Markov Switching Asymmetric GARCH Model and Artificial Neural Networks: Enhancing the volatility forecasting for S&P 500 Index

Abdellah Tahiri[a,*], Brahim Benaid[a], Hassane Bouzahir[a], Naushad Ali Mamode Khan[b]

[a] *ISTI Lab, ENSA PO Box 1136, Ibn Zohr University-Agadir, Morocco*
[b] *Department of Economics and Statistics, University of Mauritius, Mauritius*

Abstract

Financial time series exhibit different stylized facts, namely, asymmetry and nonlinearity, which require a particular specification to capture market volatility behavior. This paper suggests Back-propagation neural networks (BPNN) to improve the forecast accuracy for the S&P 500 returns volatility. The estimated volatility based on the Markov-Switching asymmetric GJR-GARCH (MS GJR-GARCH) model and the VIX index (i.e., Volatility index) series are used respectively as input and output of our neural networks model. The results reveal that the neural networks have succeeded in enhancing the forecast accuracy of the MS GJR-GARCH model according to Mean Squared Error (MSE) and Mean Absolute Error (MAE) criteria.

Keywords: Volatility; Time series; Back-propagation; Artificial Neural networks; Markov switching; GJR-GARCH.

JEL Classification: C1; C45; C5.

1. Introduction

Volatility forecasting has been a special issue regarding financial time series in general and, more precisely, for the hedging and pricing financial derivatives such as the options. Since the constant volatility model of Black & Scholes, several trails to model volatility are well documented. In particular, the Generalized Auto-Regression Conditional Heteroskedasticity (GARCH) proposed by Bollerslev (1986)[9] as the most popular econometric model, as indicated by its name, it was extended from the ARCH model of Engel (1982)[12].
The financial time series is characterized by different stylized facts, mainly persistence in conditional variance process, asymmetry, and nonlinearity, making volatility forecasting more difficult. From the business perspective, the bad news (negative shocks) affect the conditional variance more than the good news (positive shocks). Glosten-Jagannathan-Runkle (1993)[14] suggested a new specification (GJR-GARCH) to capture the asymmetry effect presenting in the conditional variance process.
Hamilton & Susmel (1994)[17] and Lamoureux & Lastrapes (1993)[20] found that the GARCH class models may lead to poor volatility forecasts. High estimated persistence can be controlled by allowing the conditional variance process to be flexible with small/ large shock in returns. Hamilton (1989)[16] introduced the Markov-switching GARCH (MS GARCH) that can adapt quickly to many variation levels. However, MS

[*] Corresponding author
Email address: abdellahtahiri01@gmail.com (Abdellah Tahiri)

GARCH models may easily lead to the path-dependence issue. Bauwens et al. (2014)[5] recommend using the Bayesian approach/Markov Chain Mont Carlo (MCMC) to estimate the MS GARCH models better. Ardia et al. (2018)[3] propose a large-scale study in which they compare different MS GARCH specifications. They suggest using the R package (MSGARCH) to implement the MS GARCH models and the Bayesian estimation approach.

The artificial neural networks (ANN) is a promising tool to deal with nonlinearity; it can assimilate the relationships between returns and variance process. Different studies used ANN to forecast the financial time series. Donaldson & Kamstra (1997)[10] suggested the use of neural networks GARCH model (NN-GARCH) in order to capture volatility in stock returns. Hamid & Iqbal (2004)[15] used the neural networks (NN) to predict the volatility of S&P 500 index futures prices. They concluded that NN outperform the implied volatility. While, Beldirici & Ersin (2009)[8] combined NN with GARCH class models to forecast the volatility series of the Istanbul stock exchange (ISE). Lahmiri (2012)[18] used a combination between NN and the asymmetric GARCH (EGARCH) model in a way the NN inputs are the estimated volatility and the trading volume. His results revealed that trading volume improved the forecast accuracy successfully. Finally, Song et al. (2018)[22] compared five NN models in predicting stock price series, and they showed the superiority of the Back-propagation NN (BPNN) model.

In this study, we believe that the volatility index (VIX) is more significant to simulate the actual volatility of the S&P 500 index, comparing with traditional volatility measures (i.e., Parkinson, Garman Klass,...).

The main goal of this paper is to evaluate the ability of NN in enhancing the forecasts of Markov switching GJR-GARCH in one hand, and to investigate on the NN architecture to select the number of recurrent connections on the other hand.

The remainder of this paper is presented as follow: describe data and estimate the volatility series in section 2, while section 3 briefly introduces artificial neural networks techniques. The empirical results will be discussed in section 4. Finally, section 5 summarizes the obtained results..

2. Data and volatility estimation

2.1. Markov switching GJR-GARCH model

One of the most valuable stylized facts presenting in financial markets is the asymmetry effect, explained by the fact that negative returns affect the conditional variance process more than positive returns. From the business perspective, bad news significantly impacts future fluctuations compared with good news. For this purpose, Glosten et al. (1993)[14] suggested new specifications (i.e., GJR-GARCH) that can respond differently to the past negative and positive shocks. In this paper, we consider only one lag in both return innovations and variance. Therefore, the GJR-GARCH(1,1) model is defined as follow:

$$r_t = \epsilon_t \mathsf{V}_t^{1/2} \qquad ; \quad \epsilon_t \overset{iid}{\sim} f(0,1), \tag{1}$$

$$\mathsf{V}_t = \alpha_0 + (\alpha + \gamma \mathbb{1}(r_{t-1} < 0)) r_{t-1}^2 + \beta \mathsf{V}_{t-1}, \tag{2}$$

where $\mathbb{1}(r_{t-1} < 0) = 1$ if $r_{t-1} < 0$ and 0 otherwise. V_t represent the log-return of a financial asset and the conditional variance at time t, respectively. (ϵ_t) is a sequence of iid random variables with zero mean and unit variance. $f(t)$ is a conditional distribution that needs to be specified. The stationary and the positivity conditions for the GJR-GARCH are assured when $\alpha + \beta + \frac{\gamma}{2} < 1$ and $\alpha_0 > 0$, $\alpha \geq 0$, $\beta \geq 0$ and $\alpha + \gamma \geq 0$. r_t, respectively.

An interesting idea is to let the mentioned above parameters switch across different regimes modeling by a Markov process $s_t \in \{1, 2, ..\}$ (see Haas, 2004[19]). Following Ardia et al. (2018)[3], the Markov switching GARCH (MS GARCH) model can be defined as :

$$r_t | (s_t = k, \mathscr{T}_{t-1}) \sim f(0, \mathsf{V}_t^{(k)}, \Phi^{(k)}), \tag{3}$$

where $\mathsf{V}_t^{(k)}$ denotes the conditional variance within regime k. $\mathscr{T}_{t-1}$ grouping the available information accumulated at time $t-1$ governed by $\{r_{t-1}, r_{t-2}, ...\}$ and $\Phi^{(k)}$ represents additional parameters within a regime k. So far, the asymmetric MS GJR-GARCH(1,1) can be defined as follow:

$$\mathsf{V}_t^{(k)} = \alpha_0^{(k)} + (\alpha^{(k)} + \gamma^{(k)} \mathbb{1}(r_{t-1} < 0)) r_{t-1}^2 + \beta^{(k)} \mathsf{V}_{t-1}, \tag{4}$$

where $\gamma^{(k)}$ is the control parameter of asymmetry in the conditional variance process across different regimes and $\alpha^{(k)}$ captures the ARCH effect within regime k.

In this work, we consider different skewed conditional distribution choices to capture the fat-tailed distribution of S&P 500 returns that exhibit a large kurtosis or skewness (see Fernández & Steel, 1998[13] and Trottier & Ardia, 2016[24]).

2.2. Data description

The data set used in this study consists of the daily adjusted closing price of the S&P 500 and VIX indices obtained from the *Yahoo finance* platform for the period from January 3, 2000, to October 10, 2019 (about 4975 observations) covering different historical shocks mainly, the attack of September 11, 2001, the Enron-scandal between July and November 2002, the Subprime crisis 2008 and the 2011 European financial crisis. The daily log-returns (r_t) of S&P 500 is calculated using the first logarithmic differentiation; $r_t = 100 \times (\log(P_t) - \log(P_{t-1}))$, where P_t is the adjusted closing price at day t.

The VIX index goal (called also fear index) is to reflect and simulate the applied volatility of S&P 500 index options. On March 26, 2004, the trading and speculations in the futures on the VIX started on CBOE (i.e., Chicago Board Options Exchange), considering as input the market price of the “Call” and “Put” options on S&P 500 index. VIX can be interpreted as the expected volatility over the next 30 days. Mathematically, VIX can be expressed as:

$$\text{VIX}^2 = \frac{2e^{r\tau}}{\tau} (\int_0^{P^*} \frac{\mathbb{P}(K)}{K^2} dK + \int_{P^*}^{\infty} \frac{\mathbb{C}(K)}{K^2} dK), \tag{5}$$

where r is the risk-free rate, τ represents the average days on one month (30 days) and P^* denotes the 30-day forward price on S&P 500. $\mathbb{C}(K)$ and $\mathbb{P}(K)$ are the ”Call” and ”Put” prices respectively on a strike K and 30 days to maturity.

Figure 1(top) shows the evolution of the S&P 500 index in parallel with the VIX index across history.

The negative correlation between the two processes becomes more significant during market crashes. We also present in Figure 1(bottom) the evolution of S&P 500 log-returns where we could clearly observe the presence of several clusters (periods) characterized by high/low level of changes in log-returns, this fluctuation can be interpreted as the volatility of the S&P 500 index.

Hence, it will be more beneficial to consider the change in levels (regimes) of the volatility governed by the change on the S&P 500 log-returns.

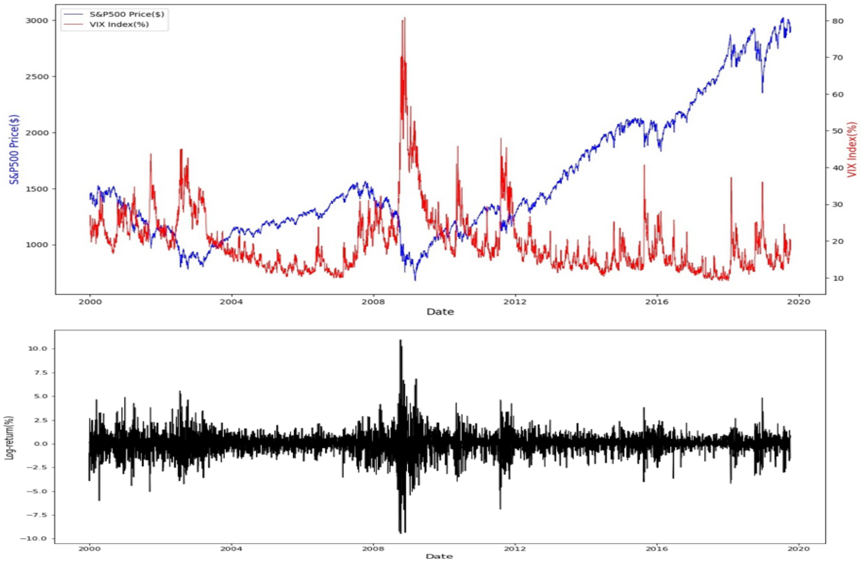

Figure 1: Historical evolution of S&P 500 index , VIX index and S&P 500 log returns.

Table 1 reports the descriptive statistics for the VIX daily adjusted closing price and the daily S&P 500 log-returns. The mean and the median are close to zero for the S&P 500 log-returns, while the standard deviation is around one. The skewness coefficient is negative and differs from zero, reporting that the distribution is tail spread to the left. The S&P 500 log-returns exhibit as well a large excess kurtosis indicating how the tails are fat. The non-normality is noticed by comparing skewness and kurtosis coefficients with the normal distribution's corresponding values. Jarque-Bera test confirmed the non-normality hypothesis (p-value $< 5\%$).

Table 1: Descriptive statistics for the daily S&P 500 log-returns and VIX Index

Statistic	S&P 500 Log-return	VIX Index
Mean (%)	0.0139	19.5575
Median (%)	0.0533	17.420
Std.Dev (%)	1.1955	8.4922
Skewness	-0.2247	2.1633
Kurtosis	8.54521	7.4716
Minimum (%)	-9.4695	9.1400
Maximum (%)	10.9572	80.860
JB-Statistic	15194.92	15465.66
JB p-value	< 0.01	< 0.01
LM (12)	1420.4	4690.5
LM p-value	< 0.01	< 0.01

Before we apply GARCH specifications, it is mandatory to verify specific hypotheses that will allow us to convince using such a Heteroskedastic model. Firstly, the volatility clustering can be checked directly from the evolution of the S&P 500 log-returns (Figure 1). Secondly, we test the nonlinearity, which can be interpreted by the presence of ARCH effect. Lagrange-Multiplier (LM) test with $q = 12$, number of lags, demonstrates the presence of ARCH effect under the null hypothesis of ”No ARCH effect”.

2.3. Volatility estimating and performing

In this section, we aim to estimate and extract the stock market volatility of the S&P 500 index using the MS GJR-GARCH(1,1) model, which has already been previously defined, where we consider the whole history of data starting from January 3, 2000, until October 10, 2019. We remove the auto-regressive effect in the data by filtering with AR(1), then estimate our models based on the residuals to ensure that (r_t) are serially uncorrelated.

Therefore, we fit $(K \times 3 = 9)$ MS GJR-GARCH(1,1) models (results not reported in this paper), we consider up to three regimes $K = 1, 2, 3$ where K is the number of regimes in the conditional variance process, and three different skewed distributions (skewed normal, skewed student's-t and skewed generalized error distribution, see Fernández & Steel, 1998) are assumed as well.

It is straightforward to use the R-package *MSGARCH* (Ardia et al. 2017[4]) to estimate our models where we use the Bayesian approach/Markov Chain Monte Carlo simulation based on the adaptive random-Walk-Metropolis-Hastings algorithm (see Vihola, 2012[25]). Berg et al. (2004)[7] and Ardia (2008)[1] have shown many advantages of the Deviance Information Criterion (DIC) provided by the Bayesian estimation MCMC (see Spiegelhalter et al., 2002[23]) concerning the selection of the most appropriate model. Positivity and stationary of the conditional variance are guaranteed within the estimation phase.

As a result, we noticed that the most appropriate model is the MS GJR-GARCH(1,1) with two-regimes and skewed GED distribution, which can provide a better trade-off in terms of fitting quality and specification complexity.

Table 2: Parameter estimates for MS GJR-GARCH skewed GED

Regime $k = 1$	Mean	Std.Dev	MCSE	Regime $k = 2$	Mean	Std.Dev	MCSE
$\alpha_0^{(1)}$	0.0547	0.0382	0.0002	$\alpha_0^{(2)}$	0.0230	0.0120	0.0001
$\alpha^{(1)}$	0.0158	0.0393	0.0002	$\alpha^{(2)}$	0.0036	0.0094	0.0000
$\gamma^{(1)}$	0.2014	0.1283	0.0006	$\gamma^{(2)}$	0.2048	0.0732	0.0004
$\beta^{(1)}$	0.7286	0.1627	0.0008	$\beta^{(2)}$	0.8749	0.0480	0.0002
$\eta^{(1)}$	1.0918	0.3402	0.0017	$\eta^{(2)}$	1.4729	0.1373	0.0007
$\xi^{(1)}$	0.9199	0.0725	0.0004	$\xi^{(2)}$	0.8658	0.0370	0.0002
p_{11}	0.9852	0.0141	0.0001	p_{21}	0.0029	0.0054	0.0000
$V_{per}^{(1)}$	0.8451	-	-	$V_{per}^{(2)}$	0.9809	-	-

Table 2 reports the parameters for the selected GJR skewed GED model; $(\alpha_0^{(k)}, \alpha^{(k)}, \gamma^{(k)}, \beta^{(k)})$ for $k = 1, 2$. The skewed GED distribution has $\eta^{(k)}$ as tail parameter and asymmetry parameter $\xi^{(k)}$. In addition, the persistence probability within regime $(s_t = k)$ is $p_{kk} = \Pr[s_t = k | s_{t-1} = k]$ where we noticed high persistence within the second regime ($p_{22} = 1 - p_{21} = 0.9971$). We also report the regime's k persistence volatility (i.e., $V_{per}^{(k)} = \alpha^{(k)} + \beta^{(k)} + \frac{\gamma^{(k)}}{2}$), we highlighted significant persistence within the second regime ($V_{per}^{(2)} \simeq 0.98 < 1$) compared to the first regime ($V_{per}^{(1)} \simeq 0.85 < 1$). The estimation results are obtained using the posterior sample of 20,000 draws.

We now turn to the performance analysis for the selected model, where we refer to Figure 2 to evaluate the prediction accuracy across the whole sample covering different peaks. The results clearly show the flexibility in capturing the market volatility within the normal/calm periods. Therefore, during the periods with high volatility (Subprime 2008, for instance), our model overestimates the market volatility.

To summarize, the selected specification shown high performance in capturing the asymmetry in the log-returns. Despite that, the model's weakness can be explained by the presence of the nonlinearity effect. So, how to capture the latter effect in our time series ?.

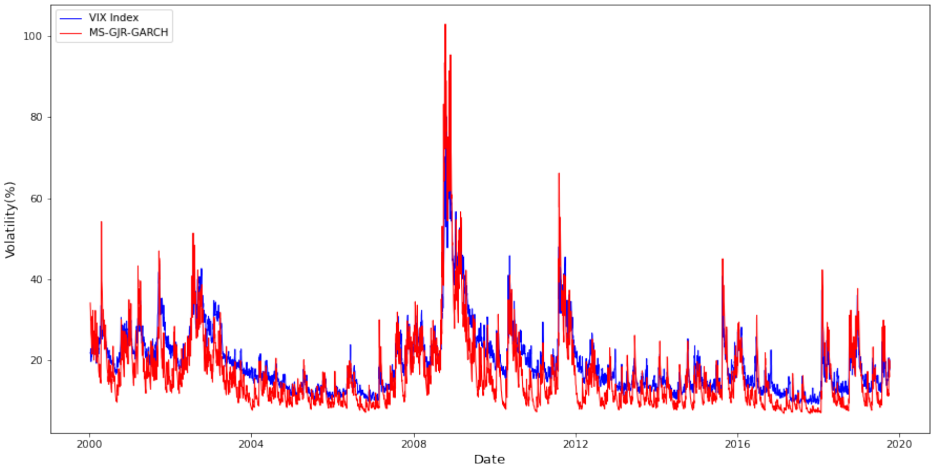

Figure 2: Markov switching GJR-GARCH(1,1) predictions

3. Neural networks and back-propagation algorithm

A neural network is a machine learning method that can be considered as a non-parametric statistical approach aiming to find the best mapping between the inputs and the corresponding outputs. Generally, NN contains three categories of layers; one input layer, one or more hidden layers, and one output layer. One layer may contain a few or plenty of neurons, While the number of hidden neurons is not an arbitrary choice. Figure 3 presents an example of NN with three layers.

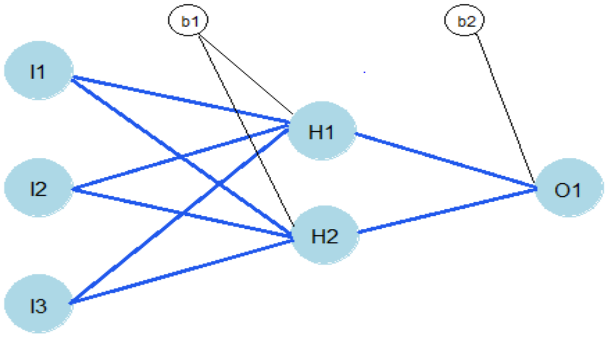

Figure 3: Neural network structure with three layers

However, several empirical studies have revealed NN's capability in forecasting the volatility of different financial assets, mainly, stock returns (Zekić-Sušac & Kliček, B., 2002[26]) and exchange rate (Dunis & Williams., 2002[11]).

In this work, to train our NN model, we aim to use the most known algorithm, namely the back-propagation algorithm (also known as gradient descent) introduced by Rumelhart et al. (1986)[21]; the procedure computes the gradient of the cost function with respect to the weights.

Suppose we have L layers, $l = 1$ indicates the first layer (input layer), $l = L$ corresponds to the output layer, and finally $1 < l < L$ corresponding to the hidden layers. In Figure 3, $L = 3$, $l = 2$ represents the hidden layer.

Hence, let us use $\omega_{ij}^{(l)}$ to denote the weights for the connection between the j^{th} neuron in the $(l-1)^{\text{th}}$ layer and the i^{th} in the l^{th} layer. The activation of the i^{th} neuron in the l^{th} layer can be expressed as :

$$A_i^{(l)} = \phi(\sum_j \omega_{ij}^{(l)} A_j^{(l-1)} + b_i^{(l)}), \tag{6}$$

where $b_i^{(l)}$ denotes the bias of the i^{th} neuron in the l^{th} layer and ϕ is called an activation function that gives the non-linearity to the NN. The widely used activation function is the logistic function (also called sigmoid):

$$\phi(z) = (1 + \mathrm{e}^{-z})^{-1}, \tag{7}$$

In this case, the Equation (6) can be rewritten as follow:

$$A_i^{(l)} = [1 + \mathrm{e}^{-(\sum_j \omega_{ij}^{(l)} + b_i^{(l)})}]^{-1}, \tag{8}$$

In the training phase, the algorithm takes a set of input-output pairs (X_i, Y_i), for each pair (X_i, Y_i), the loss of the model is the cost or the distance between the predicted output $\hat{Y}_i$ and the targeted value Y_i . Overall, for a given point i, the cost function can be presented as:

$$\begin{aligned} \mathrm{C}_i &\equiv \psi(Y_i, \hat{Y}_i) \\ &\equiv \psi(Y_i, A_i^{(L)}), \end{aligned} \tag{9}$$

The algorithm aims to minimize the global cost function C by updating the weights $\omega_{ij}^{(l)}$ according to the following expressions:

$$\begin{cases} \omega_{ij}^{(l)} = \omega_{ij}^{(l)} - \eta \dfrac{\partial C}{\partial \omega_{ij}^{(l)}} \\ b_i^{(l)} = b_i^{(l)} - \eta \dfrac{\partial C}{\partial b_i^{(l)}} \end{cases},$$

where η is a predefined constant (in this study) called the learning rate supposed to be small. $\frac{\partial C}{\partial \omega_{ij}^{(l)}}$ and $\frac{\partial C}{\partial b_i^{(l)}}$ measure the cost function sensitivity to small changes of weights and bias, respectively. By the end of the algorithm training phase, we are expecting final vector of weights ω and bias b to predict $\hat{Y}$.

4. Empirical results analysis

For the purpose of this work, we consider the data set composed of the estimated volatility series for the S&P 500 using the MS GJR-GARCH model with skewed GED distribution (see section 2) and the historical (actual) values of the VIX index. The sample is divided into a training set (70%) where we train our BPNN, and the remaining (30%) are used as a test set to evaluate the models.

4.1. In-sample analysis

We first consider an in-sample analysis, where we train different BPNN architectures with three layers, in the period from January 7, 2000, to October 24, 2013, about 3,470 records.

The number of neurons in the input layer was a topic of a particular study. While the number of the output neurons is fixed at one, there is no magic formula to determine the hidden layer's optimal number. In fact, some rules-of-thumb are available for selecting the number of the hidden neurons h for a NN with three layers. Master (1993) proposed a geometric pyramid rule: $h = \sqrt{l \times n}$, with n and l are respectively the number of the input and output neurons.

We use $(\mathsf{V}_t)_{1 \le t \le 3{,}470}$ to denote the estimated volatility at time t. We consider different NN architectures with different number of input neurons in order to select the optimal number of input neurons n^*, presented by the prior estimated volatility series. We set $X_t^{(T)} = (\mathsf{V}_t, \mathsf{V}_{t-1}, ..., \mathsf{V}_{t-T})$, where T represents the previous period to be used to predict the $Y_t = \mathrm{VIX}_t$. For instance, $T = 2$ means we use $(\mathsf{V}_t, \mathsf{V}_{t-1}, \mathsf{V}_{t-2})$ to predict the value of VIX index at time t (i.e., VIX_t).

In this paper, we aim to select the optimal number of the input neurons where $0 \le T \le 4$,by checking each model's performance (a model is given by a specified value of T), that lead to compare five NN structures.

Hence, we consider $(X_t^{(T)}, Y_t)$ pair as one sample.

In order to help our algorithm to learn fast and correctly, some techniques for scaling the numerical data are available, namely, the min-max normalization. We apply the Equation (10) on the estimated volatility series (V_t) to normalize data in the interval [0,1];

$$\mathsf{V}_t^{'} = \frac{\mathsf{V}_t - \text{Min}(\mathsf{V}_t)}{\text{Max}(\mathsf{V}_t) - \text{Min}(\mathsf{V}_t)}, \tag{10}$$

where $\mathsf{V}_t^{'}$ and V_t are the normalized and the original data, respectively. The VIX series is normalized as well using the Equation (10). The normalized data will be fed to our BPNN algorithm.

An interesting parameter to be specified is the learning rate η which is used to update the weights until the error is normalized. The learning rate should be chosen carefully from the interval [0,1]. After several tests, we set $\eta = 0.01$ to speed up the computations and to avoid the quick convergence to local minima. The implementation results are obtained using the BP algorithm in R-software (*neuralnet* package).

At a certain level, the information contained in the volatility series is assumed to be effective within one week (five working days). Then, the NN with five input neurons is more expected to predict the VIX series better. Figure 4, shows the NN structure for $T = 4$.

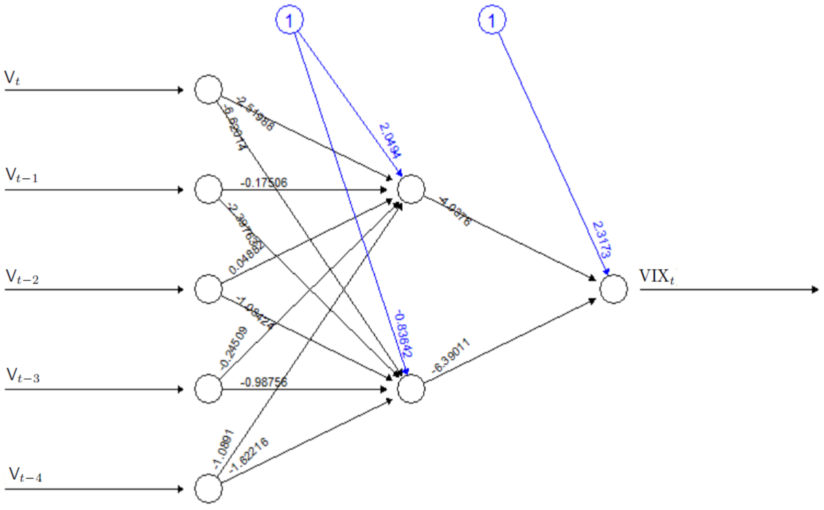

Figure 4: BPNN structure with $T = 4$.

4.2. Out-of-sample analysis

We turn now to the out-of-sample analysis, where we evaluate our models' ability to forecast the VIX index for the period from October 25, 2013, until October 10, 2019, covering 1,500 trading days.

The most appropriate model should provide better forecasting accuracy. In other words, the model with the smallest distance (i.e., error) between the actual value and the predicted value should be chosen. The error in a given model is controlled by referring to the most frequently used metrics, namely, the Mean Squared Error (MSE) and the Mean Absolute Error (MAE). The metrics are expressed as follow:

$$\text{MSE} = \frac{1}{N}\sum_{t=1}^{N}(Y_t - \hat{Y}_t)^2 \tag{11}$$

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{N}|Y_t - \hat{Y}_t| \tag{12}$$

where Y_t and $\hat{Y}_t$ are respectively, the actual value and the predicted value of the VIX index at time t, and N is the out-of-sample size.

The Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) can be considered as well to evaluate the goodness of fit of models. They are not intended for identifying the correct model. Nevertheless, AIC and BIC can be used to compare different models to identify the most appropriate one. Table 3 reports the accuracy metrics (MSE, MAE) and the information criteria (AIC, BIC) for different NN architectures. In addition, we also report the change (%) in the metrics to present the enhancing forecasts that the combination between the MS GJR-GARCH(1,1) and the NN reached.

Table 3: Comparison results of the two specifications.

		MSE	MAE	AIC	BIC
T=0	MS GJR-GARCH	0.0048	0.0518	-	-
	MS GJR-GARCH NN	0.0028	0.0390	**27.7586**	**73.3382**
	Change (%)	-42.4%	-24.7%	-	-
T=1	MS GJR-GARCH	0.0048	0.0518	-	-
	MS GJR-GARCH NN	0.0027	0.0386	31.3935	89.9958
	Change (%)	-43.9%	-25.5%	-	-
T=2	MS GJR-GARCH	0.0048	0.0518	-	-
	MS GJR-GARCH NN	0.0027	0.0384	35.1793	106.8044
	Change (%)	-44.8%	-25.9%	-	-
T=3	MS GJR-GARCH	0.0048	0.0518	-	-
	MS GJR-GARCH NN	0.0026	0.0380	38.9593	123.6072
	Change (%)	-45.7%	-26.7%	-	-
T=4	MS GJR-GARCH	0.0048	0.0518	-	-
	MS GJR-GARCH NN	0.0026	0.0379	42.8083	140.4790
	Change (%)	**-46.3%**	**-26.9%**	-	-

The results in Table 3 reveal that the more we increase the number of input neurons (previous periods), the more the prediction's accuracy increases. This result can be explained by the fact that introducing much historical information into our BPNN model increases the prediction accuracy. In parallel, AIC and BIC values show that the model complexity penalizes the goodness of fit quality; the number of parameters (i.e., weights) increases.

For instance, in term of MSE and MAE, the best model is the one with five neurons (MSE= 0.0026, MAE= 0.0379), it reduced the MSE (MAE) by almost 46% (27%) compared to the standard MS GJR-GARCH(1,1) with skewed GED distribution. On the other hand, the model with only one input neuron is chosen by AIC ($\simeq 27.76$) and by BIC ($\simeq 73.38$). Hence, the need to find the balance between the metrics (MSE, MAE) and the information criteria (AIC, BIC).

Overall, Table 3, shows that, the combination of MS GJR-GARCH(1,1) and NN clearly outperforms the standard MS GJR-GARCH(1,1), whatever the number of the input neurons. It shows as well the ability in enhancing the forecasting performance of MS GJR-GARCH(1,1) by at least 42% (from the MSE perspective). In this work, we recommend to use the NN architecture with five neurons, since we are motivated to dominate the weekly information in the volatility series. We refer to Figure 5 for an intuitive presentation of the selected architecture quality and prediction accuracy, to forecast the VIX index using the standard MS GJR-GARCH(1,1) and the combined specification of MS GJR-GARCH(1,1) and NN with five input neurons.

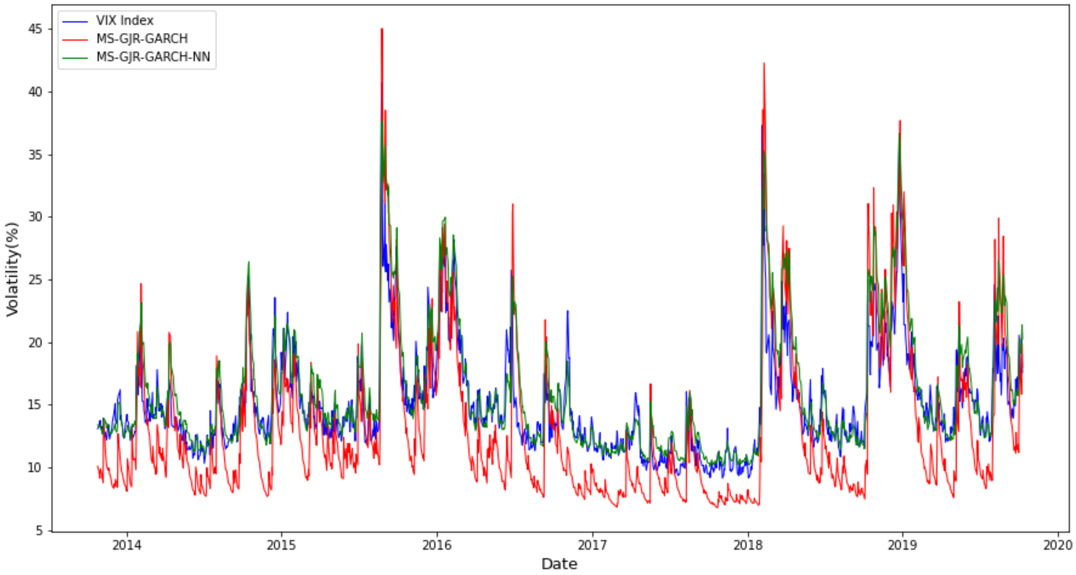

Figure 5: Forecasts of VIX index series.

It is also more helpful to validate the selected model from the stress test point of view. Figure 6, shows by evidence the ability of the MS GJR-GARCH(1,1) with NN to predict the VIX index during the 2008 global financial crisis.

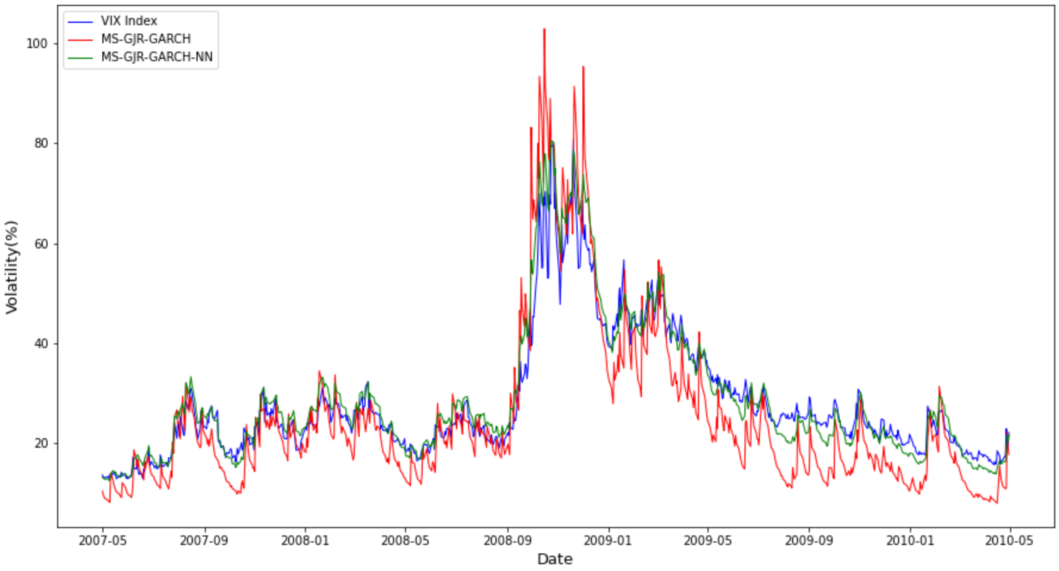

Figure 6: Models performance during the 2008 global financial crisis period.

5. Conclusion

Forecasting volatility has been a critical topic for both researchers and practitioners in the financial community. In this work, we investigated improving volatility prediction by introducing artificial neural networks. This paper can be divided into twofold; the first part consists of estimating the S&P 500 returns volatility using the Markov switching GJR-GACRH(1,1) to account for the asymmetry in the positives/negatives shocks and considering the skewed GED distribution to capture the heavy-tail. The extracted volatility series was then fed to the BPNN algorithm to simulate forecasts to capture the nonlinearity between the conditional variance and the past innovations in the second part. The actual values of the VIX index was considered as the output of our neural network models.

The results show clearly that the combination of the MS GJR-GARCH(1,1) and the BPNN for volatility forecasting is highly recommended. The selected specification's ability to describe better the market behavior, especially within high volatility periods, is confirmed as well.

The paper also has a significant contribution by investing a considerable effort in highlighting the importance of selecting the most efficient neural network architecture as a non-parametric approach to be combined with the Markov switching GARCH models. In addition, studying sophisticated neural networks training algorithms would be dedicated to the improvement of the actual results in future research.

References

[1] Ardia, D., (2008). Financial Risk Management with Bayesian Estimation of GARCH Models: Theory and Applications. Springer.

[2] Ardia, D., Bluteau, K., Rüede, M., (2018). Regime changes in Bitcoin GARCH volatility dynamics, Finance Research Letters.

[3] Ardia, D., Bluteau, K., Boudt, K., Catania, L., (2018). Forecasting risk with Markov–switching GARCH models: A large–scale performance study. International Journal of Forecasting.

[4] Ardia, D., Bluteau, K., Boudt, K., Catania, L., Trottier, D. A., (2017). Markov–switching GARCH models in R: The MSGARCH Package. Journal of Statistical Software.

[5] Bauwens, Luc ; Dufays, Arnaud ; Rombouts, Jeroen., (2014). Marginal likelihood for Markov-switching and change-point GARCH models. Journal of Econometrics, Vol. 178, Part 3, p. 508-522.

[6] B. Benaid and H. Bouzahir., (2018). A Statistical Test of Volatility Persistence in GARCH Models and their Application to Stock Exchange, Advances And Applications in Statistics, 52(6), pp 363-374.

[7] Berg, A., Meyer, R., Yu, J., (2004). Deviance information criterion for comparing stochastic volatility models. Journal of Business & Economic Statistics 22 (1), 107–120.

[8] Bildirici, M., & 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, 36, 7355-7362.

[9] Bollerslev, Y.. (1986). Generalized autoregressive conditional heteroskedasticity, Journal of Econometrics 31, 307–327.

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

[11] Dunis, C.L., Williams, M. (2002). Modelling and trading the euro/US dollar exchange rate: Do neural networks perform better?, Journal of Derivatives & Hedge Funds, 8, 3, 211–239.

[12] Engle, R.F., (1982), Autoregressive conditional heteroskedasticity with estimates of the variance of U.K. inflation, Econometrica 50, 987-1008.

[13] Fernández, C., Steel, M. F. J., Mar. (1998). On Bayesian modeling of fat tails and skewness. Journal of the American Statistical Association 93 (441), 359–371.

[14] Glosten, L. R., Jagannathan, R., Runkle, D. E., (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. Journal of Finance 48 (5), 1779–1801.

[15] Hamid, S. A., & Iqbal, Z. (2004). Using neural networks for forecasting volatility of S&P 500 Index futures prices. Journal of Business Research, 57, 1116–1125.

[16] Hamilton, J. (1989). A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. Econometrica, 57(2), 357-384.

[17] James Hamilton and Raul Susmel, (1994), Autoregressive conditional heteroskedasticity and changes in regime, Journal of Econometrics, 64, (1-2), 307-333.

[18] Lahmiri, S. (2012). An EGARCH-BPNN system for estimating and predicting stock market volatility in Morocco and Saudi Arabia: The effect of trading volume.Management Science Letters , 2(4), 1317-1324.

[19] Markus Haas, (2004), A New Approach to Markov-Switching GARCH Models, Journal of Financial Econometrics, 2, (4), 493-530.

[20] Lamoureux, C., & Lastrapes, W. (1993). Forecasting Stock-Return Variance: Toward an Understanding of Stochastic Implied Volatilities. The Review of Financial Studies, 6(2), 293-326.

[21] Rumelhart, D., Hinton, G. & Williams, R. (1986). Learning representations by back-propagating errors. Nature 323, 533–536.

[22] Song, Y., Zhou, Y., & Han, R. (2018). Neural networks for stock price prediction. ArXiv, abs/1805.11317.

[23] Spiegelhalter, D. J., Best, N. G., Carlin, B. P., van der Linde, A., (2002). Bayesian measures of model complexity and fit. Journal of Royal Statistical Society B 64, 585–616.

[24] Trottier, D.-A., Ardia, D., Aug. (2016). Moments of standardized Fernández-Steel skewed distributions: Applications to the estimation of GARCH-type models. Finance Research Letters 18, 311–316.

[25] Vihola, M. Robust., (2012). Adaptive Metropolis algorithm with coerced acceptance rate. Stat Comput 22, 997–1008.

[26] Zekić-Šušac, M. and Kličєk, B. (2002). A nonlinear strategy of selecting NN architectures for stock return predictions, Finance, Proceedings from the 50th Anniversary Financial Conference Svishtov, Bulgaria, 11–12 April, Svishtov, Veliko Tarnovo, Bulgaria: ABAGAR, 325–355.

[27] Zhang, X. (2020). Analysis of financial market trend based on autoregressive conditional heteroscedastic model and BP neural network prediction. J. Intell. Fuzzy Syst., 39, 5845-5857.