Errors			Unconditional moments			
	MSE	NMSE	NSR	Mean	Variance	Skewness	Kurtosis
PJM				0.22	1.21	0.38	5.52
GARCH	1.19	0.98	−0.09	1.53	0.24	0.53	8.10
RBF	0.71	0.58	−2.36	−0.11	1.64	−0.01	2.96
ANFIS	0.64	0.53	−2.76	−0.21	1.24	0.26	5.58
MoG	0.57	0.47	−3.28	−0.41	1.77	0.28	4.61

MSE, mean, and variance are scaled by 10^{-3}.

Table 7: Numerical results for PJM return series in Q4 2008.

Model	Errors			Unconditional moments			
	MSE	NMSE	NSR	Mean	Variance	Skewness	Kurtosis
PJM				−1.08	0.85	−1.14	8.98
GARCH	0.78	0.92	−0.38	1.66	0.13	−2.97	18.09
RBF	0.30	0.35	−4.55	−1.08	0.61	−0.51	9.54
ANFIS	0.27	0.31	−5.01	−1.20	0.51	−1.28	9.95
MoG	0.33	0.38	−4.21	−0.72	1.09	−0.65	8.44

MSE, mean, and variance are scaled by 10^{-3}.
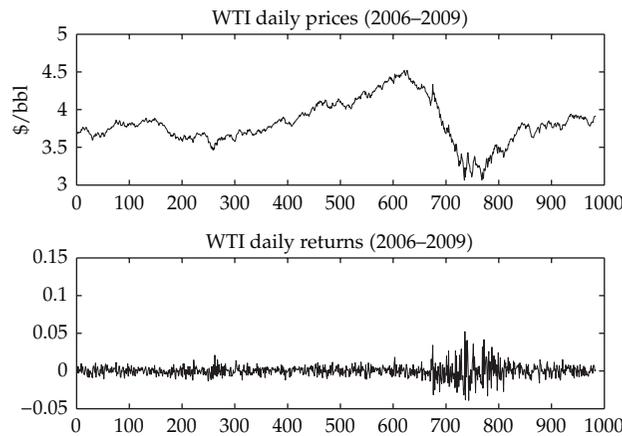
Figure 9: Daily prices and returns of WTI crude oil index.
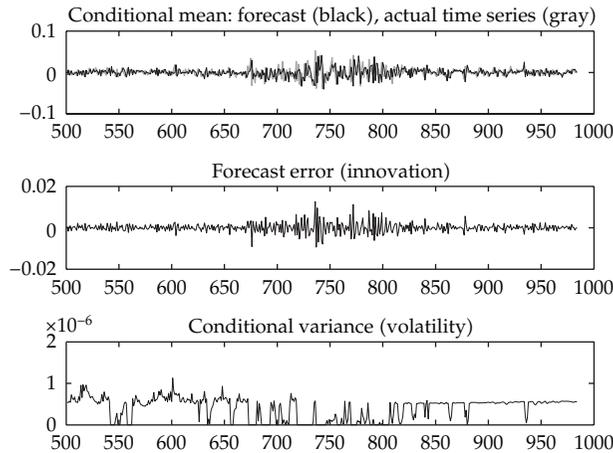
Figure 10: Prediction of crude oil returns using the MoG neural network.
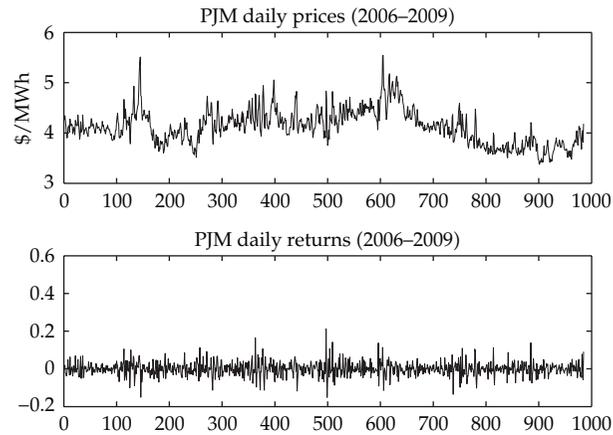
Figure 11: Daily prices and returns of PJM electricity index.

the numerical results obtained in the considered subperiod are summarized in Table 7. These results prove that the proposed neural approach is able to model complex time series showing high kurtosis, as in the presence of spikes, especially when compared to GARCH models. Nonetheless, further comparisons of neural models with respect to stochastic models (such as jump-diffusion, regime-switching, etc.) can be considered in the future, taking into account the potentiality of neural networks to obtain good performances independently of the considered process, whilst stochastic models must be carefully chosen *a priori* also relying on the skill of expert users.

6. Conclusion

In this paper, a new neural network approach is proposed for modeling time series associated with energy commodity prices. It is based on a pure data-driven estimation of nonlinear regression models, which are applied to the estimation of both conditional mean and volatility. In this regard, we focused on MoG neural networks, which are based on GMM for

kernel-based regression and pdf approximation. MoG neural networks can be considered as a nonlinear generalization of GARCH models, for which volatility clustering can be pursued in a straightforward manner.

The use of a constructive procedure to train MoG neural networks has been proposed. It aims to the regularization of the network complexity, determining automatically the optimal number of components in the GMM in order to avoid overfitting and maximize the generalization capability of the network. The proposed approach provides very accurate description of energy prices dynamics, allowing to estimate daily prices for energy commodities over a long time horizon. The validation performed on historical data shows that the neural network approach generates prices that are able to replicate the daily data and to reproduce the same probability distribution for the various series.

The proposed approach does not rely on information about seasonality in the processed time series. Actually, both natural gas and coal exhibit seasonality in the price and volatility, thereby suggesting that there is a nonrandom component to the signals which can be used to improve backtesting of the model using daily returns. However, seasonality is not explicitly accounted for using neural networks, since they can be used as a black-box where such an information is inherently processed using neither specific preprocessing tasks nor the personal skill of the users. Nevertheless, unlike coal, natural gas, and crude oil, electricity is nonstorable and exhibits large jumps in daily spot price returns. Also in this case, the proposed approach and the generalization capability of the considered neural models are able to take into account these phenomena without using specific models as, for example, the well-known mean-reverting jump diffusion model.

The accurate prediction of daily prices on a short-term basis will provide a useful tool to estimate the relationship existing between related commodities, that is, correlation between natural gas and crude oil or between natural gas and electricity. These relationships represent a key element for the pricing of energy derivatives as well as spread options which have become a largely used instrument to efficiently manage risk in the energy markets. Currently, we are investigating more advanced techniques for the application of the proposed approach to a multivariate time series analysis and for the automatic and more reliable selection of the samples to be used for prediction, including the order of regression models and the resulting complexity of neural models.

References

[1] J. D. Hamilton, "Understanding crude oil prices," *Energy Journal*, vol. 30, no. 2, pp. 179–206, 2009.

[2] R. Gibson and E. Schwartz, "Stochastic convenience yield and the pricing of oil contingent claims," *The Journal of Finance*, vol. 45, pp. 959–976, 1990.

[3] M. Barlow, Y. Gusev, and M. Lai, "Calibration of multifactor models in electricity markets," *International Journal of Theoretical and Applied Finance*, vol. 7, no. 2, pp. 101–120, 2004.

[4] Á. Cartea and M. G. Figueroa, "Pricing in electricity markets: a mean reverting jump diffusion model with seasonality," *Applied Mathematical Finance*, vol. 12, no. 4, pp. 313–335, 2005.

[5] C. Mari, "Random movements of power prices in competitive markets: a hybrid model approach," *The Journal of Energy Markets*, vol. 1, no. 2, pp. 87–103, 2008.

[6] H. Geman, *Commodities and Commodity Derivatives: Modeling and Pricing for Agriculturals, Metals and Energy*, The Wiley Finance Series, John Wiley & Sons, Chichester, UK, 2005.

[7] R. G. Donaldson and M. Kamstra, "An artificial neural network-GARCH model for international stock return volatility," *Journal of Empirical Finance*, vol. 4, no. 1, pp. 17–46, 1997.

[8] F. Miranda and N. Burgess, "Modelling market volatilities: the neural network perspective," *European Journal of Finance*, vol. 3, pp. 137–157, 1997.

[9] K. J. Kim, "Financial time series forecasting using support vector machines," *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.

[10] B. R. Szkuta, "Electricity price short-term forecasting using artificial neural networks," *IEEE Transactions on Power Systems*, vol. 14, no. 3, pp. 851–857, 1999.

[11] B. L. Zhang and Z. Y. Dong, "An adaptive neural-wavelet model for short term load forecasting," *Electric Power Systems Research*, vol. 59, no. 2, pp. 121–129, 2001.

[12] W.-H. Chen, J.-Y. Shih, and S. Wu, "Comparison of support-vector machines and back propagation neural networks in forecasting the six major asian stock markets," *Electronic Finance*, vol. 1, no. 1, pp. 49–67, 2006.

[13] W. Xie, L. Yu, L. Xu, and S. Wang, "A new method for crude oil price forecasting based on support vector machines," in *Proceedings of the 6th International Conference on Computational Science (ICCS '06)*, V. N. Alexandrov, G. D. van Albada, P. M. A. Sloot, and J. Dongarra, Eds., vol. 3994 of *Lecture Notes in Computer Science*, pp. 444–451, Springer, Heidelberg, Germany, 2006.

[14] I. Haidar, S. Kulkarni, and H. Pan, "Forecasting model for crude oil prices based on artificial neural networks," in *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '08)*, pp. 103–108, Sydney, Australia, 2008.

[15] S. Kulkarni and I. Haidar, "Forecasting model for crude oil price using artificial neural networks and commodity futures prices," *International Journal of Computer Science and Information Security*, vol. 2, no. 1, pp. 81–88, 2009.

[16] A. K. Dhamija and V. K. Bhalla, "Financial time series forecasting: comparison of neural networks and ARCH models," *International Research Journal of Finance and Economics*, vol. 49, pp. 185–202, 2010.

[17] R. Dase and D. Pawar, "Application of artificial neural network for stock market predictions: a review of literature," *International Journal of Machine Intelligence*, vol. 2, no. 2, pp. 14–17, 2010.

[18] T. Mills, *Time Series Techniques for Economists*, Cambridge University Press, Cambridge, UK, 1990.

[19] A. Pankratz, *Forecasting with Dynamic Regression Models*, John Wiley & Sons, New York, NY, USA, 1991.

[20] D. B. Percival and A. T. Walden, *Spectral Analysis for Physical Applications*, Cambridge University Press, Cambridge, UK, 1993.

[21] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis*, Prentice-Hall, Englewood Cliffs, NJ, USA, 3rd edition, 1994.

[22] P. Brockwell and R. Davis, *Time Series: Theory and Methods*, Springer, New York, NY, USA, 2nd edition, 2009.

[23] M. Panella, F. Barcellona, V. Santucci, and R. L. D'Ecclesia, "Neural networks to model energy commodity price dynamics," in *Proceedings of the USAEE/IAEE North American Conference*, Washington, DC, USA, 2011.

[24] M. Panella, F. Barcellona, and R. L. D'Ecclesia, "Subband prediction of energy commodity prices," in *Proceedings of the IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2012)*, pp. 495–499, IEEE, Cesme, Turkey, 2012.

[25] M. Panella, F. Barcellona, and R. L. D'Ecclesia, "Modeling energy markets using neural networks and spectral analysis," in *Proceedings of the IAEE European Energy Conference*, IAEE, Venice, Italy, 2012.

[26] H. D. I. Abarbanel, *Analysis of Observed Chaotic Data*, Institute for Nonlinear Science, Springer, New York, NY, USA, 1996.

[27] S. Haykin and J. Principe, "Making sense of a complex world," *IEEE Signal Processing Magazine*, vol. 15, no. 3, pp. 66–81, 1998.

[28] B. Widrow and R. Winter, "Neural nets for adaptive filtering and adaptive pattern recognition," *Computer*, vol. 21, no. 3, pp. 25–39, 1988.

[29] F. Masulli and L. Studer, "Time series forecasting and neural networks," in *Proceedings of International Joint Conference on Neural Networks (IJCNN '99)*, Washington, DC, USA, 1999.

[30] M. Panella, A. Rizzi, and G. Martinelli, "Refining accuracy of environmental data prediction by MoG neural networks," *Neurocomputing*, vol. 55, no. 3-4, pp. 521–549, 2003.

[31] M. Panella, "Advances in biological time series prediction by neural networks," *Biomedical Signal Processing and Control*, vol. 6, no. 2, pp. 112–120, 2011.

[32] B. D. Moor, "Singular value decomposition and long and short spaces of noisy matrices," *IEEE Transactions on Signal Processing*, vol. 41, no. 9, pp. 2826–2838, 1993.

[33] M. E. Tarter and M. D. Lock, *Model-Free Curve Estimation*, vol. 56 of *Monographs on Statistics and Applied Probability*, Chapman & Hall, New York, NY, USA, 1993.

[34] L. Xu and M. I. Jordan, "On convergence properties of the EM algorithm for gaussian mixtures," *Neural Computation*, vol. 8, no. 1, pp. 129–151, 1996.

[35] A. Figueiredo and A. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 381–396, 2002.

[36] J. Li and A. Barron, "Mixture density estimation," in *Advances in Neural Information Processing Systems*, vol. 12, MIT Press, Cambridge, Mass, USA, 2002.

[37] G. McLachlan and D. Peel, *Finite Mixture Models*, Wiley Series in Probability and Statistics: Applied Probability and Statistics, Wiley-Interscience, New York, NY, USA, 2000.

[38] C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, New York, NY, USA, 2006.

[39] M. Panella, A. Rizzi, F. M. Frattale Mascioli, and G. Martinelli, "A constructive EM approach to density estimation for learning," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN'01)*, vol. 4, pp. 2608–2613, Washington, DC, USA, July 2001.

[40] G. Acciani, E. Chiarantoni, G. Fornarelli, and S. Vergura, "A feature extraction unsupervised neural network for an environmental data set," *Neural Networks*, vol. 16, no. 3-4, pp. 427–436, 2003.

[41] R. Parisi, A. Cirillo, M. Panella, and A. Uncini, "Source localization in reverberant environments by consistent peak selection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 1, pp. I37–I40, Honolulu, Hawaii, USA, April 2007.

[42] C. Y. JIA, X. Y. SHAN, and Z. P. NIU, "High precision prediction of rolling force based on fuzzy and nerve method for cold tandem mill," *Journal of Iron and Steel Research International*, vol. 15, no. 2, pp. 23–27, 2008.

[43] M. Panella and G. Martinelli, "Neurofuzzy networks with nonlinear quantum learning," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 3, pp. 698–710, 2009.

[44] M. Panella and G. Martinelli, "Neural networks with quantum architecture and quantum learning," *International Journal of Circuit Theory and Applications*, vol. 39, no. 1, pp. 61–77, 2011.

[45] S. Salcedo-Sanz, J. L. Camacho, A. M. Perez-Bellido, E. G. Ortiz-Garcia, A. Portilla-Figueras, and E. Hernandez-Martin, "Improving the prediction of average total ozone in column over the Iberian Peninsula using neural networks banks," *Neurocomputing*, vol. 74, no. 9, pp. 1492–1496, 2011.

[46] M. Panella and G. Martinelli, "The quantum approach leading from evolutionary to exhaustive optimization," *Journal of Applied Sciences*, vol. 12, pp. 1995–2005, 2012.

[47] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[48] R. Neal, "Monte carlo implementation of gaussian process models for bayesian regression and classification," Tech. Rep. 9702, Department of Statistics, University of Toronto, 1997.

[49] S. Seo, M. Wallat, T. Graepel, and K. Obermayer, "Gaussian process regression: active data selection and test point rejection," in *International Joint Conference on Neural Networks (IJCNN'2000)*, pp. 241–246, July 2000.

[50] S. Haykin, *Neural Networks, a Comprehensive Foundation*, Prentice-Hall, Englewood Cliffs, NJ, USA, 2nd edition, 1999.

[51] M. Panella, "A hierarchical procedure for the synthesis of ANFIS networks," *Advances in Fuzzy Systems*, vol. 2012, Article ID 491237, 12 pages, 2012.

[52] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[53] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982.

[54] T. Bollerslev, "A conditionally heteroskedastic time series model for speculative prices and rates of return," *Review of Economics & Statistics*, vol. 69, no. 3, pp. 542–547, 1987.

[55] J. M. Matías, M. Febrero-Bande, W. González-Manteiga, and J. C. Reboredo, "Boosting GARCH and neural networks for the prediction of heteroskedastic time series," *Mathematical and Computer Modelling*, vol. 51, no. 3-4, pp. 256–271, 2010.

[56] L. D. S. Coelho and A. A. P. Santos, "A RBF neural network model with GARCH errors: application to electricity price forecasting," *Electric Power Systems Research*, vol. 81, no. 1, pp. 74–83, 2011.

[57] C. Schittenkopf, G. Dorffner, and E. J. Dockner, "Forecasting time-dependent conditional densities: a semi-non-parametric neural network approach," *Journal of Forecasting*, vol. 19, no. 4, pp. 355–374, 2000.

[58] P. Tiňo, C. Schittenkopf, and G. Dorffner, "Financial volatility trading using recurrent neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 865–874, 2001.

[59] R. T. Baillie and T. Bollerslev, "Prediction in dynamic models with time-dependent conditional variances," *Journal of Econometrics*, vol. 52, no. 1-2, pp. 91–113, 1992.

[60] T. Bollerslev, R. F. Engle, and D. B. Nelson, "ARCH models," in *Handbook of Econometrics, volume IV*, vol. 2 of *Handbooks in Economics*, chapter 49, pp. 2959–3038, North-Holland, Amsterdam, The Netherlands, 1994.

[61] C. M. Bishop, *Neural Networks for Pattern Recognition*, The Clarendon Press Oxford University Press, New York, NY, USA, 1995.

[62] Z. Ghahramani, "Solving inverse problems using an EM approach to density estimation," in *Proceedings of the Connectionist Models Summer School*, Erlbaum Associates, Hillsdale, NJ, USA, 1994.

[63] M. Panella, G. Grisanti, and A. Rizzi, "A probabilistic PCA clustering approach to the SVD estimate of signal subspaces," in *WIRN VIETRI 2004*, B. Apolloni, M. Marinaro, and R. Tagliaferri, Eds., Biological and Artificial Intelligence Environments, pp. 271–279, Springer, Amsterdam, The Netherlands, 2005.

[64] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302–309, 1991.

[65] A. G. Borş and I. Pitas, "Median radial basis function neural network," *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1351–1364, 1996.

[66] S. Chiu, "Fuzzy model identification based on cluster estimation," in *Journal of Intelligent & Fuzzy Systems*, vol. 2, pp. 267–278, 1994.

[67] J.-S. Jang, C. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*, Prentice-Hall, Upper Saddle River, NJ, USA, 1997.

[68] F. Audrino and P. Buhlmann, "Volatility estimation with functional gradient descent for very high-dimensional financial time series," *Journal of Computational Finance*, vol. 6, pp. 65–89, 2003.