Time Series Model for Forecasting Intraday Volatilities

Dominykas Serksnas
St. Anne's College
University of Oxford

Supervised by Dr Patrick McSharry

The dissertation for MSc Mathematical and Computational Finance
Trinity Term, 2013

Abstract

The aim of this dissertation is to construct different intraday volatility forecasting techniques for the future contracts and to evaluate their accuracies in terms of the forecasting errors. In order to achieve these goals the intraday volatility levels are assumed to be reflected by the absolute values of the intraday returns. The periodic intraday volatility factor is modelled by different regression methods and the future intraday absolute returns are forecasted. Since the intraday returns depend on the overall volatility of the particular day, Exponential Weighted Moving Average and GARCH(1,1) techniques are implemented to predict these quantities. Finally, the results reveal that the most accurate method for the intraday volatility forecasting is based on modelling each 5 min interval during the day as an independent variable in the periodic intraday component and this holds not only for the next day forecasting but also for the longer time horizon.

Acknowledgements

I would like to thank my supervisor Dr Patrick McSharry for his support, important suggestions and advices while I was writing this dissertation. I believe that his guidance made a big difference in the quality of this project. Also, I am really grateful to Dr Lajos Gergely Gyurko for his help to get the financial data used for my analyses.

Contents

Chapter 1

Introduction

The rapid development of the algorithmic trading systems and the increase in the volume of the financial contracts traded by the computers has been a common phenomenon in the recent decade. The ability to capture the best price now strongly depends on how fast you can send your order to the market and how volatile the market is at that time. Hence, a lot of importance in the financial area should be given to the volatility levels throughout the day, in other words to the intraday volatility.

The intraday volatility modelling has already been the area of interest by Andersen and Bollerslev (ref. [1] and [2]), Bollerslev et al. (ref. [4]), Engle and Sokalska (ref. [11]), Fuertes, Izzeldin and Kalotychou (ref. [12]) and many other academics but most of their analyses were attempting to estimate the fluctuations of the returns in the particular market (e.g. US or Japan equity market). Hence, in this dissertation I will use two different instruments, i.e. S&P 500 equity index futures and Gold futures, which represent different kinds of financial markets, and I will try to investigate what techniques give us the highest accuracy in terms of the forecasting errors.

This thesis is split into the several parts. In Chapter 2 I will try to recover the consistent time series for the returns of the previously mentioned future contracts by rolling from one to another. Chapter 3 will analyse the behaviour of the intraday metrics such as returns and autocorrelations and lay the foundations for modelling of the intraday volatility forecasting methods described in Chapter 4. I will evaluate these methods in Chapter 5 by using the out-of-sample data and determine the accuracy in terms of the forecasting errors. Finally, the concluding remarks will be given which will summarise the results of all the analyses done in this paper.

Chapter 2

Getting Time Series

2.1 Futures Data Description

As it has been mentioned before, the analysis in this dissertation will be concentrated on the future contracts. The main reason for this choice is the continuous intraday trading of these contracts. Unlike the equities, futures are usually traded 24 hours a day and that could help us find out the relationships between the changes in the volatility levels and the trading activities in three biggest world financial centres, i.e. Hong Kong, London and New York. It has already been discussed that for most of the securities the volatility increases a lot during the opening and closing hours of the market (Lockwood and Linn, 1990) so I will try to determine how the openings and closings of the previously mentioned financial centres affect the volatility and try to forecast it for the nearest future.

In order to develop the intraday volatility forecasting model the future contracts of gold (GC[1]) and S&P 500 (ES[2]) equity index will be used in this analysis. The choice of these contracts could be supported by the fact that they represent the underlying securities from different types of financial markets. S&P 500 is the equity index representing 500 US stocks which are largest by the market capitalisation while gold contracts represent the commodities' market as well as the FX market as gold is also traded as the pair against the major world currencies (e.g. XAUUSD, XAUAUD, XAUCHF, etc.)

One of the problems while analysing financial time series is the decision how much data should be used in order to be sure that we do not employ the data which might be irrelevant these days but at the same time not to miss the important one. Hence, the future contracts maturing in years 2011 and 2012 have been included. These two years have been chosen because they include both upward and downward trends and prevents from the possibility

[1] GC is the ticker of the gold future contracts
[2] ES is the ticker of the S&P500 future contracts

that the intraday volatility changes might be distorted during the long periods of the increases or decreases of the securities' prices associated to the bull or bear markets respectively (Daly, 2011).

In this dissertation I will be using GC contracts maturing in February (G[3]), April (J), June (M), August (Q), October (V) and December (Z), i.e. every 2 months, and ES contracts maturing in March (H), June (M), September (U) and December (Z), i.e. every 3 months. As each of the future contract, i.e. maturing in a particular month, is traded just for the short period of time we must roll from one contract to the other one in order to get the continuous financial time series which might be used for the forecasting purposes. The rolling procedure from one contract to the other is described in the following section of this chapter.

Finally, it must be mentioned, that I will use the future contracts which have the data subsampled on a 5 min scale. This gives us 288 data points for each day (there are 288 five minutes intervals in 24 hours). Such a sampling has been used in a number of time series analyses of the intraday volatilities (e.g. refs. [1] and [2]) and it is enough to identify intraday periodicity of the volatility fluctuations of the analysed future contracts.

2.2 Future contracts rollover

All future contracts have their expiry dates so investors willing to hold their position in these contracts for a longer period of time have to close one contract before its expiration and open a new one expiring later. The rollover from one contract to the other one lets the investors keep their positions in the future markets for the longer time than the life of a particular contract. Hence, it becomes possible to reproduce the continuous time series for the futures prices.

There have been a lot of empirical studies done about the trading volumes of different contracts (e.g. Holmes and Rougier, 2005; Daigler and Willey, 1999; Grammatikos and Saunders, 1986, etc.) and most of them analyse the interaction between the changes in trading volumes of different futures contracts approaching their expirations, the open interest in these contracts and their absolute returns. Hence, due to the lack of the open interest data I will

[3] The letters next to each month is a standard notation of the maturity month of the future contract, i.e. F means January, G – February, etc.

concentrate on the changes in trading volumes trying to determine the rollover dates from one contract to another.

The graphs a) and b) in Figure 1 below show the cumulated daily volumes of different S&P 500 and gold futures[4]. It might be seen from these graphs that the volume of the contract approaching its expiration decreases sharply while the volume of the next one increases at the similar pace. Also, there is only one crossing point between the consecutive contracts near the expiration of the first one. As a result, I will assume that the rollover day from one contract to the following one is determined by looking when the latter contract becomes more liquid than the former one, i.e. when the daily trading volumes of these contracts cross.

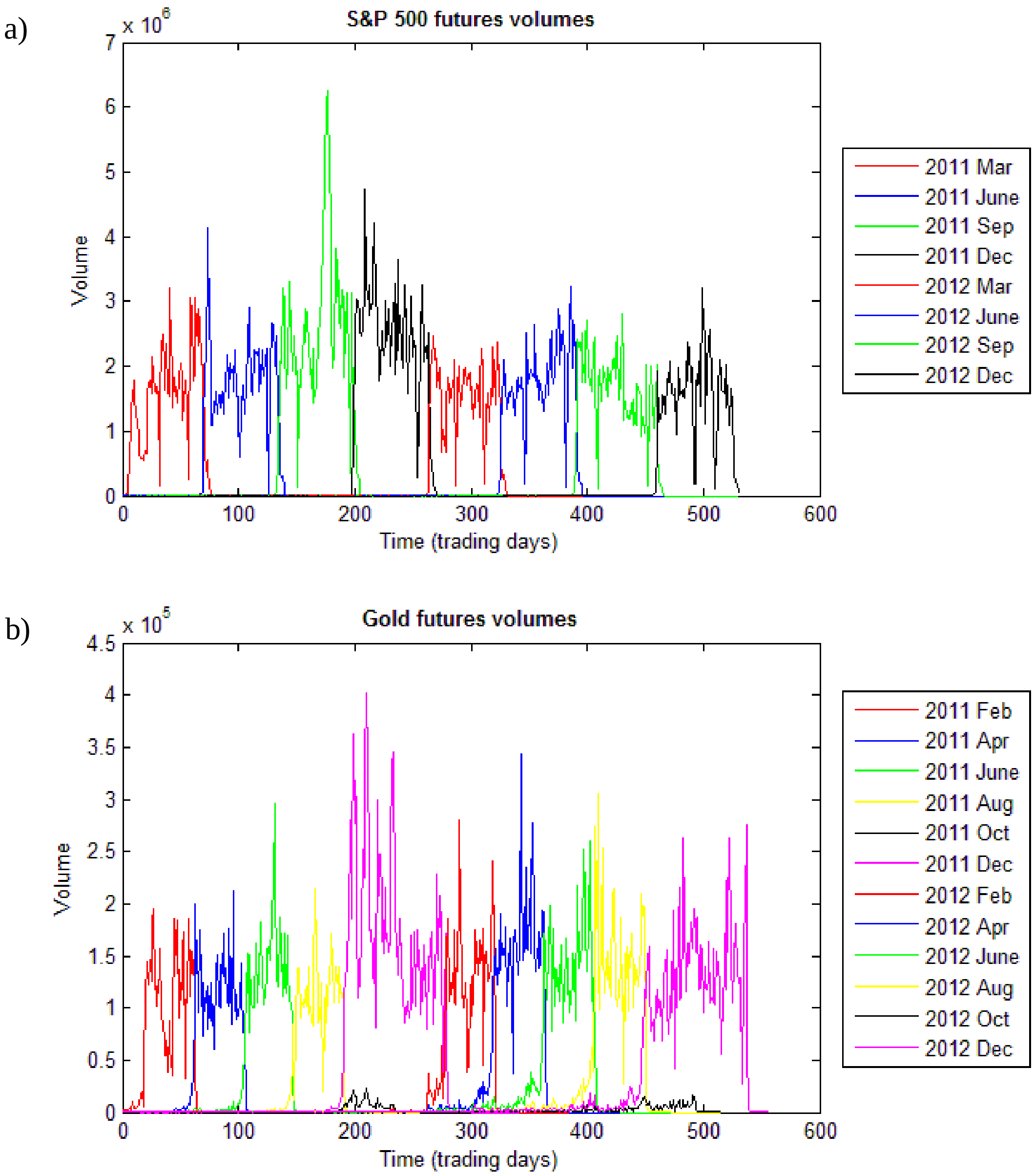


Figure 1: *Daily volumes of a) S&P 500 and b) gold future contracts*

[4] All MATLAB codes can be found in the Appendices

It is important to note here, that for the gold futures I will roll from the contract maturing in August straight to the contract maturing in December because the contract maturing in October is very illiquid and the daily trading volumes of it are much lower than the volumes of either of the other two in each day of the contract's existence. It means, that in order to get the time series for the gold futures I won't include the October contract. There is the full list of the future contracts used for the intraday volatility forecasting in the Table 1 below.

S&P 500			Gold		
Contract	**Start date**	**End date**	**Contract**	**Start date**	**End date**
2011 March	10-Dec-10	10-Mar-11	**2011 February**	29-Nov-10	27-Jan-11
2011 June	11-Mar-11	09-Jun-11	**2011 April**	28-Jan-11	28-Mar-11
2011 September	10-Jun-11	08-Sep-11	**2011 June**	29-Mar-11	26-May-11
2011 December	09-Sep-11	08-Dec-11	**2011 August**	27-May-11	27-Jul-11
2012 March	09-Dec-11	08-Mar-12	**2011 December**	28-Jul-11	28-Nov-11
2012 June	09-Mar-12	07-Jun-12	**2012 February**	29-Nov-11	27-Jan-12
2012 September	08-Jun-12	13-Sep-12	**2012 April**	30-Jan-12	28-Mar-12
2012 December	14-Sep-12	13-Dec-12	**2012 June**	29-Mar-12	29-May-12
			2012 August	30-May-12	27-Jul-12
			2012 December	30-Jul-12	28-Nov-12

Table 1: *The full list of the future contracts used and the rolling dates*

Note, this dissertation is aiming to forecast the intraday volatility of the future contracts, not their prices, so I will roll from one contract to another during one day and not by weighting the contracts over the several days. Since the returns of the different futures at the same date are almost the same (what is not the case for the prices), rolling instantly from one contract to the other do not distort the returns time series. Also, there are only 7 rolling dates for the S&P 500 contracts and 9 for the gold, so these do not affect the time series which include 2 years of data.

2.3 Time series of the intraday returns

In the financial time series analysis the analysis of the volatilities are usually done by considering the returns and more specifically the absolute returns (see [1]). Hence, as mentioned before I will create the time series for the ES and GC contracts considering the changes in the intraday prices. I will use the logarithmic returns notation in my study for finding the returns of each 5 min interval as it lets to capture the continuously compounding feature of the returns over that period. Hence,

$$r_{n,n+1} = ln(S_{n+1}) - ln(S_n) \tag{2.1}$$

where $r_{n,n+1}$ is the return over the time interval $(t_n, t_{n+1}]$ and S_i is the price of the particular contract at time t_i.

Furthermore, in spite of the fact that futures are traded 24 hours a day, the futures market is partially closed during the weekends. Also, for the analysis of the intraday volatilities it is crucial to have the data for each 5 min interval over the day. As a result, I will exclude Saturdays and Sundays from the time series which I will consider later. That won't create any discontinuities or jumps in the data as the returns and not the prices are considered. The exclusion of the weekends from the time series data are done by removing the days which have less than 288 data points.

After leaving only the days when the trading of the futures has been continued for 24 hours, the time series for the S&P 500 index has 149,472 observations which represent 519 trading days. As mentioned before, the data involves the time period from 10 December, 2010 till 13 December, 2012. Regarding the gold future contracts, the period from 29 November, 2010 till 28 November, 2012 is considered and it gives 149,184 observations representing 518 trading days.

Chapter 3

Analysing time series

The analysis in this chapter will be done just for the two thirds of the data, i.e. for 346 days which are represented by the 99,648 observations. The rest of the data will be left for the evaluation of the forecasting accuracy and for finding out the forecasting errors. Hence, the out-of-sample data which will be used for checking the accuracy won't be included to the calibration of the model and it should not give us the biased results.

3.1 Periodicity of the intraday returns

In order to find out the periodicity of the intraday returns, the average returns for each 5 min interval are calculated. As it might be seen from Figures 2 and 3, the average returns for both ES and GC contracts do not reflect any patterns in the intraday returns and may look like a standard white noise process.

a)

b)

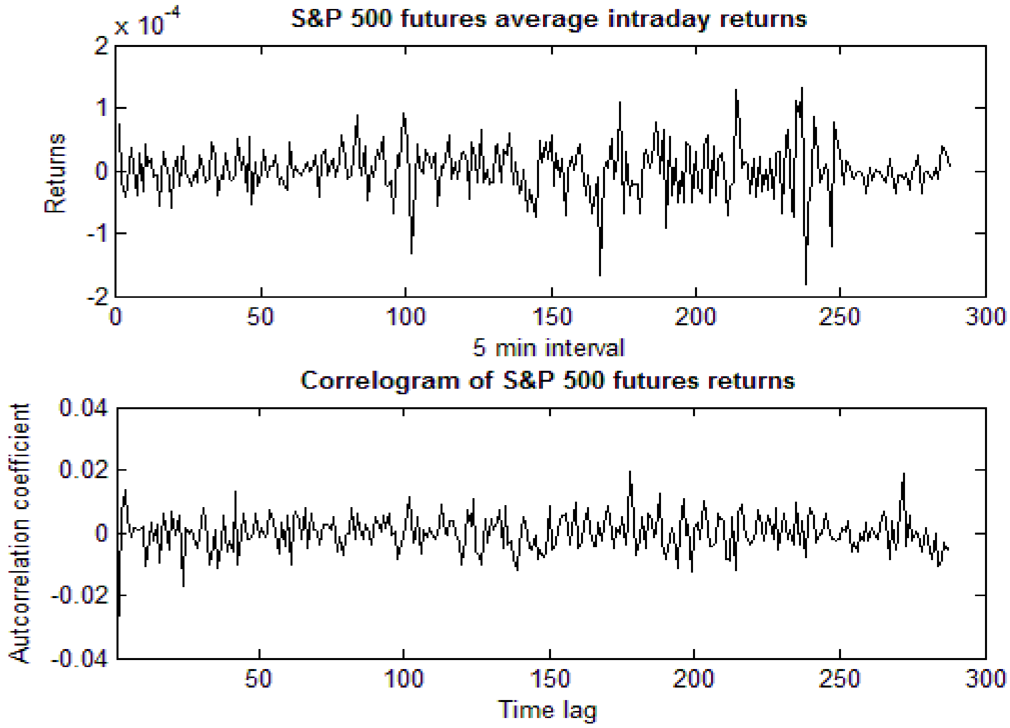


Figure 2: *S&P 500 futures a) average intraday returns and b) correlogram*

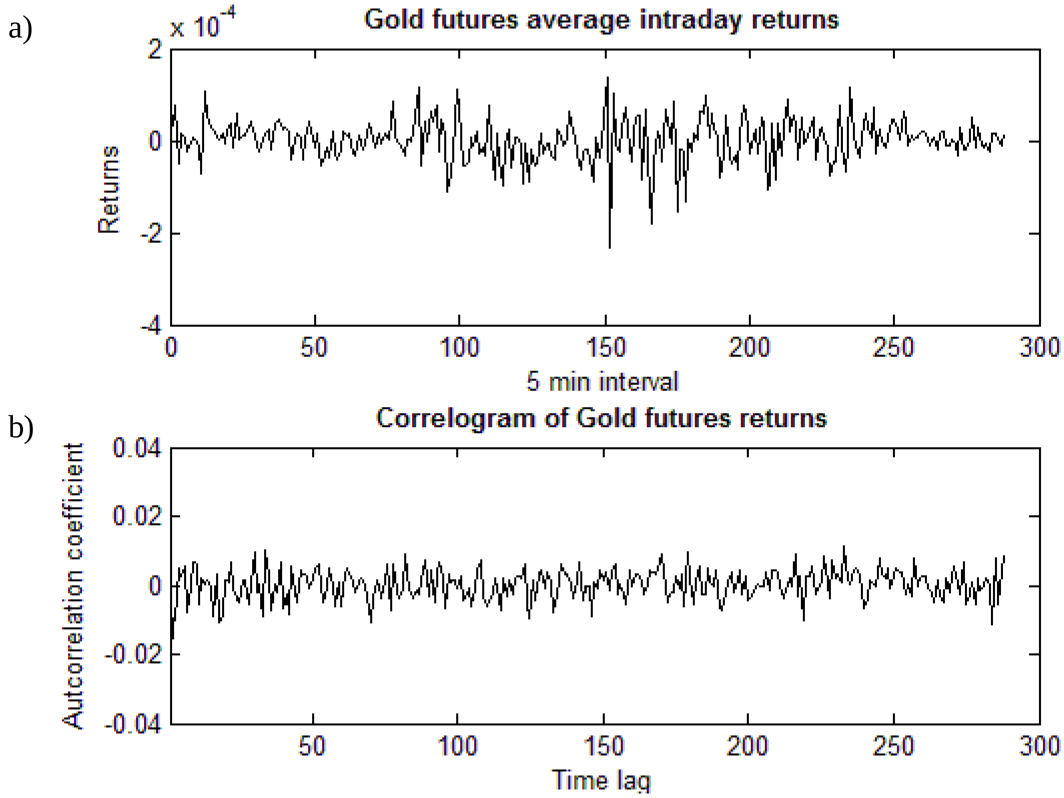


Figure 3: *Gold futures a) average intraday returns and b) correlogram*

However, the observed processes do not follow the main property of the white noise process. Although the returns have the mean value almost equal to zero, the skewness and the kurtosis for both of the time series are significantly different from the ones of the normally distributed variables. The main metrics for the returns of the S&P 500 and gold future contracts are presented in the table below:

	S&P 500	**Gold**
Mean	0.00014%	0.00013%
Standard Deviation	0.075%	0.073%
Skewness	-0.027	-0.789
Kurtosis	27.75	26.53

Table 2: *Main metrics of the intraday returns*

The other important measure for analysing the financial time series is the autocorrelation coefficients. They measure the correlation between the observed values of the same time series which are at different distances from each other. The autocorrelation coefficient at lag k is defined in [6] as

$$r_k = \frac{\frac{1}{N-k}\sum_{t=1}^{N-k}(x_t-\bar{x})(x_{t+k}-\bar{x})}{\frac{1}{N}\sum_{t=1}^{N}(x_t-\bar{x})^2} \quad (3.1)$$

where N is the size of the time series, x_t is the observation at time t, $\bar{x}$ is the average value of the observations in the time series and k is the distance at which we want to find out the correlation between the observations.

Hence, regarding the correlations between the 5 min returns, the negative first order autocorrelation coefficients of -0.035 for the ES contracts and -0.02 for the gold futures suggests that the traders are managing their market quotes asymmetrically to the market determined prices. According to Holmes and Rougier (2005), this helps to manage their inventory positions by bouncing the price of the contract up and down. Comparing the first order autocorrelation coefficients to the ones with larger lags, it appears that the latter ones do not reveal any important periodicities as their absolute values are much smaller.

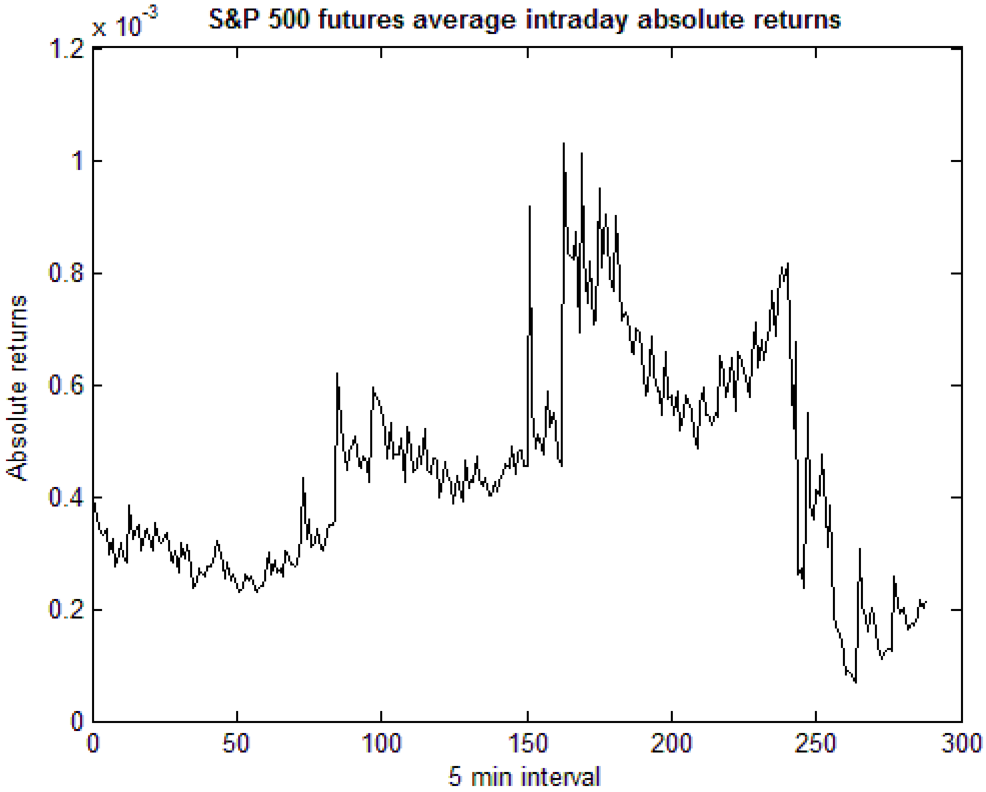
One more important thing which might be noticed from the average intraday returns is the fact that the S&P 500 returns become more volatile somewhere around the intervals 96, 162 and 240, corresponding to 8:00 GMT, 13:30 GMT and 20:00 GMT. For the gold contract, the highest returns and the largest volatility appears near the interval 150 representing 12:30 GMT. It means that the volatility of the stock index contracts are mostly affected by the opening and the afternoon trading in the European markets and the opening and closing bells in the US while gold contracts are traded much smoother except 1 hour before the opening of the US exchanges.

3.2 Periodicity of the absolute intraday returns

As suggested by Ding and Granger (1996) the good starting point for modelling the volatility from the time series perspective is to consider $|r_t|^d$ for $d = 1$, i.e. the absolute returns of the financial instrument. Even though the original paper by Ding and Granger considered the daily returns, the similar analyses have been implemented successfully by Andersen et al. for the intraday data and the highly correlated patterns have been discovered. Also, the absolute returns let us investigate the fluctuations in the financial markets considering just the magnitudes of these swings but neglecting the direction of them.

The graphs below depict the average absolute 5 min intraday returns of the S&P 500 and gold futures respectively. Comparing these graphs to the ones of the average returns, we notice the triple U-shaped patterns which suggest some correlated behaviour of the intraday returns as the approximations for the intraday volatilities.

a)



b)

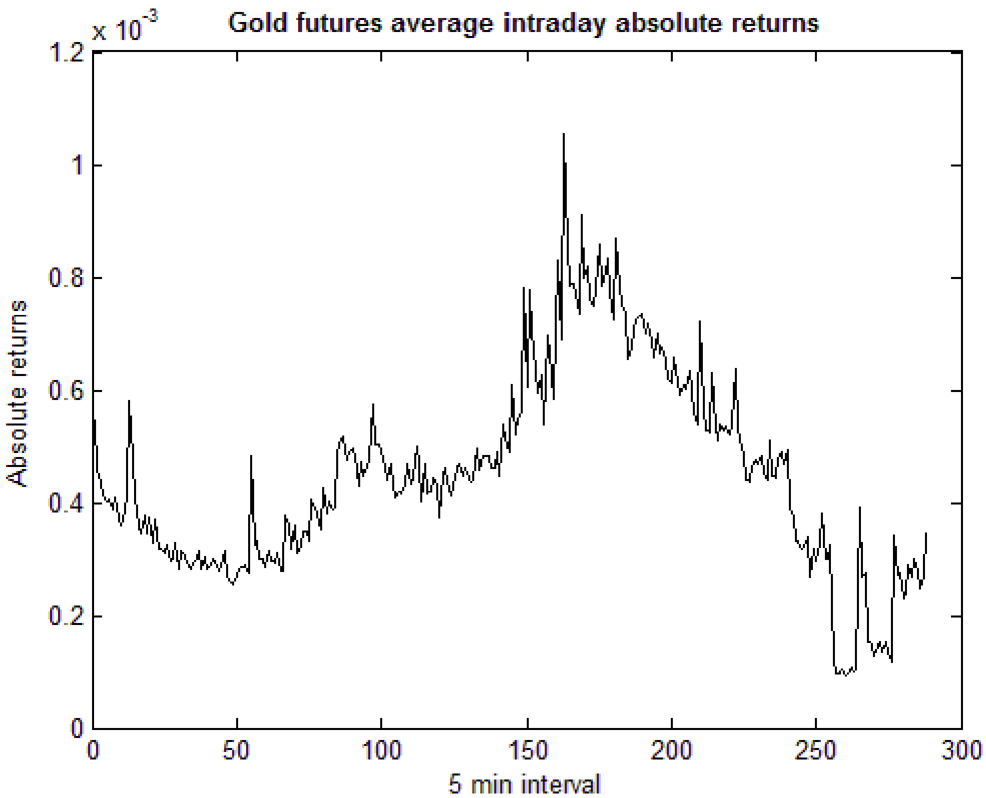
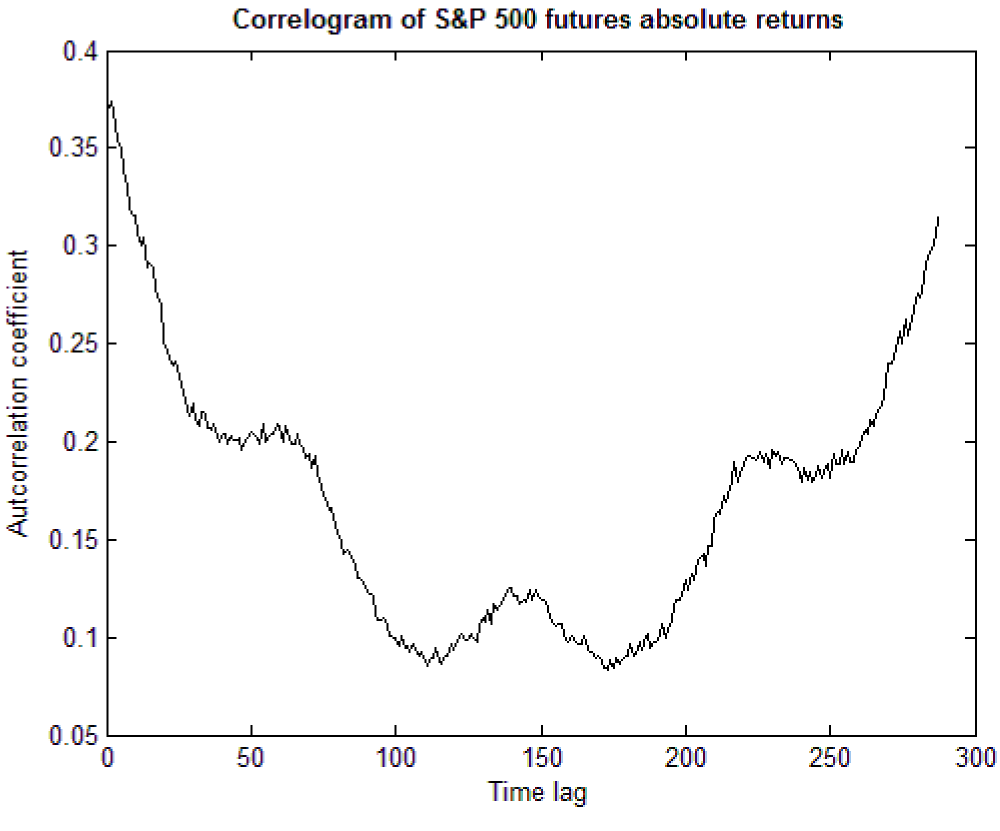


Figure 4: *Average absolute intraday returns for a) S&P 500 and b) gold futures*

The common feature of the absolute intraday average returns is that they reveal the strong dependence on the financial activity across the world. Firstly, the opening of the European financial markets at 8:00 GMT (interval 96) has a strong impact on the absolute returns making them increase to a large extent. Later, the lunch hour in London from 12:30 to 13:30 GMT (intervals 150-162) is represented by a drop in the volatilities which retrieve their losses towards the end of the European trading hours and the beginning of the trading activity in the US. However, the distinct feature of the gold market is that the magnitude of the absolute returns do not recover towards the time of the US closing bell but decreases all the time until it is affected by the activity in the Far East at the end of the day. At the same time, there is strong increase in the volatility of the S&P 500 futures but it might be mainly explained by the fact that Standard & Poors 500 is the US equity index and the activity in the US financial market mostly affects the "local" rather than "global" securities.

As the absolute returns graphs do not reflect the random behaviour at all, there should be a strong autocorrelation between them, too. Following the same procedure as in the previous section we calculate the autocorrelations for the absolute returns time series and notice the strong dependence between the data throughout the whole day. The correlograms of the ES and GC contracts are presented in Figure 5.

a)



b)

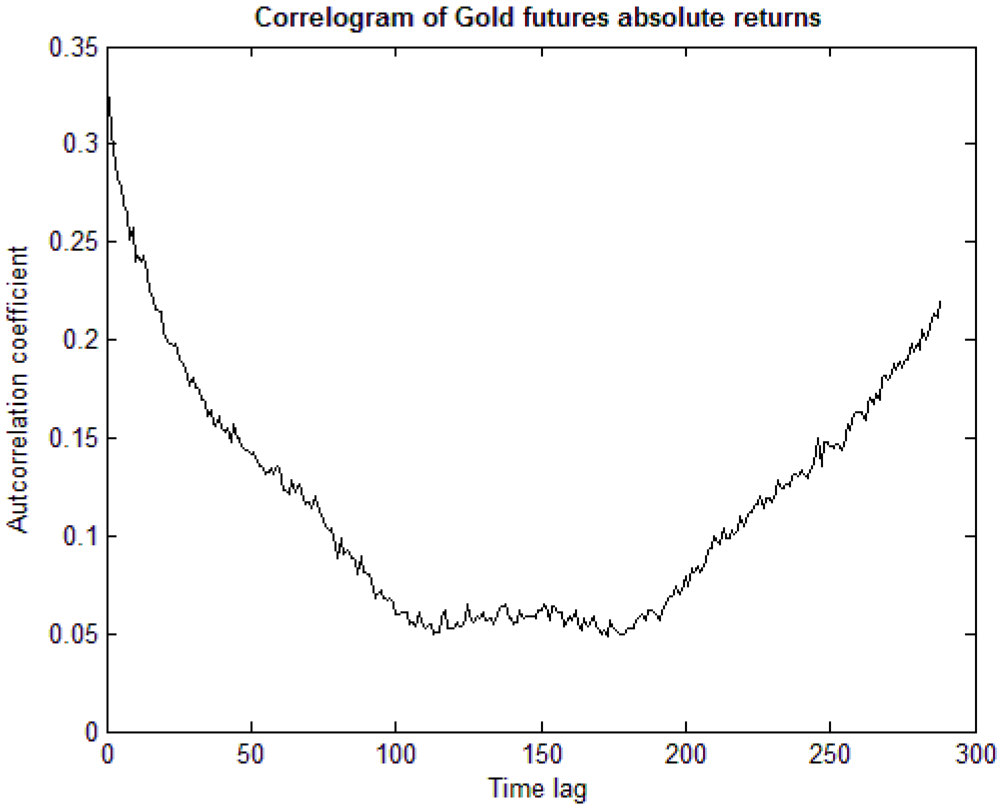
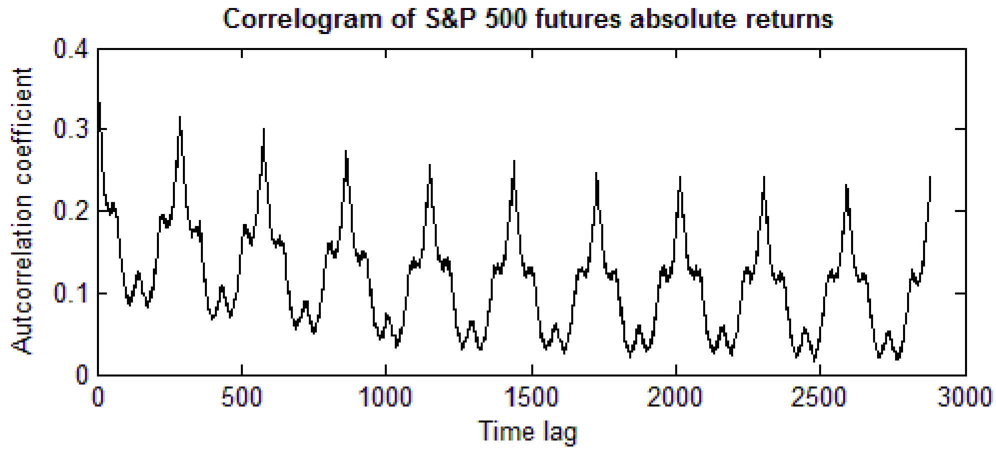


Figure 5: *Correlograms of a) S&P 500 and b) gold futures absolute intraday returns*

The correlograms depict the similar story as the absolute returns graphs but from the different perspective. The existence of the several local minima and maxima in the S&P 500 data suggests the stronger intraday volatility dependence on the opening/ closing times of the different world financial markets while the smoother correlogram of the GC absolute returns explains the smother volatility changes in the gold market. The autocorrelation is the highest for the smallest and largest lags suggesting that there is the strong dependence between the consecutive absolute returns throughout the day.

Also, looking to the autocorrelations for up to 10 days we also notice the strong intraday effects. Although the autocorrelations seem to be asymptotically decreasing, the intraday shape of them are preserved. The graphs of the autocorrelation up to lag 2,880 (representing 10 trading days period, i.e. two weeks) are presented in Figure 6.

a)



b)

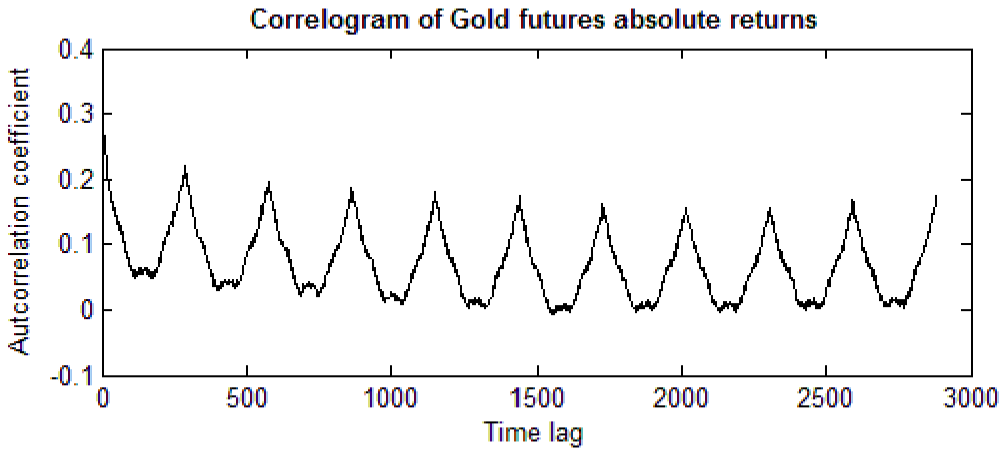


Figure 6: *Correlograms of a) S&P 500 and b) gold futures absolute intraday returns for the lags up to 10 trading days*

The fact that the correlogram is U-shaped means that the modelling of the intraday volatility by using the ARCH model should not produce any relevant results as the ARCH models imply the geometric decay in autocorrelation (see [2]). Hence, I will use the GARCH(1,1) model in the later section as it has been assumed to be one of the most effective model for volatility forecasting purposes. Also, I will develop the exponentially weighted moving average model in the following chapter and compare the forecasting errors later.

.

Chapter 4

Modelling intraday volatilities

4.1 Intraday periodic component

As it has been mentioned before, in order to observe the intraday volatility structure we should observe the distribution of the absolute intraday returns. The clear periodicity of the absolute intraday returns has been observed in the previous chapter. Hence, the model for the intraday volatilities should examine the intraday periodic component.

The model involving all these components, i.e. the intraday returns, the volatility and the intraday periodic component, has been observed by Andersen and Bollerslev in [1]. The daily returns are expressed as

$$R_t = \sum_{n=1}^{N} R_{t,n} = \sigma_t \frac{1}{\sqrt{N}} \sum_{n=1}^{N} s_n Z_{t,n} \tag{4.1}$$

where R_t is the daily continuously compounded return, $R_{t,n}$ is the intraday return from the 5 min interval $(n-1, n]$ of the day t, N is the number of intraday 5 min intervals, i.e. $N = 288$, σ_t is the conditional day t volatility factor, $Z_{t,n}$ is an i.i.d. mean zero, unit variance error term and the deterministic periodic component is denoted by s_n for the interval $(n-1, n]$. So, from the formula (4.1) we get that

$$R_{t,n} = \sigma_t \frac{1}{\sqrt{N}} s_n Z_{t,n} \tag{4.2}$$

Note, that we are considering the absolute values of the intraday returns, so $Z_{t,n}$ is also of the absolute value. Now, expressing $x_{t,n}$ as

$$x_{t,n} = \frac{R_{t,n}}{\sigma_t} \sqrt{N} \tag{4.3}$$

we get that $x_{t,n} = s_n Z_{t,n}$ and hence we can model the periodic intraday component $x_{t,n}$ by using the Fourier or polynomial regressions.

Note, that $R_{t,n}$ are the observed values of the absolute intraday returns, so the only variable needed to calculate the values of $x_{t,n}$ is the conditional daily volatility component σ_t. As the next chapter will evaluate all the forecasting models analysed here, the volatility component should be forecasted in advance. The two mostly used forecasting methods for the daily volatility factors are based on Exponentially Weighted Moving Average (EWMA) and Generalised Autoregressive Conditional Heteroskedasticity (GARCH) techniques. The next two sections will consider these two techniques and just then we will move to modelling the periodic intraday component itself.

4.2 EWMA volatility estimate

The Exponentially Weighted Moving Average of the volatility parameter is expressed as

$$\sigma_t^2 = \alpha\varepsilon_{t-1}^2 + (1-\alpha)\sigma_{t-1}^2 \tag{4.4}$$

Here, $0 < \alpha < 1$ is the parameter determining how fast the weight of the particular day's volatility decays and ε is the measure of the returns. In our case it is the continuously compounded daily return because we are trying to determine the daily volatility factor.

Note, that EWMA estimate is more convenient for the volatility estimation purposes than the simple moving average, as the decay factor α ensures that the returns of the particular day t^* have less and less weight on the overall volatility of any day $t > t^*$. In the moving average estimation all days have equal weights, so the unimportant information might be included in our estimation. By using the EWMA we ensure that the weight of the information which might be irrelevant is reduced to the minimum.

Regarding the decay factor, [18] suggests that the reasonable choice for the daily returns is $\alpha = 0.06$, and hence our model for the EWMA volatility estimation becomes

$$\sigma_t^2 = 0.06\varepsilon_{t-1}^2 + 0.94\sigma_{t-1}^2 \tag{4.5}$$

It means that the volatility estimate approximately depends on the last 17 days returns since $\frac{1}{0.06} \approx 17$.

4.3 GARCH volatility estimate

The GARCH model has been introduced by Bollerslev (see ref. [3]) and it was the generalisation of the Autoregressive Conditional Heteroskedasticity (ARCH) model developed by Engel (ref. [10]). These models let the conditional variance change in time but unconditional variance is kept constant. The historical conditional variances are also included into the calculation of the current variance, hence, the variance dependence on the historical fluctuations is preserved. Formally, the GARCH(p,q) model is expressed as

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \, \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \, \sigma_{t-j}^2 \tag{4.6}$$

where ε_i^2 are the squared daily returns as in the EWMA model. The parameters α_i and β_j for $0 \le i \le p$ and $0 \le j \le q$ are estimated by using the common techniques such as ordinary least squares.

According to Bollerslev (1986), the GARCH(1,1) model is a good approximation for the daily volatilities of most of the financial instruments. Since my analysis is considering the futures contracts of the S&P 500 equity index and gold, this model will be used for volatility estimation throughout this paper. The GARCH(1,1) volatility estimate is expressed form the formula (4.6) and is given by

$$\sigma_t^2 = \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \tag{4.7}$$

Note, the values of α_1 determines how fast the volatility reacts to the shocks in the volatility levels. It means, that for large α_1 the shock will be almost instantly reflected in the daily volatility forecast for the next period, whereas small values of this coefficient predicts the smoother volatility pattern for the future. The large values of β_1 represents the cases when the large changes in the volatility will affect the future volatilities for a long period of time as their decay is slower.

As S&P 500 and gold contracts have different volatility structures, the GARCH(1,1) estimates for both of them have different model coefficients. After applying this GARCH model to the financial time series used for the model estimation for both future contracts[5], we

[5] The GARCH(1,1) coefficients are estimated by using the MFE MALAB toolbox created by Kevin Sheppard, University of Oxford

get that the coefficients for the S&P 500 futures are $\alpha_0 = 2.58 \times 10^{-6}$, $\alpha_1 = 0.1459$ and $\beta_1 = 0.8433$ and for the gold contract they are $\alpha_0 = 1.66 \times 10^{-5}$, $\alpha_1 = 0.2158$ and $\beta_1 = 0.676$. Hence, for the gold contracts the shock in the volatility decays for a shorter period of time but the volatility reacts faster to these shocks. Also, the sums of the GARCH coefficients are less than 1 for both of the contracts which ensures that the GARCH processes are covariance stationary.

4.4 Polynomial regression

The first method to approximate the intraday periodic component is done by using the linear polynomials regression. In this case, the values of $x_{t,n}$ are approximated by the polynomials of order m, i.e.

$$x_{t,n} = \sum_{i=0}^{m} a_i \left(\frac{n}{N}\right)^i = a_0 + a_1\left(\frac{n}{N}\right) + a_2\left(\frac{n}{N}\right)^2 + \cdots + a_m\left(\frac{n}{N}\right)^m \tag{4.8}$$

where n represents the n^{th} interval in the day and $N = 288$ is the number of 5 min intervals in the day.

The linear regression is done by using the ordinary least squares method, i.e. by minimising the sum of the squared errors of the estimates from the observed values of $x_{t,n}$. The regression returns the vector of the regression coefficients A such that $X = NA$ where

$$X = \begin{pmatrix} x_{1,1} \\ x_{1,2} \\ \vdots \\ x_{1,288} \\ x_{2,1} \\ \vdots \\ x_{346,288} \end{pmatrix}, \quad N = \begin{pmatrix} 1 & \frac{1}{288} & \left(\frac{1}{288}\right)^2 & \cdots & \left(\frac{1}{288}\right)^m \\ 1 & \frac{2}{288} & \left(\frac{2}{288}\right)^2 & \cdots & \left(\frac{2}{288}\right)^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \frac{288}{288} & \left(\frac{288}{288}\right)^2 & \cdots & \left(\frac{288}{288}\right)^m \\ 1 & \frac{1}{288} & \left(\frac{1}{288}\right)^2 & \cdots & \left(\frac{1}{288}\right)^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \frac{288}{288} & \left(\frac{288}{288}\right)^2 & \cdots & \left(\frac{288}{288}\right)^m \end{pmatrix}, \quad A = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{pmatrix} \tag{4.9}$$

i.e. X is $tn \times 1$ vector of $x_{t,n}$ values, N is $tn \times (m+1)$ regression matrix and A is $(m+1) \times 1$ vector of regression coefficients.

Note, the representation of $x_{t,n}$ in (4.8) does not depend on the particular day t, but depends only on the intraday interval n. Hence, the entries of the regression matrix are periodic, so the regression will produce the unique set of the periodic intraday estimates $x_1, x_2, \dots, x_{288}$ which will be used to forecast the intraday absolute returns by using (4.2).

In order to find out m, I will compare the Mean Absolute Errors (MAE) for the different degrees of the polynomial. The MAE is the average absolute error between the forecasted intraday returns and the observed values. It is defined as

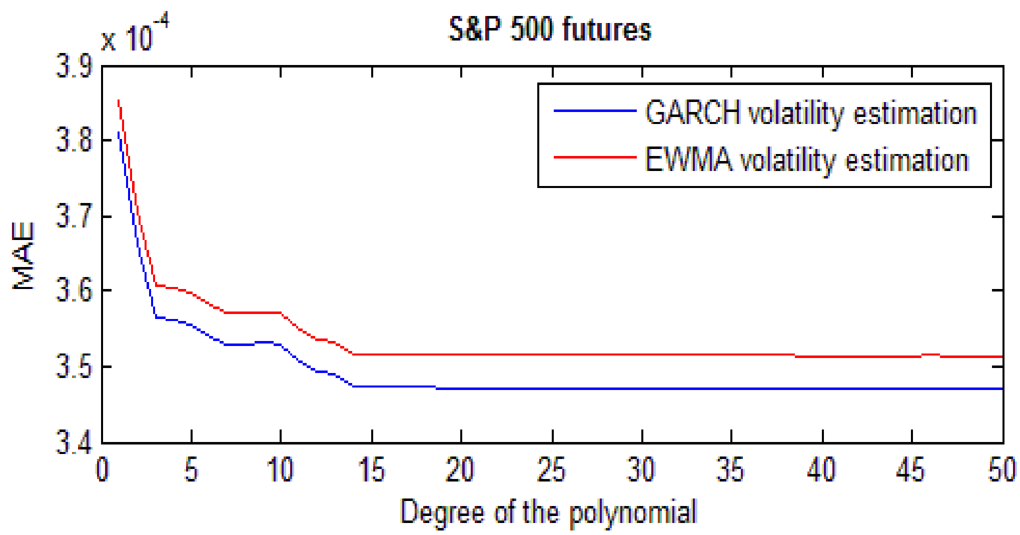
$$MAE = \frac{1}{N}\sum_{i=1}^{N} |\hat{r}_i - r_i| \tag{4.10}$$

where $\hat{r}_i$ is the estimated return, r_i is the observed return and N is the total number of observations.

Note that this analysis will be done on the same data set used for estimating GARCH and EWMA volatilities, i.e. for the in-sample data. Only when the polynomial degree is determined, the same analysis will be repeated on the out-of-sample data in order to evaluate the accuracies of the different forecasting methods.

The graphs representing the MAE versus the degree of the polynomial for S&P 500 and gold futures are presented in Figure 7 below.

a)



b)

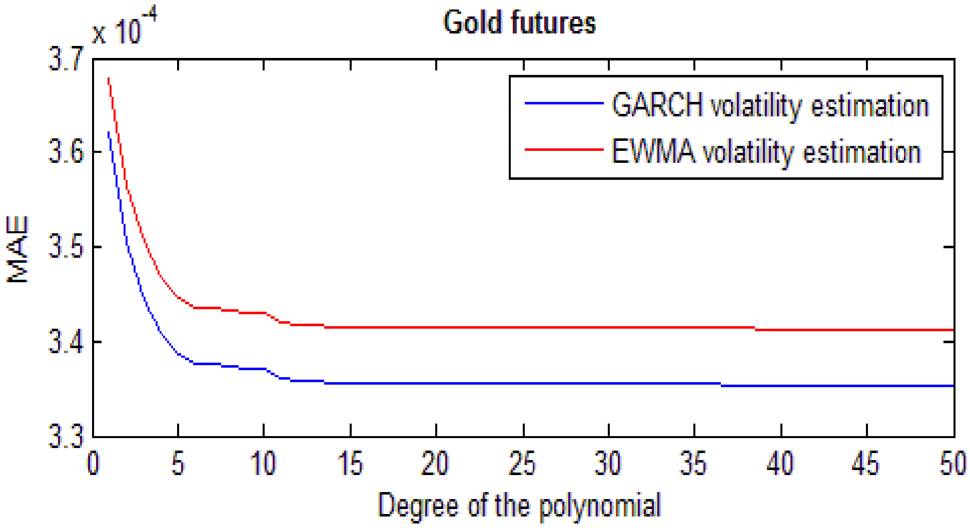


Figure 7: *comparison of MAE vs. degree of the polynomial for a)S&P 500 and b) gold futures*

It is clear, that the polynomial of degree more than 14 do not add any significant improvement in the accuracy to our model, so the final regression is done for the polynomials of the degree $m = 14$, i.e. the periodic intraday coefficient $x_{t,n}$ is estimated as

$$x_{t,n} = \sum_{i=0}^{14} a_i \left(\frac{n}{N}\right)^i \tag{4.11}$$

4.5 Fourier regression

The Fourier regression described in this section is similar to the polynomial one from before, but instead of trying to approximate the periodic intraday component by the polynomial we will try to use the Fourier representation of it, i.e. the one involving sines and cosines. According to [14], the Fourier representation of $x_{t,n}$ in this case will be given by

$$x_{t,n} = a + \sum_{i=1}^{m} \left(a_i \cos\left(2\pi i \frac{n}{N}\right) + b_i \sin\left(2\pi i \frac{n}{N}\right)\right) \tag{4.12}$$

The regression is done in a similar way as in the previous section, but the regression matrix N and the regression coefficients vector A are different:

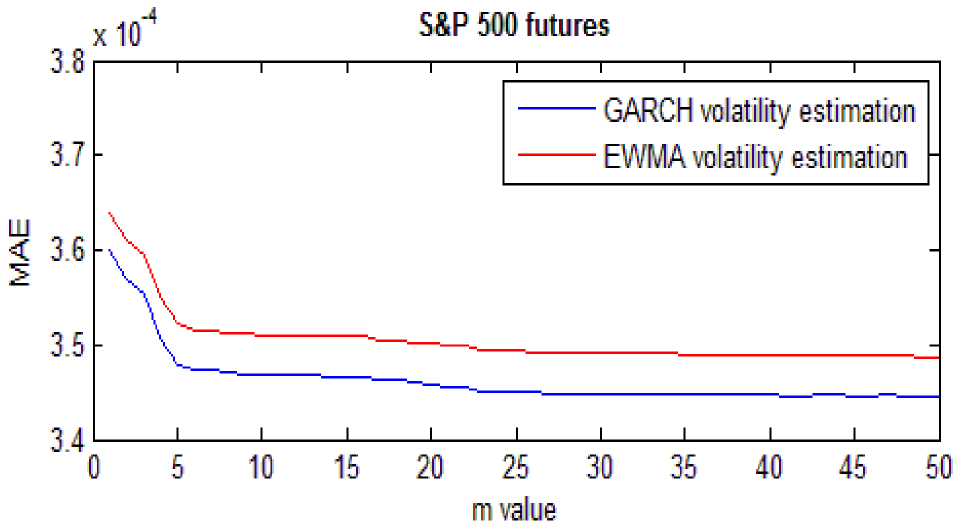
$$N = \begin{pmatrix} 1 & \cos\left(2\pi\frac{1}{288}\right) & \sin\left(2\pi\frac{1}{288}\right) & \cdots & \cos\left(2\pi m\frac{1}{288}\right) & \sin\left(2\pi m\frac{1}{288}\right) \\ 1 & \cos\left(2\pi\frac{2}{288}\right) & \sin\left(2\pi\frac{2}{288}\right) & \cdots & \cos\left(2\pi m\frac{2}{288}\right) & \sin\left(2\pi m\frac{2}{288}\right) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \cos\left(2\pi\frac{288}{288}\right) & \sin\left(2\pi\frac{288}{288}\right) & \cdots & \cos\left(2\pi m\frac{288}{288}\right) & \sin\left(2\pi m\frac{288}{288}\right) \\ 1 & \cos\left(2\pi\frac{1}{288}\right) & \sin\left(2\pi\frac{1}{288}\right) & \cdots & \cos\left(2\pi m\frac{1}{288}\right) & \sin\left(2\pi m\frac{1}{288}\right) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \cos\left(2\pi\frac{288}{288}\right) & \sin\left(2\pi\frac{288}{288}\right) & \cdots & \cos\left(2\pi m\frac{288}{288}\right) & \sin\left(2\pi m\frac{288}{288}\right) \end{pmatrix} \quad (4.13)$$

$$A = \begin{pmatrix} a \\ a_1 \\ b_1 \\ a_2 \\ \vdots \\ a_m \\ b_m \end{pmatrix} \quad (4.14)$$

i.e. N is $tn \times (2m+1)$ regression matrix and A is $(2m+1) \times 1$ vector of regression coefficients.

In order to determine the coefficient m, the similar analysis of the Mean Absolute Errors is done as in the previous section. The results are presented in Figure8:

a)



b)

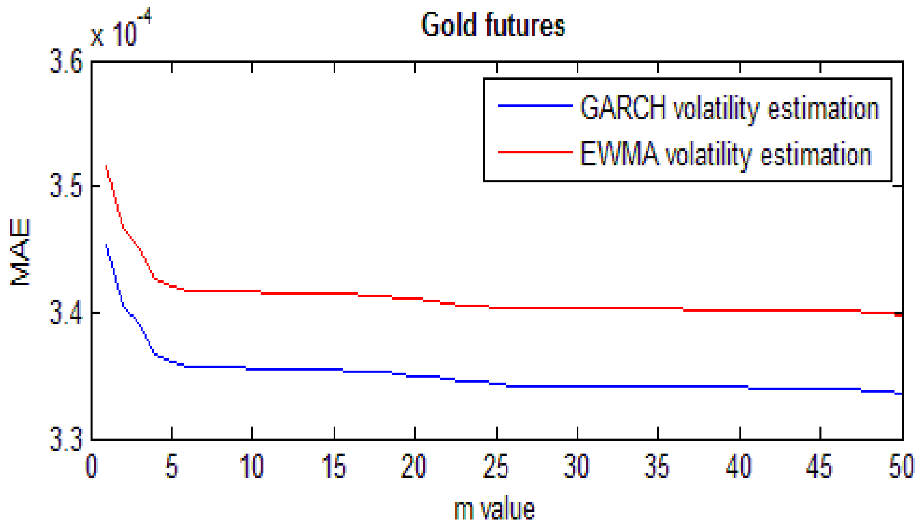


Figure 8: *comparison of MAE vs. the value of m in Fourier series for a) S&P 500 and b) gold futures*

As the values of m greater than 6 do not significantly improve the accuracy of the approximation of the intraday periodic components by the Fourier series, the regression is done by using the Fourier approximation up to the factor $m = 6$. Hence, (4.12) becomes

$$x_{t,n} = a + \sum_{i=1}^{6} \left(a_i \cos\left(2\pi i \frac{n}{N}\right) + b_i \sin\left(2\pi i \frac{n}{N}\right)\right) \tag{4.15}$$

4.6 Flexible Fourier Form (FFF) regression

The third method for modelling the periodic intraday factor is known as the Flexible Fourier Form. Developed by Gallant (see [13]) this method has been also proved to be valid for modelling intraday periodic returns for the financial securities by Andersen and Bollerslev (1997, 1999). The FFF regression method involves both the polynomial and Fourier methods described above hence it might produce the better forecasting results than any of them. The intraday periodic component in [2] is expressed as:

$$x_{t,n} = \sum_{q=0}^{Q} a_i \left(\frac{n}{N}\right)^q + \sum_{p=1}^{P} \left(b_i \cos\left(2\pi p \frac{n}{N}\right) + c_i \sin\left(2\pi p \frac{n}{N}\right)\right) \tag{4.16}$$

where the variables are the same as in (4.8) and (4.12).

The regression matrix and the vector of regression coefficients in this case are

$$N = \begin{pmatrix} 1 & \cdots & \left(\frac{1}{288}\right)^Q & \cos\left(2\pi\frac{1}{288}\right) & \cdots & \sin\left(2\pi P\frac{1}{288}\right) \\ 1 & \cdots & \left(\frac{2}{288}\right)^Q & \cos\left(2\pi\frac{2}{288}\right) & \cdots & \sin\left(2\pi P\frac{2}{288}\right) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \cdots & \left(\frac{288}{288}\right)^Q & \cos\left(2\pi\frac{288}{288}\right) & \cdots & \sin\left(2\pi P\frac{288}{288}\right) \\ 1 & \cdots & \left(\frac{1}{288}\right)^Q & \cos\left(2\pi\frac{1}{288}\right) & \cdots & \sin\left(2\pi P\frac{1}{288}\right) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \cdots & \left(\frac{288}{288}\right)^Q & \cos\left(2\pi\frac{288}{288}\right) & \cdots & \sin\left(2\pi P\frac{288}{288}\right) \end{pmatrix}, \quad A = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_Q \\ b_1 \\ c_1 \\ \vdots \\ b_P \\ c_P \end{pmatrix} \qquad (4.17)$$

i.e. N is $tn \times (2P + Q + 1)$ regression matrix and A is $(2P + Q + 1) \times 1$ vector of regression coefficients.

Andersen, Bollerslev and Cai suggest in [2] that the Fourier terms in (4.16) do not add any significant estimates for $Q > 2$ and $P > 6$, so the model precision by using $Q = 2$ and $P = 6$ is enough to capture the behaviour of the intraday periodicities. In this case our approximation (4.16) becomes

$$x_{t,n} = \sum_{q=0}^{2} a_i \left(\frac{n}{N}\right)^q + \sum_{p=1}^{6} \left(b_i \cos\left(2\pi p\frac{n}{N}\right) + c_i \sin\left(2\pi p\frac{n}{N}\right)\right) \qquad (4.18)$$

4.7 Dummy variables regression

The last technique for modelling the periodic intraday component is done via modelling each time interval within the day itself. It means that each 5 minutes interval is considered as an independent variable. In [16] it is defined as the dummy variables technique. Theoretically,

$$x_{t,n} = a_n, \qquad \text{for } 1 \leq n \leq 288 \qquad (4.19)$$

Hence, out regression matrix and the vector of coefficients are defined as

$$N = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix}, \quad A = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{288} \end{pmatrix} \tag{4.20}$$

i.e. N is $tn \times 288$ regression matrix and A is 288×1 vector of regression coefficients.

The dummy variables technique lets us approximate the sharp increases and decreases in the volatility levels related to the openings or closings of different financial markets discussed in Chapter 3. Unlike the polynomial or Fourier cases, the dummy variables approach does not smoothen the intraday periodic component, hence it might give us the more accurate predictions.

Chapter 5

Forecasting and evaluation

5.1 One-step intraday volatility forecasting

The forecasting of the fluctuations in the intraday absolute returns will be based on our modelling techniques of the daily volatilities and the intraday periodic factor discussed in the Chapter 4. The daily volatility will be forecasted using EWMA and GARCH(1,1) techniques for the coefficients found in section 4.3. It will let us compare the different techniques and to find out which ones are working the best for the S&P 500 and gold future contracts.

According to (4.2) – (4.3), the intraday absolute returns (as the approximations to the intraday volatility levels) might be forecasted by using the formula

$$R_{t,n} = \hat{\sigma}_t \frac{1}{\sqrt{N}} \times x_n \tag{5.1}$$

where $\hat{\sigma}_t$ is the forecast of the daily volatility and x_n is the intraday periodic factor modelled by using techniques explained in Chapter 4.

The comparison of the forecasting accuracy will be done by using the Mean Absolute Error (MAE) defined in (4.10). It is not possible to compare the Mean Absolute Percentage Errors (MAPE) of the different techniques as the observations have a lot of intervals with zero returns (it means that no contracts had been traded during these 5 min intervals) and calculating percentage errors includes the division by the observed value. Hence, for the observed zero returns the percentage errors are undefined.

Moreover, in order to evaluate all of the previously mentioned forecasting techniques I will use the out-of-sample data and not the same one used for the estimation of the daily volatility models and the calibration of the periodic intraday coefficient regression techniques. As mentioned before, the out-of-sample data includes 49,824 observations (representing 173 trading days) for the S&P 500 futures and 49,536 observations (representing 172 trading days) for the gold contracts.

In Figure 9 you can see the similar graphs as in Section 3.2, but this time the average absolute intraday returns are calculated for the out-of-sample data and the average forecasted intraday returns are also presented.

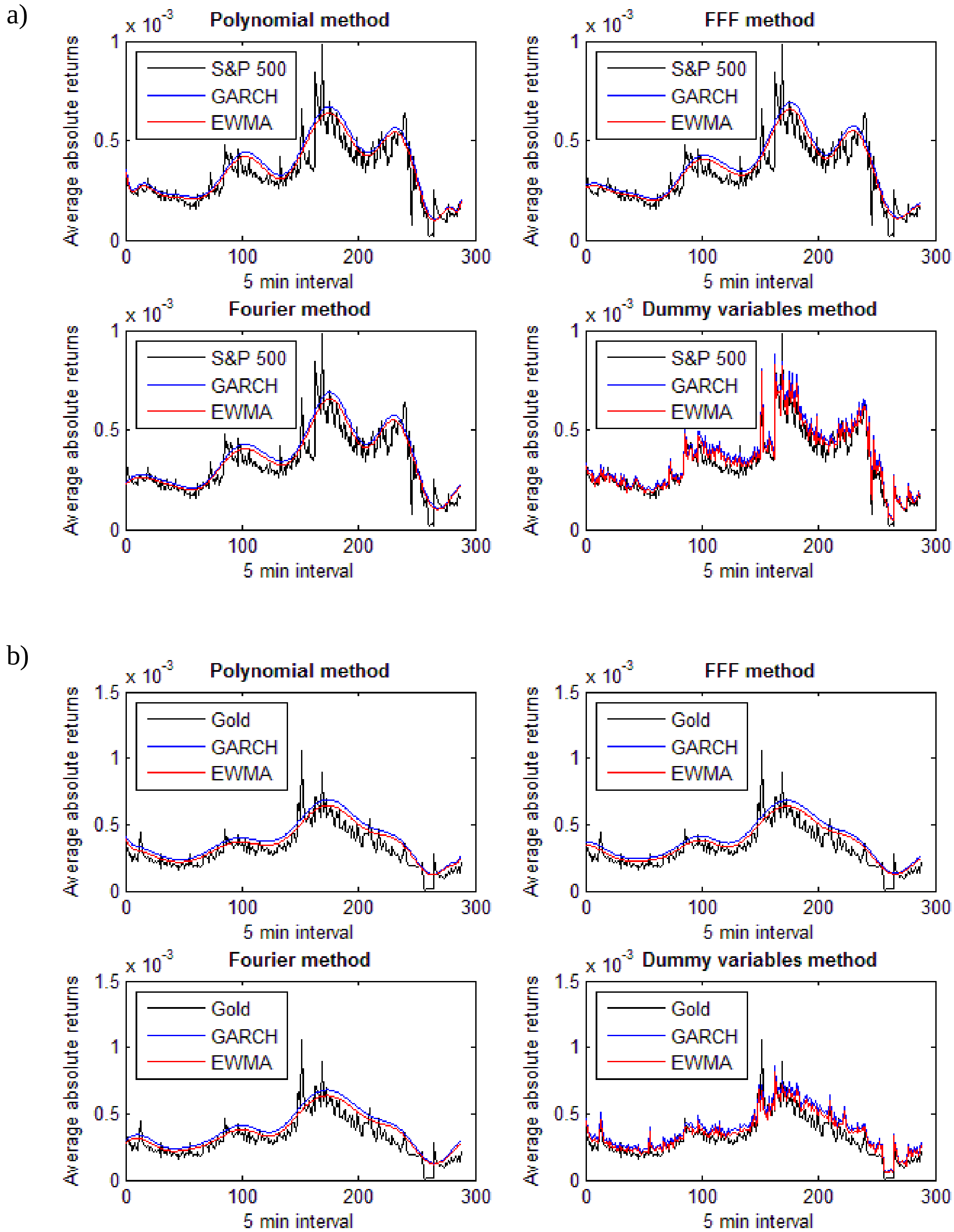


Figure 9: *Average forecasted intraday absolute returns for a) S&P 500 and b) gold futures*

As expected, the first three methods (Polynomial, Fourier and FFF) give the smoothened values of the average returns while the dummy variables technique also captures the shocks in the volatilities of the traded instruments. Note, that in all of the cases the GARCH volatility estimation gives the higher returns than the EWMA but the difference is not significant. Despite of that, both EWMA and GARCH methods give almost the same forecasts for the intraday absolute returns.

When it comes to the MAEs of the forecasting, you can find the table below summarising the accuracies of all the models:

S&P 500	**GARCH**	**EWMA**	**Gold**	**GARCH**	**EWMA**
	$\times 10^{-4}$			$\times 10^{-4}$	
Polynomial	2.551	2.506	**Polynomial**	2.547	2.430
Fourier	2.553	2.509	**Fourier**	2.549	2.432
FFF	2.552	2.508	**FFF**	2.548	2.431
Dummy variables	2.521	2.477	**Dummy variables**	2.524	2.406

Table 3: *Mean Absolute Errors of different forecasting techniques*

As expected the most accurate method for the intraday volatility forecasting, which is estimated as the absolute intraday returns, is the Dummy variables method because it better captures the volatility shocks related to the openings and closings of different financial markets in the world. Note, that the difference between MAEs of the different techniques is of order 10^{-5}, so they seem to be small in their absolute value. However, if the trader manages to achieve the better returns by 10^{-5} on average for every 5 min interval, it gives the difference of almost 0.3% per day and assuming 252 trading days per year the difference is 72.5% which is a reasonable improvement in the forecasting.

Figure 10 compares the Mean Absolute Errors for each 5 min interval throughout the day and we cannot notice any major advantages of one or another model in terms of the accuracy. The MAEs are almost the same for all the techniques all the time and it is impossible to separate the intervals of time where one or another technique could give us smaller errors. As we have already noticed, the GARCH and EWMA estimates of the daily volatility gives the forecasts which repeat each other, the graphs below represent only the GARCH case. It gives us the advantage as the graph is easier to read due to the lower number of MAEs plots.

a)

b)

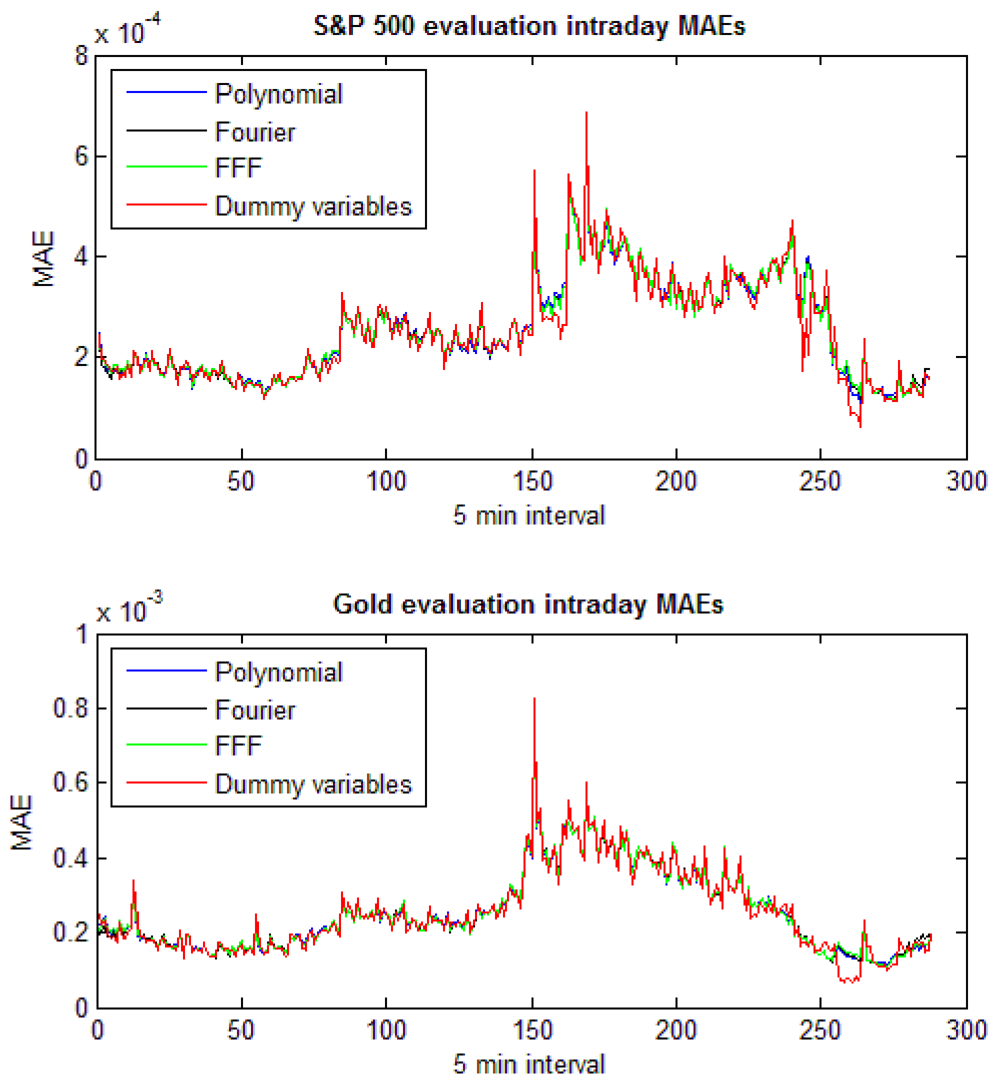


Figure 10: *MAEs for each 5 min interval for a) S&P 500 and b) gold future contracts*

5.2 Multi-step intraday volatility forecasting

The previous section considered the one-step intraday forecasting, i.e. the forecast of the intraday returns was made for the following day. Here, I will consider the forecasting for up to 5 trading days in the future. It is equal to 1 business week and if we are considering the algorithmic trading which is mainly related to short term fast trading activities, the longer forecasting horizon does not add any significant value. The reason for the multi-step forecasting is the attempt to find out if the same conclusion as in the previous section can be made if we are looking for the longer period of time in the future.

Both EWMA (formula (4.4)) and GARCH(1,1) (formula (4.7)) include ε_{t-i}^2 in their definitions. As we have discussed before, it represents squared returns so we cannot just use the value of the forecasted σ_t^2 to find the next day forecast σ_{t+1}^2 since we do not know what ε_t^2 is, i.e. we have not forecasted the next day's returns in advance yet (assuming that it is day $t-1$ at the moment). In order to make the multi-step forecasting work, I will use the Autoregressive model AR(1) to forecast the daily returns for the future.

As defined in [5], AR(1) process is such that

$$X_t = \phi X_{t-1} + Z_t, \quad t = 0, \pm 1, \ldots, \tag{5.2}$$

where $Z_t \sim N(0, \sigma^2)$, $|\phi| < 1$ and Z_t is uncorrelated with X_s for all $s < t$. Note, in the forecasting, I will use σ^2 in Z_t forecasted by the previously mentioned GARCH(1,1) or EWMA models. Hence, I will call these models AR(1)-GARCH(1,1) and AR(1)-EWMA respectively. Also, X_t will be the representation of the daily returns and by squaring it I will get ε_t^2 which will be used in the formulas (4.4) and (4.7) to forecast the daily volatilities for more than 1 day in the future, i.e.

$$X_t = \phi X_{t-1} + Z_t, \qquad \sigma_{t+1}^2 = \alpha_0 + \alpha_1 X_t^2 + \beta_1 \sigma_t^2 \tag{5.3}$$

or

$$X_t = \phi X_{t-1} + Z_t, \qquad \sigma_{t+1}^2 = X_t^2 + (1-\alpha)\sigma_t^2 \tag{5.4}$$

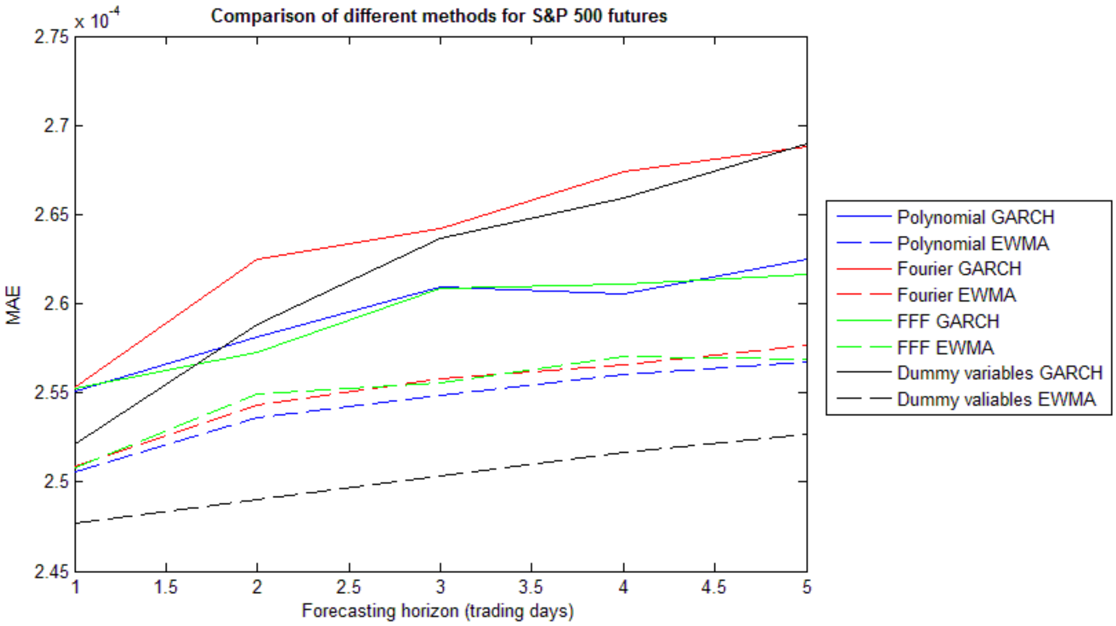
depending on which model we use to forecast the daily volatilities.

By using the in-sample data we find that the value of ϕ for the S&P 500 futures daily returns is -0.1485 and for the Gold contracts it is 0.0175[6]. Hence, by (5.2) I forecast the daily returns for up to 4 days in advance for the out-of-sample data and use these forecasts to get the daily volatility forecasts for the out-of-sample data for up to 5 trading days in advance.

The comparison of the different forecasting techniques is done in a similar way as in the previous section. The Mean Absolute Errors for all of them are calculated and plotted against the time horizon. The graphs comparing the MAEs of the intraday volatility forecasting techniques for the S&P 500 and Gold futures are presented in Figure 11.

[6] The coefficients are found by using the same MATLAB package as for the coefficients of GARCH(1,1) model

a)



b)

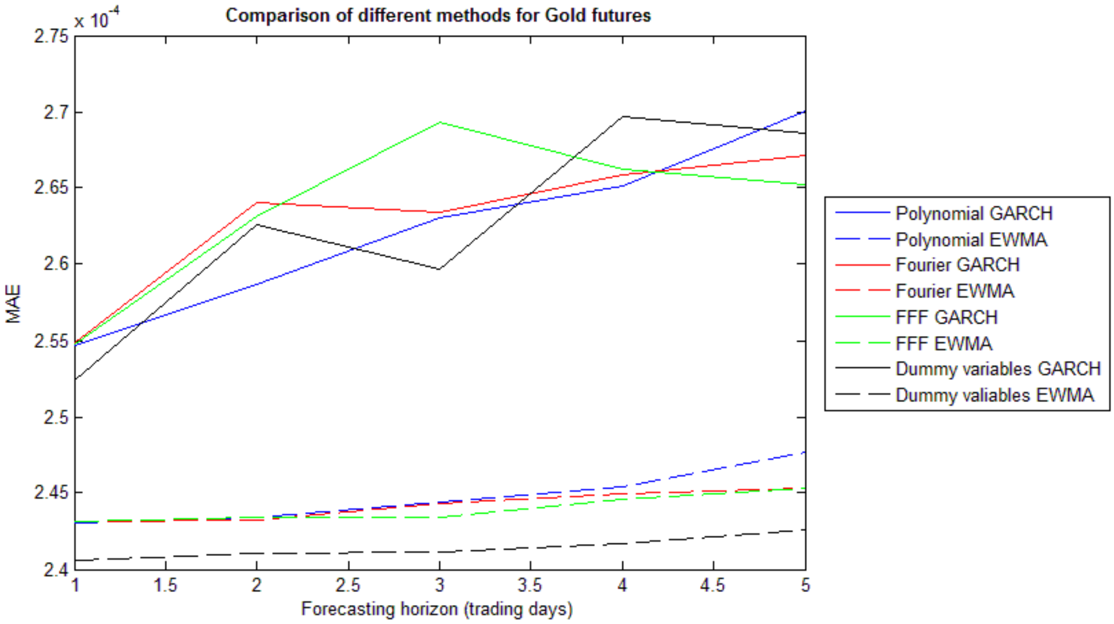


Figure 11: *Comparison of the multi-step forecasting errors for a) S&P 500 and b) gold futures*

As in the previous section, the Dummy variables technique using the EWMA volatility forecasting method gives us the most accurate results for both future contracts. Note, that other techniques using the EWMA daily volatility estimate give almost the same accuracy while all the models based on the GARCH(1,1) produce higher errors.

Chapter 6

Conclusion

The main aim of this dissertation was to analyse and to compare the different techniques of the intraday volatility forecasting for the futures contracts and to find out which one of them gives us the most accurate forecasted results. In order to achieve this aim the thesis has been split into several parts.

First of all, the future contracts of the S&P 500 index and gold have been used to create the time series for the intraday returns. Since the future contracts have different maturity dates and are usually traded during the different parts of the year, the problem of getting the consistent time series has been solved by rolling from one futures contract to the other one as soon as the latter becomes more liquid.

Moreover, the average intraday returns and the absolute values of them have been analysed. Neither the intraday returns nor their autocorrelations with different lags revealed any important information which could help us forecast the future intraday volatility levels. However, the absolute average intraday returns revealed the periodic pattern throughout the day and the autocorrelations of them suggested the strong interdependence between them. Hence, it has been decided to model the periodic intraday component which might be used to capture the intraday fluctuations in the absolute returns.

The most important part of this dissertation involved modelling the previously mentioned intraday periodic component and using it to forecast future intraday volatility levels (which are assumed to be reflected by the absolute intraday returns). The modelling has been done by using 4 different techniques, i.e. approximating it by the polynomial, by the Fourier series, by the Fast Fourier Form method and finally by modelling each intraday interval as an independent variable. Also, two different ways of forecasting the future daily volatilities have been used which are EWMA and GARCH(1,1) methods.

Finally, the comparison of all these techniques using the out-of-sample data revealed that the most accurate method for the intraday volatilities forecasting is the Dummy variables method where the daily volatilities are estimated by using EWMA model. The only disadvantage of this method comparing to the other three is that it involves modelling 288 different variables which is also reflected in the cost of the computational time. However, the higher accuracy is mainly reflected by the fact that by modelling each time interval within the day it is easier to capture the shocks in the absolute returns levels than by the smoothened polynomial or Fourier methods. Hence, the Dummy variables estimation of the intraday component gives us the best forecasting results not only for 1 day ahead but also for the longer time horizon.

References

[1] Andersen, T. G. and Bollerslev, T. (1997), 'Intraday periodicity and volatility persistence in financial markets', *Journal of Empirical Finance*, vol. 4, pages 115-158

[2] Andersen, T. G., Bollerslev, T. and Cai, J. (2000),'Intraday and interday volatility in Japanese stock market', *Journal of International Financial Markets, Institutions and Money*, vol. 10, pages 107-130

[3] Bollerslev, T. (1986), 'Generalised Autoregressive Conditional Heteroskedasticity', *Journal of Econometrics*, vol. 31, pages 307-327

[4] Bollerslev, T., Cai, J. and Song, F. M. (2000), 'Intraday periodicity, long memory volatility, and macroeconomic announcement effects in the US Treasury bond market', *Journal of Empirical Finance*, vol. 7, pages 37-55

[5] Brockwell, P. J. (2002), 'Introduction to Time Series and Forecasting', 2nd edition, New York: Springer

[6] Chatfield, C. (2004), 'The Analysis of Time Series: An Introduction', 6^{th} edition, Boca Raton, Fla.: London: Chapman & Hall/CRC

[7] Daigler, R. T. and Wiley, M. K. (1999), 'The Impact of Trader Type on the Futures Volatility-Volume Relation', *The Journal of Finance*, vol. 54, no. 6

[8] Daly, K. (2011), 'An Overview of the Determinants of Financial Volatility: An Explanation of Measuring Techniques', *Modern Applied Science*, vol. 5, no. 5

[9] Ding, Z. and Granger, C. W. J. (1996), 'Modelling volatility persistence of speculative returns: A new approach', *Journal of Econometrics*, vol. 73, pages 185-215

[10] Engle, R. F. (1982), ‘Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation’, *Econometrica*, vol. 50, no. 4, pages 987-1007

[11] Engle, R. F. and Sokalska, M. E. (2012) ‘Forecasting intraday volatility in the US equity market. Multiplicative component GARCH’, *Journal of Financial Econometrics*, vol. 10, pages 54-83

[12] Fuertes, A. M., Izzeldin, M. and Kalotychou, E. (2009), ‘On forecasting daily sock volatility: The role of intraday information and market conditions’, *International Journal of Forecasting*, vol. 25, pages 259-281

[13] Gallant, A. R. (1981), ‘On the bias in flexible functional form and an essentially unbiased form. The Fourier Flexible Form’, *Journal of Econometrics*, vol. 15, pages 211-245

[14] Gonzalez-Velasco, E. A. (1995), ‘Fourier Analysis and Boundary Value Problems’, San Diego: Academic Press

[15] Grammatikos, T. and Saunders, A. (1986), ‘Futures Price Variability: A Test of Maturity and Volume Effects’, *The Journal of Business*, vol. 59, no. 2, pages 319-330

[16] Hardy, M. A. (1993) ‘Regression with Dummy Variables’, Newbury Park, [Calif.]; London: SAGE

[17] Holmes, P. and Rougier, J. (2005), ‘Trading volume and contract rollover in futures contracts’, *Journal of Empirical Finance*, vol. 12, pages 317-338

[18] J.P. Morgan/ Reuters (1996), ‘RiskMetrics™ – Technical Document’

[19] Lockwood, L. J. and Linn, S. C. (1990), ‘An Examination of Stock Market Return Volatility During Overnight and Intraday Periods, 1964-1989’, *The Journal of Finance*, vol. 45, no. 2, pages 591-601

Appendices

Appendix A

MATLAB code for the cumulative volumes of different S&P 500 future contracts. The similar code is used for gold futures, too.

```
clear all;
volumeMatrix = (20070101:20121231);
volumeMatrix(9,size(volumeMatrix))=0;
row=2;
productCode = 'es';
frequency = 300;
type = 'trades';

for (maturityYear=2011:2012)
    for (month=['h','m','u','z'])

        maturityMonth = month;
        [price, volume, sec, day, offset] = concatSamples(productCode,
maturityMonth, maturityYear, frequency);


        % CALCULATING DAILY VOLUMES

        i=size(day,1);
        j=1;
        dayVolume(1,1)=volume(1);    % day volume
        dayVolume(1,2)=day(1);   % day
        dayVolume(1,3)=1;    % number of 5min intervals


        for (k=2:i)
            if (day(k)==day(k-1))
               dayVolume(j,1)=dayVolume(j,1)+volume(k);
               dayVolume(j,3)=dayVolume(j,3)+1;
            end

            if (day(k)~=day(k-1))
                j=j+1;
                dayVolume(j,1)=volume(k);
                dayVolume(j,2)=day(k);
                dayVolume(j,3)=1;

            end
        end
```

```
        % REMOVING DAYS WHEN MARKET WAS PARTIALLY CLOSED

         for (diffDays=size(dayVolume,1):(-1):1)
            if (dayVolume(diffDays,3)<200)
                dayVolume(diffDays,1)=0;
            end
        end

        for (number=(1:size(dayVolume,1)))
            currentDay=dayVolume(number,2);
            [R,C]= find(volumeMatrix==currentDay);
            volumeMatrix(row,C)=dayVolume(number,1);
        end

        row=row+1;

        dayVolume=[];
    end
end

for (remove=size(volumeMatrix,2):(-1):1)
    if (volumeMatrix(2:9,remove)==0)
        volumeMatrix(:,remove)=[];
    end
end

volumeMatrix(:,1:130)=[];

x=(1:size(volumeMatrix,2));
plot(x,volumeMatrix(2,:),'r-',x,volumeMatrix(3,:),'b-
',x,volumeMatrix(4,:),'g-',x,volumeMatrix(5,:),'k-',...
    x,volumeMatrix(6,:),'r-',x,volumeMatrix(7,:),'b-
',x,volumeMatrix(8,:),'g-',x,volumeMatrix(9,:),'k-')
legend('2011 Mar', '2011 June', '2011 Sep','2011 Dec','2012 Mar', '2012
June', '2012 Sep','2012 Dec','Location','eastoutside');
title('S&P 500 futures volumes','FontWeight','bold')
xlabel('Time (trading days)')
ylabel('Volume')
```

Appendix B

MATLAB code to create the return time series for the S&P 500 futures. The similar code is used for gold futures, too.

```
clear all;
returns=zeros(1,288);
returns2=zeros(1,288);
row=1;

productCode = 'es';
frequency = 300;
type = 'trades';
```

```
for (maturityYear=2011:2012)
    for (maturityMonth=['h','m','u','z'])

        % SETTING ROLL DATES BETWEEN FUTURE CONTRACTS
        if (maturityYear==2011)
            if (maturityMonth=='h')
                startDay=20101210;
                endDay=20110310;
            end
            if (maturityMonth=='m')
                startDay=20110311;
                endDay=20110609;
            end
            if (maturityMonth=='u')
                startDay=20110610;
                endDay=20110908;
            end
            if (maturityMonth=='z')
                startDay=20110909;
                endDay=20111208;
            end
        end
         if (maturityYear==2012)
            if (maturityMonth=='h')
                startDay=20111209;
                endDay=20120308;
            end
            if (maturityMonth=='m')
                startDay=20120309;
                endDay=20120607;
            end
            if (maturityMonth=='u')
                startDay=20120608;
                endDay=20120913;
            end
            if (maturityMonth=='z')
                startDay=20120914;
                endDay=20121213;
            end
        end

        [price, volume, sec, day, offset] = concatSamples(productCode,
maturityMonth, maturityYear, frequency);

        i=size(day,1);
        j=1;
        diffDays(1,1)=day(1);   % day
        diffDays(1,2)=1;    % number of 5min intervals

        % CHECKING HOW MANY 5MIN INTERVALS THERE ARE IN EACH DAY
        for (k=2:i)
            if (day(k)==day(k-1))
               diffDays(j,2)=diffDays(j,2)+1;
            end
            if (day(k)~=day(k-1))
                j=j+1;
                diffDays(j,1)=day(k);
                diffDays(j,2)=1;
            end
```

```
        end

        % REMOVING DAYS WHEN MARKET WAS PARTIALLY CLOSED

        for (count=size(diffDays,1):(-1):1)
            if (diffDays(count,2)<200)
                diffDays(count,:)=[];
            end
        end

        %FILLING THE MATRIX OF RETURNS (LINE REPRESENTS A DAY, COLLUMN
        %REPRESENTS A 5 MIN INTERVAL)

        firstDay=find(diffDays==startDay);
        lastDay=find(diffDays==endDay);

        for (index=firstDay:lastDay)
            returns(row,1)=0;
            returns2(row,1)=0;
            a=diffDays(index,1);
            validDay=find(day==a);  % Store the indices of the particular
day in a vector validDay

            for (column=1:288)
                returns(row,column)=log(price(validDay(column)))-
log(price(validDay(column)-1));
                returns2(row,column)=abs(log(price(validDay(column)))-
log(price(validDay(column)-1)));
            end

            row=row+1;
        end

    end

end

returnsForAnalysis1=returns(1:346,:);
returnsForAnalysis2=returns2(1:346,:);
returnsForEvaluation1=returns(347:519,:);
returnsForEvaluation2=returns2(347:519,:);

averageReturns=sum(returnsForAnalysis1)/size(returnsForAnalysis1,1);
averageReturns2=sum(returnsForAnalysis2)/size(returnsForAnalysis2,1);

timeSeriesDays1=size(returnsForAnalysis1,1);
timeSeriesDays2=size(returnsForEvaluation1,1);
i=1;

for (a=1:timeSeriesDays1)
    for (b=1:288)
        analysisTimeSeries(i)=returnsForAnalysis1(a,b);
        analysisTimeSeries2(i)=returnsForAnalysis2(a,b);
        i=i+1;
    end
end

i=1;
```

```
for (a=1:timeSeriesDays2)
    for (b=1:288)
        evaluationTimeSeries(i)=returnsForEvaluation1(a,b);
        evaluationTimeSeries2(i)=returnsForEvaluation2(a,b);
        i=i+1;
    end
end
```

Appendix C

MATLAB code for the estimation of the autocorrelations.

```
c=0;
c2=0;
cVector=zeros(288,1);
cVector2=zeros(288,1);

timeSeries=analysisTimeSeries;
timeSeries2=analysisTimeSeries2;

average=sum(timeSeries)/size(timeSeries,2);
average2=sum(timeSeries2)/size(timeSeries2,2);

for (i=1:size(timeSeries,2))
    c=c+(timeSeries(i)-average)^2;
    c2=c2+(timeSeries2(i)-average2)^2;
end

for (k=1:288)
    for (t=1:(size(timeSeries,2)-k))
        cVector(k,1)=cVector(k,1)+(timeSeries(t)-average)*(timeSeries(t+k)-
average);
        cVector2(k,1)=cVector2(k,1)+(timeSeries2(t)-
average2)*(timeSeries2(t+k)-average2);
    end
end

autoCorrelation=cVector./c;
autoCorrelation2=cVector2./c2;

figure(1);
subplot(211);
plot(averageReturns,'k')
title('S&P 500 futures average intraday returns','FontWeight','bold')
xlabel('5 min interval')
ylabel('Returns')
subplot(212);
plot(autoCorrelation,'k')
title('Correlogram of S&P 500 futures returns','FontWeight','bold')
xlabel('Time lag')
ylabel('Autcorrelation coefficient')
set(gca,'YTick',[-0.04 -0.02 0 0.02 0.04])
axis([-Inf 300 -0.04 0.04])

figure(2);
```

```
plot(averageReturns2,'k')
title('S&P 500 futures average intraday absolute
returns','FontWeight','bold')
xlabel('5 min interval')
ylabel('Absolute returns')

figure(3);
plot(autoCorrelation2,'k')
title('Correlogram of S&P 500 futures absolute
returns','FontWeight','bold')
xlabel('Time lag')
ylabel('Autcorrelation coefficient')
```

Appendix D

MATLAB code for the EWMA and GARCH(1,1) volatility estimations and forecasts

```
numberOfDays=size(returnsForAnalysis1,1);
numberOfDays2=size(returnsForEvaluation1,1);
dailyReturns=sum(returnsForAnalysis1,2);
forecastDailyReturns=sum(returnsForEvaluation1,2);
garchCoefficients = tarch(dailyReturns,1,0,1);  % GARCH(1,1) coefficients
alpha=0.06;

% Volatility estimates for modelling

garchEstimate=zeros(numberOfDays,1);
garchEstimate(1)=garchCoefficients(1)+garchCoefficients(2)*(dailyReturns(1)
^2);
ewmaEstimate=zeros(numberOfDays,1);
ewmaEstimate(1)=alpha*dailyReturns(1)^2;


for (i=2:numberOfDays)

garchEstimate(i)=garchCoefficients(1)+garchCoefficients(2)*(dailyReturns(i-
1)^2)+garchCoefficients(3)*garchEstimate(i-1);
end

for (i=2:numberOfDays)
    ewmaEstimate(i)=alpha*(dailyReturns(i-1)^2)+(1-alpha)*ewmaEstimate(i-
1);
end

garchVector=zeros(numberOfDays*288,1);
ewmaVector=zeros(numberOfDays*288,1);
position=1;
for (i=1:numberOfDays)
    for(j=1:288)
        garchVector(position)=garchEstimate(i);
        ewmaVector(position)=ewmaEstimate(i);
        position=position+1;
    end
```

```
end

xVectorGARCH=sqrt(288)*transpose(analysisTimeSeries2)./sqrt(garchVector);
xVectorEWMA=sqrt(288)*transpose(analysisTimeSeries2)./sqrt(ewmaVector);

xVectorGARCH(1:50*288)=[];
xVectorEWMA(1:50*288)=[];
garchVector(1:50*288)=[];
ewmaVector(1:50*288)=[];
analysisTimeSeries(1:50*288)=[];
analysisTimeSeries2(1:50*288)=[];

% Volatility estimates for forecasting (out of sample)

forecastGarchEstimate=zeros(numberOfDays2,1);
forecastGarchEstimate(1)=garchCoefficients(1)+garchCoefficients(2)*(dailyRe
turns(numberOfDays)^2)+garchCoefficients(3)*garchEstimate(numberOfDays);
forecastEwmaEstimate=zeros(numberOfDays2,1);
forecastEwmaEstimate(1)=alpha*(dailyReturns(numberOfDays)^2)+(1-
alpha)*ewmaEstimate(numberOfDays);

for (i=2:numberOfDays2)

forecastGarchEstimate(i)=garchCoefficients(1)+garchCoefficients(2)*(forecas
tDailyReturns(i-1)^2)+garchCoefficients(3)*forecastGarchEstimate(i-1);
end

for (i=2:numberOfDays2)
    forecastEwmaEstimate(i)=alpha*(forecastDailyReturns(i-1)^2)+(1-
alpha)*forecastEwmaEstimate(i-1);
end

forecastGarchVector=zeros(numberOfDays2*288,1);
forecastEwmaVector=zeros(numberOfDays2*288,1);

position=1;
for (i=1:numberOfDays2)
    for(j=1:288)
        forecastGarchVector(position)=forecastGarchEstimate(i);
        forecastEwmaVector(position)=forecastEwmaEstimate(i);
        position=position+1;
    end
end
```

Appendix E

MATLAB code for the estimation of the polynomial degree used in polynomial regression.

```
errors=zeros(2,50);

for (m=1:50)

    dim=size(xVectorGARCH);
    regressionMatrix=zeros(dim,m+1);
    regressionMatrix(:,1)=1;
```

```
    days=size(xVectorGARCH)/288-1;

    for(i=0:days)
        for(j=1:288)
            for (k=1:m)
                regressionMatrix(i*288+j,k+1)=(j/288)^k;
            end
        end
    end

    garchRegress=regress(xVectorGARCH,regressionMatrix);
    ewmaRegress=regress(xVectorEWMA,regressionMatrix);

    kof1=zeros(288,1);
    kof2=zeros(288,1);

    for(j=1:288)
        for(k=0:m)
            kof1(j)=kof1(j)+garchRegress(k+1)*(j/288)^k;
            kof2(j)=kof2(j)+ewmaRegress(k+1)*(j/288)^k;
        end
    end

    seriesSize=size(xVectorGARCH,1);
    vector1=garchVector;
    vector2=ewmaVector;
    realReturns=analysisTimeSeries2;

    Forecasting;

    errors(1,m)=MAE1;
    errors(2,m)=MAE2;
end

x=(1:50);
plot(x,errors(1,:),'b',x,errors(2,:),'r')
xlabel('Degree of the polynomial')
ylabel('MAE')
legend('GARCH volatility estimation','EWMA volatility
estimation','Location','Northeast')
title('S&P 500 futures','FontWeight','bold')
```

Appendix F

MATLAB code for the estimation of the m value used in Fourier regression.

```
errors=zeros(2,50);

for (m=1:50)

    dim=size(xVectorGARCH);
    regressionMatrix=zeros(dim,2*m+1);
    regressionMatrix(:,1)=1;
    days=size(xVectorGARCH)/288-1;
```

```
    for(i=0:days)
        for(j=1:288)
            for (k=1:m)
                regressionMatrix(i*288+j,k*2)=cos(2*k*pi*j/288);
                regressionMatrix(i*288+j,k*2+1)=sin(2*k*pi*j/288);
            end
        end
    end

    garchRegress=regress(xVectorGARCH,regressionMatrix);
    ewmaRegress=regress(xVectorEWMA,regressionMatrix);

    kof1=zeros(288,1);
    kof2=zeros(288,1);

    for(j=1:288)

        kof1(j)=garchRegress(1);
        kof2(j)=ewmaRegress(1);

        for(k=1:m)

kof1(j)=kof1(j)+garchRegress(2*k)*cos(2*k*pi*j/288)+garchRegress(2*k+1)*sin
(2*k*pi*j/288);

kof2(j)=kof2(j)+ewmaRegress(2*k)*cos(2*k*pi*j/288)+ewmaRegress(2*k+1)*sin(2
*k*pi*j/288);
        end

    end

    seriesSize=size(xVectorGARCH,1);
    vector1=garchVector;
    vector2=ewmaVector;
    realReturns=analysisTimeSeries2;

    Forecasting;

    errors(1,m)=MAE1;
    errors(2,m)=MAE2;
end

x=(1:50);
plot(x,errors(1,:),'b',x,errors(2,:),'r')
xlabel('m value')
ylabel('MAE')
legend('GARCH volatility estimation','EWMA volatility
estimation','Location','Northeast')
title('S&P 500 futures','FontWeight','bold')
```

Appendix G

MATLAB code for the polynomial regression.

```
dim=size(xVectorGARCH);
regressionMatrix=zeros(dim,15);
regressionMatrix(:,1)=1;
days=size(xVectorGARCH)/288-1;

for(i=0:days)
    for(j=1:288)
        for (k=1:14)
            regressionMatrix(i*288+j,k+1)=(j/288)^k;
        end
    end
end

garchRegress=regress(xVectorGARCH,regressionMatrix);
ewmaRegress=regress(xVectorEWMA,regressionMatrix);

kof1=zeros(288,1);
kof2=zeros(288,1);

for(j=1:288)
    for(k=0:14)
        kof1(j)=kof1(j)+garchRegress(k+1)*(j/288)^k;
        kof2(j)=kof2(j)+ewmaRegress(k+1)*(j/288)^k;
    end
end

seriesSize=size(evaluationTimeSeries2,2);
vector1=forecastGarchVector;
vector2=forecastEwmaVector;
realReturns=evaluationTimeSeries2;

Forecasting;
```

Appendix H

MATLAB code for the Fourier regression.

```
dim=size(xVectorGARCH);
regressionMatrix=zeros(dim,13);
regressionMatrix(:,1)=1;
days=size(xVectorGARCH)/288-1;

for(i=0:days)
    for(j=1:288)
        for (k=1:6)
            regressionMatrix(i*288+j,k*2)=cos(2*k*pi*j/288);
            regressionMatrix(i*288+j,k*2+1)=sin(2*k*pi*j/288);
        end
    end
end

garchRegress=regress(xVectorGARCH,regressionMatrix);
ewmaRegress=regress(xVectorEWMA,regressionMatrix);

kof1=zeros(288,1);
```

```
kof2=zeros(288,1);

for(j=1:288)

    kof1(j)=garchRegress(1);
    kof2(j)=ewmaRegress(1);

    for(k=1:6)

kof1(j)=kof1(j)+garchRegress(2*k)*cos(2*k*pi*j/288)+garchRegress(2*k+1)*sin
(2*k*pi*j/288);

kof2(j)=kof2(j)+ewmaRegress(2*k)*cos(2*k*pi*j/288)+ewmaRegress(2*k+1)*sin(2
*k*pi*j/288);
    end

end


seriesSize=size(evaluationTimeSeries2,2);
vector1=forecastGarchVector;
vector2=forecastEwmaVector;
realReturns=evaluationTimeSeries2;

Forecasting;
```

Appendix I

MATLAB code for the FFF regression.

```
dim=size(xVectorGARCH);
regressionMatrix=zeros(dim,15);
days=size(xVectorGARCH)/288-1;

for(i=0:days)
    for(j=1:288)
        for (k=1:3)
            regressionMatrix(i*288+j,k)=(j/288)^(k-1);
        end
        for (k=1:6)
            regressionMatrix(i*288+j,k*2+2)=cos(2*k*pi*j/288);
            regressionMatrix(i*288+j,k*2+3)=sin(2*k*pi*j/288);
        end

    end
end

garchRegress=regress(xVectorGARCH,regressionMatrix);
ewmaRegress=regress(xVectorEWMA,regressionMatrix);

kof1=zeros(288,1);
kof2=zeros(288,1);

for(j=1:288)
```

```
    for(k=0:2)
        kof1(j)=kof1(j)+garchRegress(k+1)*(j/288)^k;
        kof2(j)=kof2(j)+ewmaRegress(k+1)*(j/288)^k;
    end
    for(k=1:6)

kof1(j)=kof1(j)+garchRegress(2*k+2)*cos(2*k*pi*j/288)+garchRegress(2*k+3)*s
in(2*k*pi*j/288);

kof2(j)=kof2(j)+ewmaRegress(2*k+2)*cos(2*k*pi*j/288)+ewmaRegress(2*k+3)*sin
(2*k*pi*j/288);
    end
end

seriesSize=size(evaluationTimeSeries2,2);
vector1=forecastGarchVector;
vector2=forecastEwmaVector;
realReturns=evaluationTimeSeries2;

Forecasting;
```

Appendix J

MATLAB code for the Dummy variables regression.

```
dim=size(xVectorGARCH);
regressionMatrix=zeros(dim,288);

days=size(xVectorGARCH)/288-1;

for(i=0:days)
    for(j=1:288)
        regressionMatrix(i*288+j,j)=1;
    end
end

garchRegress=regress(xVectorGARCH,regressionMatrix);
ewmaRegress=regress(xVectorEWMA,regressionMatrix);

kof1=garchRegress;
kof2=ewmaRegress;

seriesSize=size(evaluationTimeSeries2,2);
vector1=forecastGarchVector;
vector2=forecastEwmaVector;
realReturns=evaluationTimeSeries2;

Forecasting;
```

Appendix K

MATLAB code for the single step forecasting.

```
forecastedGarchReturns=zeros(seriesSize,1);
forecastedEwmaReturns=zeros(seriesSize,1);

garchPeriodicVector=zeros(seriesSize,1);
ewmaPeriodicVector=zeros(seriesSize,1);

position=1;

for (i=1:(seriesSize/288))
    for (j=1:288)
        garchPeriodicVector(position)=kof1(j);
        ewmaPeriodicVector(position)=kof2(j);
        position=position+1;
    end
end

forecastedGarchReturns=sqrt(vector1).*garchPeriodicVector/sqrt(288);
forecastedEwmaReturns=sqrt(vector2).*ewmaPeriodicVector/sqrt(288);

MAE1=abs(forecastedGarchReturns-transpose(realReturns));
MAE2=abs(forecastedEwmaReturns-transpose(realReturns));

MAE1=mean(MAE1);
MAE2=mean(MAE2);

averageForecastedReturns1=zeros(288,1);
averageForecastedReturns2=zeros(288,1);


position=1;
for (j=1:seriesSize/288)
    for (i=1:288)

averageForecastedReturns1(i)=averageForecastedReturns1(i)+forecastedGarchRe
turns(position);

averageForecastedReturns2(i)=averageForecastedReturns2(i)+forecastedEwmaRet
urns(position);
        position=position+1;
    end
end

avg1=averageForecastedReturns1/(seriesSize/288);
avg2=averageForecastedReturns2/(seriesSize/288);
averageEvaluationReturns=mean(returnsForEvaluation2,1);
```

Appendix L

MATLAB code for the multistep forecasting.

```
phi=armaxfilter(dailyReturns,0,1);

garchForecastedReturns1=zeros(numberOfDays2,1);
garchForecastedReturns2=zeros(numberOfDays2,1);
garchForecastedReturns3=zeros(numberOfDays2,1);
garchForecastedReturns4=zeros(numberOfDays2,1);
ewmaForecastedReturns1=zeros(numberOfDays2,1);
ewmaForecastedReturns2=zeros(numberOfDays2,1);
ewmaForecastedReturns3=zeros(numberOfDays2,1);
ewmaForecastedReturns4=zeros(numberOfDays2,1);

for (i=2:4)
    garchForecastedReturns1(i)=phi*forecastDailyReturns(i-
1)+normrnd(0,sqrt(forecastGarchEstimate(i-1)));

garchForecastedReturns2(i+1)=phi*garchForecastedReturns1(i)+normrnd(0,sqrt(
forecastGarchEstimate(i)));

garchForecastedReturns3(i+2)=phi*garchForecastedReturns2(i+1)+normrnd(0,sqr
t(forecastGarchEstimate(i+1)));

    ewmaForecastedReturns1(i)=phi*forecastDailyReturns(i-
1)+normrnd(0,sqrt(forecastEwmaEstimate(i-1)));

ewmaForecastedReturns2(i+1)=phi*ewmaForecastedReturns1(i)+normrnd(0,sqrt(fo
recastEwmaEstimate(i)));

ewmaForecastedReturns3(i+2)=phi*ewmaForecastedReturns2(i+1)+normrnd(0,sqrt(
forecastEwmaEstimate(i+1)));
end

for (i=5:numberOfDays2)
    garchForecastedReturns1(i)=phi*forecastDailyReturns(i-
1)+normrnd(0,sqrt(forecastGarchEstimate(i-1)));
    garchForecastedReturns2(i)=phi*garchForecastedReturns1(i-
1)+normrnd(0,sqrt(forecastGarchEstimate(i-1)));
    garchForecastedReturns3(i)=phi*garchForecastedReturns2(i-
1)+normrnd(0,sqrt(forecastGarchEstimate(i-1)));
    garchForecastedReturns4(i)=phi*garchForecastedReturns3(i-
1)+normrnd(0,sqrt(forecastGarchEstimate(i-1)));

    ewmaForecastedReturns1(i)=phi*forecastDailyReturns(i-
1)+normrnd(0,sqrt(forecastEwmaEstimate(i-1)));
    ewmaForecastedReturns2(i)=phi*ewmaForecastedReturns1(i-
1)+normrnd(0,sqrt(forecastEwmaEstimate(i-1)));
    ewmaForecastedReturns3(i)=phi*ewmaForecastedReturns2(i-
1)+normrnd(0,sqrt(forecastEwmaEstimate(i-1)));
    ewmaForecastedReturns4(i)=phi*ewmaForecastedReturns3(i-
1)+normrnd(0,sqrt(forecastEwmaEstimate(i-1)));
end

length=numberOfDays2-3;
multistepGarch=zeros(4,numberOfDays2);
multistepEwma=zeros(4,numberOfDays2);

for (i=3:length)

multistepGarch(1,i)=garchCoefficients(1)+garchCoefficients(2)*(garchForecas
tedReturns1(i-1)^2)+garchCoefficients(3)*forecastGarchEstimate(i-1);
```

```
multistepGarch(2,i+1)=garchCoefficients(1)+garchCoefficients(2)*(garchForec
astedReturns2(i)^2)+garchCoefficients(3)*multistepGarch(1,i);

multistepGarch(3,i+2)=garchCoefficients(1)+garchCoefficients(2)*(garchForec
astedReturns3(i+1)^2)+garchCoefficients(3)*multistepGarch(2,i+1);

multistepGarch(4,i+3)=garchCoefficients(1)+garchCoefficients(2)*(garchForec
astedReturns4(i+2)^2)+garchCoefficients(3)*multistepGarch(3,i+2);

    multistepEwma(1,i)=alpha*(ewmaForecastedReturns1(i-1)^2)+(1-
alpha)*forecastEwmaEstimate(i-1);
    multistepEwma(2,i+1)=alpha*(ewmaForecastedReturns2(i)^2)+(1-
alpha)*multistepEwma(1,i);
    multistepEwma(3,i+2)=alpha*(ewmaForecastedReturns3(i+1)^2)+(1-
alpha)*multistepEwma(2,i+1);
    multistepEwma(4,i+3)=alpha*(ewmaForecastedReturns4(i+2)^2)+(1-
alpha)*multistepEwma(3,i+2);
end

multistepGarchMatrix=zeros(4,numberOfDays2*288);
multistepEwmaMatrix=zeros(4,numberOfDays2*288);
position=1;
for (i=1:numberOfDays2)
    for(j=1:288)
        multistepGarchMatrix(:,position)=multistepGarch(:,i);
        multistepEwmaMatrix(:,position)=multistepEwma(:,i);
        position=position+1;
    end
end

garchPeriodicVector=zeros(numberOfDays2*288,1);
ewmaPeriodicVector=zeros(numberOfDays2*288,1);
position=1;
for (i=1:numberOfDays2)
    for (j=1:288)
        garchPeriodicVector(position)=kof1(j);
        ewmaPeriodicVector(position)=kof2(j);
        position=position+1;
    end
end

MAEs1=zeros(1,5);
MAEs2=zeros(1,5);

forecastedGarchReturns=sqrt(vector1).*garchPeriodicVector/sqrt(288);
forecastedEwmaReturns=sqrt(vector2).*ewmaPeriodicVector/sqrt(288);
MAE1=abs(forecastedGarchReturns-transpose(realReturns));
MAE2=abs(forecastedEwmaReturns-transpose(realReturns));
MAEs1(1)=mean(MAE1);
MAEs2(1)=mean(MAE2);

multistepGarchMatrix(:,1:(5*288))=[];
multistepEwmaMatrix(:,1:(5*288))=[];
garchPeriodicVector(1:(5*288))=[];
ewmaPeriodicVector(1:(5*288))=[];
realReturns(1:(5*288))=[];
length=size(multistepGarchMatrix,2);
multistepGarchMatrix(:,(length-3*288+1):length)=[];
multistepEwmaMatrix(:,(length-3*288+1):length)=[];
```

```
garchPeriodicVector((length-3*288+1):length)=[];
ewmaPeriodicVector((length-3*288+1):length)=[];
realReturns((length-3*288+1):length)=[];
multistepForecastedGarchReturns=zeros(4,size(garchPeriodicVector));
multistepForecastedEwmaReturns=zeros(4,size(garchPeriodicVector));

for (i=1:4)

multistepForecastedGarchReturns(i,:)=sqrt(multistepGarchMatrix(i,:)).*trans
pose(garchPeriodicVector)/sqrt(288);

multistepForecastedEwmaReturns(i,:)=sqrt(multistepEwmaMatrix(i,:)).*transpo
se(ewmaPeriodicVector)/sqrt(288);
    MAE1=abs(multistepForecastedGarchReturns(i,:)-realReturns);
    MAE2=abs(multistepForecastedEwmaReturns(i,:)-realReturns);
    MAEs1(i+1)=mean(MAE1);
    MAEs2(i+1)=mean(MAE2);
end
```