A REPLICATION STUDY
OF THE RECURRENT SUPPORT VECTOR MACHINE-BASED GARCH (1,1) MODEL ON THE EXCHANGE RATE OF TURKISH LIRA TO USD DOLLAR

Replication of Chen et. al. (2010),
"Forecasting Volatility with Support Vector Machine-Based GARCH Model"

Batuhan Ipekci

Humboldt-Universität zu Berlin
Economics and Management Science

Final Paper for the Course "Selected Topics in Econometrics"
17.09.2018

ABSTRACT

Shiyi Chen, Wolfgang K. Härdle, and Kiho Jeong (2010) have developed a new forecasting scheme for volatility, recurrent support vector machines (recurrent SVM), which is shown to perform better than the traditional GARCH models in most situations of one-period-ahead volatility. In this paper, the results of this study are replicated through the exchange rate of Turkish Lira to USD Dollar, which experiences a fast depreciation in years between 2014 and 2018 coupled with a free fall most recently. Especially for this special case, the conditional variance estimation of the recurrent SVM procedure yields the least performance among other candidate models, in contrast to findings of the original paper. The models in comparison to the recurrent support vector machine are simple moving average, GARCH, EGARCH, TGARCH, GJRGARCH, and a recurrent neural network model, Jordan networks. Performance measurement is done through mean absolute error (MAE), directional accuracy (DA), and Diebold-Mariano (DM) test.

Key words: (recurrent) support vector machine; GARCH model;
volatility forecasting; Diebold-Mariano test

INTRODUCTION

The aim of this study is to present an interesting application of the original ideas of the Chen et. al.'s article (2010) with conflicting results, their proposed model being the worst performer in terms of mean absolute error (MAE) measured for a very restricted case. However, this is not an evaluation of model performance in general. Rather, a personal interest on the volatility of the exchange rate of Turkish Lira to USD Dollar is vested in the study of the proposed model in a difficult situation.

Volatility forecasting is an important task in financial markets with its various applications in investment, security valuation, risk management, and monetary policy making. To estimate volatility, numerous variations of Engle's (1982) autoregressive conditional heteroscedasticity (ARCH) model and Bollershev's (1986) generalized ARCH (GARCH) have been developed. The models compared in this study are ARCH, GARCH, exponential GARCH (EGARCH), GJR-GARCH, and threshold GARCH (TGARCH) along with the recurrent SVM-GARCH, and an AR(1) - ARCH(1) model estimated by a recurrent neural networks framework, Jordan networks.

The data set selected for the analysis has many outliers, many of which coincide with the testing sample for out-of-sample forecasts. Therefore, all models yielded higher MAE scores than they yielded during Monte Carlo simulations. Jordan network based ARCH(1) model (Jordan-ARCH) and recurrent support vector machine based GARCH(1,1) (SVM-GARCH) were the worst performers against GARCH family models, although during Monte Carlo simulations Jordan-ARCH and SVM-GARCH models were better performers in most cases. Across the GARCH family model

DATA EXPLORATION

The characteristics of financial time series are well-documented and stylized facts are widely drawn. The most relevant of them to our case are listed below (Cont, 2001):

1. Linear autocorrelations of returns are often insignificant except small intraday time scales. That means, application of the difference operator at the first order is usually sufficient to deal with the autocorrelations in the data.

2. The unconditional distribution of returns displays heavy tails, i.e. they are often leptokurtic, characterized by high excess kurtosis.

3. Returns exhibit large downs but not equally large ups. There exists a gain/loss asymmetry. Past returns are correlated with future volatility negatively. The variance of returns increases with a decrease in prices. This phenomenon is referred to as the leverage effect.

4. Returns are intermittent, displaying high degree of variability and irregular bursts.

5. High-volatility events tend to cluster in time. Different measures of volatility have autocorrelation over periods and this observation is documented as volatility clustering. Clustering is also seen in residuals even after correcting returns for volatility clustering, the observation which led to the development of a family of GARCH models.

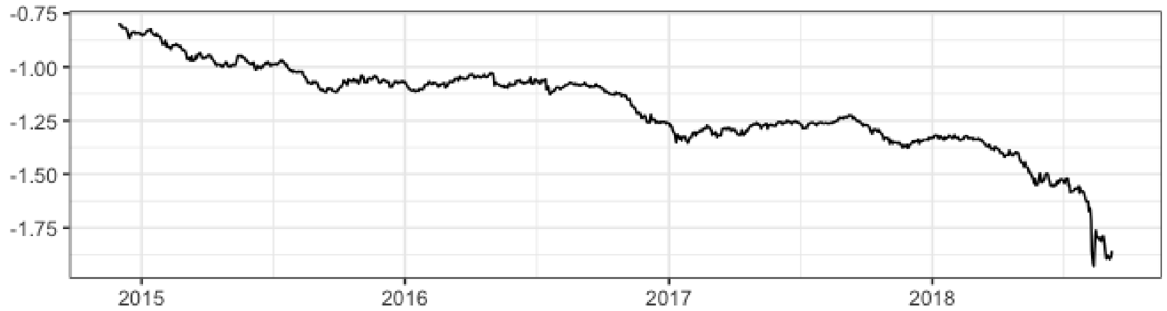

Figure 1. The exchange rate TRY/USD from 02.12.2014 to 07.09.2018, log levels

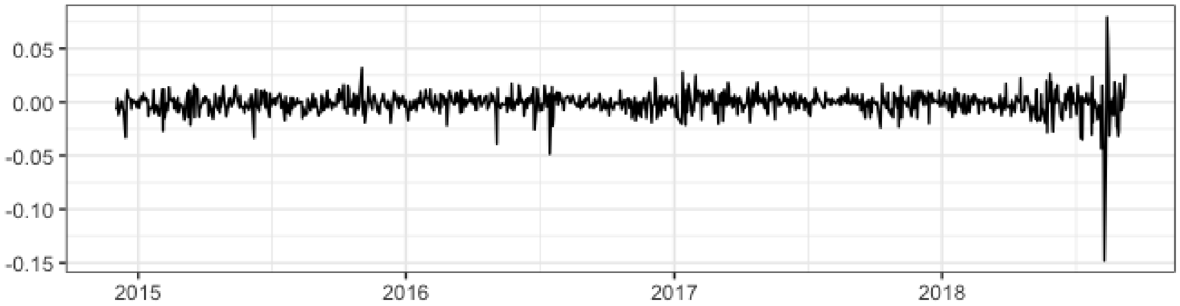

Figure 2. The returns of the exchange rate TRY/USD from 02.12.2014 to 07.09.2018, log levels

Figure 1 shows log levels of the daily nominal bilateral exchange rate of Turkish Lira against US Dollar for the period from December 2, 2014 to September 7, 2018. The data is downloaded from Investing.com as historical data table in CSV format and consists of 1000 data points. Figure 2 shows the log return rates. As seen in Figure 3, returns exhibit a burst in the four-days period between August 10, 2018 and August 14, 2018, when the differenced series have reached both its minimum (-0.147) and maximum (0.079) values. In Figure 4 we have taken a closer snapshot of the returns by excluding the days after August 4, 2018 to better capture the intermittency of the series. By graphical examination, the fourth stylized fact is observed.

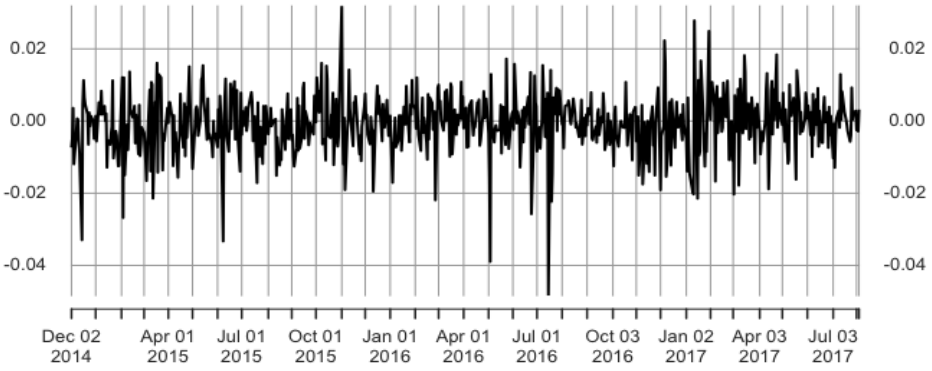

Figure 3. The returns of the exchange rate TRY/USD from 02.12.2014 to 04.08.2018

Table 1. Summary Statistics

Summary	Original Series		Differenced Series	
	Statistic	p-value	Statistic	p-value
Minimum	-1.92895		-0.14789	
Maximum	-0.79518		0.07935	
Mean	-1.1967		-0.00106	
Variance	0.04347		0.00012	
Skewness	-0.75957		-2.53357	
Kurtosis	0.82963		39.92136	
JB	125.65	< 1.11e-16	67696.69	< 1.11e-16
KS	5641.33	< 1.11e-16	74.60	< 1.11e-16
Q(6)	5557.26	< 1.11e-16	212.29	< 1.11e-16
Q*(6)	1905.517	< 1.11e-16	341.32	< 1.11e-16
ARCH(4)	-1.92895	< 1.11e-16	-0.14789	< 1.11e-16
ADF	-0.79518	> 0.99	0.07935	< 0.01
KPSS	-1.1967	< 0.01	-0.00106	> 0.1

Table 1 gives summary statistics and some tests about the original series and the differenced series. The asymmetry of the series is already apparent in both Figure 2 and Figure 3. Returns are characterized by left skewness and a very large excess kurtosis. We can consider skewness as a measure of symmetry and kurtosis as a measure of tailedness.

The first inspection of the summary statistics requires us to test the distribution of the series. Jarque-Bera Test is a Lagrange Multiplier test of normality (Jarque 2011). The null hypothesis is a joint hypothesis of both skewness and excess kurtosis being equal to zero. The test statistic is χ^2-distributed with (n-k) degrees of freedom under the null hypothesis:

$$JB = \frac{n-k+1}{6}\left(S^2 + \frac{1}{4}(C-3)^2\right)$$

Jarque-Bera Test is rejected for the differenced series. The non-normality can also be observed by the two-sample Kolmogorov-Smirnov Test (2008) which is a nonparametric test of whether two underlying one-dimensional probability distributions differ:

$$D_{n,m} = \sup_{\mathrm{x}} \left| F_{1,n}(x) - F_{2,m}(x) \right|$$

Null hypothesis is rejected at level α if $D_{n,m} > c(\alpha)\sqrt{\frac{n+m}{nm}}$. Kolmogorov-Smirnov Test is also rejected for the differenced series. In Figure 4, the quantile-quantile plot (QQ-plot) of the differenced series is seen. QQ-plot is a nonparametric graphical tool to compare the normal theoretical quantiles on the x-axis and sample quantiles on the y-axis. The shape of the QQ-Plot confirms the leptokurtic distribution of our sample by showing the outliers at large distance from the red line. By examining the JB and D statistics, skewness, kurtosis, and QQ-plot we observe the second and third stylized facts.

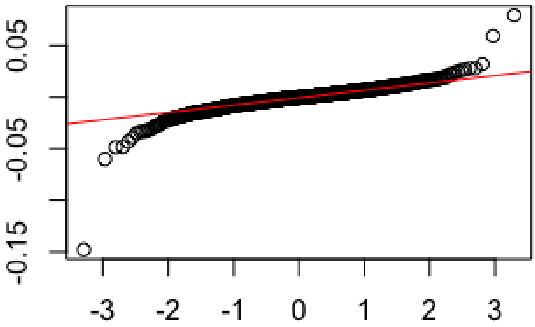

Figure 4. QQ-plot of the differenced series

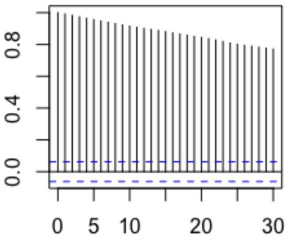

Figure 5. Autocorrelation function of the original series relative to lags

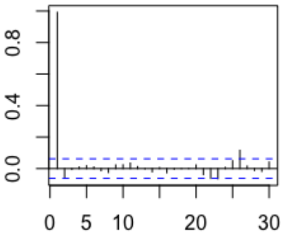

Figure 6. Autocorrelation function of the original series relative to lags

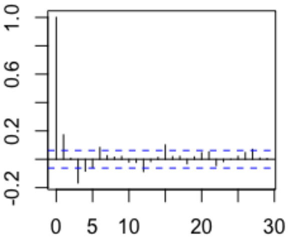

Figure 7. Autocorrelation function of the differenced series relative to lags

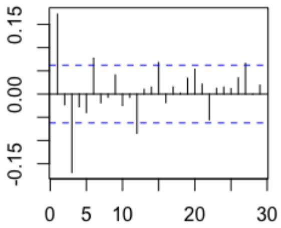

Figure 8. Partial Autocorrelation function of the differenced series relative to lags

The autocorrelation is persistently high at many lags for the original series, as seen in Figure 5, while it cuts off for the differenced series in Figure 7. By examining Figure 7, we can partially arrive the stylized fact 1, autocorrelations become much rare when we apply the first order difference operator to the series. The Ljung-Box Q Test (1978) is a portmanteau test for the randomness based on a number of lags:

$$Q = n(n+2)\sum_{k=1}^{h}\frac{\hat{\rho}_k^2}{n-k}$$

where n is the sample size, $\hat{\rho}_k^2$ is the sample autocorrelation at lag k and h is the number of lags being tested. The null hypothesis requires $\hat{\rho}_k^2$ to be equal to zero for all lags. In our sample, the Q-test is rejected at until lag 6 for both the original and differenced series, thereby suggesting autocorrelation for both original and differenced series. Q-test is also applied to the squared returns, the result is denoted by Q*(6) and also suggests autocorrelation in squared terms, hence heteroscedasticity. This is an indication for the autocorrelation in the variance of the time series, as in the fifth stylized fact. Another indication for the fifth stylized fact is the test of Engle's ARCH effects (1982).

In order to check heteroscedasticity effects, the ARCH(4) is applied to the residuals following ARMA(3,0) of the differenced series and following ARIMA(3,1,0) of the original series. The squared series of the residuals of a conditional mean model, $a_t = r_t - \mu_t$ is put in the equation below:

$$\mathrm{\alpha_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \ \ldots + \alpha_m\ a_{t-m}^2 + e_t, \quad t\ =\ m+1, \ldots, T}$$

Engle (1982) proposed the Lagrange Multiplier procedure on the autocorrelation of the squared OLS residuals, equivalent to applying an F-statistic for testing $\alpha_i = 0$, $(\mathrm{i}\ =\ 1, \ldots, \mathrm{m})$ which is asymptotically distributed as a chi-squared distribution with m degrees of freedom under the null hypothesis. Both of the test statistics, Q*(6) and ARCH(4) are significant at level 0.01, i.e. the conditional heteroscedasticity of a_t^2 is detected.

As the last step of the data exploration, stationarity checks are applied. The series must be stationarity for an ARIMA estimation of the conditional mean equation and for a GARCH-family estimation of the conditional variance equation. A series is weakly stationary if its mean at any time t is constant and its autocovariance is time-invariant (Tsay 2010, p.23). Two test results for stationarity are observed in Table 1 and Table 2. The Augmented Dickey Fuller test (Dickey 2010) is a unit root test whose null hypothesis is that the series has a unit root, hence it is nonstationary. The other test, KPSS test (Kwiatkowski et. al. 1992) is used for testing the null hypothesis that the time series is stationary around a deterministic trend versus an alternative hypothesis that the series has a unit root. These two tests have contrasting null hypotheses, and by using them simultaneously we apply a sound test for stationarity. For the original series, the ADF fail to reject the null hypothesis while KPSS rejects its null hypothesis at 0.01 level. That means, original series are not weakly stationary and have possibly a unit root. However, differencing the series leads ADF test to reject non-stationarity at 0.01 level and KPSS to fail to reject stationarity at 0.1 level. As a result, the differenced series is stationary. Since we obtain a stationary series through the first differencing, the series is said to be integrated of order 1.

MODEL SPECIFICATION

The real data analysis is done on daily log returns,

$$y_t = \log(I_{t+1} - \log I_t)$$

1. Simple Moving Average

Being the simplest model, the moving average of the five most recent observations is used as a benchmark model, which is expressed as

$$\hat{u}_{t+1}^2 = \frac{1}{5} \sum_{j=t-4}^{t} u_j^2$$

for simulated data and

$$\hat{u}_{t+1}^2 = \frac{1}{5} \sum_{j=t-4}^{t} \left(y_j - \bar{y}_{5,t}\right)^2$$

for the real data.

2. Autoregressive-moving-average Model (ARMA)

Standard ARCH and GARCH-family models are estimated on the residuals obtained from ARMA estimation of the conditional mean equation of returns.

$$r_t = \varphi_0 + \varphi_1 r_{t-1} + \cdots + \varphi_p r_{t-p} + \theta_1 \varepsilon_{t-1} + \ \ldots + \theta_q \varepsilon_{t-q}$$

The suitable model is chosen by following the Box-Jenkins Methodology (Rao et. al. 1972) with model identification, estimation and residual diagnostics. Model identification is done with the order selection of differencing order, p and q. The differencing order is already selected as 1 at the data exploration part. By examining the ACF and PACF of the differenced series, there are significant autocorrelations at lags 1 and 3, while there is significant partial autocorrelation at lag 3. Candidates by visual inspection are ARMA(3,1) and ARMA(3,3). Since the ACF and PACF is well-behaved after integrating and since the frequency of the series is daily, there is no need to check for seasonality.

Model estimation is done by maximum likelihood estimation, as it is the default choice in the R package. Akaike Information Criterion (AIC) and Schwarz - Bayesian Information Criteria (SBIC or BIC) are compared to select the most suitable models.

$$AIC(\rho) = \ln\left(\hat{\sigma}_\rho^2\right) + \frac{2\rho}{T}$$

$$SBIC(\rho) = \ln\left(\hat{\sigma}_\rho^2\right) + \frac{\rho}{T}\ln(T)$$

where $\hat{\sigma}_\rho^2$ is the maximum likelihood estimate of the variance of the error term, and T is the sample size and ρ is the total number of orders (Brooks p.232). Since $\ln(T) > 2$ when $T > 7$, SBIC penalizes high orders stiffer.

Table 2. AIC and BIC scores for different ARMA models

Parameters	AIC	BIC
(p = 3, q = 0)	2.784351	2.808885
(p = 3, q = 1)	2.787565	2.812094
(p = 4, q = 0)	2.785605	2.815046
(p = 1, q = 5)	2.782693	2.821947

BIC is lowest when (3,0) is applied and AIC is lowest when (1,5) is applied. By the parsimony principle, we avoid choosing the autoregressive order bigger than 3, as this could lead to overfitting. Two candidate models are selected as (3,0) and (3,1) for further scrutiny of the residual diagnostics.

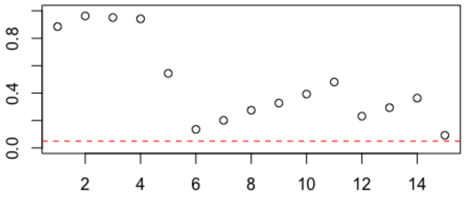

Figure 9. p-values of Ljung Box Q-test on the residuals of ARMA(3,0) model relative to lags

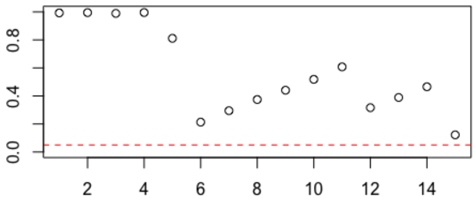

Figure 10. p-values of Ljung Box Q-test on the residuals of ARMA(3,1) model relative to lags

Both ARMA(3,0) and ARMA(3,1) pass residual diagnostics, since Ljung-Box test is not rejected for either of them until 16th lag. We have chosen the ARMA(3,1) model, since its p-values are slightly higher than the other model, and for the sake of a little more complexity in model selection.

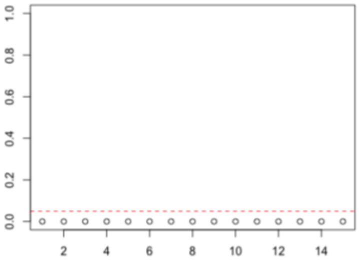

Figure 11. p-values of Ljung Box Q-test on the squared residuals of ARMA(3,1) model relative to lags

In Figure 11, when we applied Ljung-Box test to squared residuals, heteroscedasticity effects are persistent, as it is also shown in Table 1, where the null hypothesis is rejected for all lags until the 16th.

3. Autoregressive Conditional Heteroskedasticity (ARCH)

An ARCH(m) model is expressed as follows:

Let $\varepsilon_t = r_t - \mu_t$ be the mean-corrected log return,

$$\varepsilon_t = \sigma_t \epsilon_t, \quad \sigma_t^2 = \varepsilon_0 + \alpha_1 \varepsilon_{t-1}^2 + \ \ldots + \alpha_m\, \varepsilon_{t-m}^2 \ \ where \ \{\epsilon_t\} \sim iid\ (0,1),\ \alpha_0 \geq 0\ for\ i > 0$$

where the asymptotic stationarity of the volatility term σ_t^2 is assured through the assumption of $\Sigma_i^p \alpha_i < 1$.

Autoregressive conditionally heteroscedastic (ARCH) models are first introduced by Engle in the aim of improving the performance of previous models which assume "a constant one-period forecast variance" (1982, p. 987). As we have explored during data exploration, our sample is leptokurtic and skewed, hence the usual models assuming homoscedasticity were simply not suitable. The observed volatility clustering can be imitated throughout this improvement.

The fourth unconditional moment of α_t is in general greater than 3, i.e. ARCH model leads to a heavier tail distribution of α_t than that of a normal distribution (Tsay 2010, p. 85). Hence, outliers appear more often in returns than implied by an iid sequence of normal random variates. Nevertheless, as a criticism, ARCH models likely to overpredict the volatility because they respond slowly to large isolated shocks (Tsay 2010, p. 86).

Another weakness of ARCH Models is that they assume symmetry in volatility, since volatility depends on the square of previous shocks. This property is also seen in GARCH models. This implicit assumption does not hold for our sample. Alternative models, such as EGARCH, have been proposed to deal with this issue, to them we will come later on.

The order determination of the ARCH model is done through looking at the PACF graph. Once the order is determined, the model can be checked through the standardized shocks

$$\tilde{\varepsilon}_{\mathrm{t}} = \frac{\varepsilon_{\mathrm{t}}}{\sigma_{\mathrm{t}}}$$

which should be iid random variates following either a standard normal or standardized Student-t distribution. Forecasts of the ARCH model can be obtained recursively.

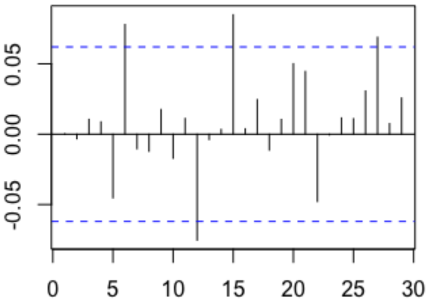

Figure 12. Partial Autocorrelation function of the squared residuals of ARMA(3,1) relative to lags

As it is seen in Figure 12, PACF of ARMA(3,1) residuals shows significant partial autocorrelations only after the 6th order. ARCH models demand a high order.

4. Generalized Autoregressive Conditional Heteroskedasticity (GARCH)

The GARCH model is developed by Bolleshev (1986). The current volatility depends on past volatilities and observations of the model residuals. Bollershev extended the ARCH model to include the ARMA structure.

Let $\varepsilon_t = r_t - \mu_t$ be the mean-corrected log return,

$$\varepsilon_{\mathrm{t}} = \sigma_{\mathrm{t}}\epsilon_{\mathrm{t}}, \qquad \sigma_{\mathrm{t}}^2 = \alpha_0 + \sum_{\mathrm{i=1}}^{\mathrm{m}} \alpha_{\mathrm{i}}\varepsilon_{\mathrm{t-i}}^2 + \sum_{\mathrm{j=1}}^{\mathrm{s}} ß_{\mathrm{j}}\sigma_{\mathrm{t-j}}^2$$

where $\{\epsilon_t\} \sim iid\ (0,1),\ \alpha_0 \geq 0, ß_j \geq 0\ for\ i,j > 0,\ \Sigma_{i=1}^{\max(m,s)}(\ \alpha_i +\ ß_i) < 1$

It is obvious that a large ε_{t-1}^2 or σ_{t-1}^2gives rise to a large σ_t^2, namely a large ε_{t-1}^2 tends to be followed by another large a_t^2, imitating volatility clustering in financial time series. Similar to the ARCH process, GARCH process also yields heavy tails. This is an appropriate characterization of financial time series in general, let alone being suitable to our case (Ghose and Kroner 1995). Furthermore, GARCH models are usually more parsimonious than ARCH models, avoiding overfitting (Brooks 2008, p.393).

As in the case of ARCH models, GARCH models suffer from an implicit assumption of symmetry as well. Therefore, GARCH models may not capture leverage effects adequately.

5. Exponential GARCH (EGARCH)

Nelson (1991) pointed out three major drawbacks of GARCH when introducing an improvement to the model. Firstly, GARCH models do not account for asymmetry, i.e. the observation that 'volatility tends to rise in response to 'bad news' (excess returns lower than expected) and to fall in response to 'good news' (excess returns lower than expected)'. Secondly, GARCH models impose parameter restrictions that are often violated by estimated coefficients. And thirdly, persistence of shocks is ambiguous in GARCH models. To overcome these drawbacks, exponential GARCH (EGARCH) is proposed.

Let $\varepsilon_t = r_t - \mu_t$ be the mean-corrected log return, an EGARCH(m, s) can be written as

$$\varepsilon_t = \sigma_t \epsilon_t, \ \ln(\sigma_t^2) = \alpha_0 + \frac{1 + ß_1 L + \ldots + ß_s L^s}{1 - \alpha_1 L - \ldots - \alpha_m L^m} g(\epsilon_{t-1}),$$

$$g(\epsilon_t) = \theta \epsilon_t + \gamma[|\epsilon_t| - E(|\epsilon_t|)], \qquad where\ \theta\ and\ \gamma\ are\ real\ constants$$

L is the lag operator and the multiplier of $g(\epsilon_{t-1})$ has nominator and denominator polynomials having roots outside the unit circle and having no common factors (Tsay 2010). Since $\ln(\sigma_t^2)$ is modelled, even if the parameters are negative, σ_t^2 will be positive. Therefore, there is no need to impose non-negativity constraints to model parameters α, ß, θ, and γ (Brooks 2008, p. 406).

6. GJR-GARCH and Threshold GARCH (TGARCH)

GJR-GARCH model is introduced by Glosten, Jagannathan and Runkle (1993) while TGARCH by Zakoian (Rabemananjara and Zakoian 1993) along with a similar concern that of Engle's that GARCH model does not allow positive and negative unanticipated returns to have different impacts on the conditional variance. Indeed, as they estimated, there exists a negative relationship between conditional mean and conditional variance of the excess return of stocks (1993, p.1799). Therefore, these models are designed to capture leverage effects. Both models are essentially the same except TGARCH uses standard deviation in specification instead of variance and it relaxes the linearity assumption to capture non-linear behaviour in volatility (Rabemananjara and Zakoian 1993, p.32).

In order to deal with this misspecification in the standard GARCH model, they added a dummy variable I_t to the conditional equation which switches along the sign of the return. Zakoian and Rabemananjara pointed out limitations of the EGARCH as the fixed-in-time effects on volatility of positive ϵ_t values relative to negative ones and its reliance on a moving average equation on the $\ln(\sigma_t^2)$ process (1993, p.33).

Let $\varepsilon_t = r_t - \mu_t$ be the mean-corrected log return, GJR-GARCH(m,s) can be written as

$$\varepsilon_t = \sigma_t \epsilon_t, \qquad \sigma_t^2 = \alpha_0 + \sum_{i=1}^{m} (\alpha_i + \gamma I_{t-i})\, \varepsilon_{t-i}^2 + \sum_{i=1}^{s} ß_j \sigma_{t-j}^2$$

While TGARCH(m,s) can be written as,

$$\varepsilon_t = \sigma_t \epsilon_t, \qquad |\sigma_t| = \alpha_0 + \sum_{i=1}^{m} (\alpha_i + \gamma I_{t-i}) |\varepsilon_{t-j}| + \sum_{i=1}^{s} ß_j |\sigma_{t-j}|$$

$$where \quad I_{t-1} = 1 \quad if \;\; \varepsilon_{t-1} < 0, \qquad otherwise \;\; I_{t-1} = 0$$

Leverage effects are captured by $\gamma > 0$.

7. ANN – GARCH (Jordan Network)

The artificial neural networks (ANN) is a versatile semiparametric tool in many estimation problems. It is fundamentally a series of regression models stacked on top of each other with the final layer being a linear one (Murphy 2012, p.563). In the context of financial time series, the recurrent subset of models is commonly used. The selected model by Chen et. al. is described as "the feedback multilayer perceptrons with the addition of a global feedback connection from the output layer to its input space" which has one nonlinear hidden layer and one linear output layer (Chen et. al., p.415). In the recurrent neural networks literature, the predecessor of this architecture is Jordan and Elman networks. In this paper results of Jordan networks are presented because its architecture is more similar than Elman networks to the recurrent support vector machine procedure.

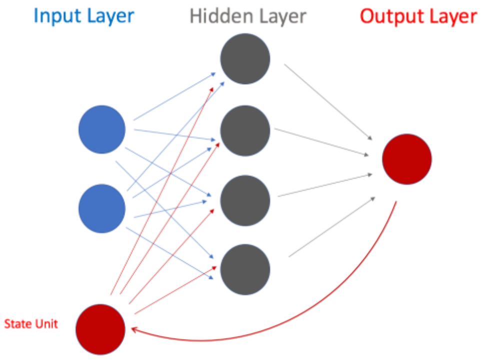

Figure 13. Jordan networks

Following the Chen et. al.'s specification, the recurrent ANN (Jordan network) in Figure 13 has one nonlinear hidden layer with four neurons, each using a tan-sigmoid differentiable transfer function to generate the output, and one linear output layer with one neuron. The number of neurons is also selected as 4 by cross validation during real data analysis.

A neural network "activates" information from one layer to the next by a transfer function f (in our case this function is the tan-sigmoid). The jth node in the hidden layer is defined as

$$h_j = f_j\left(\alpha_{0j} + \sum_{i \to j} \omega_{ij} x_i + \sum_{i \to j} u_{ij} o_{j-1}\right)$$

where x_i is the value of the ith input node and the input units are fed from the output layer. For the output layer, the node is defined as

$$o = f_o\left(\alpha_{0o} + \sum_{j \to o} \omega_{jo} h_j\right)$$

where α is the bias vector and ω is the weights vector. Nodes, biases and weights are unknown while the functional form is known. By the universal approximation property of multilayer perceptrons, they can approximate any continuous function uniformly on compact sets by increasing the number of nodes in the hidden layer (Tsay 2010, p.248).

The Jordan networks is applied with the following specifications: One hidden layer with 4 neurons, as chosen by 10-fold cross validation. The learning rate for the conditional mean equation is chosen by cross validation as 0.04 and it updates itself with cross validation at each steps of the conditional variance estimation. The transfer function is tanh-sigmoid and the training algorithm is stochastic gradient descent. By using the time series vector with its lagged term as the only explanatory variable for both the conditional mean and conditional variance equations, a Jordan-AR(1)-ARCH(1) model is estimated. The reason why not a Jordan-GARCH model is not implemented is the belief that the artificial neural networks might do a better job in discovering non-linearities in autoregression to minimize model residuals, having optimized bias and weights vectors. As we will see during the Monte Carlo simulations, the Jordan-AR(1)-ARCH(1) model yielded the best results.

8. Recurrent Support Vector Machine

The support vector machine (SVM) maps input vectors into a high-dimensional feature space through some nonlinear mapping. In this space, an optimal separating hyperplane is constructed (Vapnik 1998, p.133). On this hyperplane, an empirical error minimization process is undertaken according to the principles of structural risk minimization (Vapnik 1998).

It has a design of a feedforward network with an input layer, a single hidden layer of nonlinear units, and an output layer. Similar to MLP, SVM has a universal approximation property, such that if the dimensions of hidden space is high enough, any nonlinear mapping relations can be approximated. Differently from MLP, SVM estimates functions by convex optimization with a unique solution. Therefore, the optimization process does not end at local minima. SVM's are much more immune to overfitting than neural networks by its design.

Given a training dataset (x_t, y_t) where input vector $x_t \in \mathbb{R}^p$ and output scalar $y_t \in \mathbb{R}^1$, we aim to find a sample regression function $f(x)$ to approximate the unknown decision function $g(x)$:

$$f(x) = w^T \phi(x) + b$$

where $\phi(x)$ is known nonlinear transfer function. The dimension of the feature space is l (the higher l, the better approximation), the parameter w denotes linear weights and b is the threshold.

In order to find $f(x)$, the optimal w* and b* have to be estimated by the constrained optimization problem below:

$$\min_{\substack{w \in \mathbb{R}^t, \\ \xi(\,') \in \mathbb{R}^{2T}, \\ b \in \mathbb{R}}} C(w, b, \xi_t, \xi_t') = \frac{1}{2} \left|\left|w\right|\right|^2 + C \sum_{t=1}^{T} (\xi_t + \xi_t')$$

such that

$$\begin{gathered} w^T \phi(x) + b - y_t \leq \varepsilon + \xi_t \\ y_t - w^T \phi(x) - b \leq \varepsilon + \xi_t' \\ \xi_t \geq 0, \xi_t' \geq 0, t = 1,2, \ldots, T \end{gathered}$$

where the non-negative slack variables ξ and ξ' denote the data points on or outside the ε-tube which comes from the $\varepsilon -$ insensitive loss function L_ε :

$$L_\varepsilon\left(x, y, f(x)\right) = \begin{cases} |y - f(x)| - \varepsilon & for\ |y - f(x)| \geq \varepsilon \\ 0 & otherwise \end{cases}$$

The dual optimization problem above can be derived from the primal problem by using the Karush-Kuhn-Tucker conditions as follows:

$$\min_{\alpha_t(\prime) \in \mathbb{R}^{2T}} \frac{1}{2} \sum_{s=1}^{T} \sum_{t=1}^{T} (\alpha_s' - \alpha_s)(\alpha_t' - \alpha_t)\ K(x_s . x_t) + \varepsilon \sum_{t=1}^{T} (\alpha_t' - \alpha_t) - \sum_{t=1}^{T} y_t (\alpha_t' - \alpha_t)$$

such that

$$\sum_{t=1}^{T} (\alpha_t - \alpha_t') = 0$$

$$0 \leq \alpha_t, \quad \alpha_t' \leq Cs, \quad t = 1,2, \ldots, T$$

where α_t and α_t' are Lagrange multipliers. For any solution of α_t and α_t', the optimal w* can be calculated unique as follows:

$$w^* = \sum_{t=1}^{T} (\alpha_t - \alpha'_t)\, \phi(x_t)$$

However, $b^*$is not unique:

$$b^* = y_t - \sum_{t=1}^{T} (\alpha_t - \alpha'_t)\, K(x_t.x_i) + \varepsilon, \qquad if \quad i \in \{t|\alpha_t \in (0,C)\}$$

$$b^* = y_j - \sum_{t=1}^{T} (\alpha_t - \alpha'_t)\, K\left(x_t.x_{ij}\right) + \varepsilon, \qquad if \quad j \in \{t|\alpha'_t \in (0,C)\}$$

Thus, the regression function $f(x)$ will be computed using w* and b* in the following forms:

$$f(x) = {w^*}^T \phi(x) + b^* = \sum_{t=1}^{T} (\alpha_t - \alpha'_t)\, \phi^T(x_t)\phi(x) + b^* = \sum_{t=1}^{T} (\alpha_t - \alpha'_t)\, K(x_t,x) + b^*$$

where $K(x_t,x) = \phi^T(x_t)\phi(x)$ is the inner-product kernel function. Without specifying $\phi(x)$ and without computing corresponding inner products, only the form of $K(x_t,x)$ is taken into consideration. It is referred as the "Kernel trick", by means of which the computational complexity of the high-dimensional hidden space is significantly reduced (Schölkopf et.al. 2002). Chen et. al. investigates three different kernels:

$$Linear\colon K(x_t,x) = x_t^T x$$

$$Polynomial\colon K(x_t,x) = (x_t^T x + 1)^d$$

$$Gaussian\colon K(x_t,x) = exp \frac{-\left||x - x_t|\right|^2}{2\sigma^2}$$

We will only be concerned with the Gaussian kernel and the polynomial kernel of second degree during this application. The procedure of Chen et. al. does not converge for all the kernels in the case of our analysis.

Before implementation of SVM, appropriate values of $\varepsilon, C, d\ and\ \sigma^2$ must be determined in advance through cross-validation. Too high values of C places a high penalty for nonseparable points causing many support vectors and overfitting, while too small C causes underfitting (Alpaydin, 2011). The parameter ε controls the width of the $\varepsilon - insensitive$ zone. The bigger ε, the fewer support vectors are selected, and results become flatter (Vapnik, 1997). The parameters $d\ and\ \sigma^2$ are kernel parameters, with very large value of sigma, the Gaussian kernel becomes almost linear and with d = 1, the polynomial kernel becomes linear. The tuning is done through the validation sets that are recursively separated from and added to the training set (cross validation principle). The parameters at the end of tuning sessions are used only on the training set, by leaving the test set out for the final validation.

In order it to capture the time series sequences, we need to adjust SVM to be recurrent in a way that output should update the input, as in the the case of Jordan network. Chen et. al. proposes an iterative algorithm, following a Box-Jenkins-like procedure which relies on residual diagnostics (p. 416). No converging algorithm for their proposed recurrent SVM model has been introduced. During this study, the convergence seemed to be dependent upon the relationship between the kernel function and the empirical data. For instance, the data in logarithms but no further preprocessing converged on a Gaussian kernel whereas the data came out of outlier removal and standardization converged on a second-degree polynomial kernel.

The algorithm is as follows:

The letter i indicates the iterative epoch and t denotes the period.

- Step 1: set i = 1 and start with all residuals at zero: $w_t^{(1)} = 0$.
- Step 2: Run an SVM procedure to get the decision function $f^{(i)}$ $to\ the\ points\ \{x_t, y_t\} = \{u_{t-1}^2, u_t^2\} with\ all\ inputs\ x_t = \left\{u_{t-1}^2, w_t^{(i)}\right\}$.
- Step 3: Compute the new residuals $w_t^{(i+1)} = \{u_{t-1}^2, w_{t-1}^{(i)}\}$.
- Step 4: Terminate the computational process when the stopping criterion is satisfied; otherwise, set i = i+1 and go back to Step 2.

where the first iterative epoch is a feedforward SVM process and results in an AR(1) estimation and following epochs provide results of the ARMA(1,1) model, being estimated by the recurrent SVM.

The procedure stops when the corresponding residuals have no autocorrelation. Ljung-Box-Pierce Q-test at first lag is used to investigate autocorrelation. Only if the p-values of the Q-test for five consecutive epochs are simultaneously higher than 0.1, the iterative computational process is stopped. The recurrent SVM model that is proposed by Chen et. al. here embeds the principles of Box-Jerkins methodology. It is the very novel part of the approach which combines the SVM's kernel structure that allows statistical and computational flexibility with a major econometric technique.

FORECASTING SCHEME

Forecasting is done through separating 1000 data points into 940 training and 60 test sets for both the real and artificial data, that means out-of-sample method is used to avoid overfitting. The fundamental problem in forecasting is the fact that volatility is unobservable. For the artificially generated data, the estimation of the same model that is employed to generate the data is used, while for the real data Chen et. al.'s auxiliary assumption about the actual ex-post volatility for the real data is followed:

$$u_t^2 = (y_t - \bar{y})^2$$

The recursive forecasting scheme of SVM is employed with an updating sample window. the estimating and forecasting process is carried out recursively by updating the sample with one observation each time, rerunning the SVM approach and recalculating the model parameters and corresponding forecasts. In total, 60 one-period-ahead forecast volatilities are estimated separately and then combined together (Chen et. al.).

Forecasting accuracy is tested by the Diebold and Mariano (1995) for the difference of mean absolute forecast error (MAE) between two different models:

$$H_0: MAE_1 - MAE_0 = 0 \ \ versus \ H_{1a}: MAE_1 - MAE_0 \ \neq 0 \ \ \ or \ \ H_{1b}: MAE_1 - MAE_0 < 0$$

where subscript denotes the benchmark model and 1 the competing model. In the original paper, the DM statistic is in a robust form where $\hat{S}^2$ denotes a heteroscedasticity and autocorrelation consistent (HAC) robust covariance matrix which is estimated according to Newey-West procedure (1987**).** In application we used the function *dm.test* from forecast package of R, and the results seemed to be robust, since dividing by the HAC covariance matrix did not affect the results. The DM statistic which is used by Chen et. al. follows:

$$DM := \frac{1}{\sqrt{n}}\frac{1}{\sqrt{\hat{S}^2}}\sum_{t=T_1}^{T-1}\left(|u_{t+1}^2 - \ \hat{u}_{1,t+1}^2| - |u_{t+1}^2 - \hat{u}_{0,t+1}^2|\right) \ \sim \ N(0,1)$$

Forecasting performance is also compared using MAE and directional accurracy (DA), as follows;

$$MAE := \frac{1}{n}\sum_{t=T_1}^{T-1} |u_{t+1}^2 - \hat{u}_{t+1}^2|$$

$$DA(\%) := \frac{100}{n}\sum_{t=T_1}^{T-1} a_t \quad where \quad a_t = \begin{cases} 1 & (u_{t+1}^2 - u_t^2)(\hat{u}_{t+1}^2 - \ \hat{u}_t^2) \\ 0 & \qquad\qquad otherwise \end{cases}$$

MONTE CARLO SIMULATION

The data generation process is completely in parallel with Chen et. al., in order to be sure that the models are well-understood and well-implemented. Another advantage of generating data is to prepare a sound design environment to compare across a normal random sample, a skewed student-t sample and our real data which exhibits high skewness and kurtosis.

To generate the data, the ARMA(1,0) - GARCH (1,1) model is parametrized by the following settings,

$$(c, \phi_1, \kappa, \delta_1, \alpha_1) = (0, 0.5, 0.0005, 0.8, 0.1)$$

in addition, the disturbance term is first distributed as Gaussian distribution, and secondly, as Student's t distribution with five degrees of freedom (kurtosis is equal to 5). The Student's t distribution exhibits a relatively degree of skewness and excess kurtosis. Using the same specified models, two artificial samples of size 500 and 1000 are created. Each situation is replicated 50 times, that means there are in total 200 samples.

Table 3. Mean Absolute Error and Directional Accuracy (in percentage) scores for different models and scenarios

Models	Normal				Student's - t			
	n = 500		n = 1000		n = 500		n = 1000	
	MAE	DA(%)	MAE	DA(%)	MAE	DA(%)	MAE	DA(%)
GARCH(1,1)	0.008968	45.8	0.006998	46.7	0.007533	44.5	0.007966	46
EGARCH(1,1)	0.008855	45.6	0.007016	46.4	0.007833	44.3	0.007909	46.2
GJRGARCH(1,1)	0.008869	45.7	0.007036	46.5	0.007737	44.4	0.007962	46.1
TGARCH(1,1)	0.008888	45.6	0.007036	46.4	0.007835	44.4	0.007924	46.1
SVM-GARCH	0.006687	50.9	0.006359	52.3	0.007648	51.7	0.008376	48.7
Jordan-ARCH	0.004678	50.1	0.004426	49	0.004915	50.1	0.005822	49.9
Moving Average	0.005594	50.7	0.004796	48.8	0.005808	48.4	0.004796	48.8

The analysis of random samples is done in a different manner than that in the original paper. At first, we skip the sensitivity analysis of SVM parameters by Monte Carlo samples for the recurrent SVM model, since it would be a bare repetition. We saved this analysis for the next section on the real data, and tracked the process while the data is fitted to the model. Secondly, Chen et al. analyzed the effect of kernel type to forecasting performance by evaluating one-step-ahead forecast errors. Here, we evaluate 60-step-ahead errors to compare the forecasting scheme of the real data, which has many irregularities, with the controlled environment of the artificial data. The aim of this section is to create a control environment for the forecast of the real data, and studying the characteristics of the competing models, rather than studying the effects of kernels or any other model parameters. Lastly, we applied SVM-GARCH in a feedforward manner, as it is done in the Monte Carlo analysis of the original paper.

In this section, the out-of-sample MAE and DA scores of the 60-step-ahead forecast and the p-values of the DM test are reported. Two-sided version of the DM test is calculated by choosing four different benchmark models, moving average, GARCH(1,1), SVM-GARCH, and Jordan-ARCH, denoted by DM1, DM2, DM3, and DM4, respectively.

Table 3 represents the average MAE and DA scores of 50 different samples from 4 different scenarios. The moving average of 5th order is considered as a crossover, rather than a benchmark as in the original paper. Since the moving average is calculated on the known variance of the artificially generated data, it is expected to be the strongest. Note that moving average is the least performing benchmark in the original paper because Chen et. al. compares one-step-ahead performances instead of 60-step-ahead as we have done here.

The performance of Jordan-ARCH model is overstepping the crossover in all cases except the case of the sample of 1000 length drawn from t distribution. SVM-ARCH is a second-runner when the distribution of t is normal. Note that the performance of the SVM-ARCH reduces importantly when the distribution is changed from normal to Student's t. The most important reason is the Gaussian kernel of the SVM model, which does not perform not optimally under this situation. There is also a convergence in performances of GARCH family models and SVM-ARCH model when the sample is t-distributed.

All GARCH family models have important performance improvement when the length of sample having normal distribution increases, and performance degradation when the length of a sample having t distribution increases. There are not large differences between GARCH family models across the scenarios. Even though the data satisfy the normality assumption that is required for maximum likelihood estimation in the GARCH family models, the SVM-ARCH and Jordan-ARCH models still outperform them. Moreover, models that are based on SVM and Jordan networks have always better DA scores then standard GARCH family models.

Table 4. Diebold-Mariano tests for different models and scenarios

Distribution	Models	Sample size = 500				Sample size = 1000			
		DM1: moving average	DM2: GARCH	DM3: SVM	DM4: Jordan	DM1: moving average	DM2: GARCH	DM3: SVM	DM4: Jordan
Normality	Moving Average	-	0.893	0.852	0.664	-	0.897	0.833	0.719
	GARCH(1,1)	0.106	-	0.418	0.118	0.102	-	0.478	0.146
	EGARCH(1,1)	0.131	0.529	0.430	0.129	0.090	0.471	0.467	0.134
	TGARCH(1,1)	0.130	0.502	0.425	0.126	0.091	0.497	0.465	0.134
	GJRGARCH(1,1)	0.115	0.506	0.422	0.108	0.111	0.603	0.526	0.147
	SVM-GARCH	0.147	0.581	-	0.12	0.066	0.521	-	0.147
	Jordan-GARCH	0.335	0.881	0.852	-	0.280	0.853	0.852	-
Student's t	Moving Average	-	0.748	0.839	0.516	-	0.823	0.829	0.656
	GARCH(1,1)	0.251	-	0.590	0.263	0.176	-	0.525	0.194
	EGARCH(1,1)	0.245	0.361	0.564	0.264	0.169	0.568	0.564	0.218
	TGARCH(1,1)	0.244	0.391	0.567	0.264	0.172	0.500	0.534	0.215
	GJRGARCH(1,1)	0.251	0.438	0.580	0.262	0.178	0.486	0.526	0.212
	SVM-GARCH	0.160	0.409	-	0.126	0.170	0.474	-	0.215
	Jordan-ARCH	0.483	0.736	0.873	-	0.343	0.805	0.784	-

In Table 4 we observe p-values of one-sided Diebold-Mariano (DM) tests, small p-values meaning that the benchmark model is significantly better performing in terms of MAE. The Jordan ARCH model performs the best in all situations, being multiple times around the 10 level of significance against each other models. EGARCH, TGARCH, and GJRGARCH have higher changes against GARCH when the sample size is small and the data is t-distributed. It is interesting that Jordan ARCH model is almost a better model than SVM-GARCH model at 0.01 significance level in every scenario, even in the cases where the sample is normally distributed and the kernel of the SVM is Gaussian.

PARAMETER TUNING OF JORDAN NETWORK AND SVM MODELS

1. SVM Conditional Mean

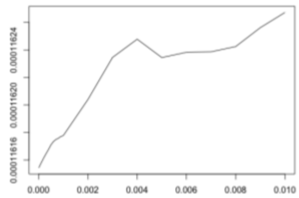

Figure 13. Mean squared error performances of chosen cost levels for conditional mean equation

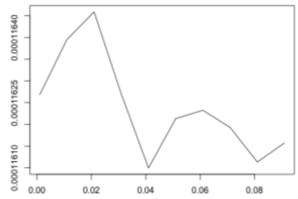

Figure 14. Mean squared error performances of chosen epsilon levels for conditional mean equation

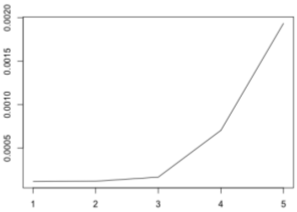

Figure 15. Mean squared error performances of chosen polynomial order levels for conditional mean equation

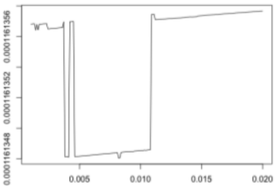

Figure 16. Mean squared error performances of chosen gamma levels for conditional mean equation

The mean equation of the SVM-GARCH (1,1) model is estimated by a feedforward SVM procedure, as suggested. The cost parameter and epsilon parameter are needed to be tuned for the convex optimization problem while the sigma should be tuned for the Gaussian kernel and the degree of polynomial, d, for the polynomial kernel. The option of tuning sigma was not directly available in the R-package e117017, although gamma, $\frac{1}{2\sigma^2}$, exists in the model options. As seen from Figure 13 to Figure 16, the cost parameter needed to be as low as possible, we have opted for 10^{-4}. The epsilon parameter is optimal at 0.04 level, whereas the gamma at 0.008. Suitable choices of the polynomial parameter d are 1 (where the d=1 is almost the same as the linear kernel case), 2, 3, although with neither of them the procedure of Chen et. al. converges. This is an issue that is also observed with some Monte Carlo simulations.

Our personal experience is that if the data cannot be modeled by a linear or polynomial structure, the Ljung-Box tests during residual diagnostics never yields a p-value higher than 0.1. It is important to note that this convergence issue is only relevant to the recurrent version of the SVM which is proposed by Chen et. al.. The feedforward structure is mechanically immune to convergence issues, since its convex optimization problem has always a global solution. That means, even in the case of not converging recurrent SVM models, a feed-forward variant can be used. We suppose that this is exactly what Chen et. al. did in their Monte Carlo analysis.

2. SVM Conditional Variance

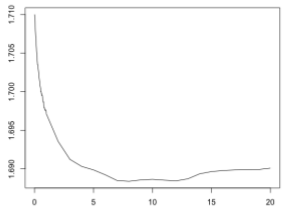

Figure 17. Mean squared error performances of chosen cost levels for conditional variance equation

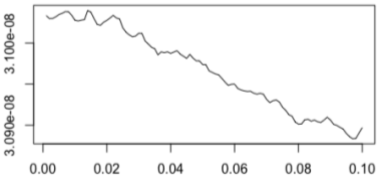

Figure 18. Mean squared error performances of chosen epsilon levels for conditional variance equation

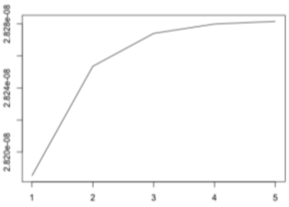

Figure 19. Mean squared error performances of chosen polynomial order levels for conditional variance equation

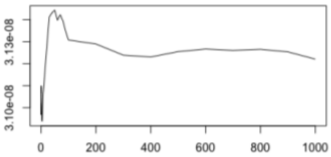

Figure 20. Mean squared error performances of chosen gamma levels for conditional variance equation

As a general tendency of machine learning models being highly data-dependent, the residuals of the feedforward SVM model demand almost a completely new set of parameters. This time the least error is obtained by the cost being equal to 9, epsilon to 0.1, gamma to 0.008 and d to 1. The best gamma parameter of the conditional variance model did not change importantly from that of the conditional mean model. Whereas epsilon and cost lies completely opposite ends of the spectrum in this comparison. This could be attributed to the fact that gamma is a kernel parameter that is concerned more with the structure of the data, and the cost and epsilon for the new optimization problem of conditional variance that arises following the conditional mean estimation.

As for the stopping criteria of the recurrent SVM algorithm, luckily the convergence is established at the first epoch. For all the 60 one-step-ahead forecasts, from the first try on, the p-values of the Ljung-Box test are always higher than 0.1. That means it only takes 5 iterations to finish one one-step-ahead forecast, and the residuals following the application of the model are white noise. Should the parameters not be tuned carefully, the algorithm never converges, and autocorrelations of the residuals persist. During the Monte Carlo simulations, there were cases which converged in 100, 200, or even 300 epochs, as Chen et. al. experienced in their real data analysis. In our real data, high iterations were not the case.

3. Jordan Network Conditional Mean and Variance

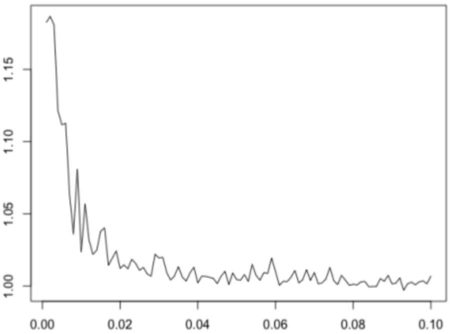

Figure 21. Mean squared error performances of chosen learning rate levels for conditional mean equation

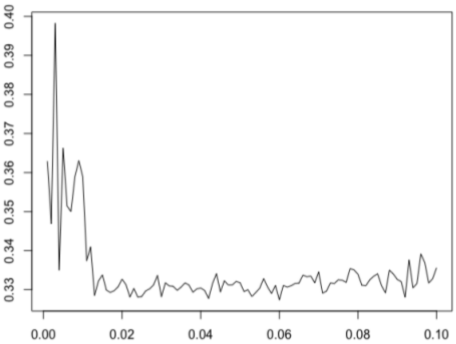

Figure 22. Mean squared error performances of chosen learning rate levels for conditional variance equation

The Jordan networks are tuned following the same logic, using 10-fold cross validation. The optimal number of neurons are 4 for solving both the conditional mean and the conditional variance equations, while the performance of learning rate beyond 0.02 does not change that much. However, we let the RNN to tune itself at each consequent iteration since they can get very complex as the time window is sliding, hence tuning the learning rate in each iteration might be a good idea.

REAL DATA ANALYSIS

The choice of the data, as seen during the section Data Exploration, is obviously not the most suitable for making good comparisons across models. Undergoing a crash, the Turkish Lira / US Dollar returns are very heavy-tailed and skewed, having unexpected bursts even in the nontechnical sense. Furthermore, the period which is to be forecasted is almost a mission impossible because the outliers of the data are located mostly within the forecast horizon. This is a task of forecasting when an outlier appears. The author is completely aware of the specificity of the task that might be better captured by some other probabilistic or binary model than GARCH and extensions.

All models perform bad for the real data, as it is the intention of the author to present a contrasting case for the original study, where it is shown that the SVM-GARCH model performs in almost all situations better than GARCH family models.

By close inspection of the Figure 1 and 2, it can be seen that the test sample separated for evaluating results coincides with a massive burst in the daily returns of the TRY/USD which makes it the worst-case scenario for any predictive model. The MAE results in Monte Carlo experiments have almost been reversed. SVM-GARCH and Jordan-ARCH are the worst models by far, and all GARCH and ARCH models are very close to pass the threshold of moving average in terms of MAE. Nevertheless, we observe DA scores of the SVM-GARCH model are the best. During turbulent days, DA accuracy is also something that should be taken into consideration. As the results are this unsuccessful in terms of MAE, we tried to run the models in the preprocessed data. We removed outliers by removing points which lie beyond the 1.5 times the interquartile range (IQR) of the 25th and 75th quartiles. Then, as the next step, we subtracted the mean of the returns and divided them by their standard deviation. The result did not change significantly, except the Jordan-ARCH model loses DA scores for unknown reasons.

Note that during application of preprocessing, the tuning kernel for the SVM conditional mean equation has changed. Before preprocessing, the converging kernel is Gaussian, while after preprocessing the converging kernel becomes polynomial of second order. Similarly, after outlier removal, the ARMA order of the real data becomes ARMA(0,1), while it is ARMA(3,1) before removal.

Table 5. Model performances on real data, without preprocessing and with preprocessing

Models	Without Preprocessing		With Preprocessing	
	MAE	DA	MAE	DA
ARCH(2)	7.351	0.3728814	0.6314564	0.03389831
ARCH(3)	7.354538	0.3728814	0.6311587	0.03389831
ARCH(4)	7.353157	0.3728814	0.6290647	0.03389831
GARCH(1,1)	7.352622	0.3728814	0.6241123	0.03389831
GARCH(2,1)	7.352623	0.3728814	0.6240958	0.03389831
EGARCH(1,1)	7.356087	0.3728814	0.6374778	0.01694915
EGARCH(2,1)	7.356204	0.3728814	0.6192406	0.03389831
EGARCH(2,2)	7.356034	0.3728814	0.6273376	0.03389831
GJRGARCH(1,1)	7.356493	0.3728814	0.6292502	0.03389831
GJRGARCH(1,2)	7.357633	0.3728814	0.6299123	0.03389831
GJRGARCH(2,1)	7.356758	0.3728814	0.6292826	0.03389831
GJRGARCH(2,2)	7.357433	0.3728814	0.6298394	0.03389831
TGARCH(1,1)	7.357578	0.3728814	0.6293884	0.03389831
TGARCH(1,2)	7.357881	0.3728814	0.6346957	0.03389831
TGARCH(2,1)	7.358083	0.3728814	0.6294861	0.03389831
TGARCH(2,2)	7.360413	0.3728814	0.6294973	0.03389831
SVM-GARCH	8.052639	0.440678	1.7323	0.440678
Jordan-ARCH	7.965949	0.559322	2.001958	0.3728814
Moving Average	7.239113	0.3898305	1.639371	0.5254237

The SVM-GARCH model, although being the least successful in terms of MAE scores, have still some promising advantages when we inspect the forecast plots in Figure 23 and Figure 24. A reason why all the models failed considerably in read data analysis might be the intermittent peaks which are observed in the data. In the close-up graph in Figure 24, comparing the SVM-GARCH result and the test sample, we observe that the SVM-GARCH model surprisingly captures the some off the structures which might have caused a peak day. Although the magnitude is not that well estimated, the Figure 24 is encouraging further research.

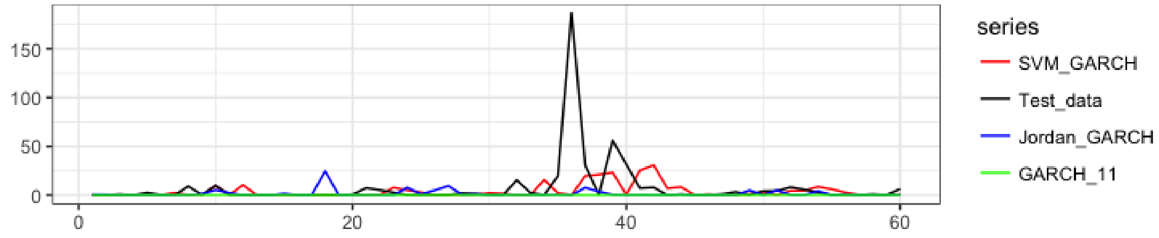

Figure 23. Forecast performances on real data, all models

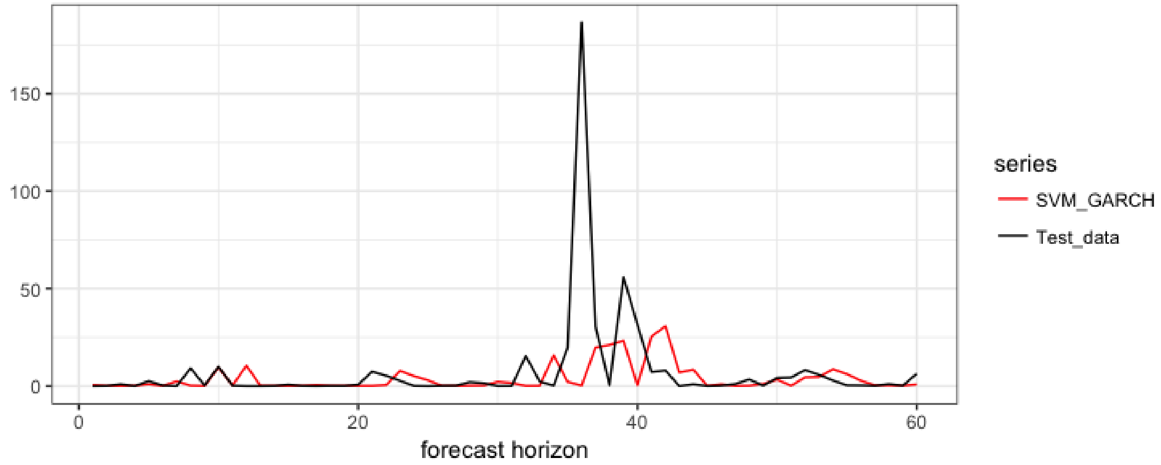

Figure 24. Forecast performances on real data, only SVM-GARCH

This study only refers to the specific case of a sample that is drawn from a currency in crisis. An argument in general cannot be made. Furthermore, the SVM's kernel functions are modifiable, thus have still a room for improvement. The same also holds for the RNN-GARCH models, of which one of the oldest model is here presented (Jordan networks). Lastly, the application of long memory models such as Markov Switching GARCH could also make an interesting case.

CONCLUSION

The Lira/Dollar exchange rate presents an interesting case to both the study of Chen. et. al. and their section of Monte Carlo simulation. At first, SVM-GARCH model is defeated by GARCH family models on the measure of MAE under an unfair singular situation, although the Monte Carlo studies in both papers show that SVM-GARCH is a better model for most scenarios. Nevertheless, SVM-GARCH stays strong on the DA measure, and have promisingly captured the timing of peak moments. Secondly, none of the GARCH extensions brought about significant performance improvements of GARCH model under these circumstances. Lastly, the recurrent artificial network architecture Jordan Network, seems to be different and might be better performing than what Chen. et. al. has used, although it is run with the same structure of training algorithm and transfer function. To confirm this, further studies are needed.

A conclusion that can be drawn from this study that it might be better to stay with the GARCH family models for 60-days-ahead forecasts during very high volatility bursts, and during the times that no model can be reliable. The direction of the change in data is no surprise to people who follow Turkish politics. Still, the magnitude of the changes is difficult to capture either from viewpoint of a technical analyst nor a policy analyst, given the accumulated evidences of brutal abolishment of legal, political, and economic reason in Turkey under hands of an unpredictable and relentless corrupt few.

BIBLIOGRAPHY

(2008) Kolmogorov–Smirnov Test. In: *The Concise Encyclopedia of Statistics*. Springer, New York, NY.

Akaike H. (2011) Akaike's Information Criterion. In: Lovric M. (eds) *International Encyclopedia of Statistical Science*. Springer, Berlin, Heidelberg.

Alpaydın, E. (2011). Machine learning. *Wiley Interdisciplinary Reviews: Computational Statistics,3*(3), 195-203.

Bollerslev, T. (1986). Generalized Autoregressive Conditional heteroskedasticity. *Journal of Econometrics*, 31(3), 307-327.

Brooks, C. (2008). *Introductory Econometrics for Finance*. Cambridge: Cambridge University Press.

Cont, R. (2001). Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance,1*(2), 223-236.

Chen, S., Härdle, W. K., & Jeong, K. (2010). Forecasting volatility with support vector machine-based GARCH model. Journal of Forecasting, 29, 406-433.

Dickey, D. G. (2011). Dickey-Fuller Tests. *International Encyclopedia of Statistical Science,*385-388.

Diebold, F., & Mariano R. (1995). Comparing predictive accuracy*. Journal of Business and Economic Statistics*, 13, 253–265.

Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica,50*(4), 987.

Ghose, D., & Kroner, F. (1995). The relationship between GARCH and symmetric stable processes: Finding the source of fat tails in financial data. *Journal of Empirical Finance,2*(3), 225-251.

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. *The Journal of Finance,48*(5), 1779-1801.

Jarque C.M. (2011) Jarque-Bera Test. In: Lovric M. (eds) *International Encyclopedia of Statistical Science*. Springer, Berlin, Heidelberg

Kwiatkowski, D., Phillips, P. C., Schmidt, P., & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics,54*(1-3), 159-178.

Ljung, G.M. & Box, G. E. P. (1978). On a Measure of a Lack of Fit in Time Series Models. *Biometrika,* 65, 297-303.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. Cambridge, Massachusetts, London: MIT Press.

Nelson, D. B. (1991). Conditional Heteroskedasticity in Asset Returns: A New Approach. *Econometrica,59*(2), 347. doi:10.2307/2938260

Rabemananjara, R., & Zakoian, J. M. (1993). Threshold arch models and asymmetries in volatility. *Journal of Applied Econometrics,8*(1), 31-49. doi:10.1002/jae.3950080104

Rao, J. N., Box, G. E., & Jenkins, G. M. (1972). Time Series Analysis Forecasting and Control. *Econometrica,40*(5), 970.

Schölkopf, B., & Smola, A. (2002). Support Vector Machines and Kernel Algorithms.

Tsay, R. S. (2010). Analysis of Financial Time Series. *Wiley Series in Probability and Statistics.*