Comparison of finite mixture of ARMA-GARCH, back propagation neural networks and support-vector machines in forecasting financial returns

*Altaf Hossain, M. Nasser** and M. Arifur Rahman***

*Department of Statistics, Rajshahi University, Rajshahi-6205, Bangladesh
**Institute of Mathematical Sciences, University of Malaya, Kuala Lumpur 50603, Malaysia/ Department of Statistics, Rajshahi University, Rajshahi-6205, Bangladesh
*** Department of Finance & Banking, Rajshahi University, Rajshahi-6205, Bangladesh

Abstract— The use of GARCH type models and computational intelligence based techniques for forecasting financial time series has been proved extremely successful in recent times. In this article we apply finite mixture of ARMA-GARCH model instead of AR or ARMA model to compare with the standard BP and SVM in forecasting financial time series (daily stock market index returns and exchange rate returns). We do not apply pure GARCH model as the finite mixture of ARMA-GARCH model outperforms the pure GARCH model. These models are evaluated on five performance metrics or criteria. Our experiment shows that SVM model outperforms both the finite mixture of ARMA – GARCH and BP models in deviation performance criteria. In direction performance criteria, the finite mixture of ARMA – GARCH model performs better. The memory property of these forecasting techniques is also examined using the behavior of forecasted values vis-à-vis the original values. Only the SVM model shows long memory property in forecasting financial returns.

***Index Terms*—**Autoregressive Moving Average (ARMA), Generalized Autoregressive Conditional Heteroskedastic (GARCH), Back Propagation (BP) Artificial Neural Network (ANN), Support Vector Machine (SVM).

1. Introduction

In the business and economic environment, it is very important to predict various kinds of financial variables accurately in order to develop proper strategies and avoid the risk of potential large losses. In the literature, there are some classical methods that have been developed in predicting financial time series. Among them, there is a powerful method, available in the literature for univariate time series forecasting, known as Box-Jenkins (1976) ARMA approach on stationary time-series. In conventional econometric models, the variance of the disturbance is assumed to be constant. But many economic and financial time series such as exchange rates, stock market indices, market returns, inflation rate etc exhibit periods of unusual large volatility, followed by periods of relative tranquility (i.e., time series exhibits clustering of large and small disturbances). Such circumstances suggest a form of heteroskedasticity in which the variance of the disturbance depends on the size of preceding disturbance and hence the conditional variance is non-constant over the sample period. Engle (1982) showed that it is possible to model the mean and the variance of a series simultaneously by a model named Autoregressive Conditional Heteroskedastic (ARCH); whereas Box-Jenkins (1976) modeled only the mean series. In latter, Bollerslev (1986) extended Engle's original work by developing a technique that allows the conditional variance to be an ARMA process and that extended process is known as the GARCH process. To deal with the intricacy specially, Wong et al. (1998) adopted the well – known GARCH model in the form of the so – called mixture of AR – GARCH model in exchange rate prediction. Again Tang et al. (2003) explored the mixture of ARMA – GARCH model for stock price prediction. These time series models capture several important features of financial series, such as leptokurticity and volatility clustering– see Mikosch (2001) for a recent paper on GARCH and stochastic volatility models. Evidence on the forecasting ability of GARCH model is somewhat mixed. Anderson and Bollerslev (1998) showed that GARCH model provides good volatility forecast. Conversely, some empirical studies showed that GARCH model tends to give poor forecasting performances; for examples, Figlewski (1997), Cumby et al. (1993), Jorion (1995, 1996), Brailsford and Faff (1996), and McMillan et al. (2000).

To improve the forecasting ability of GARCH model, some alternative approaches have been advocated from the perspective of estimation and forecasting. Neural network (NN) is one such method. The classical methods are based on some specific assumptions, such as linearity; or on error distributions, such as normality. In this circumstance, ANNs were developed to meet the increasing demand that can predict, detect, classify and summarize the structure of variables and define the relationships between them – without relying too much on such assumptions. Not all relationships in economics and finance are direct. But the hidden layer of an ANN can capture all non-direct relationships between input and output variables. ANNs have been successfully applied in different areas, including marketing, retail, banking and finance, insurance telecommunications and operations management (Smith and Gupta (2000)). In recent years, NNs have been successfully used for forecasting financial time series. The reason for its popularity in finance can be attributed to the ability to approximate any nonlinear relationship with a reasonable degree of accuracy. However, NN suffers from a number of weaknesses including the need for a large number of controlling parameters, difficulty in obtaining a global solution and the danger of over-fitting (Tay and Cao (2001a)).

Recently, a novel Artificial Intelligence (AI) algorithm, called support vector machine (SVM), was developed by Vapnik and his co-workers (1995, 1997) and since then it has been gaining popularity due to many attractive features. While the traditional NN implements the empirical risk minimization (ERM) principle, SVM implements the structural risk minimization (SRM) principle which seeks to minimize the upper bound of the population risk using the concept of the Vapnik-Chervonenkis (VC) dimension, as opposed to ERM that minimizes the error on the in-sample estimating data. Based on SRM principle, SVM achieves a balance between the training error and generalization error, leading to better forecasting performance than traditional NN. Selecting the best model in SVM is equivalent to solving a quadratic programming, which gives SVM another merit of a unique global solution. SVM was originally developed for classification problems defined as support vector classification (SVC) and then extended to regression problems defined as support vector regression (SVR). Our problem is SVR problem; so both the terms, SVM and SVR carry the same meaning in our paper. A discussion of SVMs is given by Scholkopf and Smola (2002).

Many researchers have already published huge number of papers comparing Autoregressive (AR) or ARMA or GARCH model with NN and SVM in financial time series forecasting. There are some recent papers in this combination. For example, Chen and Wu (2006) compared SVMs and BPs taking AR as a benchmark in forecasting the six major Asian stock markets, and Chen et al. (2008) proposed recurrent SVR based on GARCH model comparing with moving average (MA), recurrent NN and parametric GARCH model in terms of their ability to forecast volatility. But they did not account of the finite mixture ARMA – GARCH model in this combination.

In this paper, our first contribution is to examine the capacity of finite mixture of ARMA-GARCH model comparing with the standard NN and simple SVR models as this combination brings new dimension in forecasting returns of the stock index and exchange rate. Second contribution is to consider both the stock market index and exchange rate in our empirical study and try to find out the answers of the two questions – whether there is any difference between stock market index and exchange rate in nature/property, and whether there is any difference in the used techniques' performance. The last contribution is to check the memory property of the underlying models.

The rest of the paper is organized as follows. The related works are reviewed in Section 2. The methodology including three prediction methods in this paper is discussed in Section 3. The data source and their properties are introduced in Section 4. The results and discussion are presented in Section 5. Finally, our conclusion is in Section 6.

2. Review of related works

A number of research articles used ANN and SVM methods to forecast the price behavior of the financial commodities and compared the results with the traditional statistical methods. Refenes et al. (1995), for example, modeled stock returns with ANNs in the framework of Arbitage Pricing Theory (APT) and compared their study with regression models. Steiner and Wittkemper (1995) investigated the performance of several ANN models that forecast the return of a single stock. Tay and Cao (2001b) examined the feasibility of SVM in financial time series forecasting by comparing it with a multilayer BP neural Network (BPNN) and they showed that SVM outperforms the BP neural network. Kim and Han (2001) showed that SVM provides a promising alternative to stock market prediction comparing it with BP neural networks. Kamruzzaman and Sarker (2003) modeled and forecasted currency of exchange rates using three ANN based model and a comparison was made with traditional Autoregressive Integrated Moving Average (ARIMA) model, and they showed that all the ANN based models outperform ARIMA model. Ince and Trafalis (2006) proposed a two stage forecasting model which incorporates parametric techniques such as ARIMA, Vector Autoregressive (VAR) and co-integration techniques, and non parametric techniques such as SVR and ANN. Comparison of these models showed that the SVR outperformed the ANN. Chen et al. (2006) compared SVMs and BPs taking AR as a benchmark in forecasting the six major Asian stock markets. They showed that both the SVMs and BPs outperform the traditional models, ARs.

There are huge numbers of recent articles, where authors successfully tried to improve the forecasting ability of GARCH family models with ANNs and SVMs in forecasting stock return volatility. Some of these important papers are reviewed as follows. Donaldson and Kamstra (1997) constructed a semi-parametric nonlinear GARCH model based on the ANNs in forecasting stock return volatility. It was revealed that ANN model captures volatility effects over looked by GARCH, EGARCH (Exponential GARCH) and

GJR (Glosten, Jagannathan and Runkle) models. Hamid and Iqbal (2004) compared volatility forecasts from NNs with implied volatility from S&P 500 Index futures options using Barone – Adesi and Whaley (BAW) American futures options pricing model. Forecasts from NNs outperformed implied volatility forecasts and were not found to be significantly different from realized volatility like implied volatility forecasts. Wang (2009) integrated a new hybrid asymmetric volatility approach, Grey – GJR – GARCH into ANNs option – pricing model to improve forecasting ability and then the nonlinear NN forecast models with Grey – GJR –GARCH volatility approach achieved better forecasting performance than that with GARCH and traditional GJR – GARCH volatility approach in forecasting Taiwan stock index option price. Liang et al. (2009) modified three conventional parametric methods, the binomial tree, the finite difference and the Monte Carlo, allowing them to forecast the option prices and they employed the NNs and SVRs at the second stage to further decrease the forecasting errors of the parametric methods. Although the forecasting errors given by the conventional methods were already small, their experiment showed that the NNs and the SVRs are generally able to reduce them further still. Bildirici and Ersin (2009) enhanced GARCH family model, APGARCH (Asymmetric Power GARCH) with ANN to evaluate the volatility of the daily returns in Istanbul Stock Exchange. The proposed extended version ANN – APGARCH model improved forecast results.

In most of studies like the above, they showed that SVM (even simple version) outperformed both BP neural network and traditional statistical models like GARCH type. The simple neural learning procedures, such as BP algorithms easily outperformed the best practice of traditional statistical models. Again the forecasting ability of the

GARCH type models can be improved by hybridizing with AI models. Wong et al. (1998) proposed the finite mixture of AR-GARCH model and it performed better than pure GARCH in financial prediction. Again Tang et al. (2003) extended the mixture of AR-GARCH model to the mixture of ARMA-GARCH model and it performed better than the mixture of AR-GARCH models in prediction. Chen et al. (2008) proposed recurrent SVR based on GARCH model comparing with a moving average (MA), a recurrent NN and a parametric GARCH in terms of their ability to forecast; they did not account of the finite mixture ARMA-GARCH. There are several improved versions of ANN and SVM that we could use in our comparative study. But our goal is to examine the capacity of finite mixture of ARMA-GARCH comparing with the standard NN and simple SVM as this combination brings new dimension in forecasting financial returns.

3. Methodology

3.1. Finite mixture of ARMA-GARCH for time series forecasting

3.1.1. Introduction to ARCH and GARCH modeling

Engle (1982) showed that the serial correlation in squared returns, or conditional heteroskedasticity, can be modeled using an autoregressive conditional heteroskedasticity (ARCH) model of the form

$$y_t = E_{t-1}[y_t] + \in_t \tag{1}$$

$$\in_t = z_t \sigma_t \tag{2}$$

$$\sigma_t^2 = \delta_0 + \sum_{q=1}^{Q} \delta_q \in_{t-q}^2 \tag{3}$$

where $E_{t-1}[.]$ represents expectation conditional on information available at time t – 1 and z_t is a sequence of iid random variables with mean zero and unit variance. In the ARCH model z_t is assumed to be iid standard normal. The representation (1) - (3) is convenient for deriving properties of the model as well as for specifying the likelihood function for estimation.

An important extension of the ARCH model proposed by Bollerslev (1986) replaces the AR (P) representation in (3) with an ARMA (Q, P) formulation

$$\sigma_t^2 = \delta_0 + \sum_{q=1}^{Q} \delta_q \in_{t-q}^2 + \sum_{p=1}^{P} \beta_p \sigma_{t-p}^2 \tag{4}$$

The model in (4) together with (1)-(2) is known as the generalized ARCH or GARCH (Q, P) model. A high-order ARCH model may have a more parsimonious GARCH representation that is much easier to identify and estimate. Usually a GARCH (1,1) model with only three parameters in the conditional variance equation is adequate to obtain a good model fit for financial time series. Indeed, Hansen and Lunde (2004), provided compelling evidence that it is difficult to find a volatility model that outperforms the simple GARCH (1,1). So we use GARCH (1,1) in this paper. For testing ARCH/GARCH effects, assessing the fit, diagnostic checks for model adequacy and estimation see Enders (2004).

3.1.2. The finite mixture of ARMA-GARCH model

The linear time series model like AR and ARMA can be combined with GARCH to model the dynamics of stock indices and their volatilities. The mixture of ARMA-GARCH model is similar to the mixture of AR-GARCH model proposed by Wong et al.

(1998). Specifically, each component of the mixture model can be denoted as a normal ARMA series

$$y_{t,j} = \sum_{r=1}^{R} b_{rj} y_{t-r,j} + \sum_{s=1}^{S} a_{sj} \in_{t-s,j} + \in_{t,j} \ , \tag{5}$$

Furthermore, each residual term $\in_{t,j}$ is assumed Gaussian white noise with variance denoted by the GARCH model

$$\sigma_{t,j}^2 = \delta_{0j} + \sum_{q=1}^{Q} \delta_{qj} \in_{t-q,j}^2 + \sum_{p=1}^{P} \beta_{pj} \sigma_{t-p,j}^2 \ , \tag{6}$$

where $\delta_{qj} > 0$ for q = 1, …, Q and $\beta_{pj} > 0$ for p = 1, …, P.

Mathematically, the finite mixture of ARMA – GARCH model can be denoted as a K – component Gaussian mixture model

$$P(y_t) = \sum_{j=1}^{K} \alpha_j G(y_t; \hat{y}_{t,j}, \sigma_{t,j}^2), \tag{7}$$

$$\hat{y}_{t,j} = \sum_{r=1}^{R} b_{rj} y_{t-r,j} + \sum_{s=1}^{S} a_{sj} \in_{t-s,j} \ , \tag{8}$$

where $\alpha_j > 0$ and $\sum_{j=1}^{K} \alpha_j = 1$.

Once the model has been learned, one – step ahead prediction can be done via taking expectation of y_t

$$E(y_t) = \alpha_1 \hat{y}_{t,1} + \ldots + \alpha_K \hat{y}_{t,K} \tag{9}$$

A generalized expectation-maximization (GEM) algorithm is used to learn the mixture model. See Tang et al. (2003) for details.

According to the above model principle, steps required for forecasting stock market indices or exchange rates are given below:

(1) **Data analysis:** Based on the historical stock market index or exchange rate data, analyze data feature, if series is non – stationary series, stabilizes it by log difference or periodical difference.

(2) **ARMA** model identification and parameter estimation: Identify Model and estimate parameter(s) according to series autocorrelation and partial correlation plot after stabilizing. For the detail theory of ARMA modeling see Box & Jenkins (1976) and Pankaraz (1991).

(3) **ARMA** model test. Test model by statistical hypothesis testing method, if model is effective, then go into forth step, otherwise come back second step to adjust model's order and establish model again.

(4) **ARCH** effect test, model identification and parameter estimation. Do **GARCH** effect test for residual series, identify model's order, estimate parameter, establish **ARMA – GARCH** model.

(5) **ARMA – GARCH** model test. Test model by statistical hypothesis testing method, if model is effective, go into sixth step, otherwise come back forth step to adjust **GARCH** model' order again. To select the order of **GARCH** model finally, we will also check the performance of that model on the validation set, which is well illustrated in subsection 5.1.

(6) Stock market indices or exchange rates forecasting. Forecast stock market indices or exchange rates according to established **ARMA – GARCH** model.

3.2. ANNs for time series forecasting

3.2.1. Introduction to ANNs

An ANN is a biologically inspired form of distributed computation. It simulates the functions of the nervous system by a composition of interconnected simple elements (artificial neurons) operating in parallel, as shown in Figure 1. An element is a simple structure that performs three basic functions: input, processing and output. ANNs can be organized into several different connection topologies and learning algorithms (Lippmann (1987)). In case of time-series, the series at different lags act as inputs of ANNs. The number of inputs to the network is constrained by the problem type, whereas the number of neurons in the output layer is constrained by the number of outputs required by the problem type. However, the number of hidden layers and the size of the layers are decided by the designer.

[Figure 1]

There are several different BP training algorithms with a variety of different computation and storage requirements. Kamruzzaman and Sarker (2003) used three algorithms, Standard Backpropagation (SBP), Scaled Conjugate Gradient (SCG) and Backpropagation with Baysian Regularization (BPR), in training the ANN. It was found that SCG based model performs best when measured on the two most commonly used metrics and showed competitive results when compared with BPR based model on other three metrics. BP is one of the most commonly used algorithms in financial research. Chen and Wu (2006) therefore used only the SBP and mentioned, “No single algorithm is best suited to all the problems.” So we also consider only the SBP for training the ANN models in our paper.

3.2.2 Back propagation learning

Backpropagation, or propagation of error, is a common method of teaching artificial neural networks how to perform a given task. It was first described by Paul Werbos in 1974, but it wasn't until 1986, through the work of Rumelhart et al. (1986), that it gained recognition, and it led to a "renaissance" in the field of artificial neural network research. It is a supervised learning method, and is an implementation of the Delta rule. It requires a teacher that knows, or can calculate, the desired output for any given input. It is most useful for feed-forward networks (networks that have no feedback, or simply, that have no connections that loop). The term is an abbreviation for "backwards propagation of errors". Backpropagation requires that the activation function used by the artificial neurons (or "nodes") is differentiable. Actual algorithm for a 3-layer network (only one hidden layer):

Gradient-Decent

Each training example is a pair with the form $<\vec{x}, y>$, *where* $\vec{x}$ *is the vector of input values, o is the target output value and* η *is the learning rate*

Begin

Initialise each w_{ij} to some small random value

While

For each $<\vec{x}, y>$ in the training data set

Input the instance to the unit x and compute the output o

For each weight w_{ij}, compute and update

$$\Delta w_{ij}(t+1) = \Delta w_{ij}(t) + \eta \delta_j x_i'$$

if node j is an output node, then

$$\delta_j = o_j(1-o_j)(y_j - o_j)$$

if node j is an internal hidden node, then

$$\delta_j = x_j'(1-x_j')\sum_k \delta_k w_{jk}$$

Until convergence

End

Backpropagation is used to calculate the gradient of the error of the network with respect to the network's modifiable weights. This gradient is almost always then used in a simple stochastic gradient descent algorithm to find weights that minimize the error. Backpropagation usually allows quick convergence on satisfactory local minima for error in the kind of networks to which it is suited. See Rumelhart et al. (1986), Kamruzzaman and Sarker (2003), and Chen and Wu (2006) for details

3.3. SVMs for time series forecasting

3.3.1 Introduction to SVMs

The SVM, originally developed as an implementation of Vapnik's Structural Risk Minimization (SRM) principle (Vapnik (1995)), is now being used to solve a variety of learning, classification and prediction problems. It has the following advantages over ANN: (1) it can obtain the global optimum and (2) the overfitting problem can be easily controlled.

The SVM deals with the classification and regression problems by mapping the input data into the higher-dimensional feature spaces. Its central feature is that the regression surface can be determined by a subset of points or Support-Vectors (SV); all other points are not important in determining the surface of the regression. Vapnik introduced a ε -insensitive zone in the error loss function (Figure 2). Training vectors that lie within this zone are deemed correct, whereas those that lie outside the zone are deemed incorrect and contribute to the error loss function. As with classification, these incorrect vectors also become the support vector set. Vectors lying on the dotted line are SV, whereas those within the ε -insensitive zone are not important in terms of the regression function.

[Figure 2]

3.3.2. Support-vector Regression

The SVR algorithm tries to construct a linear function such that training points lie within a distance ε (Figure 2).

Given a set of training data $\{(x_1, y_1),...,(x_l, y_l)\} \subset X \times R$, where X denotes the space of the input patterns, the goal of SVR is to find a function $f(x)$ that has at most ε deviation from the targets y_i for all the training data and, at the same time, is as flat as possible.

Let the linear function f takes the form:

$$f(x) = < w, x > + b; w \in X, b \in R \tag{10}$$

The optimal regression function is given by the minimum of the functional,

$$\Phi(w, \xi) = \frac{1}{2}\|w\|^2 + C\sum_i (\xi_i^- + \xi_i^+), \tag{11}$$

where C is pre-specified value, and ξ^-, ξ^+ are slack variables representing upper and lower constraints on the outputs of the system. Flatness in (10) means a smaller $\|w\|$.

Using an ε-insensitive loss function,

$$L_\varepsilon(y) = \begin{cases} 0 & for\ |f(x) - y| < \varepsilon \\ |f(x) - y| - \varepsilon & otherwise \end{cases} \tag{12}$$

the solution is given by,

$$\begin{aligned} \max_{\alpha,\alpha^*} W(\alpha, \alpha^*) &= \max_{\alpha,\alpha^*} -\frac{1}{2}\sum_{i,j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) < x_i, x_j > \\ &+ \sum_{i=1}^{l} \alpha_i (y_i - \varepsilon) - \alpha_i^*(y_i + \varepsilon) \end{aligned} \tag{13}$$

with constraints,

$$0 \le \alpha_i, \alpha_i^* \le C, i = 1,2,...,l$$
$$\sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0. \tag{14}$$

Solving equation of (13) with constraints equation (14) determine the Lagrange multipliers, α, α^* and the regression function is given by (10), where

$$\bar{w} = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) x_i$$
$$\bar{b} = -\frac{1}{2} < \bar{w}, (x_r + x_s) > . \tag{15}$$

w is determine by training patterns x_i, which are SVs. In a sense, the complexity of the SVR is independent of the dimensions of the input space because it only depends on the number of SV.

To enable the SVR to predict a non-linear situation, we map the input data into a *feature space*. The mapping to the feature space F is denoted by

$$\Phi : \Re^n \to F$$
$$x \mapsto \Phi(x)$$

The optimization equation (13) can be written as

$$\max_{\alpha,\alpha^*} W(\alpha, \alpha^*) = \max_{\alpha,\alpha^*} -\frac{1}{2} \sum_{i,j=1}^{l} (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) < \Phi(x_i), \Phi(x_j) >$$
$$+ \sum_{i=1}^{l} \alpha_i (y_i - \varepsilon) - \alpha_i^* (y_i + \varepsilon)$$

The decision can be computed by the inner products, $< \Phi(x_i), \Phi(x_j) >$ without explicitly mapping to a higher dimension which is a time – consuming task. Hence the kernel function is as follows:

$$K(x, z) \equiv < \Phi(x), \Phi(z) >$$

By using a kernel function, it is possible to compute the SVR without explicitly mapping in the feature space. The condition for choosing kernel functions should confirm to Mercers condition, which allows the kernel substitutions to represent dot products in some Hilbert space.

3.4. Direct vs indirect multistep forecasting

When a one uses a model with a given periodicity but wishes to forecast at several, say, > 1, periods into the future, one is faced with a choice between iterating one-step ahead forecasts or directly modelling the relation between the end-of-sample observation and its lth successor in order to forecast the latter. This direct technique was first suggested by Cox (1961) in the context of exponential smoothing; Klein (1971) applied it to dynamic forecasting. Johnston (1974) was about to put end to the analysis by concluding that, when a quadratic loss function is used as criterion for both estimation and forecast accuracy, the former constitutes a "reliable indicator of prediction efficiency" and hence direct methods are inefficient for forecasting. But after of the works of Findley (1983) and then Weiss (1991) debate flared up again. Now-a-days the theoretical literature on this problem tends to emphasize the advantages of the direct over indirect forecasts. Iterated forecasts are more efficient if the model is correctly specified, but direct forecasts are more robust to model misspecification. Results of empirical investigations are mixed.

In this article we undertake iterative method.

4. Data and their properties

Two international leading daily stock market indices and two internationally important daily exchange rates are considered in our empirical study, which are collected from Thomson Datastream®. The names of the stock markets are Nikkei 225 from 1986.01.06 to 2009.04.30 and S & P 500 from 1986.01.02-2009.04.30, and the two exchange rate series are US – JP from 1986.01.03 to 2009.04.30 and US - U.K. from 1986.01.06 to 2009.04.30.

To illustrate the main empirical properties often observed in high – frequency financial time series, Table 1 contains descriptive statistics (including time period for each market) of four financial return series observed daily. The time periods cover many important economic events, which we believe are sufficient for the training models.

Let p_t be the observed daily price at time t and y_t the corresponding daily return defined by

$$y_t = 100 \times (\ln p_t - \ln p_{t-1})$$

[Table 1]

In Table 1, it is possible to observe that all the series show almost zero means and excess kurtosis (always above 3) for the normal distribution value. It can also be observed that the standard deviation, skewness and kurtosis for index returns are larger than that for the exchange rate returns. That is, stock market indices are generally more volatile than the exchange rates.

The return series of Nike 255 and S&P 500 are plotted in Figure 3. It can readily be seen that the volatility concentrates itself in clusters, i.e. periods of high and low volatility can be observed in the data. We have also plotted the return series of US – JP and US – UK exchange rates in Figure 4. In case of currency rates, it can also be seen that the series are highly volatile like the return series of stock market indices but the index returns are more volatile than exchange rate returns.

[Figure 3]

[Figure 4]

Finally, ACFs (Autocorrelation Functions) of the return series for each market are depicted in Figure 5 and ACFs of the squared return series for each market are also depicted in Figure 6. The volatility clustering is reflected in the significant correlations of squared returns. The autocorrelation coefficients of squared returns are larger and last longer (persistent) than those of the return series (non - squared). We must point out that the return series show little or no correlation, but its squares show high – correlation; which indicates the ARCH or GARCH effect.

[Figure 5]

[Figure 6]

5. Results and discussion

5.1. Preprocessing and design of the models

We have transformed the stock market index and exchange rate into return series for each market. White noise series is the first requirement for a data series to which we can apply the classical statistical techniques as well as the AI models for better fitting. According to Thomason (1999a, b) and Tay and Cao (2001b), this transformation has many advantages. The most important is that the distribution of the transformed data becomes more symmetrical so that it follows a normal distribution more closely. This also improves the predictive power of neural network.

Each of the four data sets is partitioned into three subsets according to the time sequence in the ratio 95: 2.5: 2.5 in forecasting returns. The first part is used for training; the second is a validating set that selects optimal parameters for the SVR, identifies the orders of ARMA-GARCH models and prevents the overfitting found in the BP neural networks, and the last is used for testing.

In Standard BP models, we use one hidden layer of different units for each market. We select the number of inputs for the ANNs, and units for the hidden layer of that ANNs which give minimum Mean Squared Error (MSE) on the validation sets. Regression techniques like SVR or Radial Basis Function (RBF) nets can be used to estimate the prediction function on the basis of time – delay coordinates. For stationary dynamical systems the embedding parameters can be found e.g. by the method of Liebert et al. (1991).

5.2. Performance criteria

Although the MSE is a perfectly acceptable measure of performance, in practice the ultimate goal of any testing strategy is to confirm that the results of models are robust and capable of measuring the profitability of a system. It is important, therefore, to design a test from the outset. According to Tay and Cao (2001b) and Thomason (1999a, b), the prediction performance is evaluated using the following statistics: MSE, Normalized Mean Squared Error (NMSE), Mean Absolute Error (MAE), Directional Symmetry (DS) and Weighted Directional Symmetry (WDS). These criteria are defined in the Table 2. See Chen and Wu (2006) for more details.

[Table 2]

5.3. BP and SVM implementation

The BP models used in this experiment are implemented using the R-2.8.1-win32's tsDyn Package. We use the conjugate gradient approach. The architecture of the BP model is as follows: the optimum number of hidden nodes is 4 for each market in forecasting returns that minimize the error rate on the validation sets is determined market – wise. The activate function of the hidden layer is sigmoid and the output node uses the linear transfer function.

We apply Vapnik's SVM for regression by using the R-2.8.1-win32's e1071 Package. We use the radial basis kernel function $k(x, y) = \exp(-\gamma \|x - y\|^2)$ from several typical functions as it performs well under general smoothness assumptions. The set of parameters to be determined is (γ, C, ε). Method of choosing these parameters is an area of active research (Duan et al. (2003); Rossi and de Carvallo (2008)). We choose them following a version of the most popular method of cross validation. The optimum sets ((.005, 1, 0.001) for NIKE 255 and (.001, 1, 0.01) for S & P 500 index, and (.001, 1, 0.001) for US - JP and (.01, 1, 0.001) for US - UK exchange rates) in forecasting returns that minimize the error rate on the validation sets are determined market – wise.

5.4. Comparison and discussion

For each market, we also design different ordered ARMA models as to show the necessity of GARCH model. The ACF of squared residuals for each fitted ARMA indicates the presence of GARCH effects. The forecasting results of the ARMA-GARCH, BP and SVR for the test set are collated in Table 3. The results of ARMA are not reported in the table to make it simple and clear.

[Table 3]

In general, ARMA-GARCH, BP and SVR models outperform the ARMA model in the deviation performance criteria. However, in the either direction and weighted direction performance criteria, the ARMA model sometimes shows better results comparing with BP and SVR models, possibly because SVRs and BPs are trained in terms deviation performance; the former to find the error bound and the latter to minimize MSE. Thus, in the deviation performance criteria, they (including ARMA-GARCH models) perform better than the ARMA. With regard to the direction criteria, the time-series model has an intrinsic merit because it emphasizes the trend of the series. Therefore, the BP and SVR models, even ARMA-GARCH model sometimes cannot outperform ARMA in the direction criteria.

Our experiment also shows that SVR model outperforms both the finite mixture of ARMA – GARCH and BP models in the deviation performance criteria, MSE, MAE, and NMSE. In direction performance criteria, DS and WDS, the finite mixture of ARMA – GARCH models perform better. If we consider the deviation and direction criteria together, the BP model performs well with the SVR model.

It is remarkable that all the models comparatively perform better in forecasting exchange rate returns than that in forecasting stock market index returns under different performance criteria. These results support the statement that exchange rates are less volatile than stock market indices.

The deviation criteria, such as NMSEs are less than 1 for both the BP and SVR, except for the US - UK exchange rate in the BPs. But the NMSE of ARMA-GARCH is greater than 1 for each market.

Table 4 gives the comparison of the result of the ARMA-GARCH, BP and SVR models. In the NIKE 255 and US-JP data sets, the SVR models perform better than the ARMA-GARCH and BP models. In the remaining two markets, the BP method performs better than SVM and ARMA – GARCH models. Table 4 demonstrates that out of twenty places SVR wins in 9 places, ARMA-GARCH wins in 2 places, BP wins in 3 places and in the remaining 6 places, BP and ARMA-GARCH win together; which indicates that the fitted BP and ARMA-GARCH models are very close in prediction.

[Table 4]

In the Table 4, there are 8 places on the basis of directional performance criteria where ARMA-GARCH wins in 2 places and BP and ARMA – GARCH win together in the remaining 6 places; which indicates that the BP has also intrinsic merit like the time – series models, ARMA and ARMA-GARCH. But the SVR model does not have such intrinsic merit as it does not win in any place based on the directional criteria.

The forecasted returns of ARMA – GARCH models are plotted in Figures 7 and 8, BPs in Figures 9 and 10, and finally SVRs in Figures 11 and 12. It can be easily observed that SVR forecasts consistently over the whole period with time varying property for each market and SVRs are able to forecast stock and exchange returns well capturing volatility. That is, SVR holds the long memory property. However, on the other hand, both the GARCH-type and BP models give only the constant forecasts.

[Figure 7], [Figure 8]

[Figure 9], [Figure 10]

[Figure 11], [Figure 12]

6. Summary, conclusion and further research

In this article, we examine the feasibility of applying two AI models, SVM/SVR and standard BP, and one classical statistical model, finite mixture of ARMA – GARCH to financial returns series forecasting. Our experiments demonstrate that both models, SVR and BP perform better than the ARMA and ARMA - GARCH models in the deviation measurement criteria. But in the directional criteria, the ARMA – GARCH outperform both the SVR and BP models. Again BP model comparatively perform better than the SVR in the directional criteria. That is, BP perform well both in the deviation and directional criteria. All the models perform better in forecasting exchange rates than that in forecasting stock market indices under different performance criteria. It can be easily observed that SVR forecasts consistently over the whole period with time varying property for each market and SVRs are able to forecast stock market indices and exchange rates returns well capturing volatility; which property supports that SVR is a long memory model in forecasting financial returns. However, on the other hand, both the GARCH – type and BP models give only the constant forecasts.

The statistical methods, ARMA and ARMA-GARCH, require large number of sample size for better forecasting and these models drastically reduce the original sample size when the high order model is fitted. Though these statistical methods have not performed well in deviation performance criteria, our results support that these models have both the important interpretability and intrinsic properties. To improve the forecasting ability of ARMA - GARCH model, we can use NN methods as our results show that NN performs well in both the deviation and direction performance criteria. But the ANN structures contain hidden layers which are not interpretable. And the selection of optimal parameters for ANN is very difficult, that is, in the selection procedure, there is no art. To avoid the problems raised by ANN, we have also used simple autoregressive (AR) model based SVR model in this article. SVR model is quite simple and has the interpretability properties as opposed to the complex GARCH type and NN models. The simple AR model is estimated by maximum likelihood estimation (MLE) (which is usually affected by potential outliers), while the SVR model is estimated by robust estimation procedure. And the selection of optimal parameters for SVR is easier than NN. Rather, SVR model additionally shows long memory property supporting the statement that SVR model have the property of generalization power. In summary, our result suggest that the simple SVR model could be still used fairly successfully as a long memory model in forecasting financial returns

Our future study is to apply more complex GARCH methods and ANN with other training methods to the forecasting of Bangladesh stock markets' indices and also compare it with the SVMs. We will also use recurrent ANN and SVM in that comparative study. The estimation procedure of ARMA-GARCH model is not robust. But we want to use robust estimation procedure like MM to estimate the ARMA-GARCH models for more efficient prediction and then making comparison with the other methods.

Acknowledgment

We would like to express our special thanks to the authors of fGarch, tsDyn and e1071 packages of R which we have used to implement our results. In addition, we would like to thank the reviewers of the Journal of Applied Statistics and the audiences of the conferences, ICCIT, 2008 as well as IASC2008 for giving their valuable suggestions; as a result we have been able to improve and enhance the paper.

References

[1] Andersen, T.G. and Bollerslev, T. (1998), *Answering the Skeptics: Yes, Standard Volatility Models do Provide Accurate Forecasts*. International Economic Review, 39, pp. 885-905.

[2] Bildirici, M. and Ersin, Ö.Ö. (2009), Improving forecasts of GARCH family models with the *artificial neural networks*: An application to the daily returns in *Istanbul Stock Exchange.* Expert Systems with Applications, Vol. 36, Issue 4, pp. 7355 – 7362.

[3] Bollerslev, T. (1986), *Generalized autoregressive conditional heteroscedasticity.* Journal of Econometric 31, pp. 307-27.

[4] Box, G. E. P and Jenkins, G. M. (1976), *Time Series Analysis Forecasting and Control*. San Francisco: Holden-Day.

[5] Brailsford, T.J. and Faff, R.W. (1996), *An evaluation of volatility forecasting techniques.* Journal of Banking and Finance 20, pp. 419-438.

[6] Chen, W-H., Shih, J-Y. and Wu, S. (2006), *Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets.* Int. J. Electronic Finance, Vol. 1, No. 1, pp.49-67.

[7] Chen, S., Jeong, K. and Härdle, W. (2008), *Support Vector Regression Based GARCH Model with Application to Forecasting Volatility of Financial Returns.* SFB

649 Discussion Paper 2008-014, http://edoc.hu-berlin.de/series/sfb-649-papers/2008-14/PDF/14.pdf

[8] Cox, D.R. (1961), *Prediction by Exponentially Weighted Moving Averages and Related Methods*. JRSS, B23, pp. 414-422.

[9] Cumby, R., Figlewski, S. and Hasbrouck, J. (1993), *Forecasting volatility and correlations with EGARCH models*. Journal of Derivatives winter, pp. 51-63.

[10] Donaldson, R.G. and Kamstra, M. (1997), *An artificial neural network-GARCH model for international stock return volatility*. Journal of Empirical Finance, Vol. 4, Issue 1, pp. 17-46.

[11] Duan, K., Keerthi, S.S., and Poo, A.N. (2003), *Evaluation of simple performance measures for tuning SVM hyperparameters*. Neurocomputing 51, pp. 41 – 59.

[12] Enders, W. (2004), *Applied Econometric Time Series*, John Wiley & Sons, Inc..

[13] Engle, R.F. (1982), *Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation*. Econometrica 50, pp. 987-1007.

[14] Figlewski, S. (1997). *Forecasting volatility*. Financial Markets, Institutions and Instruments, Vol. 6, pp 1–88.

[15] Findley, D. F. (1983), *On the use of multiple models for multi-period forecasting. Proceedings of Business and Economic Statistics*. American Statistical Association, pp. 528–531.

[16] Hamid, S.A. and Iqbal, Z. (2004), *Using neural networks for forecasting volatility of S&P 500 Index futures prices*. Journal of Business Research, Vol. 57, Issue 10, pp. 1116-1125.

[17] Hansen, P. and Lunde, A. (2004), *A Forecast comparison of volatility models: does anything beat a GARCH (1,1) model?*. Journal of Applied Econometrics, Vol. 20, pp. 873-889.

[18] Ince, H. and Trafalis, T.B. (2006), *A hybrid model for exchange rate prediction. Decision Support Systems*, Vol. 42, Issue 2, pp.1054-1062, ISSN 0167-9236.

[19] Johnston, H. N. (1974), *A note on the estimation and prediction ine ciency of 'dynamic' estimators*. International Economic Review 15, pp. 251–255.

[20] Jorion, P. (1995), *Predicting volatility in the foreign exchange market*. Journal of Finance, Vol. 50, pp. 507–528.

[21] Jorion, P. (1996), *Risk and turnover in the foreign exchange market*. In The Microstructure of Foreign Exchange Markets, Franke JA, Galli G, Giovannini A (eds). Chicago University Press: Chicago.

[22] Kamruzzaman, J. and Sarker, R. (2003), Forecasting of currency exchange rates using ANN: a case study. Proc. IEEE Intl. Conf. on Neur. Net. & Sign. Process. (ICNNSP03), China.

[23] Kim, K-S. and Han, I. (2001), *The cluster-indexing method for case-base reasoning using self-organizing maps and learning vector quantization for bond rating cases*. Expert Systems with Applications, Vol. 21, pp. 147-156.

[24] Klein, L. R. (1971), *An essay on the theory of economic prediction*. Chicago, IL: Markham.

[25] Liang, X., Zhang, H. and Chen, Y. (2009), *Improving option price forecasts with neural networks and support vector regressions*. Neurocomputing, Vol. 72, Issues 13-15, pp. 3055-3065.

[26] Lippmann, R.P. (1987), *An introduction to computing with neural nets.* IEEE ASSP Magazine, pp.36-54.

[27] Liebert, W., Pawelzik, K. and Schuster, H.G. (1991), *Optimal embeddings of chaotic attractors from topological considerations*. Europhys. Lett., Vol. 14, pp. 521 – 526.

[28] McMillan, D.G., Speight, A.E.H. and Gwilym, O. (2000). *Forecasting UK stock market volatility: a comparative analysis of alternate methods*. Applied Financial Economics 10, pp. 435-448.

[29] Mikosch, T. (2001), Modeling financial time series. Notes for lecture on '*Modeling heavy tails and dependence in financial data*' presented to the meeting on 'New Directions in Time Series Analysis', Centre International de Rencontres Mathe ´matiques, Luminy, France.

[30] Pankratz, A. (1991), *Forecasting with Dynamic Regression Models*. John Wiely & Sons Inc..

[31] Refenes, A.P., Zapranis, A.D. and Francis, G. (1995), *Modeling stock returns in the framework of APT: a comparative study with regression models.* Neural Networks in the Capital Markets, pp.101-125.

[32] Rossi, A.L.D, and Carvalho, A.C.P.L.F. (2008), *Bio-inspired Optimization Techniques for SVM Parameter Tuning*. 10th Brazilian Symposium on Neural Networks.

[33] Rumelhart, D.E., McClelland, J.L. and the PDP research group (1986), *Parallel Distributed Processing*. vol. 1, MIT Press, 1986.

[34] Smith, K.A. and Gupta, G.N.D. (2000), *Neural networks in business: techniques and applications for the operations research*. Computers and Operations Research, Vol. 27, pp.1023-1044.

[35] Scholkopf, B. and Smola, A.J. (2002), *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, Cambridge: Massachusetts.

[36] Steiner, M. and Wittkemper, H-G. (1995), *Neural Networks as an alternative stock market model*. Neural Networks in the Capital Markets, pp.135-147.

[37] Tay, F. and Cao, L. (2001a). *Financial forecasting using support vector machines.* Neural Computation and Application 10: pp. 184-192.

[38] Tay, F.E.H. and Cao, L. (2001b), *Application of support vector machines in financial time-series forecasting*. Omega, Vol. 29, pp.309-317.

[39] Tang, H., Chun, K.C. and Xu, L. (2003), *Finite Mixture of ARMA-GARCH Model for Stock Price Prediction*, Proc. Of 3rd International Workshop on Computational Intelligence in Economics and Finance (CIEF2003), North Carolina, USA, pp. 1112-1119.

[40] Thomason, M. (1999a), *The practitioner method and tools: a basic neural network-based trading system project revisited (parts 1 and 2)*. Journal of Computational Intelligence in Finance, Vol. 7, No. 3, pp. 36-45.

[41] Thomason, M. (1999b). *The practitioner method and tools: a basic neural network-based trading system project revisited (parts 3 and 4)*. Journal of Computational Intelligence in Finance, Vol. 7, No. 3, pp. 35-48.

[42] Vapnik, V.N. (1995), *The Nature of Statistical Learning Theory*. 2nd ed., New York: Sringer-Verlag.

[43] Vapnik, V.N. (2001), *Statistical Learning Theory*. Wiley, New York.

[44] Wang, Y.H. (2009), *Nonlinear neural network forecasting model for stock index option price: Hybrid GJR–GARCH approach*. Expert Systems with Applications, Vol. 36, Issue 1, pp. 564-570.

[45] Weiss, A.A. (1991), *Multi-step Estimation and Forecasting in Dynamic Models.* Journal of Econometrics, Vol. 48, pp. 135-149.

[46] Wong, W.C., Yip, F. and Xu, L. (1998), Financial *Prediction by Finite Mixture GARCH Model*. in Proceeding of Fifth International Conference on Neural Information Processing, pp. 1351-1354.

Appendix

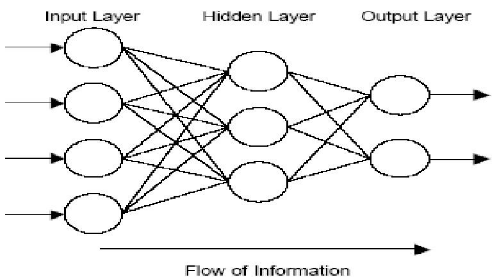

Figure 1 A neural network with one hidden layer

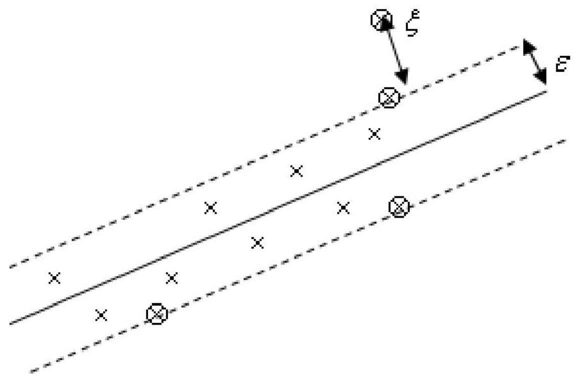

Figure 2 Approximation function (solid line) of the SV regression using a ε -insensitive zone

Table 1: Summary statistics of index and exchange rate returns for each market

SERIES	TIME PERIOD	N	MEAN	STDEV	SKEWNESS	KURTOSIS
NIKE 255	1986.01.06-2009.04.30	5745	− 0.00685	1.5135	− 0.23	**7.97**
S & P 500	1986.01.02-2009.04.30	5879	0.0241	1.2033	− 1.40	**30.31**
US-JP	1986.01.03-2009.04.30	6085	0.01151	0.71934	0.51	**7.16**
US-UK	1986.01.06-2009.04.30	6084	0.000452	0.62549	−0.31	**3.30**

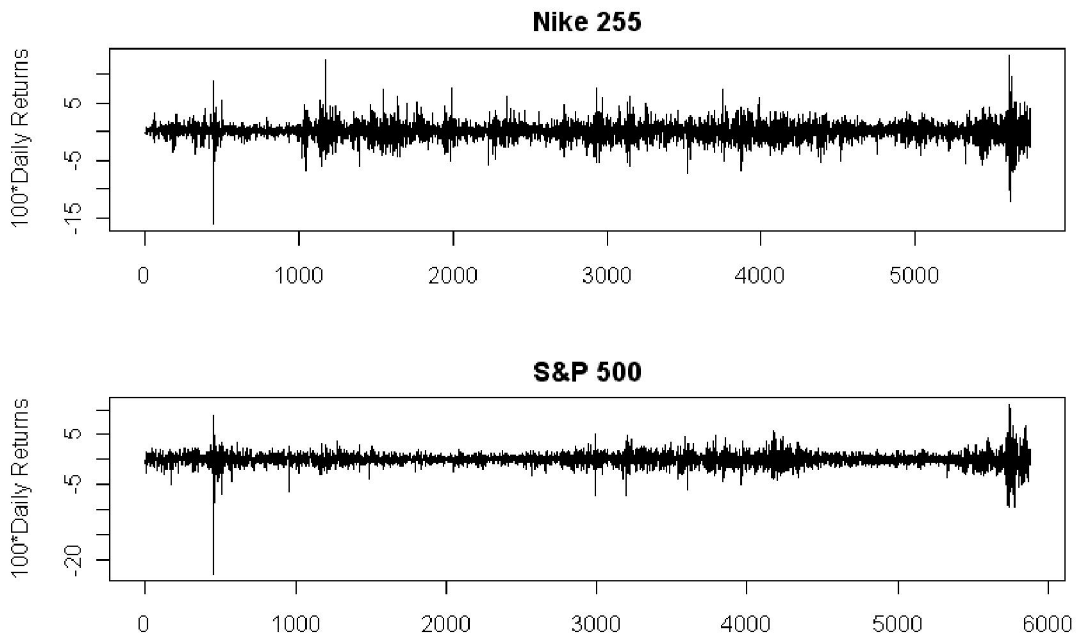

Figure 3: Daily returns of index price for two markets

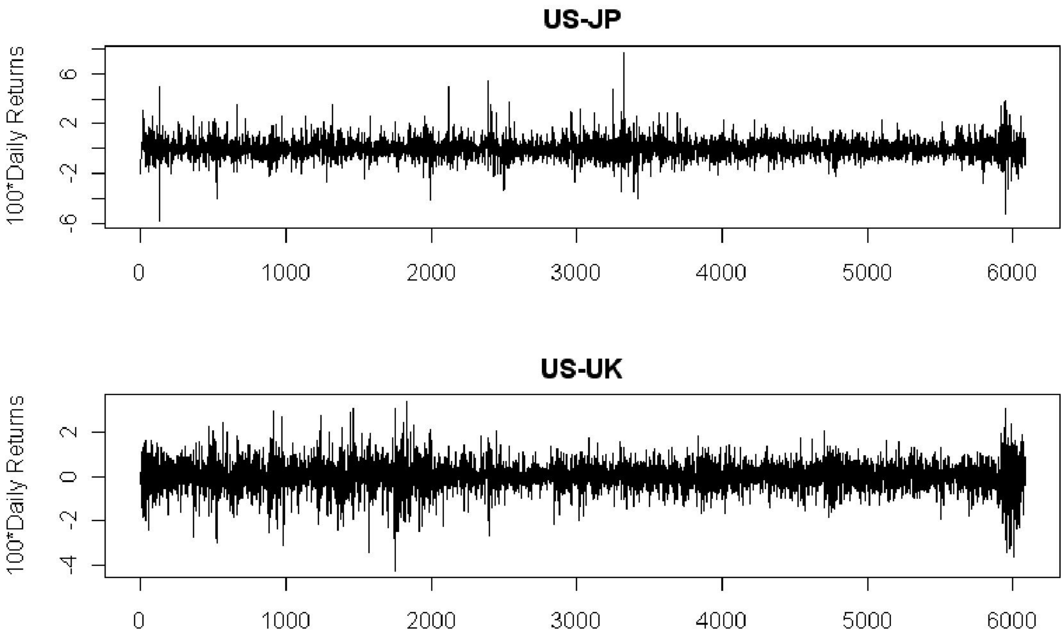

Figure 4: Daily returns of exchange rate for two currencies

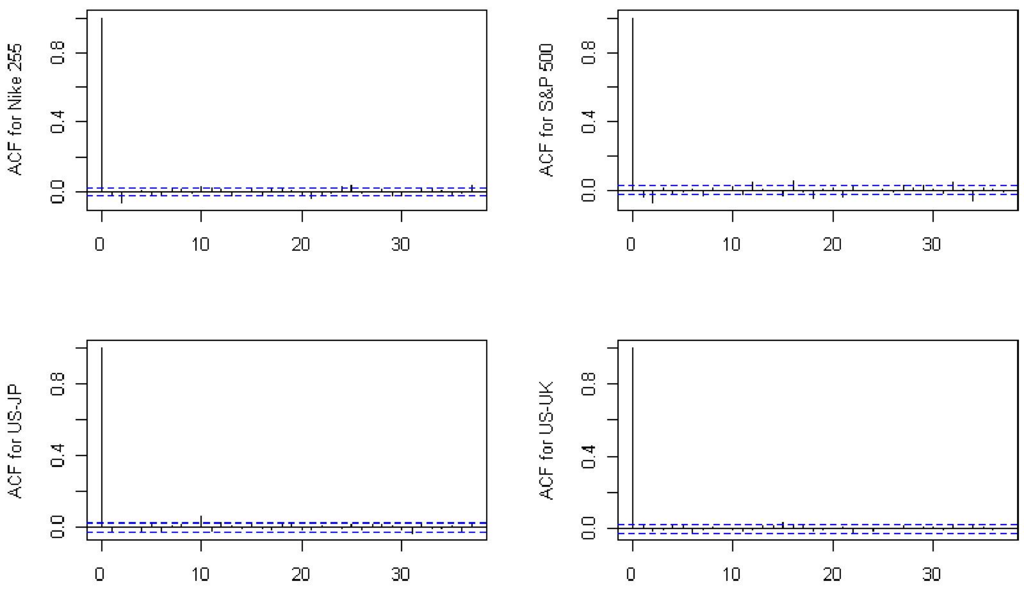

Figure 5: ACFs of daily returns for each market

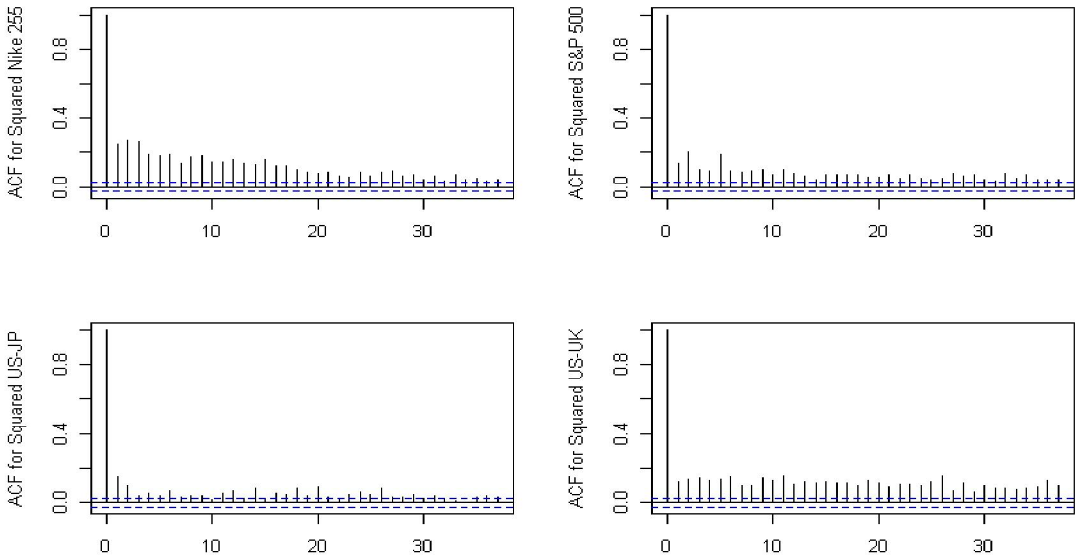

Figure 6: ACFs of daily squared returns for each market

Table2: Performance metrics and their calculation

METRICS	CALCULATION
MSE	$MSE = \frac{1}{n}\sum_{i=1}^{n}(a_i - p_i)^2$
NMSE	$NMSE = \frac{1}{\delta^2 n}\sum_{i=1}^{n}(a_i - p_i)^2$ $\delta^2 = \frac{1}{n-1}\sum_{i=1}^{n}(a_i - \bar{a})^2$
MAE	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert a_i - p_i \rvert$
DS	$DS = \frac{100}{n}\sum_{i=1}^{n} d_i$ $d_i = \begin{cases} 1 & (a_i - a_{i-1})(p_i - p_{i-1}) \geq 0 \\ 0 & otherwise \end{cases}$
WDS	$WDS = \sum_{i=1}^{n} d_i \lvert a_i - p_i \rvert / \sum_{i=1}^{n} d'_i \lvert a_i - p_i \rvert$ $d_i = \begin{cases} 0 & (a_i - a_{i-1})(p_i - p_{i-1}) \geq 0 \\ 1 & otherwise \end{cases}$ $d'_i = \begin{cases} 1 & (a_i - a_{i-1})(p_i - p_{i-1}) \geq 0 \\ 0 & otherwise \end{cases}$

Table 3: Comparison of the results of ARMA-GARCH, NN and SVR on the test set

SERIES	MODELS	SQRT MSE	MAE	NMSE	DS	WDS
NIKE 255	ARMA(2,0) – GARCH(1,1)	2.835186	1.987818	1.002047	99.651567	0.00
	BP	0.028308	0.019842	0.998972	66.550522	0.587614
	SVR	0.028148	0.019804	0.987713	57.839721	0.806003
S & P 500	ARMA(2,0) – GARCH(1,1)	2.681041	1.869955	1.002267	99.659863	0.00
	BP	0.026789	0.018699	1.000684	99.659863	0.00
	SVR	0.026632	0.018703	0.988980	62.585034	0.657628
US-JP	ARMA(2,0) – GARCH(1,1)	1.055699	0.782228	0.996894	99.671052	0.00
	BP	0.010556	0.007822	0.996757	99.671052	0.00
	SVR	0.010534	0.007797	0.992651	67.763157	0.577243
US-UK	ARMA(2,0) – GARCH(1,1)	1.011421	0.738053	1.008306	99.671052	0.00
	BP	0.010108	0.007379	1.007129	99.671052	0.00
	SVR	0.010534	0.007797	0.992651	67.763157	0.577243

Table 4: Comparison of ARMA-GARCH, BP and SVR models on the test set

	SQRT MSE	MAE	NMSE	DS	WDS	SUMMARY
NIKE 255	√	√	√	−	−	√
S & P 500	√	+	√	− +	− +	+
US-JP	√	√	√	− +	− +	√
US-UK	+	+	√	− +	− +	+

'−' represents ARMA-GARCHs that perform well and '+' represents BPs that perform well '√' represents SVRs that perform well.

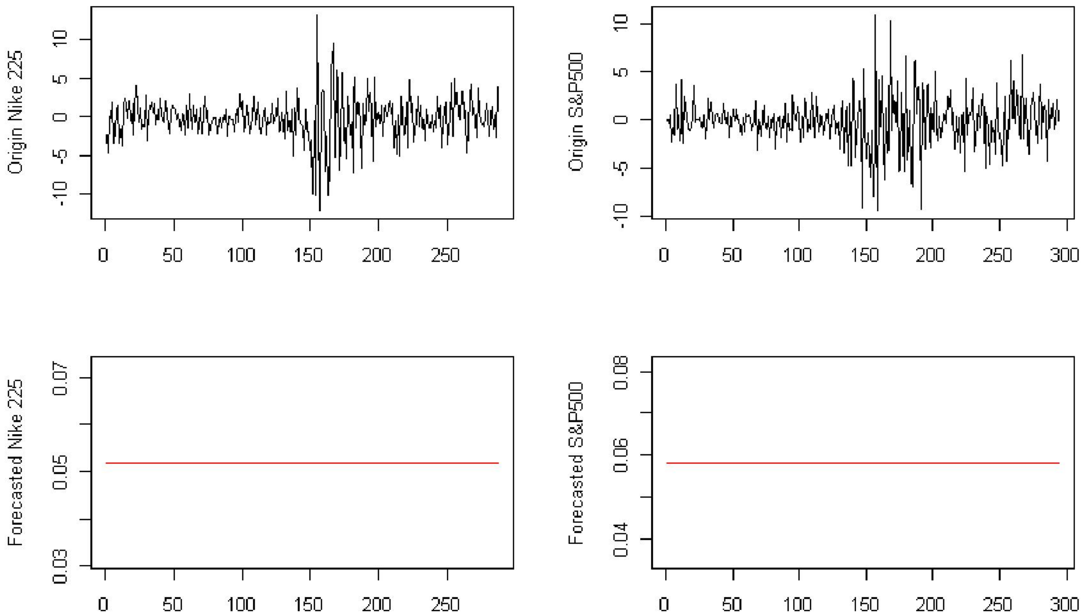

Figure 7: Returns forecasts of stock indices for ARMA – GARCH model

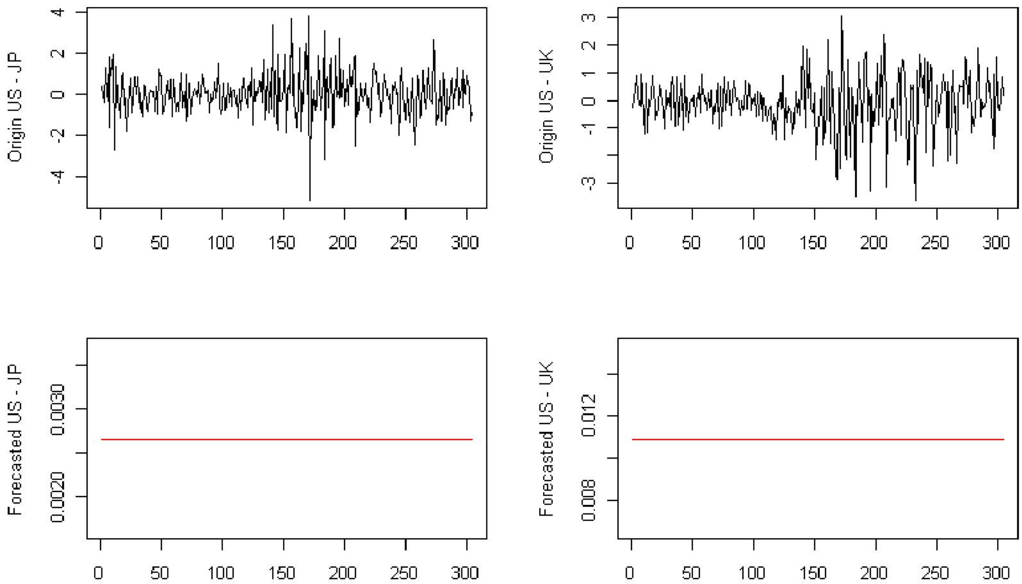

Figure 8: Returns forecasts of exchange rates for ARMA – GARCH model

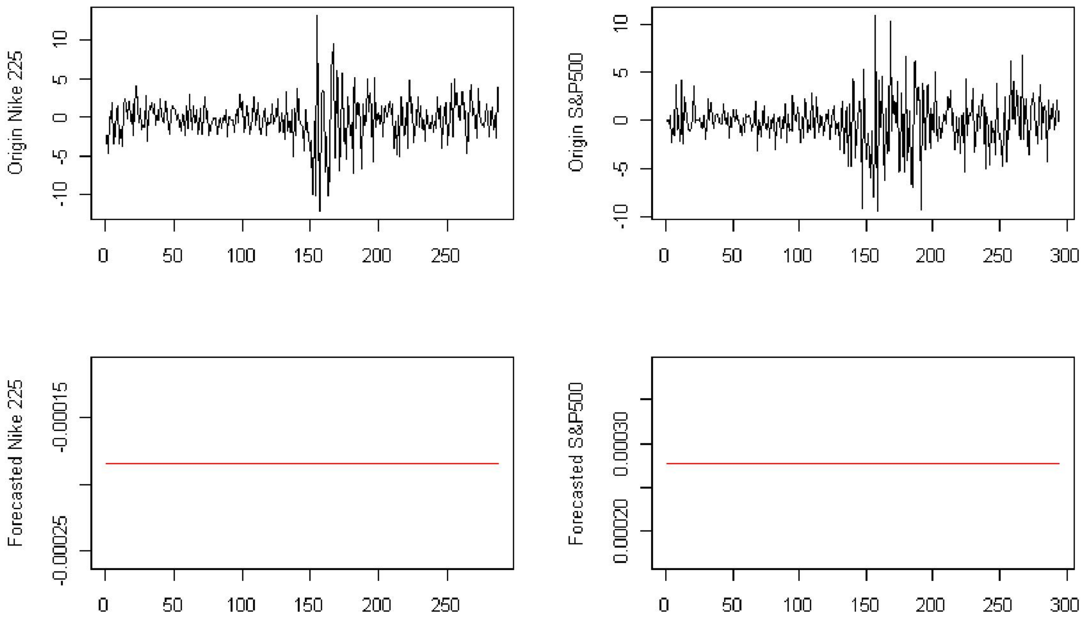

Figure 9: Returns forecasts of stock indices for BP NN model

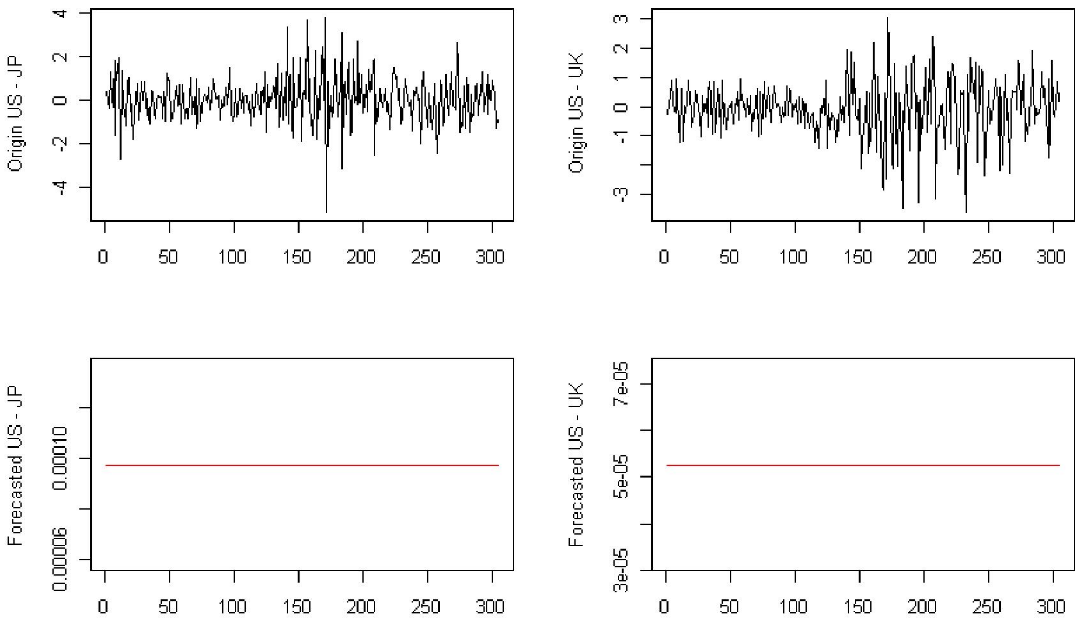

Figure 10: Returns forecasts of exchange rates for BP NN model

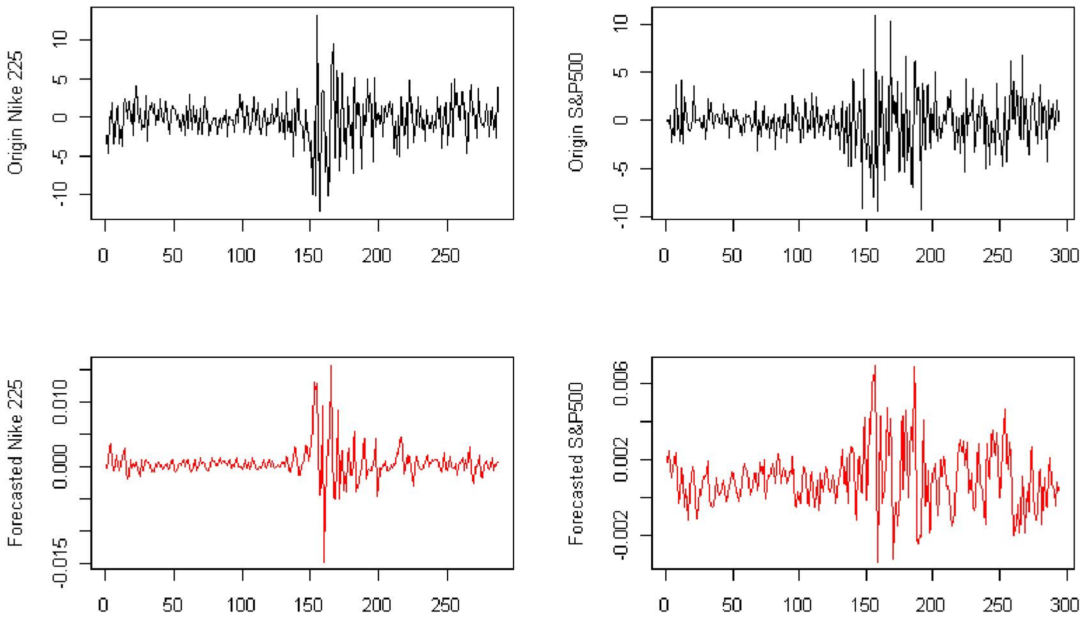

Figure 11: Returns forecasts of stock indices for SVR model

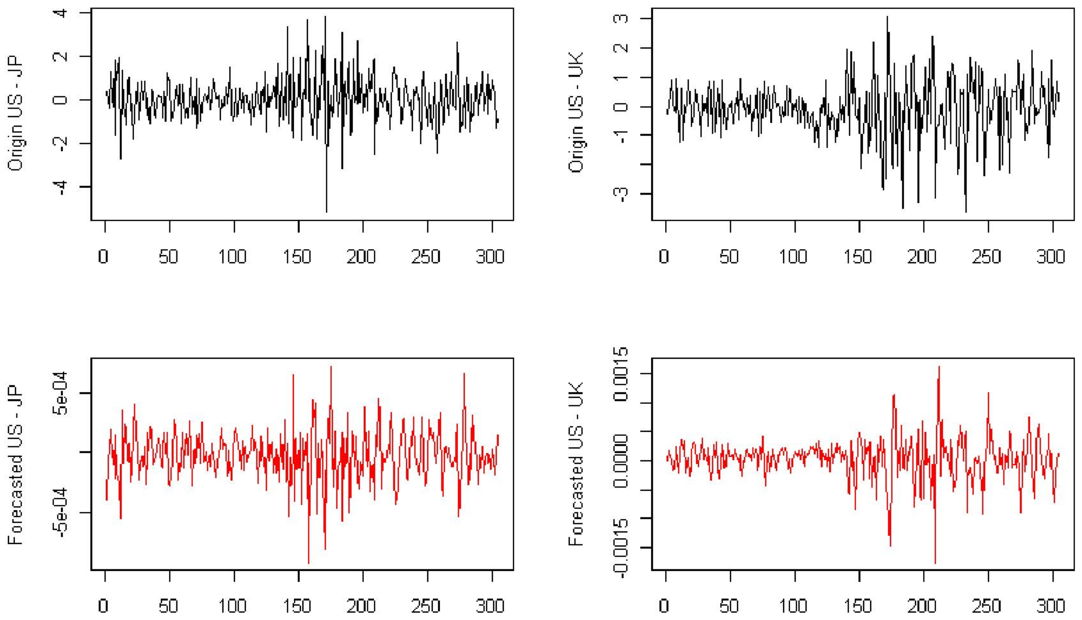

Figure 12: Returns forecasts of exchange rates for SVR model