



Operations Research Society in  
Bosnia and Herzegovina

Southeast Europe Journal of Soft Computing

Available online: <http://scjournal.ius.edu.ba>



IUS Soft Computing  
Research Group

## Forecasting Conditional Variance of S&P100 Returns Using Feedforward and Recurrent Neural Networks

Sadi Fadda

Mehmet Can

International University of Sarajevo,  
Faculty of Engineering and Natural Sciences,  
Hrasnicka Cesta 15, Ilidža 71210 Sarajevo,  
Bosnia and Herzegovina  
[sfadda@ius.edu.ba](mailto:sfadda@ius.edu.ba)  
[mcan@ius.edu.ba](mailto:mcan@ius.edu.ba)

### Article Info

#### Article history:

Article received on 17 Jan. 2017

Received in revised form 17 Feb.2017

#### Keywords:

Key words: conditional variance,  
GARCH, multivariate regression,  
ANN, volatility models

### Abstract

It is shown that time series about financial market variables are highly nonlinearly dependent on time. Fluctuations or volatility of returns on assets is one of them. Portfolio managers, option traders and market makers are all interested in volatility forecasting in order to get higher profits and less risky positions. The nonlinear dependence on time is very complex and parametric approaches, and linear models fail. Therefore as nonparametric tools artificial neural networks (ANNs) are candidates to deal with the volatility and/or return forecasting problems. On the other hand, based on the fact that volatility is time varying and that periods of high volatility tend to cluster, the most popular models in modeling volatility are GARCH type models because they can account excess kurtosis and asymmetric effects of financial time series. A standard GARCH(1,1) model usually indicates high persistence in the conditional variance, which may originate from structural changes. Hence it is natural that artificial neural networks (ANN) will be constructed to capture the nonlinear relationship between past return innovations and conditional variance which may be missed by linear regression models. First a usual feedforward, back propagation network is used. The structure of the return data makes FFANN difficult to converge. To overcome this difficulty a neural network with appropriate recurrent connections in the context of nonlinear ARMA models are used. These are the Jordan neural networks (JNN). Then Elman recurrent networks (ENN) and a mixture of the two (EJNN) are also used. The data set consists of returns of the S&P100 index daily closing prices obtained from the S&P100 website. The results indicate that the selected JNN(1,1,1) model has superior performances compared to the standard GARCH(1,1) model. The contribution of this paper can be seen in determining the appropriate NN that is comparable to the standard GARCH(1,1) model and its application in forecasting conditional variance of stock returns. Moreover, from the econometric perspective, NN models are used as a semi-parametric method that combines flexibility of nonparametric methods and the interpretability of parameters of parametric methods.

## 1. INTRODUCTION

For several decades market makers, portfolio managers, and option traders are all interested in forecasting return fluctuations in order to maximize their profits, and minimize their risks. Volatility is the name given to these return fluctuations. The relation between the market parameters and volatility is a very complex one. Black-Scholes pricing formula's in their analytical simplicity determine the price of a European call options  $c$ , and puts  $p$  on a non-dividend paying asset by

$$c = SN(d_1) - Ke^{-rT}N(d_2) \quad (1)$$

$$p = -SN(-d_1) + Ke^{-rT}N(-d_2) \quad (2)$$

with

$$d_1 = \left( \ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T \right) / \sigma\sqrt{T},$$

and

$$d_2 = d_1 - \sigma\sqrt{T} \quad (3)$$

where  $N$  is the cumulative normal distribution,  $S$  is the price of the underlying security,  $K$  is the strike price,  $r$  is the prevailing risk-free interest rate,  $T$  is the time-to-maturity and  $\sigma$  is the volatility of the underlying asset. This model is helpful when some assumptions are fulfilled, and equations (1) - (3) do contain neither preferences of individuals nor the preferences of the aggregate market (Hull, 1993). For this reason nonparametric methods like artificial neural networks may be the most appropriate device to infer volatility from market observations.

### *ANNs to Infer Volatility from Market Observations*

Forecasting (Sharda, 1994) in financial markets (Kuan, and Liu 1995; Zhang et. al. 1997; Bekiros, and Georgoutsos 2004; Chen, and Lai 2008; Samur, and Temur 2009; Wang et. al., 2011; Arneric et. al., 2014) is one of the major application areas of ANNs. They are valuable for forecasting tasks since they are data-driven self-adaptive methods departing from the traditional model-based methods. As in the case of prices, returns, and the volatility of the market, while underlying relationships between market parameters are unknown or hard to describe, ANNs learn from examples, capture inherent functional relationships in the data, and generalize these relationships to the untouched test data.

For this reason ANNs are well suited for problems whose solutions require analytic relations like Black-Scholes formulas. When these relations are difficult to specify, and when there are enough data or observations, ANNs are

implemented as multivariate nonlinear nonparametric statistical methods (White, 1989; Ripley, 1993; Cheng and Titterton, 1994). In some situations ANNs provide the only feasible solutions to real-world problems.

It has also been shown that a network can approximate any smooth function to any desired accuracy, hence they are universal functional approximators (Irie and Miyake, 1988; Hornik et al., 1989; Cybenko, 1989; Funahashi, 1989; Hornik, 1991, 1993). ANNs have more general and flexible functional forms than the traditional statistical methods can effectively deal with. Forecasting models assume that there exists an underlying relationship between the inputs in the forms of the past values of the time series, regressors, and other relevant independent variables, and outputs in the form of future values, and dependent variables. Traditional statistical forecasting models assume that the underlying relations are linear which cannot capture the complexity of the real system.

There is no reason to assume a priori that a particular realization of a given time series, or an output data is generated by a linear process. Although in 1990's, several nonlinear time series models are created, success of these nonlinear models were limited since the explicit relationship for the data series at hand has to be hypothesized with little knowledge of the underlying mechanism.

On the other hand, artificial neural networks, as nonlinear data-driven approaches, are capable of performing nonlinear modeling without a priori knowledge about the relationships between input and output variables.

The first application the idea of using ANNs for forecasting dates back to 1964. Hu (1964), in his thesis, uses the Widrow's adaptive linear network to weather forecasting. Due to the lack of a training algorithm for general multi-layer networks at the time, the research was quite limited. Backpropagation algorithm was introduced (Rumelhart et al., 1986b) after that there had been much development in the use of ANNs for forecasting. Werbos (1974), (1988) first formulates the backpropagation and finds that ANNs trained with backpropagation outperform the traditional statistical methods such as regression and Box-Jenkins approaches. Lapedes and Farber (1987) conduct a simulated study and conclude that ANNs can be used for modeling and forecasting nonlinear time series. Weigend et al. (1990), (1992); Cottrell et al. (1995) address the issue of network structure for forecasting real-world time series. Tang et al. (1991), Sharda and Patil (1992), and Tang and Fishwick (1993), among others, report results of several forecasting comparisons between Box-Jenkins and ANN models. In a recent forecasting competition organized by Weigend and Gershenfeld (1993) through the Santa Fe Institute, winners of each set of data used ANN models (Gershenfeld and Weigend, 1993).

The literature on ANNs for forecasting is vast and growing. Marquez et al. (1992) and Hill et al. (1994) review the literature comparing ANNs with statistical models in time series forecasting and regression-based forecasting. However, their review focuses on the relative performance of ANNs and includes only a few papers. In G. Zhang et al (1998), authors attempted to provide a more comprehensive review of the status of research in this area in their time.

## 2. A BRIEF NOTE ON ANNS

This brief presentation of artificial neural networks will focus on a particular structure of ANNs, multi-layer feedforward networks, which is the most popular and widely-used network paradigm in many applications including forecasting volatilities and prices in markets. For a general introductory account of ANNs, readers are referred to Wasserman (1989); Hertz et al. (1991); Smith (1993). Rumelhart et al. (1986a), (1986b), (1994), (1995); Lippmann (1987); Hinton (1992); Hammerstrom (1993); Haykin 1999 illustrate the basic ideas in ANNs.

Hush and Horne (1993) summarize some theoretical developments in ANNs since Lippmann (1987) tutorial article. Masson and Wang (1990) give a detailed description of five different network models. Wilson and Sharda (1992) present a review of applications of ANNs in the business setting. Sharda (1994) provides an application bibliography for researchers in Management Science/Operations Research. A bibliography of neural network business applications research is also given by Wong et al. (1995). Kuan and White (1994) review the ANN models used by economists and econometricians and establish several theoretical frames for ANN learning. Cheng and Titterington (1994) make a detailed analysis and comparison of ANNs paradigms with traditional statistical methods.

Basic structures of artificial neural networks, originally developed to mimic the human brain, are composed of a number of interconnected simple processing elements called neurons or nodes. Each node receives an input signal which is the total "information" from other nodes or external stimuli. The node processes incoming data locally through an activation function and produces a transformed output signal to other nodes or external outputs. Although each individual neuron implements its function rather slowly and imperfectly, collectively a network can perform a surprising number of tasks quite efficiently (Reilly and Cooper, 1990). This information processing characteristic makes ANNs a powerful computational device and able to learn from examples and then to generalize to examples never before seen.

Many different ANN models have been proposed since 1980s. Perhaps the most influential models are the multi-layer perceptrons (MLP), Hopfield networks, and Kohonen's self organizing networks. Hopfield (1982) proposes a recurrent neural network which works as an associative memory. An associative memory can recall an example from a partial or distorted version. Hopfield networks are non-layered with complete interconnectivity between nodes. The outputs of the

network are not necessarily the functions of the inputs. Rather they are stable states of an iterative process.

## 3. MULTI LAYER PERCEPTRONS FOR FORECASTING

Especially in forecasting the MLP networks are used because of their inherent capability of arbitrary input-output mapping. Other types of ANNs are radial-basis functions networks (Park and Sandberg, 1991, 1993; Chng et al., 1996), ridge polynomial networks (Shin and Ghosh, 1995), and wavelet networks (Zhang and Benveniste, 1992; Delyon et al., 1995) are also very useful in some applications due to their function approximating ability.

An MLP is composed of several layers of nodes. The lowest layer is an input layer where external information is received. The last layer is an output layer where the problem solution is obtained. hidden layers separate the input layer and output layer. The nodes in adjacent layers are usually fully connected from a lower layer to a higher layer. Fig. 1 gives an example of a fully connected MLP with one hidden layer.

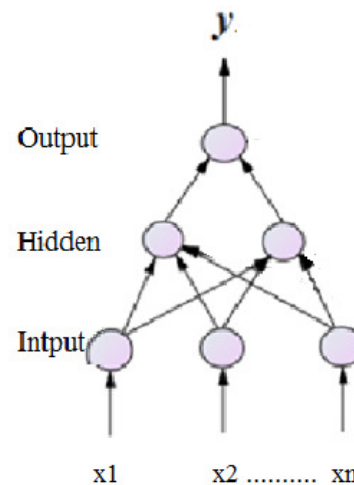


Fig. 1. A typical feedforward neural network (MLP).

For a forecasting problem, the inputs to an ANN are usually the independent variables. The functional relationship estimated by the ANN can be written as

$$y = f(\mathbf{x}), \quad \mathbf{x} = (x_1, x_2, \dots, x_p) \quad (4)$$

where  $(x_1, x_2, \dots, x_p)$  is the vector of  $p$  independent variables and  $y$  is a dependent variable. In this sense, the neural network is functionally equivalent to a nonlinear regression model. On the other hand, for an extrapolative or time series forecasting problem, the inputs are typically the past observations of the data series and the output is the future value. The ANN performs the following function mapping

$$y_{t+1} = f(\mathbf{y}), \quad \mathbf{y} = (y_t, y_{t-1}, \dots, y_{t-p}) \quad (5)$$

where  $y_t$  is the observation at time  $t$ . Thus the ANN is equivalent to the nonlinear autoregressive model for time series forecasting problems.

### 3.1 Multi-Layer Feed-Forward Networks (FNN)

Multi-layer feed-forward networks (FNN) which forward information from the input layer to the output layer through a number of hidden layers. Neurons in a current layer connect to a neuron of the subsequent layer by weights and an activation function (Figure 1.a). In order to modify weights, the backpropagation (BP) learning algorithm is adopted. This iterative mechanism works by feeding the error back through the network. The synaptic weights are iteratively updated until there is no improvement in the error function. This process requires the derivative of the error function with respect to the network weights. The sum of squared error  $E$  is the conventional least square objective function in a NN, defined as:

$$\min_w Error \quad (6)$$

where

$$Error = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2 \quad (7)$$

where  $y_t$  denote observed values of time series and  $\hat{y}_t$  are fitted outputs.

In forecasting time series, in general, alongside feed-forward neural networks there are a second type of ANNs which are called recurrent neural networks.

FNNs in Figure 1 are highly non-parsimonious requiring an infinite amount of past observations as inputs to achieve the same accuracy in forecasting comparing to RNN. Moreover, in practical applications, recurrent neural networks provide a significantly better prediction than a feed-forward network.

### 3.2 Recurrent Neural Networks (RNN)

Financial time series mostly dependent nonlinearly on time and hence recurrent neural networks (RNN) are particularly useful. They are constructed by taking a feedforward network and adding feedback connections from output and/or hidden layers to input layers. The standard backpropagation algorithm also trains these networks conditional that patterns must always be presented in time sequential order. The one difference in the structure is that there are extra neurons in the input layer that is connected to the hidden layer and/or output layer just like the other input neurons. These extra neurons hold the contents of one of the layers as it existed when the previous pattern was trained. In this way, the network takes into account previous knowledge it has about previous inputs. These extra neurons are called the context unit and it represents the network's long-term memory (Balkin 1997).

There are three types of RNNs: Jordan, Elman, and Jordan/Elman recurrent networks. A Jordan neural network

(JNN) has additional neurons in the input layer, which are fed back from output layer. While an Elman neural network (ENN) has additional neurons in the input layer, which is fed back from hidden layer. The mixture of the two, Jordan/Elman recurrent networks (JENN) has additional neurons in the input layer, which is fed back from hidden layer, and output layer.

#### 3.2.1 Jordan Recurrent Neural Networks (JNN)

A Jordan neural network (JNN) has several feedback connections from the output layer to the input layer. The input layer has additional neurons, which are fed back from the output layer. Econometric interpretation of such feedback connection lies in the fact that in this way the model is expanded by lagged error terms (Figure 2).

Using JNN, the problem of convergence can be solved by a more complex model. Although this network is more complicated than a multi-layer feed-forward network, the characteristics of feeding back data to the network are similar to a GARCH model, having the previous variance in current forecasts (Dechpichai, 2010).

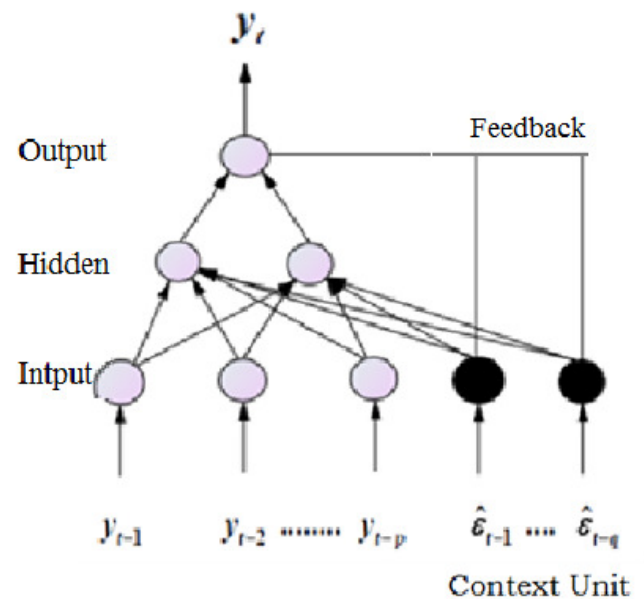


Figure 2. JNN with a single hidden layer representing a nonlinear ARMA(p,q) model

#### 3.2.2 Elman Recurrent Neural Network (ENN)

Although properties of feedforward networks make them attractive econometric tools in nonparametric applications, in a dynamic context, it is natural to include lagged dependent variables as explanatory variables in a feedforward network to capture dynamics. Analogous to the problem of determining the order of an autoregression, this approach suffers the drawback that the correct number of lags needed is unknown. Hence, the lagged dependent variables in a network may not be enough to characterize the behavior of  $y$  in some applications. To overcome this deficiency, various recurrent networks, i.e. networks with feedbacks, have been proposed.

A recurrent network has a richer dynamic structure and is similar to a linear time series model with moving average terms. A recurrent network may capture more dynamic characteristics of  $y$ , than does a feedforward network.

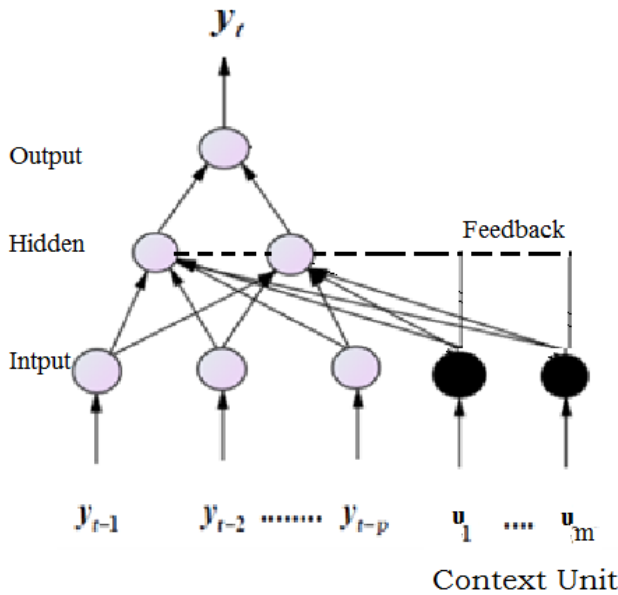


Figure 3. A simple Elman (1990) network with hidden-unit activations feedback to the input layer with delay and serve to 'memorize' the past information.

3.2.2 Jordan/Elman Recurrent Neural Network (JENN)

The mixture of the two, Jordan/Elman recurrent networks (JENN) has additional neurons in the input layer, which is fed back from hidden layer, and output layer.

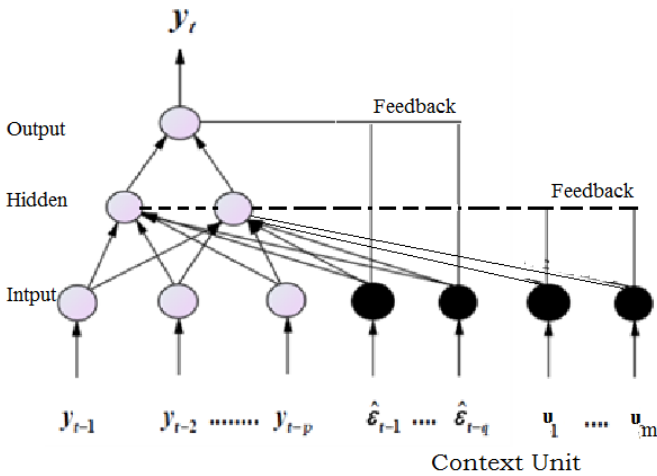


Figure 4. A simple Jordan/Elman network with output and hidden unit activations feedback to the input layer with delay and serve to remember the past information.

4. THE CONDITIONAL VARIANCE PROCESS

The most popular models in modeling volatility are generalized autoregressive conditional heteroskedasticity (GARCH) type models which can account for excess kurtosis and asymmetric effects of high frequency data, time varying

volatility and volatility clustering. The first autoregressive conditional heteroscedasticity model (ARCH) was proposed by Engle (Engle, 1982) who won a Nobel Prize in 2003 for his contribution to modeling volatility. The model was extended by Bollerslev (Bollerslev, 1986) by its generalized version (GARCH). However, the standard GARCH(1,1) model usually indicates high persistence in the conditional variance, which may originate from structural changes in the variance process. Hence, the estimates of a GARCH model suffer from a substantial upward bias in the persistence parameters. In addition, it is often difficult to predict volatility using traditional GARCH models because the series is affected by different characteristics: non-stationary behavior, high persistence in the conditional variance and nonlinearity. Due to practical limitations of these models, different approaches have been proposed in the literature, some of which are based on artificial neural networks (ANN).

Neural networks are a valuable tool for modeling and prediction of time series in general (Balkin, 1997; Ghiassi, et., al, 2005; Kuan, and White 2007). Time dependence of most financial time series is nonlinear, that is, current values of the time series are nonlinearly conditioned on the information set consisting of all relevant information up to and including the period (Aminian, et. al., 2006; Gonzales, 2000; Hwarng, 2001; Zhang, 2003). The feed-forward neural networks (FNN), i.e., multilayer perceptron, are most popular and commonly used. They are criticized in the literature for the high number of parameters to estimate and they are sensitive to over fitting (Franses, and van Dijk 2003; Lawrence, et. al., 1997). A property added to recurrent networks feed-forward neural networks to allow feed-back form a cycle within the network architecture which can be analyzed as a nonlinear extension of traditional linear models, such as auto regressive moving average model ARMA. Recurrent neural networks (RNN) preserve long memory of the series and allow adequate forecasts of volatility with a smaller number of parameters to estimate (Balkin 1997; Dechpichai 2010). Therefore, recurrent neural networks are more appropriate than feed-forward neural networks in forecasting nonlinear time series.

4.1 ARCH

ARCH stands for autoregressive conditional heteroskedasticity. Changes in the scale of a variable give us the word heteroskedastic. A scale parameter is a standard deviation or a variance and the variable of interest here is the return from an asset. The variance of a return, conditional on the information in previous returns, is found to depend on this information. Engle (1982) defined a stochastic process whose variables have conditional mean zero and conditional variance given by a linear function of previous squared variables. The squared variables follow an autoregressive process in his pioneering and influential research.

Subsequent research has provided many alternative functions that specify the conditional variance of a variable at time  $t$  as a function of information known at time  $t-1$ . For any specification that also gives us the conditional density function at time  $t$ , we will call the stochastic process an ARCH model provided

that the standardized residuals of the process are independent and identically distributed. In particular, we allow the conditional mean to vary though time; unlike some authors who prefer to restrict the acronym ARCH to processes whose conditional means are always zero.

There is a multitude of ARCH specifications and many of them have their own acronyms, the best known being generalized ARCH, GARCH, from Bollerslev (1986) and exponential, generalized ARCH, EGARCH from Nelson (1991). The popularity of the models can be explained by their compatibility with the major stylized facts for asset returns, by efficient methods for estimating model parameters and by the availability of useful volatility forecasts. The specification of conditional densities provides the likelihood function for a dataset, which can be maximized to give optimal parameter estimates. Several software packages will maximize the likelihood function and thus estimation of an ARCH model is now a routine activity. Likelihood theory allows specifications to be compared and choices to be made from among the many functions that have been proposed for conditional variances.

The literature on ARCH models is considerable. Bollerslev, Chou, and Kroner (1992) provide a review of theory and ten years of empirical evidence for financial markets, which covers applications to equities, foreign exchange, and interest rates. The authors describe an impressive number of interesting studies without requiring the reader to understand many equations. The subsequent survey by Bollerslev, Engle, and Nelson (1994) is suitable for those readers who wish to see more theory than is presented in this book. It also contains detailed examples of the specification of conditional densities for daily returns from US equity indices, going back as far as 1885. A neglected precursor to the ARCH literature is a working paper by Rosenberg (1972), while early likelihood estimates of an integrated ARCH specification can be found in Taylor and Kingsman (1979).

Some basic ARCH models and the general ARCH framework are described here. There are many applications of ARCH methods in finance research, including investigations into asset pricing, hedging, and microstructure effects. Option pricing, volatility forecasting, and density estimation are also important application areas.

#### 4.2 ARMA(p,q)

The structure of RNN representing nonlinear ARMA(p,q) is comparable to the GARCH(p,q) model with appropriate lag selection. JNN can be represented as

$$\hat{y}_t = \phi \left( \alpha_0 + \sum_{i=1}^q \alpha_i \psi \left( \beta_{0i} + \sum_{j=1}^p \beta_{ij} y_{t-j} + \gamma_i \varepsilon_{t-1} \right) \right) \quad (8)$$

where  $t$  is a time index,  $\hat{y}_t$  is the output vector,  $y_{t-j}$  is the input matrix with  $t-j$  time lags,  $\phi$  and  $\psi$  are activation functions.  $\alpha_0$  denotes the constant term in the output layer,  $\beta_{0i}$  denotes the constant term in the hidden layer. The weights  $\beta_{ij}$  and  $\alpha_i$  denote the weights for the connections between the inputs and hidden neurons and between the hidden neurons and the output,  $\gamma_i$  denotes the weight for connections between

the context unit and hidden neurons and  $\varepsilon_{t-1}$  denotes the difference between observed values of time series and fitted values of time series from the previous period. JNN with  $p$  inputs,  $q$  hidden neurons and one target unit has the abbreviation JNN(p,q,1) (Balkin, 1997).

To define appropriate JNN based on GARCH innovations we concentrate on a nonlinear version of ARMA(1,1) originated from equation (7):

$$\begin{aligned} \hat{\varepsilon}_t^2 &= \phi \left( \alpha_0 + \alpha_1 \psi \left( \beta_0 + \beta_1 e_{t-1}^2 + \gamma \hat{v}_{t-1} \right) \right), \\ \hat{v}_{t-1} &= e_{t-1}^2 - \hat{\varepsilon}_{t-1}^2 \end{aligned} \quad (9)$$

In equation (8), function  $\phi$  is the linear activation function in the output layer and  $\psi$  is a nonlinear activation function in the hidden layer, which processes information from the input layer to the output layer. A recurrent neural network in (8) assumes one neuron in a single hidden layer network, a nonlinear activation function in a hidden layer, i.e., a sigmoid, a linear activation function in the output layer, one squared innovations with one time lag as input, one current squared innovations as target, while the error term  $\hat{\varepsilon}_{t-1}$  is added through feed-back connection from the output layer to the input layer.

The parameters,  $\alpha_1$ ,  $\beta_1$  and  $\gamma$  are the called weights, while  $\alpha_0$  and  $\beta_0$  are the biases, that is constant terms of the hidden layer and output layer, respectively. These parameters are estimated from a training sample by minimizing the sum of the squares residuals of innovations using the gradient descent procedure known as “backpropagation” (BP). The network, which takes into account all assumptions above, can be presented as a Jordan neural network with memory.

A Jordan net without memory “remembers” only the output from the previous time step. A Jordan net with memory remembers past values as an exponentially decaying weighted average of inputs, that is no outputs are forgotten; they just diminish away. The context unit is called a long-term memory unit in a RNN since it remembers past events. According to equation (7), there is only one context unit which accounts for a moving average structure of a time series.

Many researchers rely on automatically chosen ANN provided by various software tools. But in this paper since ANNs are used as semi-parametric methods that combine flexibility of nonparametric models, with less restricted assumptions and the interpretability of synaptic weights, authors will rely on ANN architectures written in MATHEMATICA programming language by them.

#### 4.3 ARCH(1)

The simplest example of an ARCH process is the ARCH(1) specification presented by Engle (1982). The distribution of the return for period  $t$ , conditional on all previous returns, is normal with constant mean  $\mu$  and time-varying conditional variance  $h_t$  defined by

$$h_t = \omega + \alpha(r_{t-1} - \mu)^2. \quad (10)$$

The volatility parameters are  $\omega > 0$  and  $\alpha > 0$  (Taylor 2005). The volatility of the return in period  $t$  then depends solely on the previous return. Either a large positive or a large negative return in period  $t - 1$  implies higher than average volatility in the next period when  $\alpha$  is positive; conversely, returns near the mean level  $\mu$  imply lower than average future volatility.

The residual at time  $t$  is

$$e_t = r_t - \mu \quad (11)$$

and the forecast error when predicting squared residuals is

$$v_t = e_t^2 - h_t \quad (12)$$

These forecast errors are uncorrelated. Replacing  $h_t$  in (12) by  $e_t^2 - v_t$  gives

$$e_t^2 = \omega + \alpha e_{t-1}^2 + v_t \quad (13)$$

and hence squared residuals follow an AR(1) process. This explains the AR part of the ARCH acronym.

The ARCH(1) model is stationary when  $\alpha < 1$ . Any satisfactory AR(p) process for squared residuals must have a high order  $p$ . A natural alternative is an ARMA process and this explains the interest in GARCH models.

#### 4.4 GARCH(1, 1)

The GARCH(1,1) model with conditional normal distributions is the most popular ARCH specification in empirical research, particularly when modeling daily returns. The letter “G” appears in the acronym of this model because it is generalized from ARCH(1) by including a lagged variance term in the conditional variance equation. The popularity of GARCH(1, 1) may be explained by three observations. First, the model has only four parameters and these can be estimated easily. Second, it provides an explanation of the major stylized facts for daily returns. Third, it is often found that the volatility forecasts from this specification have similar accuracy to forecasts from more complicated specifications. Initially, we assume conditional normal distributions following Bollerslev (1986) and Taylor (1986), who independently defined and derived properties of the GARCH(1, 1) model (Taylor 2005).

The distribution of the return for period  $t$ , conditional on all previous returns, is normal with mean  $\mu$ , and variance  $h_t$  where

$$h_t = \omega + \alpha(r_{t-1} - \mu)^2 + \beta h_{t-1}. \quad (14)$$

There are four parameters, namely  $\mu$ ,  $\alpha$ ,  $\beta$ , and  $\omega$ . The constraints  $\omega > 0$ ,  $\alpha > 0$ , and  $\beta > 0$  are required to ensure that the conditional variance is never negative. The possibility  $\alpha = 0$  is of no interest and so we assume  $\alpha$  is positive. The model is styled GARCH(1, 1) because one previous squared residual and one previous value of the conditional variance are used to define the conditional variance for period  $t$ . Calculations of

conditional variances from the recursive definition (14) are straightforward, providing an initial value is available for the first time period.

The major properties of a GARCH(1,1) stochastic process can be summarized as follows. The process is stationary if  $\alpha + \beta < 1$  and then

- The unconditional variance is finite;
- The unconditional kurtosis always exceeds three and can be infinite;
- The correlation between the returns  $r_t$  and  $r_{t+\tau}$  is zero for all  $\tau > 0$ ; and
- The correlation between the squared residual  $s_t = (r_t - \mu)^2$  and  $s_{t+\tau}$  is positive for all  $\tau > 0$  and equals  $C(\alpha + \beta)^\tau$ , with  $C$  positive and determined by both  $\alpha$  and  $\beta$ .

## 5. RESEARCH METHODOLOGY AND RESULTS

The data set consists of returns of the S&P100 index daily closing prices obtained from the S&P100 index website in the period from September 6, 2005 until June 7, 2016. However, for the long term forecasting for the purpose of this research, the first 1000 trade days of the data is neglected, and the remaining data is divided into two parts: the in-the-sample part consists of 1000 observations in the period from August 26, 2009 until August 14, 2013 which is used for the training and the estimation of parameters in the GARCH(p,q), JNN(p,q,r), ENN(p,q,r) and the EJNN(p,q,r,s) models; and the out-of-the-sample part which consists of the 500 observations in period from August 14, 2013 to August 10, 2015 which is used for the two years ahead forecasting purposes. To show that ANNs predict better for shorter term forecasts we also took in-the-sample part which consists of 500 observations in the period from August 8, 2011 until August 14, 2013 which is used for the training and the estimation of parameters, 250 observations in period from August 14, 2013 to August 8, 2014 are used for the one year ahead forecasting. We have seen that the forecasting mean absolute percentage error MPE reduces below 10%.

### 5.1. Forecasting by a Typical Feedforward Back Propagation Neural Network (MLP).

Conditional volatility forecasted by GARCH(p,q) with several values of  $p$ , and  $q$ . A typical feedforward back propagation neural network (MLP) is used with 5-10-20 neurons in one hidden layer (Figure 1). In Figure 5 the result is given for GARCH(10,10), target values are shown in blue, forecasted values are in purple. Absolute percentage error is 10.25%. To get these accuracies, 1000 trading days are used for training, and neural network successfully predicted the volatility of the next 250 trading days which is around a year.

To forecast conditional volatilities by the model GARCH(p,q) typical feedforward back propagation neural networks (MLP) are used with  $p+q$  neurons in one hidden layer. Authors wrote the codes for MLPs, and they are run with optimum experimental learning rates. Learning rate is the rate used to update the synaptic weights. When the learning rate is too

large, the network may diverge. Therefore, an appropriate learning rate should be chosen from the interval [0,1]. In this paper, an adaptive approach is used to decide about the learning rates.

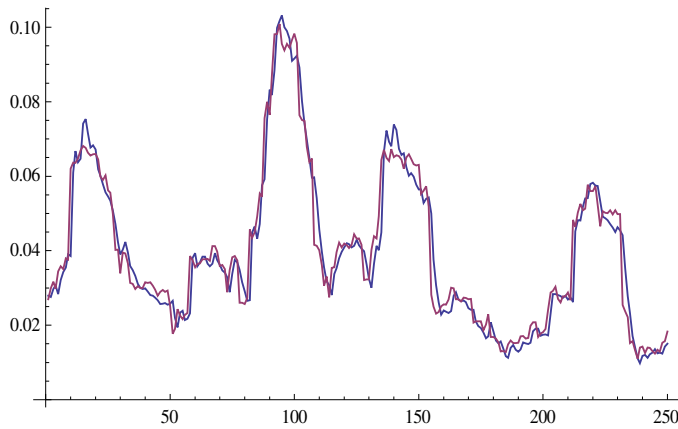


Figure 5. One year ahead forecast of conditional volatility by the model GARCH(10,10) using a typical feedforward back propagation neural network (MLP) with 20 neurons in one hidden layer.. Target values blue, forecasted values purple. Absolute percentage error 10.25%.

In this paper 3 MLPs are chosen to compute the conditional volatilities by the model GARCH(p,q). The results are presented in Table 1. Mean squared errors (MSE) of the training phase, and mean percentage absolute error of testing are calculated and presented for the train and for the out-of-sample period.

Table 1. Mean absolute percentage errors (MPE) in predicting volatilities in four years and one year future by various GARCH(p,q) models regarding different architectures.

Forecast	GARCH(5,5)	GARCH(10,10)	GARCH(20,20)
2 years	16.40	16.86	16.19
1 year	13.90	10.25	8.65

### 5.2 Forecasting Conditional Volatility, Using JNNs

In order to estimate the conditional volatility, the model GARCH(p,q) is used with r context neurons that are feedback from output. Therefore they are of JNN(p,q,r) model with several values of p, q, and r (Figure 2). In the sample, log returns of the S&P100 index daily closing prices are used for the calculation. The input for JNN is squared mean corrected returns with several time lags. In Figure 6, target values are shown in blue, forecasted values are in purple. To get these accuracies, 1000 trading days are used for training, and neural network successfully predicted the volatility of the next 500 trading days with the given accuracies.

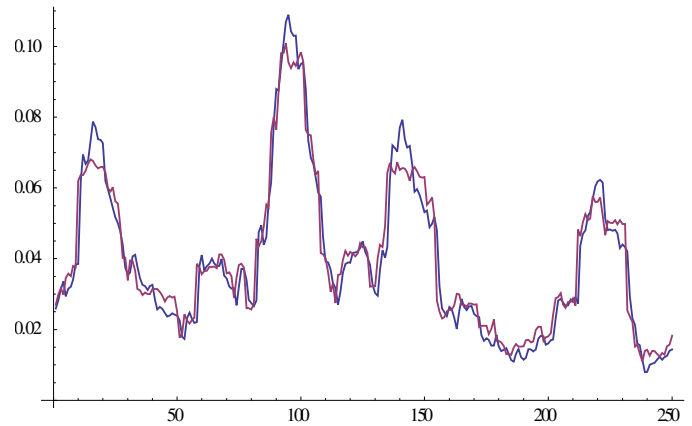


Figure 6. One year ahead forecast of conditional volatility by JNN(10,10,10). Target values blue, forecasted values purple. Absolute percentage error 11.52%.

Table 2. Mean absolute percentage errors (MPE) in predicting volatilities in four years and one year future by various JNN(p,q,r) models regarding different architectures.

Forecast	2 years %	1 year %
JNN (3,3,2)	16.39	11.36
JNN (5,5,5)	15.98	13.44
JNN (10,10,10)	20.38	11.20
JNN (20,20,20)	14.42	9.74
JNN (30,30,2)	15.92	13.26

### 5.3 Forecasting Conditional Volatility, Using ENNs

In order to estimate the conditional volatility, the model GARCH(p,q) is used with r context neurons that are feedback from hidden layer. Therefore they are of Elman artificial neural network ENN(p,q,r) model with several values of p, q, and r (Figure 3). In the sample, log returns of the S&P100 index daily closing prices are used for the calculation. The input for ENN is squared mean corrected returns with several time lags. In Figure 7, target values are shown in blue, forecasted values are in purple. To get these accuracies, 1000 trading days are used for training, and neural network successfully predicted the volatility of the next 500 trading days with the given accuracies.



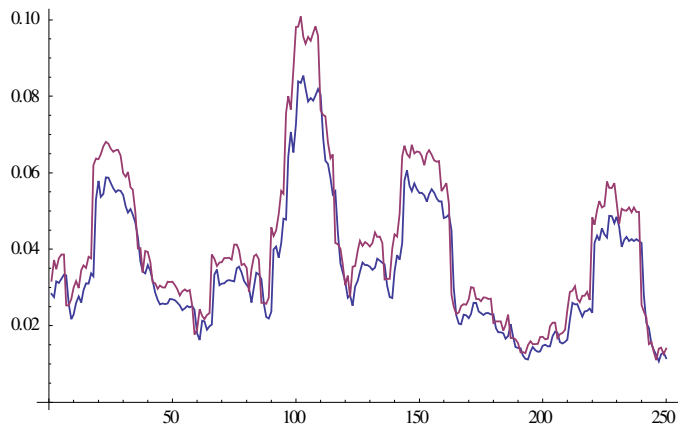


Figure 7. One year ahead forecast of conditional volatility by ENN(2,2,2,2). Target values blue, forecasted values purple. Absolute percentage error 15.14%.

Table 3. MSE in training, and MPE in testing obtained from various ENNGARCH(p,q) models regarding different architectures.

Future	ENN(2,2,2,2)	ENN(5,5,5,5)
2 years	12.51	20.41
1 year	15.14	14.97

#### 5.4 Forecasting Conditional Volatility, Using JENNs

In order to estimate the conditional volatility, the model GARCH(p,q) is used with r context neurons that are feedback from hidden layer, and s context neurons that are feedback from output. Therefore they are of JENN(p,q,r,s) model with several values of p, q, r, and s. In the sample, log returns of the S&P100 index daily closing prices are used for the calculation. The input for JENN is squared mean corrected returns with several time lags. In Figure 8 target values are shown in blue, forecasted values are in purple. To get these accuracies, 1000 trading days are used for training, and neural network successfully predicted the volatility of the next 500 trading days with the given accuracies.

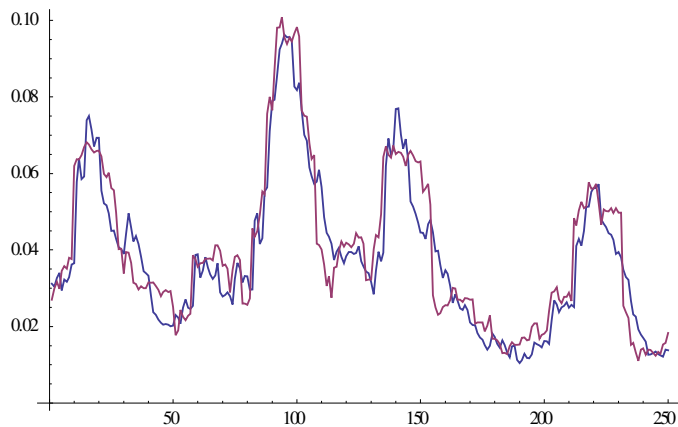


Figure 8. One year ahead forecast of conditional volatility by JENN(10,10,10,5,1). Target values blue, forecasted values purple. Absolute percentage error 12.65%.

Table 4. MSE in training, and MPE in testing obtained from various GARCH(p,q) models regarding different architectures.

Future	JENN(5,5,5,10,1)	JENN(10,10,10,5,1)
2 years	19.80	17.14
1 year	28.96	12.65

#### 6. CONCLUDING REMARKS

The focus of the paper is the volatility forecasting of returns.. Paper begins with a brief introductory to artificial neural networks and its extension Jordan, and Elman recurrent networks. Then most widespread models of the volatility are discussed. In order to estimate the conditional volatility, the model GARCH(p,q) is used with r context neurons that are feedback from output (JNN), and/or hidden neurons (ENN, and JENN). In the sample, log returns of the S&P100 index daily closing prices are used for the calculation.

The data set consists of returns of the S&P100 index daily closing prices obtained from the S&P100 index website in the period from September 6, 2005 until June 7, 2016. However, for the purpose of this research, the first 1000 trade dats of the data is neglected, and the remaining data is divided into two parts: the in-the-sample part consists of 1000 observations in the period from August 26, 2009 until August 14, 2013 which is used for the training and the estimation of parameters in the GARCH(p,q), JNN(p,q,r), ENN(p,q,r) and the EJNN(p,q,r,s) models; and the out-of-the-sample part which consists of the 500 observations in period from August 14, 2013 to August 10, which is used for the forecasting purposes. To show that ANNs predict better for shorter term forecasts we also took in-the-sample part which consists of 500 observations in the period from August 8, 2011 until August 14, 2013 which is used for the training and the estimation of parameters, 250 observations in period from August 14, 2013 to August 8, 2014 which is used for the one year ahead forecasting. We have seen that the forecasting mean absolute percentage error MPE reduces below 10%.

Table 1-4 summarizes the prediction power of the method that consists of GARCH(p,q) volatility model supported by ANNs. Mean absolute percentage error ranging from 12 to 20 % is very promising. If the same accuracy prevails in the forecasting asset prices, it may be a very powerful prediction tool in the markets.

## REFERENCES

- Aminian, F., Suarez, E.D., Aminian, M., Walz, D.T. (2006). Forecasting economic data with neural networks, *Computational Economics*, 28, 71–88.
- Arnerić, J., Poklepović, T., and Aljinović, Z. (2014) GARCH based artificial neural networks in forecasting conditional variance of stock returns, *Croatian Operational Research Review* 329 *CRORR* 5(2014), 329–343
- Balkin, S.D. (1997). Using recurrent neural networks for time series forecasting, Working Paper Series number 97–11, International Symposium on Forecasting, Barbados
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity, *Journal of Econometrics*, 31, 307–327.
- Cheng, B., Titterton, D.M. (1994) Neural networks: A review from a statistical perspective. *Statistical Science* 9 (1), 2–54.
- Ching-Ping W., Shin-Hung L., Hung-Hsi H., Pei-Chen W. (2011) Using neural network for forecasting TXO price under different volatility models, *Expert Systems with Applications* xxx (2011) xxx–xxx
- Chng, E.S., Chen, S., Mulgrew, B. (1996) Gradient radial basis function networks for nonlinear and nonstationary time series prediction. *IEEE Transactions on Neural Networks* 7 (1), 190–194.
- Cottrell, M., Girard, B., Girard, Y., Mangeas, M., Muller, C. (1995) Neural modeling for time series: a statistical stepwise method for weight elimination. *IEEE Transactions on Neural Networks* 6 (6), 1355–1364.
- Cybenko, G. (1989) Approximation by superpositions of a sigmoidal function. *Mathematical Control Signals Systems* 2, 303–314.
- Dechpichai, P. (2010). Nonlinear neural network for conditional mean and variance forecasts, Doctor of Philosophy thesis, University of Wollongong, School of Mathematics and Applied Statistics, University of Wollongong, Dubai.
- Delyon, B., Juditsky, A., Benveniste, A. (1995) Accuracy analysis for wavelet approximations. *IEEE Transactions on Neural Networks* 6 (2), 332–348.
- Elman, J. L. (1990), 'Finding structure in time', *Cognitive Science*, 14, 179–211.
- Engle, R.F. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of UK inflation, *Econometrica*, 41, 135–155.
- Franses, P.H., van Dijk, D. (2003). *Nonlinear Time Series Models in Empirical Finance*, Cambridge University Press.
- Funahashi, K. (1989) On the approximate realization of continuous mappings by neural networks. *Neural Networks* 2, 183–192.
- Gershenfeld, N.A., Weigend, A.S. (1993) The future of time series: learning and understanding. In: Weigend, A.S., Gershenfeld, N.A. (Eds.), *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley, Reading, MA, pp. 1–70.
- Ghiassi, M., Saidane, H., Zimbra, D.K. (2005). A dynamic artificial neural network model for forecasting time series events, *International Journal of Forecasting* 21, 341–362
- Gonzales, S. (2000). Neural networks for macroeconomic forecasting: A complementary approach to linear regression models, Working Paper 2000–07.
- Hammerstrom, D. (1993) Neural networks at work, *IEEE Spectrum*, June, 26–32.
- Haykin, S. (1999) *Neural Networks, a Comprehensive Foundation*, Prentice Hall.
- Hertz, J., Krogh, A., Palmer, R.G., 1991. *Introduction to the Theory of Neural Computation*. Addison-Wesley, Reading, MA.
- Hill, T., Marquez, L., O'Connor, M., Remus, W. (1994) Artificial neural networks for forecasting and decision making. *International Journal of Forecasting* 10, 5–15.
- Hinton, G.E. (1992) How neural networks learn from experience, *Scientific American*, September, 145–151.
- Hopfield, J.J. (1982) Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of the Sciences of the U.S.A.* 79, 2554–2558.
- Hornik, K. (1991) Approximation capabilities of multilayer feed-forward networks. *Neural Networks* 4, 251–257.
- Hornik, K. (1993) Some new results on neural network approximation. *Neural Networks* 6, 1069–1072.
- Hornik, K., Stinchcombe, M., White, H. (1989) Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 359–366.
- Hu, M.J.C. (1964) Application of the adaline system to weather forecasting. Master Thesis, Technical Report 6775-1, Stanford Electronic Laboratories, Stanford, CA, June.
- Hush, D.R., Horne, B.G. (1993) Progress in supervised neural networks: What's new since Lippmann? *IEEE Signal Processing Magazine*, January, 8–38.
- Hwang, H.B. (2001). Insights into neural-network forecasting of time series corresponding to ARMA(p,q) structures, *Omega* 29, 273–289.
- Irie, B., Miyake, S. (1988) Capabilities of three-layered perceptrons. In: *Proceedings of the IEEE International Conference on Neural Networks*, I, pp. 641–648.
- Kuan, C.-M., Liu, T. (1995) Forecasting exchange rates using feedforward and recurrent neural networks. *Journal of Applied Econometrics* 10, 347–364.
- Kuan, C.M., and White, H. (2007). Artificial neural networks: An econometric perspective, *Econometric Reviews*, 13, 1–92

- Kuan, C.M., Hornik, K. and H. White (1994) A convergence result for learning in recurrent neural networks', *Neural Computation*, 6, 620-640.
- Kuan, C.M. and H. White (1994) Adaptive learning with nonlinear dynamics driven by dependent processes, *Econometrica*, vol. 62 No 5 pp. 1087-1114.
- Lapedes, A., Farber, R. (1988) How neural nets work. In: Anderson, D.Z., (Ed.), *Neural Information Processing Systems*, American Institute of Physics, New York, pp. 442-456.
- Lawrence, S., Giles, C.L., Tsoi, A.C. (1997) Lessons in neural network training: Overfitting may be harder than expected, *Proceedings of the Fourteenth National Conference on Artificial Intelligence, AAAI-97*, 540-545.
- Lippmann, R.P., 1987. An introduction to computing with neural nets, *IEEE ASSP Magazine*, April, 4-22.
- Marquez, L., Hill, T., O'Connor, M., Remus, W. (1992) Neural network models for forecast a review. In: *IEEE proceedings of the 25th Hawaii International Conference on System Sciences.*, 4, pp. 494-498.
- Masson, E., Wang, Y.-J., 1990. Introduction to computation and learning in artificial neural networks. *European Journal of Operational Research* 47, 1-28.
- Pack, D.C., El-Sharkawi, M.A., Marks II, R.J. (1991a) An adaptively trained neural network. *IEEE Transactions on Neural Networks* 2 (3), 334-345.
- Pack, D.C., El-Sharkawi, M.A., Marks II, R.J., Atlas, L.E., Damborg, M.J. (1991b) Electric load forecasting using an artificial neural network. *IEEE Transactions on Power Systems* 6 (2), 442-449.
- Pankratz, A. (1983) *Forecasting with Univariate Box-Jenkins Models: Concepts and Cases*. John Wiley, New York.
- Park, J., Sandberg, I.W., 1991. Universal approximation using radial basis function networks. *Neural Computation* 3, 246-257.
- Park, J., Sandberg, I.W. (1993) approximation and radial basis function networks. *Neural Computation* 5, 305-316.
- Reilly, D.L., Cooper, L.N., 1990. An overview of neural networks: early models to real world systems. In: Zornetzer, S.F., Davis, J.L., Lau, C. (Eds.), *An Introduction to Neural and Electronic Networks*. Academic Press, New York, pp. 227-248.
- Ripley, B.D. (1993) Statistical aspects of neural networks. In: Barndorff-Nielsen, O.E., Jensen, J.L., Kendall, W.S. (Eds.), *Networks and Chaos-Statistical and Probabilistic Aspects*. Chapman and Hall, London, pp. 40-123.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J. (1986) Learning representations by backpropagating errors. *Nature* 323 (6188), 533-536.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J. (1986) Learning internal representation by back-propagating errors. In: Rumelhart, D.E., McClelland, J.L., the PDP Research Group (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, MA.
- Rumelhart, D.E., Widrow, B., Lehr, M.A. (1994) The basic ideas in neural networks. *Communications of the ACM* 37 (3), 87-92.
- Rumelhart, D.E., Durbin, R., Golden, R., Chauvin, Y. (1995) Backpropagation: the basic theory. In: Chauvin, Y., Rumelhart, D.E. (Eds.), *Backpropagation: Theory, Architectures, and Applications*. Lawrence Erlbaum Associates, New Jersey, pp. 1-34.
- Samur, Z. I., and Temur G. T. (2009) The Use of Artificial Neural Network in Option Pricing: The Case of S&P 100 Index Options, *World Academy of Science, Engineering and Technology* 54.
- Bekiros, S.D., Georgoutsos, D.A. (2004) Forecasting stock index returns using a volatility based recurrent neural network
- Sharda, R., 1994. Neural networks for the MS/OR analyst: An application bibliography. *Interfaces* 24 (2), 116-130.
- Sharda, R., Patil, R.B. (1992) Connectionist approach to time series prediction: An empirical test. *Journal of Intelligent Manufacturing* 3, 317-323.
- Shin, Y., Ghosh, J. (1995) Ridge polynomial networks. *IEEE Transactions on Neural Networks* 6 (3), 610-622.
- Sietsma, J., Dow, R. (1988) Neural net pruning-Why and how? In: *Proceedings of the IEEE International Conference on Neural Networks*, 1, pp. 325-333.
- Tang, Z., Almeida, C., Fishwick, P.A. (1991) Time series forecast-ing using neural networks vs Box-Jenkins methodology. *Simulation* 57 (5), 303-310.
- Tang, Z., Fishwick, P.A. (1993) Feedforward neural nets as models for time series forecasting. *ORSA Journal on Computing* 5 (4), 374-385.
- Taylor S. J. (2005) *Asset Price Dynamics, Volatility, and Prediction*, Princeton University Press.
- Wasserman, P.D. (1989) *Neural Computing: Theory and Practice*. Van Nostrand, Reinhold, New York.
- Weigend, A.S., Huberman, B.A., Rumelhart, D.E. (1990) Predict-ing the future: A connectionist approach. *International Journal of Neural Systems* 1, 193-209.
- Weigend, A.S., Huberman, B.A., Rumelhart, D.E. (1992) Predict-ing sunspots and exchange rates with connectionist networks. In: Casdagli, M., Eubank, S. (Eds.), *Nonlinear Modeling and Forecasting*. Addison-Wesley, Redwood City, CA, pp. 395- 432.
- Weigend, A.S., Rumelhart, D.E., Huberman, B.A. (1991) Generalization by weight-elimination with application to forecasting. *Advances in Neural Information Processing Systems* 3, 875- 882.
- Werbos, P.J. (1974) *Beyond regression: new tools for prediction and analysis in the behavioral sciences*. Ph.D. thesis, Harvard University.

Werbos, P.J. (1988) Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks* 1, 339–356.

White, H. (1989) Learning in artificial neural networks: A statistical perspective. *Neural Computation* 1, 425–464.

Wilson, R., Sharda, R. (1992) Neural networks. *OR/MS Today*, August, 36–42.

Wong, S.Q., Long, J.A. (1995) A neural network approach to stock market holding period returns. *American Business Review* 13 (2), 61–64.

Zhang, Q., Benveniste, A. (1992) Wavelet networks. *IEEE Transactions on Neural Networks* 3 (6), 889–898.

Zhang, G.P. (2003) Time series forecasting using hybrid ARIMA and neural network model, *Neurocomputing* 50, 159–175.

Zhang, G., Patuwo, B.E., M, Hu, M.Y. (1998) Forecasting with artificial neural networks: The state of the art, *International Journal of Forecasting* 14 (1998) 35–62