Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Pattern Recognition Letters 26 (2005) 2093–2103
Pattern Recognition Letters
www.elsevier.com/locate/patrec

A comparison of global, recurrent and smoothed-piecewise neural models for Istanbul stock exchange (ISE) prediction

Serdar Yümlü [a], Fikret S. Gürgen [a,*], Nesrin Okay [b]

[a] Department of Computer Engineering, Boğaziçi University, 34342 Bebek, Istanbul, Turkey
[b] Department of Management and Finance Engineering Program, Boğaziçi University, 34342 Bebek, Istanbul, Turkey

Received 10 September 2004; received in revised form 10 January 2005
Available online 23 May 2005

Communicated by W. Pedrycz

Abstract

This paper makes a comparison of global, feedback and smoothed-piecewise neural prediction models for financial *time series* (*FTS*) *prediction* problem. Each model is implemented by various neural network (NN) architectures: *global model by a multilayer perceptron* (*MLP*), *feedback model by a recurrent neural network* (*RNN*) *and smoothed-piecewise model by a mixture of experts* (*MoE*) *structure*. The advantages and disadvantages of each model are discussed by using real world finance data: 12 years data of Istanbul stock exchange (ISE) index (XU100) from 1990 to 2002. A conventional *exponential generalized* autoregressive conditional heteroskedasticity (EGARCH) volatility model is also implemented for comparison purpose. The comparison for each model is done based on well-known criterions of index return series of market: hit rate (H_R), positive hit rate (H_R^+), negative hit rate (H_R^-), mean squared error (MSE), mean absolute error (MAE) and correlation (ζ). Finally, it is observed that the *smoothed-piecewise neural model* becomes advantageous in capturing volatility in index return series when it is compared to *global* and *feedback neural model*, and also the conventional *EGARCH volatility model*.

Keywords: Financial time series (FTS) prediction; Global; Feedback; Smoothed-piecewise neural models; Exponential generalized autoregressive conditional heteroskedasticity (EGARCH) model; Risk estimation; Volatility

1. Introduction

Recently, *mixture of models* and *multiple models* have become popular research areas in machine learning and related fields. Generally in this field,

* Corresponding author. Address: Department of Computer Engineering, Boğaziçi University, 34342 Bebek, Istanbul, Turkey. Tel.: +90 212 359 6863; fax: +90 212 287 2461.
E-mail addresses: gurgen@boun.edu.tr (F.S. Gürgen), okay@boun.edu.tr (N. Okay).

0167-8655/$ - see front matter
doi:10.1016/j.patrec.2005.03.026

there has been a special interest in the development of clustering, classification, regression, prediction and parameter estimation algorithms for time series (dynamics) problems. Remarkable efforts include the directions such as *support vector machines* (*SVM*), *Bayesian networks*, *mixture of experts*, *ensembles of neural networks*, *fuzzy models*, etc. (Shafer and Vovk, 2001; Petridis and Kehagias, 1998; Petridis et al., 2001; Rao et al., 1997; Castillo and Melin, 2002; Shawe-Taylor and Cristianini, 2004; Heckerman, 1999). These ideas have further led to reinterpretation of existing network structures; proposals of new network structures; and novel learning algorithms based on optimization techniques, principles, and criteria from these areas.

This study addresses the problem of application of global, feedback and piecewise neural models to financial time series (FTS) prediction (Yümlü et al., 2003, 2004). It is an example case of choosing the proper model for a specific application of FTS prediction. Furthermore, a conventional *exponential generalized autoregressive conditional heteroskedasticity* (*EGARCH*) model is employed for comparison. A *multilayer perceptron* (*MLP*) is employed as a global predictor of FTS that uses the training samples obtained from each local part of the time series. Here, the MLP is trained with these local, fixed-size samples to receive the overall picture of the series, and then make a prediction. A *recurrent neural net* (*RNN*) is used as a feedback predictor that introduces a memory between parts of local series. In this case, a number of weighted feedback connections are added to the feedforward structure of a *MLP*, so it can encode the relationship of a number of partially local series better in the weights. Finally, smoothed-piecewise predictors such as *mixture of experts* (*MoE*) structure are used to summarize the localized series with a number of statistics such as mean, variance values and then, obtain an overall prediction smoothed by conditional probability values of each local series or *local expert*.

The global, feedback and smoothed-piecewise neural models are employed for the prediction of volatilities of certain assets in Istanbul stock exchange (ISE) index (XU100) from Turkey from years of 1990s to 2002s. The real world finance data is divided into two parts: (a) 12 years data from 1990 to 2002 and (b) 4 years data from 1998 to 2002. There is an important reason behind this separation: ISE shows different distributional effects between 1990 and 1998 because of the indeterminate economical structure, so we have decided to test our models with a smaller and more predictable data set from 01/01/1998 to 04/04/2002.

Financial markets (ISE) in Turkey have various instability sources of the economical infrastructure. Any news or rumor may cause changes in volatile movements, as a result, forecasting the volatility of the market becomes an important issue. With the described features (Table 1), we try to measure the effectiveness of above models in the prediction. No previous work has been reported in the area.

We try to make a comparison of the well-known neural models for the case of ISE data in our study: these models are global model with a MLP example, feedback model with a RNN example (Elman's model with feedback from hidden units),

Table 1
This table reports the statistics for XU100 index return series

Summary statistics for the daily returns of XU100 index (12/01/1990–26/04/2002)								
Size	Mean	Variance	Standard deviation	Skewness	Kurtosis	Min	Max	J. Bera (J-B) (p value)
2945	0.002	0.001	0.033	0.130	5.727	−0.2	0.171	918.253 (0.00)
Q (5)	Q (10)	Q^2 (5)	Q^2 (10)	$P1$	ρ^2	ARCH (Eq. (3))		
40.61	56.922	415.784	551.324	0.105	0.005	272.544		

The XU100 index data are obtained from ISE database. The sample period is from January 12, 1990 to April 26, 2002. There are 2945 daily return observations in the sample period. As this table shows like most of the financial time series, also in XU100 return distribution is negatively skewed. The kurtosis coefficient, which is a measure of the thickness of the tails, is very high. Jarque-Bera test rejects the null hypothesis of normally distributed returns. ARCH (Eq. (7)) tests the ARCH effects using Engle and Ng's (1993) test.

smoothed-piecewise model with MoE example and a well-known, conventional EGARCH model. We furthermore mention piecewise models (individual local experts) such as linear predictors (LP) and polynomial predictors (PP) and the reasons of not using them in our case. Even though a combination of various models such as neural and EGARCH models can also be considered, in this study we limit our attention to performances of above models and describe their effectiveness in the case of ISE data.

The rest of the paper is organized as follows: Section 2 introduces predictor models. Section 3 introduces volatility forecasting and the characteristics of the stock market FTS. Section 4 describes global, feedback and smoothed-piecewise neural prediction models. Section 5 mentions the implementation issues of neural models and the experiments. Section 6 presents results with discussions. Final section concludes the study with future work.

2. Predictor models

In general for a *time series* (*TS*) *prediction* problem, a predictor fits a model to given data and finds an approximate mapping function between the input and the output values (Shafer and Vovk, 2001; Petridis and Kehagias, 1998; Petridis et al., 2001). Thus, the proposed model predicts underlying patterns, trends and cycles using historical and currently observable data. A time series x_t, $t = 1, 2, 3, \ldots$; for simplicity x_t is taken to be scalar but vector-valued time series are also used. Among a number of predictors of this series, the kth predictor is obtained by

$$y_t^k (\cong x_t^k) = f(x_{t-1}, x_{t-2}, \ldots, x_{t-M}; w_k), \quad k = 1, 2, 3, \ldots, K. \tag{1}$$

The real value of x_t is predicted as y_t with a prediction error e_t. The kth predictors belong to a general family $f(.,w)$, where w is a parameter vector; the kth predictor is obtained by setting $w = w_k$. It is reasonable to assume that, if kth prediction's error $e_t^k = y_t^k - x_t^k$ forms a sequence of independent, identically distributed (*iid*) random variables with *zero* mean and σ^2 variance. The value of M is also known as the *prediction horizon* is expected to be made as large as possible by keeping the prediction error (e_t) within reasonable bounds.

Linear (*regression*) *predictors* (*LP*) are defined as single *input* (*autoregression*) *predictors* of form

$$y_t^k = a_0^k x_t + a_1^k x_{t-1} + a_2^k x_{t-2} + a_3^k x_{t-3} + \cdots + a_M^k x_{t-M} \tag{2}$$

and *multi-input predictors* with two-inputs case are defined by

$$y_t^k = a_0^k x_t + b_0^k u_t + a_1^k x_{t-1} + b_1^k u_{t-1} + a_2^k x_{t-2} + b_2^k u_{t-2} + \cdots + a_M^k x_{t-M} + b_0^k u_{t-M} \tag{3}$$

Training of the predictors means obtaining a and b coefficients using a least squares approximation method. LP has limitations especially in time series with nonstationary nature. Even though various solutions are available for certain applications such as short time linear predictive coding (LPC) of speech signal representation, etc. In the LPC, nonstationary speech signal is frozen at short frames and then is represented by LP coefficients. This becomes unsuitable for FTS prediction problem since the coverage of the representation will be limited to short windows.

Polynomial predictors (*PP*) are polynomials of time variable t, an nth *order polynomial predictor* is shown by

$$y_t^k = a_0^k + a_1^k t + a_2^k t^2 + a_3^k t^3 + \cdots + a_n^k t^n \tag{4}$$

In this case also, training for every data group means the computation of a coefficients by least squares regression method. PP has limitations in capturing main trends of the overall series: for example, when the series get longer, the order n may become too large and the prediction becomes harder.

Neural predictors are similar to linear predictors in the principle but they differ in the sense that they use a nonlinear regression implemented by various *neural networks* such as *MLP*, *RNN and MoE structure* to form

$$y_t^k = f(x_{t-1}, x_{t-2}, \ldots, x_{t-M}; w_k) \tag{5}$$

or in two-variable case of the form

$$y_t^k = f(x_{t-1}, x_{t-2}, \ldots, x_{t-M}; u_{t-1}, u_{t-2}, \ldots, u_{t-M}; w_k) \tag{6}$$

Here w_k is a matrix of weights or parameters. Training algorithm is used to determine these parameters.

Various prediction models, *modular networks* (Petridis et al., 2001), *bayesian networks* (Heckerman, 1999), *support vector machines* (*SVM*) (Nuller et al., 1997), etc. are also used in different applications: modular nets combine various predictors such as bayesian combined predictor model (Petridis et al., 2001). Bayesian nets considers the dynamics of time series such as *dynamic bayesian model* (*DBN*) (Heckerman, 1999) and *SVM*s employ support vectors in the input space (Nuller et al., 1997).

3. Financial time series problem and volatility

Generally, *FTS prediction* (Shafer and Vovk, 2001; Petridis and Kehagias, 1998; Petridis et al., 2001) is a difficult problem that has hidden variables and lacks observable data for determining the underlying structure of the series, if one exists. Our study uses conditional variance (volatility) that is time-dependent heteroskedastic variance and it is not a directly observable feature. A well-known approach of financial markets is *efficient market hypothesis* (*EMH*) in which the current market price reflects all the available information immediately. In *EMH*, people do not believe in finding evidence for the prediction of stock markets. The alternative approach, which is widely accepted by the traders' environment, is the belief that the stock market is predictable in the sense of fundamental analysis and prediction methods. In this study, we investigate the predictability of ISE index XU100 by assuming the second approach which is against the *EMH* and furthermore show the usage and comparison of neural predictors in *FTS prediction*.

Briefly, volatility is the measure of the changeability in asset returns (Tino et al., 2000; Nelson, 1991; Engle, 1982; Engle and Ng, 1993; Black, 1976). Stock prices vary with changes in volatilities of the underlying risk factor and as a consequence, accurate prediction of future stock prices requires a forecast of an asset return's volatility. This time-dependent variance is known as the heteroskedasticity. *FTS* of stock returns shows time-dependent variance and this requires us to predict the volatility. Volatility forecast of asset returns is used in market risk management, portfolio selection, market timing, etc. Market risk management plays a crucial role in financial decisions. None of the players want a volatile market. *Value at risk* has become a standard in market risk management. Value at risk estimation is based on the forecasting of the volatility of market risk factors. As a result, estimating the volatility of asset returns, which is a basic risk factor of the stock market, gives valuable information for the future risk in the market and this will make the players to consider the expected high or low volatility in the market.

Conditional variances are known to be unobservable features, but in the literature there are numerous studies and approaches to estimate volatilities using historical asset returns. This variance is time-dependent and Engle (1982) first proposed using past asset returns to model heteroskedastic behavior with an *autoregressive conditional heteroskedasticity* (*ARCH*) process. Bollerslev (1986) generalized this approach offering a *generalized autoregressive conditional heteroskedasticity* (*GARCH*) model in which conditional variances are governed by a linear autoregressive process of past squared returns and variances.

$$h_t = \omega + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{q} \beta_i h_{t-i} \tag{7}$$

In this equation, h_t represents the conditional variances and ω, α and β are constant parameters of the system. Both *ARCH* and *GARCH* models lack in modeling a volatility fact that is known as the "leverage effect" which is known as the effect of the sign of the innovations. For this purpose, asymmetric extensions of the GARCH have been proposed. One of the most widespread is Nelson (1991) who has proposed *exponential GARCH* (*EGARCH*).

$$\log(h_t) = \omega + \beta \cdot \log(h_{t-1}) + \gamma \frac{\varepsilon_{t-1}}{\sqrt{h_{t-1}}} + \alpha \left[\frac{|\varepsilon_{t-1}|}{\sqrt{h_{t-1}}} - \sqrt{\frac{2}{\pi}} \right] \quad (8)$$

where h_t represents the conditional variances and ω, α, β and γ are constant parameters of the system. EGARCH considers modeling the sign effect besides using past squared innovations and past variances. Later, the impact of news on volatility models has been measured by employing especially *EGARCH* models (Engle and Ng, 1993). As a result, *EGARCH* models have become common models of the area.

In neural predictor models, we normalize and use logarithm operators to transform the index price series y_t into r_t continuously compounded return series. A normalization process keeps the price series in a constant range. The price series, which is not stationary and contains trends, seasonality and cycles, is then converted into continuously compounded return series by the formula given at Eq. (7) to obtain an accepted stationary series:

$$r_t = \ln(y_t/y_{t-1}) \quad (9)$$

where r_t shows the compounded return at time t for given price series y_t and y_{t-1}. Return series have a constant range even when we use many years of data, but prices do vary greatly and comparing assets using return series is more accurate than using price series.

Several characteristics (Schittenkopf and Dorffner, 1999; Tino et al., 2000; Chan and Gao, 2000; Hellstrom and Holmstrom, 1997; Yao and Tan, 2001) have been confirmed by various studies on volatility model generation for the prediction of FTS. A summary can be as follows: a volatility model is expected to capture and reflect the properties of the series such as *volatility clustering*, which is related to changes of price assets, *mean reversion* in volatility, which implies that current information has no effect on the long run forecast, *asymmetric effect* or *leverage effect*, which means that changes in stock prices tend to be negatively correlated with changes in volatility. In other terms, volatility increases much more after negative shocks to asset price rather than positive shocks. Finally, the distribution of returns is fat-tailed and exhibits *leptokurtosis*, which means the kurtosis, exceeds the kurtosis of a standard gaussian distribution.

4. Global, feedback, smoothed-piecewise neural prediction models

In this paper, we interpret global, feedback and smoothed-piecewise neural prediction models for the ISE with two *FTS* examples. The task of a neural prediction model can be described as follows: given a set of input–output pairs $T = \{(x_i, y_i)\}$, where $x_i \varepsilon R^n$, $x_i \varepsilon R^m$ are drawn from an unknown distribution, design a mapping f: $R^n \Rightarrow R^m$ that minimizes the expected prediction error, in the case of squared error, given by $E[(y - f(x,w))^2]$. The function $f(x,w)$ defines corresponding mapping for the predictor. The value of m is 1 in the TS prediction case.

Global neural models like a *feedforward MLP* (Fig. 1) implements a nonlinear regression function that must fit the data well everywhere with no explicit partitioning of the input space without subdivision of the parameter set. Then, it predicts the sample y_i. It is clear that the role of individual parameters cannot be seen in the structure. In our implementation, we choose a typical sigmoid function and train the weights w of $f(x,w)$ function with a *back propagation* (*BP*) algorithm.

Feedback neural models like a *recurrent neural network* (*RNN*) (Fig. 2) structure has feedback connections that is suitable for modeling the temporal relationships in time. Unlike *feedforward MLP*, the *RNN* introduces a valuable basis for modeling time series. We implement Elman's RNN with real-time recurrent learning (RTRL) algorithm (Petridis and Kehagias, 1998) in our study. Elman's RNN stores the values of hidden units (or internal states) and feeds them back to the net. Error minimization in RTRL algorithm is done by measuring the sensitivity of the output at unit k at time t to a small change in the weight value w_{ij} (from the hidden unit j to the input unit i). The effect of a change in the weight is taken into account and propagated to the entire network during the time steps.

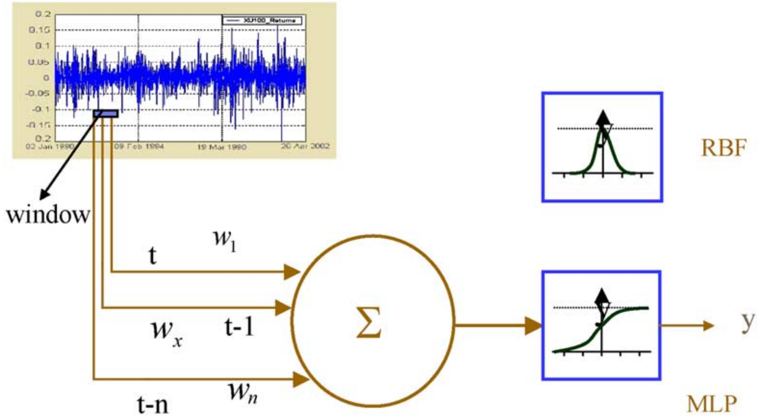

Fig. 1. A schematic for MLP and RBF structures.

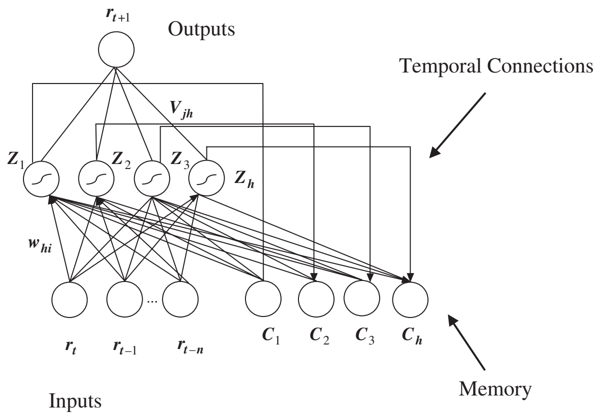

Fig. 2. A schematic for RNN structure (Type 1 RNN of Elman).

A smoothed-piecewise neural model such as MoE (Rao et al., 1997) (Fig. 3) is constructed by *local experts* with a prediction function $f(x, w_j)$, where w_j is the set of model parameters for local model j. Generally, the local experts may be constant, linear or nonlinear (polynomial of any degree) function of x. The overall prediction function of the model is defined with conditional expectation,

$$f(x,w) = \sum_j P[j/x] f(x, w_j) \tag{10}$$

where $P[j/x]$ is a nonnegative weight of association between input x and expert or local model j and it determines the degree to which expert j contributes to the overall model output. These weights are often called *gating units* and are imposed $\sum_j P[j/x] = 1$, which is a parametric function determined by a parameter set w_{gating}. Statistical interpretation of the model is as follows: the input–output pair (x_i, y_i) is randomly selected by an input density and by a local model according to probability mass function $\{P[j/x]\}$. For a selected local model k, the output is generated as a random variable whose *mean* is $f(x, w_j)$. With this viewpoint, $f(x, w)$ represents the expected value of the output for a given input x. It is known that conditional expectation is the *minimum mean-squared error* (*MMSE*) predictor.

A main advantage is that the minimization of squared error in a smoothed-piecewise neural structure improves cooperative prediction between

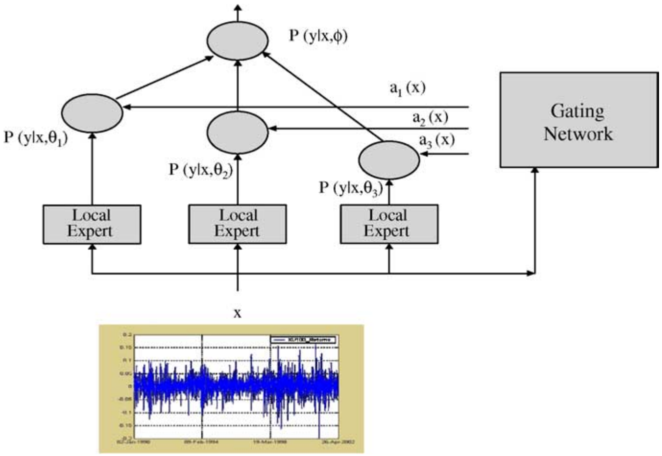

Fig. 3. The structure of the mixture of experts (MoE).

output $g(x_i)$ and predictors $f(x_i, w_j)$ while each individual predictor improves the individual prediction between output y_i and each predictor $f(x_i, w_j)$. A normalized, one hidden layer, *radial basis function* (*RBF*) (Figs. 1 and 3) structure (Petridis and Kehagias, 1998) is a first example of smoothed-piecewise neural structure in which we compute the hidden layer outputs by where $R_k(x) = \exp -[\|x - m_k\|^2/2\sigma_k^2]$ is a commonly used gaussian basis functions with m_k mean vectors (*centers of input space*) and σ_k variance. k index is for a specific node and input space is covered by all the hidden layer nodes of the architecture. The second layer generates a linear combination of hidden nodes which may be interpreted as local experts $\{f(x, w_j)\}$ and the weights to output node represent the probabilities of association $\{P[j/x]\}$ with the corresponding local experts.

Mixture of experts (*MoE*) [4] as a method of combining multiple learners becomes a general *MoE* structure. It consists of two parts: the first part is a number of parallel local experts which use the same input pattern and the second part is a gating expert which determines the output using the decisions of local experts regarding the input pattern.

The proposed *MoE* predictor (Fig. 3) predicts each data point with a probability to the local models by seeking optimal probabilistic assignments $\{P[j/x_i]\}$ as well as the model parameter set $\{f(x, w_j)\}$. The *MMSE* cost is selected for error optimization. Any choice of random branch from the output node takes us to one of the local experts. The conditional distribution for selecting the individual local decision given a node is computed at output node by a "gate" with

$$g_j(x) = \exp(w_j^{\mathrm{T}}x) \Big/ \sum_m \exp(w_m^{\mathrm{T}}x) \tag{11}$$

where weight vectors $\{w_j\}$ create a partition of input space.

Piecewise models generally fit data to local region of input space and patch the local parameters in a divide-and-conquer sense. Thus, piecewise solutions generated by individual parameters of individual submodels become easy to interpret. A modular neural network structure (Petridis et al., 2001) can be considered as an example in which individual neural net modules are trained to obtain solutions for local input regions. A smoothed-piecewise model decomposes regression problem into the learning set of local (expert) models but none of them claims exclusive ownership of the region unlike piecewise models. By using $P[j/x]$ weights that are restricted to the values $\{0, 1\}$, the parameters are added only when they require to improve the fit in a local region. Thus, the overall problem is simplified in terms of learning and modeling. Furthermore, unlike piecewise modeling that generates discontinuous regions at boundaries, smoothed-piecewise solutions are smooth everywhere due to averaging in Eq. (11). They produce no exclusive ownership of a region, thus, they simplify the learning and modeling problem by creating parsimonious solutions.

The cooperative prediction solution produced by *MoE* structure with *MMSE* criterion resembles more closely local piecewise models than global models (Rao et al., 1997). Generally in competitive models trained with *maximum likelihood* (*ML*) objective, only a few experts are activated for a given input. In cooperative models, many experts that are distributed over the input space contribute to a given output. The prediction solutions of individual local experts are combined through the $P[j/x]$ weights which defines each predicted data point associated in probability with the various local models. This bears similarity between MoE prediction and piecewise prediction model. On the other hand, it generates an averaging solution or smoothing effect for the discontinuous functions at region boundaries of piecewise prediction model. In this sense, it is claimed that MoE structure is closer to global models.

As a conventional model, we employ *EGARCH* (Bollerslev, 1986) that includes volatility (conditional variance) of a *FTS*, thus it overcomes the drawbacks of *ARCH* and *GARCH* models such as lack of leverage effect. By modeling *EGARCH*, we consider the sign effect besides using past squared innovations and past variances. This model is used as a basic model to compare the results of neural predictor experiments.

5. Implementation issues of neural models and experiments

In this section, we describe ISE data and then use it for the prediction of risk in index return series by using *MLP*, *RNN*, *MoE* and conventional *EGARCH* structures. Data consists of 2946 daily observations of ISE 100 index (Turkey) (Fig. 4). Data covers a 12-year period, from 12 January 1990 to 26 April 2002. We use two portions: (a) 12 years data and (b) 4 years data from 1998 to 2002. Besides the index close series, we have studied four supporting series: USD dollar series, two interest rate series which are simple interest rate and central bank money series have been studied as helper series to find the underlying nonlinear relationship, the unknown patterns and the effects of these series over the volatility of the index return series.

Time can be introduced to a neural architecture in various ways: in one way, we may leave the time outside of the neural model as we do in *MoE*. Here, a sliding window of last n elements is described and then is used to predict the following observation in the time series. In the other way, we may encode the time as a numerical value and use it in the structure similar to *RNN*. As a result, the time becomes an index of the state of the net. The storage of the states constructs a memory segment and various buffering and weighted buffering techniques can be applied to form the memory. In the experiments, we have used the first type of Elman's *RNN* which loops back the outputs from hidden neurons to the input layer for modeling the risk

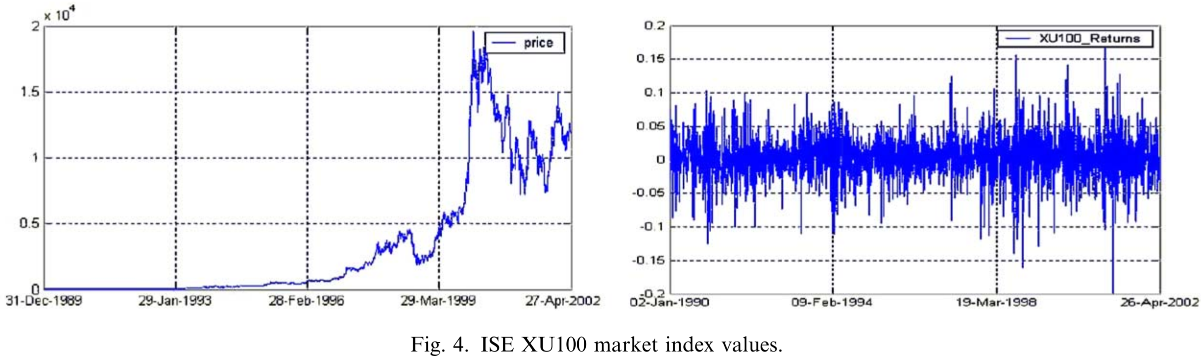

Fig. 4. ISE XU100 market index values.

in index return series and use it for the future predictions.

Financial markets in Turkey show nonlinearity with very high noise because of various instability sources of the economical infrastructure. Hidden factors such as any news or rumor may cause changes in volatile movements, as a result, forecasting the volatility of the market becomes difficult. With the described features (Table 1), we try to measure the effectiveness of above models in the prediction. There has been no previous work in the area. Thus, we study all the main stream neural models such as global, recurrent and smoothed-piecewise of FTS prediction for the ISE data. We also discuss not to apply piecewise models to the case.

In this study, we have implemented various neural models and a conventional model. A model's performance besides its efficiency is dependent on the quality and relevance of the data. For this purpose, we divide the data set into three parts for each of two cases: 12 years and 4 years. In the case of 12 years of data, training, validation and test data sets with sizes of 1857, 795 and 294 are used respectively. In this case, an input window of size 7 is used with 20 hidden units. In the case of 4 years of data, training, validation and test data sets with sizes of 649, 277 and 102 are chosen. In this case, an input window of size 7 is used again but the number of hidden units has chosen to be 15. As it is known, the number of hidden units represents the local experts in the MoE structures. The best performance giving choices were used in the experiments for both of the cases. We train our models using the training set, validate the model parameters using the validation set and use the test data set for testing the performance.

In *MoE* implementation, first the data are clustered into groups using an *expectation maximization* (*EM*) algorithm. The number of centers was set to be equal to the number of hidden units in the architecture. Each group is assumed to have a gaussian distribution with mean and variances optimized using *BP* algorithm over each local expert with *MMSE* criterion. Each trained local expert with a set of parameters specializes in its region and the outputs of all local experts are connected to the output by the gating expert which considers the input pattern at the same time by processing the outputs of each local expert. Since hidden units are gaussian, the overall system applies a smoothed-piecewise linear approximation to input data.

Identical data prediction conditions are employed for the rest of the models *MLP*, *RNN* and *EGARCH* and a comparison has been done for each model in the testing phase with various comparison criterions. These criterions are hit rate (H_{R}), positive hit rate (H_{R}^{+}), negative hit rate (H_{R}^{-}), mean squared error (MSE), root MSE (RMSE), mean absolute error (MAE) and correlation and details of them discussed in the next section. Also, the number of inputs and hidden units are kept same for all of the architectures.

6. Results

Generally, the performance of a model is dependent on the quality and relevance of the data that it represents. Thus, the data selection becomes an important component of a prediction. Here, we consider four different series besides ISE 100 close price series. Sliding windows technique, taking the last n elements in the series as input, is only applied to ISE 100 and USD series. Among the various sizes of hidden units, the best performance given size was used for structures. All the series are used as an input to neural and conventional models and one-time-ahead forecast is the expected output of the *NN*. We work on real-valued series and the outputs are also real-valued forecasts. In this study we have chosen a 12-year period of ISE 100 between 01/01/1990 and 04/04/2002. A seven lag windowed series is trained using 20 hidden units for each neural model. ISE shows different distributional effects between 1990 and 1998 because of the indeterminate economical structure, so we have tested our models with a smaller and more predictable data set from 01/01/1998 to 04/04/2002 using 15 hidden units in our models. Mean square error (MSE) is used as an error function during the training phase. For testing the out-of-sample performance and the adequacy of the model we have used several performance metrics. The reported results in Tables 2 and 3 are obtained

Table 2
Performance of the neural models: piecewise-continuous MoE, *global MLP*, *feedback RNN* and *EGARCH* for 4 years of ISE data (01/01/1998–04/04/2002)

	MoE	MLP	RNN	EGARCH
H_{R}	0.896	0.514	0.58	0.513
H_{R}^{+}	1	0.47	0.569	0.695
H_{R}^{-}	0.87	0.569	0.588	0.359
MSE	0.139	1.064	0.101	0.174
MAE	0.373	1.032	0.318	0.417
ζ	0.849	0.324	0.796	0.037

Table 3
Performance of the neural models: piecewise-continuous *MoE*, *global MLP*, *feedback RNN* and *EGARCH* for 12 years of ISE data (01/01/1990–04/04/2002)

	MoE	MLP	RNN	EGARCH
H_{R} (%)	0.622	0.538	0.524	0.404
H_{R}^{+} (%)	1	0.491	0.478	0.671
H_{R}^{-} (%)	0.609	0.598	0.579	0.298
MSE	0.499	0.988	0.912	0.129
MAE	0.706	0.994	0.955	0.359
ζ	0.546	0.347	0.378	0.021

from the production (test) data set. Hit rate (H_{R}) shows the percentage of the correct predictions of the direction of the market. Positive hit rate (H_{R}^{+}) is the percentage of the correct predictions during the increasing market and the negative hit rate (H_{R}^{-}) is the opposite. *MSE* and mean absolute error (MAE) criterions measure the ability of the model to capture the data. Also as a similarity measure, correlation (ζ) is used which should be close to 1 for a perfect fit (or strong correlation).

In the experiments, *MoE* structure becomes a favorable structure among the others for the given measures. The experimental results support that smoothed-piecewise neural *MoE* model has the strength to capture the underlying model in the data series and is able to include knowledge that can be used during trading and risk management. In Tables 2 and 3, it is observed that *MoE* has higher values for (H_{R}, H_{R}^{+} and H_{R}^{-}) hit rates and similarity measure (ζ) among others. Also *MoE* gives smaller values of MSE and MAE measures between 1998 and 2002 years. Table 2 shows results in favor of smoothed-piecewise approach. In Table 3, some MSE and MAE values are smaller for *EGARCH* model. In fact, the MSE and MAE measures show the ability to model the FTS data, but this is not reflected to the rest of the performance measures in EGARCH results. We observe a superior performance of the hit rates of the *MoE* model in both directions (H_{R}, H_{R}^{+}, and H_{R}^{-}) for the ISE data and conclude that the model is more suitable for the prediction of volatility. Furthermore, the convenience of *MoE* structure is supported by the best the similarity measure (ζ) result among the others.

As a summary, Table 3 shows the metrics obtained using the whole data set from 1990 to 2002. Because of the high volatility during the crisis in Turkey, we generally see that the models have difficulty to capture the features of data. But even in these conditions, *MoE* as a smoothed-piecewise neural structure is superior to its rivals in this study.

7. Future work and conclusion

In this study, we interpret and discuss the application of *global*, *feedback and smoothed-piecewise*, *neural MoE* structure. By using the assumption that is against the EMH hypothesis, experimental results on ISE data suggest that MoE structure specifically introduces a powerful model to predict the *volatility* of FTS data. In our work, we consider ISE XU100 and observe the prediction of volatilities (conditional variances) instead of return index prices makes a valuable contribution to this area. It was not suitable to employ *linear predictor* (*LP*) and *polynomial predictor* (*PP*) models for ISE XU data. We have observed that *global MLP* structure could not capture the *FTS data* and thus, they are not suitable for modeling the risk of the series. But, *MoE* model divides the data into segments that are dealt with local experts and it merges the results using a gating network which gives a more successful and an applicable result in order to give decision while trading. The smoothed-piecewise prediction solution that it introduces making a better focus on the areas of *FTS*. *RNN* with feedback connections from hidden units cannot provide a better result than the *MoE* structure in this case. Also, *EGARCH* method becomes inferior to the *MoE* structure. We plan to observe the prediction performance of *SVM*, *DBN* structures on ISE data. We also plan to study on intraday ISE data.

Acknowledgement

We express our thanks to Boğaziçi University Research Fund due to their support (BAP 04A103 project).

References

Black, F., 1976. Studies of stock market volatility changes. In: Proceedings of 1976, Meetings of the American Statistical Association, Business and Economic Statistics Section, pp. 177–181.

Bollerslev, T., 1986. Generalized autoregressive conditional heteroskedasticity. J. Econom. 31, 307–327.

Castillo, O., Melin, P., 2002. Hybrid intelligent systems for financial time series prediction using neural networks, fuzzy logic and fractal theory. IEEE Trans. Neural Networks 13 (6), 1395–1408.

Chan, L., Gao, X., 2000. An algorithm for trading and portfolio management using *Q*-learning and sharp ratio maximization. In: proceedings of International Conference on Neural Information Processing, ICONIP 2000, Taejon, Korea, pp. 832–837.

Engle, R., 1982. Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation. Econometrica 50, 987–1007.

Engle, R.F., Ng, V.K., 1993. Measuring and testing the impact of news on volatility. J. Finance 48 (5), 1749–1778.

Heckerman, D., 1999. A Tutorial on Learning with Bayesian Networks. MIT Press, Cambridge, MA, USA.

Hellstrom, T., Holmstrom, K., 1997. Predictable Pattern in Stock Returns, Technical Report Series, IMA-TOM, 1997-09.

Nelson, D., 1991. Conditional heteroskedasticity in asset returns: A new approach. Econometrica 59, 349–370.

Nuller, K.-R., Smola, A.J., Ratsch, G., Scholkopf, B., Kohlmorgen, J., Vapnik, V., 1997. Predicting time series with support vector machine. In: Proceedings of ICANN'97. Lecture Notes in Computer Science, 1327. Springer, pp. 999–1004.

Petridis, V., Kehagias, A., 1998. Predictive Modular Neural Networks: Time Series Applications. Kluwer.

Petridis, V., Kehagias, A., Petrou, L., Bakirtzis, A., Maslaris, N., Kiartzis, S., Panagiotou, H., 2001. A Bayesian multiple models combination method for time series prediction. J. Intell. Robot. Syst. 31, 69–89.

Rao, A.V., Miller, D., Rose, K., Gersho, A., 1997. Mixture of experts regression modeling by deterministic annealing. IEEE Trans. Signal Process. 45 (11), 1997.

Schittenkopf, C., Dorffner, G., 1999. Risk Neutral Density Extraction from Option Prices: Improved Pricing with Mixture Density Networks.

Shafer, G., Vovk, V., 2001. Probability and Finance: It's Only a Game. John Wiley, New York.

Shawe-Taylor, J., Cristianini, N., 2004. Kernel Methods for Pattern Kernel Methods for Pattern Analysis. Cambridge University Press.

Tino, P., Schittenkopf, C., Dorffner, G., 2000. Volatility Trading via Temporal Pattern Recognition in Quantized Financial Time Series.

Yao, J.T., Tan, C.L., 2001. Guidelines for Financial Forecasting with Neural Networks.

Yümlü, S., Gürgen, F., Okay, N., 2003. In: Financial Time Series Prediction Using Mixture of Experts, ISCIS 2003Lecture Notes in Computer Science, 2869. Springer Verlag, pp. 553–560.

Yümlü, S., Gürgen, F., Okay, N., 2004. Turkish Market Analysis using Mixture of Experts, Engineering in Intelligent Systems (EIS 2004) Conference Proceedings (CD), March 2004.