Expert Systems with Applications 36 (2009) 7355–7362

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Improving forecasts of GARCH family models with the artificial neural networks: An application to the daily returns in Istanbul Stock Exchange

Melike Bildirici [a,*], Özgür Ömer Ersin [b,1]

[a] Yıldız Technical University, Faculty of Economics and Business Administration, Department of Economics, Barbaros Bulvari, Besiktas, İstanbul, Turkey
[b] Yeditepe University, Faculty of Commerce, Department of International Trade and Business, Kadikoy, Kayisdagi, İstanbul, Turkey

ARTICLE INFO

Keywords:
Volatility
Stock returns
ARCH/GARCH
EGARCH
TGARCH
PGARCH
APGARCH
Artificial neural networks

ABSTRACT

In the study, we discussed the ARCH/GARCH family models and enhanced them with artificial neural networks to evaluate the volatility of daily returns for 23.10.1987–22.02.2008 period in Istanbul Stock Exchange. We proposed ANN-APGARCH model to increase the forecasting performance of APGARCH model. The ANN-extended versions of the obtained GARCH models improved forecast results. It is noteworthy that daily returns in the ISE show strong volatility clustering, asymmetry and nonlinearity characteristics.

1. Introduction

The importance of volatility has led to the development and application of many significant econometric models and found important application areas in financial markets as a result of both the need of modeling the uncertainty and the risk in financial asset returns. The three most significant characteristics of returns in financial assets could be stated as the following: *volatility clustering property*, as a result of the volatility changes over time in magnitude and in cases where prices hardly change but volatility increases in sizes of large clusters; *asymmetric relation property* of volatility to past return shocks (Engle & Ng, 1993; Glosten, Jagannathan, & Runkle, 1993; Nelson, 1991, 1992); and *nonlinearity property*, the path of volatility reacts differently in different regimes (Klaassen, 2002; Kramer, 2006).

Engle (1982) ARCH models and Bollerslev (1986) GARCH models found many important applications in the financial markets. As a result, GARCH models respond to the need felt for a foresight method that takes into account various properties of the probability distribution of return series of financial variables and had been used intensively in academic studies. Due to this effect, asymmetric GARCH models have rapidly expanded (Nelson, 1991). While Nelson (1991) developed Exponential GARCH (EGARCH) model, Zakoian (1994) and Glosten et al., 1993 working independent of one another developed the GJR-GARCH model. Zakoian (1994) introduced the Threshold GARCH (TGARCH) model and Sentana (1995) introduced the Quadratic GARCH (QGARCH) model.

GARCH models of Taylor (1986) and Schwert (1989) relate the conditional standard deviation of a series and past standard deviations of a different property compared to other models. The model was generalized by Ding, Granger, and Engle (1993) as Power GARCH. This study can be considered as the basis of the APGARCH literature. Hentschel (1995) has applied his study, in which he proposes a more general model of the Power ARCH model, to US stock market data. Tse and Tsui (2002) determined the APGARCH model. Brooks, Faff, McKenzie, and Mitchell (2000) show the leverage effect in the model and the usefulness of including a free power term.

In this study, we aim to analyze the volatility of stock return behavior of Istanbul Stock Exchange ISE 100 Index for the 23.10.1987–22.02.2008 period. The study will compare and combine a general class of Autoregressive Conditional Heteroscedasticity (G)ARCH family models (Bollerslev, 1986; Engle, 1982; Nelson, 1991) of GARCH, EGARCH, GJR-GARCH, TGARCH, NGARCH, SAGARCH, PGARCH, APGARCH, NPGARCH with Artificial Neural Network models and discuss and compare them in accordance with their forecast capabilities. The study is organized into the following sections: *Theory* and *Data Characteristics*, *Econometric Results* and lastly, *Conclusion*.

2. Theory

ANN models have found many important applications especially in the field of financial modeling and forecasting in the recent literature. Among many, Abhyankar, Copeland, and Wang (1997), Castiglione (2001), Freisleben (1992), Kim and Chun

* Corresponding author.
E-mail addresses: bildiri@yildiz.edu.tr (M. Bildirici), oersin@yeditepe.edu.tr (Ö.Ö. Ersin).
[1] Tel.: +90 216 578 0956.

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.09.051

(1998), Liu and Yao (2001), Phua, Zhu, and Koh (2003), Refenes, Zapranis, and Francies (1994) and Zhu, Wang, Xu, and Li (2007) applied nonlinear forecast methods to financial markets. Further, Black and McMillan (2004), Donaldson and Kamstra (1997), Jasic and Wood (2004), Kanas (2001), Kanas and Yannopoulos (2001) and Shively (2003) applied regarding the stock prices; whereas, Dunis and Huang (2002) and Hamid and Iqbal (2004) provide insightful applications regarding the stock volatility.

Dutta and Shektar (1988) and Surkan and Xingren (1991) applied ANN models to rate bonds, Kamijo and Tanigava (1990) to stock prices, Tam and Kiang (1992) to forecasting bank failures, Hutchinson, Lo, and Poggio (1994) to forecasting option prices and as an assessment to hedging, Grudnitsky and Osburn (1993) to gold futures prices and S&P 500 forecasting, Leung, Chen, and Daouk (2000) to FOREX forecasting, Barr and Mani (1994) to investment management, Saad, Prokhorov, and Wunsch (1998) to stock trend predicton, Udo (1993) and Wilson and Sharda (1997) to bankruptcy classification, Kaski and Kohonen (1996) to economic rating and Kong and Martin (1995) and Thiesing, Middleberg, and Vornberger (1995) to sales forecasting.

In the study, we investigated the conditional volatility by applying a hybrid modeling approach that combines various GARCH family models. Furthermore, we used the Back Propagation Artificial Neural Network model for forecasting.

2.1. Multilayer perceptron models

Neural networks represent an important class of nonlinear approximation and classification models relating a set of input variables to one or more output target variables that contain nonlinear latent units to achieve significant flexibility (Kay and Titterington, 1999: p. 2). Rosenblatt (1962) discussed the single hidden layer feedforward neural networks and called ANN models with threshold activation functions as the perceptron. The details of perceptron are analyzed in Block (1962). A similar model had been introduced and discussed in detail by Widrow and Hoff (1960) called ADALINES (ADaptive LINear Elements) that refers to a single hidden unit with threshold nonlinearity. See Bishop (1995) and Widrow and Lehr (1990) for detail.

Rosenblatt (1962) perceptron is stated as

$$o_t = f\left(\sum_{j=0}^{s} w_j \phi_j(x)\right), \quad j = 0, 1, \ldots, s \tag{1}$$

where φ_j is the activation function given in vector form $\varphi_0, \ldots, \varphi_s$; $\mathbf{x}$ is the input variable matrix, w_j is the weight, f is the output function and o is the output of the neuron. The perceptron model of Rosenblatt uses the threshold activation function of the form, $f(a) = \begin{cases} -1 & \text{if} \quad a < 0 \\ +1 & \text{if} \quad a \geq 0 \end{cases}$ The function is bounded between $[-1,+1]$.

In the NN literature, several activation functions are applied. The threshold function or step function can be expressed to be bounded between $[0,1]$ having the form, $f(a) = \begin{cases} 0 & \text{if} \quad a < 0 \\ 1 & \text{if} \quad a \geq 0 \end{cases}$ expressed similar to the *step* function. A commonly applied form of the output function f is the logistic form to achieve a bounded, continuous, sigmoidal and twice differentiable in the log-sigmoid form, $f(a) = 1/(1 + e^{-a})$. The tanh activation function is employed to achieve practical advantage in certain applications and bounded between $[-1,+1]$ similar to Rosenblatt threshold which is defined as $f(a) = \tanh = \frac{e^a - e^{-a}}{e^a + e^{-a}}$, continuous and sigmoid (Bishop, 1995: p. 98).

The multilayer perceptron model is achieved by a weighted linear combination of the d input values in the form

$$a_j = \sum_{i=0}^{d} w_{ji}^{(1)} x_i \tag{2}$$

By employing an activation function $g(\cdot)$; $z_j = g(a_j)$. The network is achieved by associating the activations of the hidden units to the second layer. For each output unit k

$$a_j = \sum_{i=0}^{M} w_{kj}^{(2)} z_j \tag{3}$$

using a nonlinear activation function, $y_k \cong g(a_k)$. The complete function of MLP can be expressed by combining (2) and (3) to give

$$y_k = \tilde{g}\left(\sum_{j=0}^{M} w_{kj}^{(2)} g\left(\sum w_{ji}^{(1)} x_i\right)\right) \tag{4}$$

If the output function is taken linear, $\tilde{g}(a) = a$, the model reduces to

$$y_k = \sum_{j=0}^{M} w_{kj}^{(2)} g\left(\sum w_{ji}^{(1)} x_i\right) \tag{5}$$

There are several training methods for Neural Networks. In NN literature, the most common method of model estimation is Backpropagation (Rumelhart, Rubin, Golden, & Chanvin, 1995), where parameters are updated so that the tuning of parameters is in accordance with the quadratic loss function; hence, the resulting *weight decay* method aims to estimate weights iteratively to achieve the lowest error. Alternative methods include Genetic Algorithms for nonlinear optimization and training of Neural Networks (Goldberg, 1989). Other second-order derivative based optimization algorithms are the Conjugate Gradient Descent, Quasi-Newton, Quick Propagation, Delta-Bar-Delta and Levenberg–Marquardt (Marquardt, 1963), which are faster and effective algorithms but more exposed to over-fitting, an important phenomenon in Neural Networks. To overcome over-learning, we applied two methods. First, the *early stopping* and second, the *algorithm cooperation*. The early stopping approach aims to stopping the training once the selection error starts to rise. The ANN models are retrained with conjugate gradient descent algorithm after the training with backpropagation.[2]

2.2. NN-GARCH models

In addition, we will investigate the negative and positive impacts of shocks on volatility by using the APGARCH model that utilizes the power parameter and then apply if for forecasting. Third, we will investigate the forecast efficiency of GARCH, EGARCH, TGARCH, GJR-GARCH, SAGARCH, PGARCH, NPGARCH, and APGARCH models.

In NN-GARCH model, the learning process is applied to GARCH(1,1) process by including the input variables defined as $\sigma_{t-1}^2 = \gamma_1 \sigma_{t-1}^2$ and $\varepsilon_{t-1}^2 = \beta_1 \varepsilon_{t-1}^2$.

On the other hand, in the NN-EGARCH model, $\ln \sigma_{t-1}^2 = \gamma_1 \ln \sigma_{t-1}^2$ and for the leverage effect, $L(leverage) = \delta \frac{\varepsilon_{t-1}}{\sigma_{t-1}}$ so that $LE(leverage\_effect) = \gamma \left| \frac{\varepsilon_{t-1}}{\sigma_{t-1}} - \sqrt{\frac{2}{\pi}} \right|$ was obtained by including β, γ and δ parameters to define L and LE.

[2] The methodology we followed has two stages to overcome local minima:

(I) Back propogation
  (i) The sample is divided into training/test/selection subsets.
  (ii) Both training and test samples are trained with large steps.
  (iii) The minimization criteria (RMSE) is checked every epoch.
  (iv) Training is early stopped if the RMSE starts to increase for the test subsample even if the same does not hold for the training sample (generalization principle).
(II) Conjugate Gradient Descent
  (v) Training is repeated with conjugate gradient descent (stages i.–iv.) but with small steps (low learning rate in ii.).Model is accepted if the global minimum is reached (see Patterson, 1996; Haykin, 1994; Fausett, 1994).