Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
Department of Economics

Gard Djupskås and Asbjørn Olsen

Employing Machine Learning and Econometric Models to Forecast Implied Volatility for EUR/USD FX Options

A comprehensive test of machine learning models ability to forecast the implied volatility of FX options

Master's thesis in Financial Economics
Supervisor: Petter Eilif De Lange
Co-supervisor: Morten Risstad
June 2022

NTNU
Kunnskap for en bedre verden

Gard Djupskås and Asbjørn Olsen

Employing Machine Learning and Econometric Models to Forecast Implied Volatility for EUR/USD FX Options

A comprehensive test of machine learning models ability to forecast the implied volatility of FX options

Master's thesis in Financial Economics
Supervisor: Petter Eilif De Lange
Co-supervisor: Morten Risstad
June 2022

Norwegian University of Science and Technology
Faculty of Economics and Management
Department of Economics

NTNU

Kunnskap for en bedre verden

Acknowledgements

This thesis concludes our master's degree in Financial Economics at the Norwegian University of Science and Technology. We would like to thank our supervisor Petter Eilif De Lange for guidance and valuable feedback throughout the semester. We would also like to thank Morten Risstad from Sparebank 1 Markets for consultation and valuable insights to a practitioner's application, including providing us with the data necessary to perform our analysis.

The authors alone are fully responsible for all content and any errors that may follow this thesis.

Abstract

In this paper we propose an empirical study of the forecasting performance of LSTM, Random Forest and AR-GARCH models on daily spot rates of implied volatility for EUR/USD exchange rate options. We apply a univariate time series of implied volatility as explanatory variables to forecast out-of-sample predictions for implied volatility, and compare the forecast performance across models based on the statistical error measurements mean squared error, root mean squared error and mean absolute error. Additionally, we conduct a Diebold-Mariano test to question the statistical differences between the models. We impose Random Forest and a Gaussian distributed AR(1)-GARCH(1,1) as benchmark models and compare their forecasting performance to the more advanced LSTM model. In addition to the benchmark AR(1)-GARCH(1,1) model, we extend the analysis with models that include an asymmetric GARCH term, moving average terms, along with Gaussian distributed residuals and Student t-distributed residuals. Our findings conclude that the LSTM model is better than the benchmark models for shorter option maturities, whilst the AR-GARCH model is superior when the maturities increase. However, when imposing other specifications and residual distribution for the GARCH models, we find that the AR-GARCH framework outperforms the more advanced machine learning models for all options. For shorter maturities the t-distributed models perform best, while ARIMA-GARCH-type models perform better for longer maturities. Implied volatility of FX options, and hereby this paper, are of interest to all market participants that are exposed to foreign exchange risk, for hedging and trading purposes.

Sammendrag

I denne artikkelen gjennomfører vi en empirisk studie av prognosenøyaktigheten til LSTM-, Random Forest- og AR-GARCH-modeller for implisitt volatilitet på daglige spotkurser for EUR/USD-valutaopsjoner. Vi bruker en univariate tidsserier med implisitt volatilitet som variabler og sammenligner prognosenøyaktigheten på tvers av modellene med de statistiske målemetodene kvadratisk gjennomsnittsfeil, rot av kvadratisk gjennomsnittsfeil og absolutt gjennomsnittsfeil. I tillegg, gjennomfører vi en Diebold-Mariano-test for å undersøke om forskjellene mellom modellene er statistisk signifikant. Vi benytter Random Forest og AR(1)-GARCH(1,1) med normalfordelte restledd som referansemodeller for den mer avanserte LSTM modellen. Utover AR(1)-GARCH(1,1) utvider vi analysen med økonometriske modeller som inkluderer asymmetriske variabler, MA-variabler, sammen med normalfordelte og t-fordelte restledd. Våre funn konkluderer med at LSTM modellen er bedre enn referansemodellene for opsjoner med korte løpetider, og AR-GARCH er bedre for lengre løpetider. Videre finner vi at AR-GARCH-rammeverket er bedre enn de mer avanserte maskinlæringsmodellene når vi utvider modellene med andre spesifikasjoner og restledd distribusjoner. For kortere løpetider er t-fordelte restledd best, mens ARIMA-GARCH modeller er bedre for lengre løpetider. Valutaopsjoners implisitte volatilitet, og herved denne artikkelen, er av interesse for alle markedsaktører som er eksponert for valutarisiko, for sikring og spekulative formål.

Contents

1.Introduction

Forecasting the implied volatility of foreign exchange (FX) options and financial time series in general, is a challenging task mainly due to incomplete information and unprecedented changes in economic trends and conditions. However, as periods of transitions from a low to high market volatility regime can be abrupt and short-lived, the development of effective modeling framework is of critical importance for the design and implementation of active portfolio immunization strategies in order to avoid sizeable drawdowns during periods of turmoil in particular (Galakis & Vrontos, 2021). As implied volatility measures the market expectations of future risk, an effective forecasting framework can identify high volatility regimes, benefiting market players exposed to currency fluctuations.
Traditional econometric time series models struggle to capture non-linearity in data, incentivizing economic researchers to adapt toward more advanced models. Machine learning methods can alleviate the complexity in time series forecasting by identifying structures and patterns of data such as non-linearity and dependency between predictors. Particularly, LSTM (Long Short-Term Memory) has received increased focus in forecasting financial time series, however, with mixed results.

Given the importance of developing an effective modeling framework to minimize FX risk, the subject of this study is:
Comparing the predictive power of LSTM models to Random Forest and AR-GARCH-type models for forecasting implied volatility for options on the EUR/USD foreign exchange rate.
We structure the analysis into two parts:

1. Statistical distribution: Analyzing the univariate time series characteristics of the implied volatility components, including dependency structure.
2. Forecasting models: Evaluating and proposing forecasting models for implied volatility.

We optimize an LSTM model on a training set of the data and compare its forecasting predictability to a RF (Random Forest) model and AR-GARCH-type models. The estimator is the daily spot rates for the implied volatility for the EUR/USD FX options. We impose RF and a Gaussian distributed AR(1)-GARCH(1,1) as benchmark models and compare their forecasting performance to the more advanced LSTM model. In addition to the benchmark AR(1)-GARCH(1,1) model, we extend the analysis with models that include an asymmetric

GARCH term, moving average terms, along with Gaussian distributed residuals and Student t-distributed residuals. The machine learning models are optimized using different hyperparameters, and we compare the best-fitted structures for each option to the benchmark models.

Our findings conclude that the LSTM model is better than the benchmark models for shorter option maturities, whilst the AR-GARCH model is superior when the maturities increase. However, when imposing other specifications and residual distribution for the GARCH models, we find that the AR-GARCH framework outperforms the more advanced machine learning models for all options. For shorter maturities the t-distributed models perform best, while ARIMA-GARCH-type models perform better for longer maturities.
Further, this thesis is organized as follows. Section 2 discusses previous publications on implied volatility and the models we apply in our analysis. Section 3 presents the data, including statistical and distribution behavior. Section 4 presents the theory behind our work and the models we use, and further describes the methodology and model architecture. In Section 5, we present results and findings. Section 6 summarizes our findings and concludes.

2. Literature Review

Garman and Kohlhagen (1983) derive an implied volatility modification to the Black-Scholes formula for option pricing, introduced by Fischer Black and Myron Scholes in 1973. According to Stan (1981), Latane and Rendleman (1976), several studies have shown that implied volatility is a better forecaster of future price variability than measurements based on history (Bharadia et al., 1996). In recent years implied volatility has become a common estimator for forecasting purposes. Ornelas and Mauad (2019) find that the slopes of currency implied volatility term structures have predictive power for the behavior of exchange rates from both cross-sectional and time series perspectives. Carr et al. (2020) build a volatility index by formulating a variance prediction model using machine learning methods such as Feedforward Neural Networks and Random Forest on the S&P 500 index options. According to Haug et al. (2010), the standard derivation of implied volatility has an evident variation over time and declines as time to maturity increases. Therefore, it is vital to be aware of the challenges that follow from the time-varying properties of implied volatility. Time-varying properties entail another challenge; volatility clustering. That is, small (big) changes in the volatility tend to be followed by small (big) changes in the volatility (Mandelbrot, 1963).

A profound approach to account for this was introduced in 1982 when Robert Engle introduced a non-linear model allowing the time-varying conditional variance to depend on the lagged values of the squared errors, the autoregressive conditional heteroskedasticity (ARCH) model. Regardless of the innovations of the ARCH model, it has a few weaknesses. It is unclear how many lags to include in the variance equation. A high number of lags results in fewer degrees of freedom, and too many lags may cause the model to produce negative estimates for the variance. An extension to the ARCH model that allows the conditional variance to depend on lags of the conditional variance is the general ARCH model, or the GARCH model, introduced by Bollerslev (1986). The GARCH model is more parsimonious than the ARCH model. The GARCH model avoids overfitting and is still today a much-applied modeling framework for financial time series data.

In 1993, Glosten et al. formulated an extension to the GARCH model that accounted for an asymmetric response to a volatility shock, i.e., “good” news and “bad” news had different impacts on the subsequent period volatility, known as the GJR-GARCH. Lim and Sek (2013) found that in “normal times”, that is in post- and pre-crisis times, the symmetric GARCH

performs well, and in times of big volatility fluctuations, i.e. times of crisis, the asymmetric model is preferred. Schmidt (2021) argues that the asymmetric models are better forecasters for financial indexes in the aftermath of the shock caused by the outbreak of the Covid-19 pandemic compared to the symmetric specifications. Poon and Granger (2001) argue that the simpler GARCH models seem to provide more extensive volatility forecasts when compared to the more sophisticated models. In contrast, the GJR-GARCH seems to forecast lower values due to its asymmetry for the financial markets, which helps this model to quickly revert from different volatility states. Ramasamy and Minusamy (2012) found that the asymmetric GJR does not improve the forecasting performance considerably compared to symmetric GARCH models. According to Javed and Mantalos (2013) the performance of information criteria for the GARCH(1,1) is satisfactory, compared to higher order GARCH specifications.

Employing machine learning models for time series predictions is a relatively new topic. To the best of our knowledge, the literature dedicated to implementing machine learning techniques for forecasting the implied volatility of FX options is scarce. However, some research exists regarding machine learning for predicting stock prices, returns and volatility. We find research dating back to 1993, when Galler and Kruzanowski (1993) implemented deep learning to classify whether stock returns are positive or negative one-year-ahead. Further, Krauss et al. (2017) used various machine learning models, such as deep learning and tree-based models, to model S&P 500 constituents. Surprisingly, Krauss et al. (2017) reported that gradient-boosted trees and Random Forest outperformed deep learning models. More interestingly, Krauss et al. (2017) revealed that deep learning models performed exceptionally well in times of market turmoil. Yu and LI's (2018) findings are consistent with the claim that deep learning networks perform well during market turmoil. Yu and Li (2018) forecasted the volatility of the Shanghai compos stock price index using LSTM and GARCH, where they only selected extreme values (highs and lows) and concluded that the LSTM model was superior. A paper somewhat similar to ours is Namin and Namini (2018). Namin and Namini (2018) compares an Arima model and a univariate multistep LSTM model imposed by Brownlee (2016) on different stock indexes. They conclude that the LSTM model outperforms the ARIMA model. Galakis and Vrontos (2021) published an interesting paper regarding implementation of machine learning techniques for implied volatility. They study whether the application of machine learning approaches can outperform traditional econometric models in forecasting implied volatility indices. They concluded that certain

machine learning techniques are strongly encouraged as they significantly improve the accuracy of the out-of-sample forecasts. However, they also report that the model accuracy is not consistent across all models.

3. Data - Distribution and Statistical Behavior

In this Section, we discuss the statistical properties and distribution of the data. The dataset, provided by Morten Risstad from Sparebank 1 Markets, consists of daily observations of implied volatilities for eleven options with distinct levels of moneyness and five different times to maturity over the period from 02.01.2007 to 31.08.2021. This provides 55 distinct time series of implied volatility consisting of 164.670 observations, enabling us to analyze the forecast performance for different maturities and the distinct moneyness levels. Table 3.1 summarizes descriptive statistics for ATM (at-the-money) put options for the five distinct maturities. The variance of the volatilities declines as the time to maturity increases. The shorter maturities have both higher peaks and lower troughs of implied volatility. In comparison, the longer maturities have higher average levels of implied volatility, measured in both mean and median (50% quantile).

Table 3.1 Descriptive statistics for ATM put options for each maturity.

	1 week	*1 month*	*3 months*	*6 months*	*1 year*
Obs	2994	2994	2994	2994	2994
Mean	9,14	9,16	9,29	9,43	9,62
Min	2,74	3,77	4,14	4,42	4,97
25 %	6,43	6,64	6,75	6,96	7,23
50 %	8,32	8,38	8,51	8,63	8,95
75 %	10,76	10,96	11,24	11,44	11,91
Max	33,58	28,88	24,65	22,29	19,91
Var	14,70	12,58	11,26	10,28	9,32
σ	3,83	3,55	3,36	3,21	3,05

Table 3.1 Descriptive statistics for ATM put options for each maturity.

The options level of moneyness is measured by the option delta and can be interpreted as the probability that the option will finish in-the-money at expiration. For the most volatile option, the one-week to maturity option, a summary of descriptive statistics is presented in Table 3.2. Likewise, for the least volatile option, the one-year to maturity, descriptive statistics are

exhibited in table 3.3. Descriptive statistics for the remaining options with a maturity of one month, three months and six months can be found in Appendix B. When the delta is equal to 50, the option is ATM, and as the delta value decreases, the option becomes increasingly OTM (out-of-the-money). For the most OTM option, the delta value is 5. In this study we use put and call options with OTM delta values of 5, 10, 18, 25, 35 and ATM put options with a delta equal 50. The level of implied volatility, measured in mean and different quantiles, is higher for options OTM than ATM or close to ATM, and it is higher for puts than for calls. This is also the case for the volatility (i.e. daily changes in the level of implied volatility). This distribution pattern is the same for all five distinct maturities and is referred to as the volatility smile and is visualized in Figure 3.1. The implied volatility is higher for OTM put options than similar call options, consistent with a negative risk reversal that measures the volatility smile's skewness. The most common is to measure the risk reversal for call and put options with a delta of 25 (McDonald, 2014). When looking at the 25-delta risk reversal, on average for the whole data sample, all risk reversals are negative and increasingly negative as the time to maturity increases. The risk reversal also becomes more and more negative as the options become increasingly OTM (see Appendix B Figure B.2) and can be interpreted as a market-based measure of implied skewness.

Table 3.2. Descriptive statistics of implied volatility for options with one week to maturity

	Put 5	*Put 10*	*Put 18*	*Put 25*	*Put 35*	*Put 50*	*Call 35*	*Call 25*	*Call 18*	*Call 10*	*Call 5*
Obs	2994	2994	2994	2994	2994	2994	2994	2994	2994	2994	2994
Mean	10,39	10,02	9,70	9,50	9,28	9,14	9,11	9,19	9,29	9,48	9,72
Min	3,02	2,92	2,83	2,79	2,75	2,74	2,79	2,87	2,94	3,06	3,19
25 %	7,17	6,94	6,74	6,64	6,51	6,43	6,45	6,57	6,65	6,82	6,97
50 %	9,31	9,00	8,72	8,56	8,41	8,32	8,32	8,39	8,48	8,62	8,85
75 %	12,35	11,93	11,49	11,24	10,99	10,76	10,70	10,76	10,83	11,01	11,26
Max	38,94	37,09	36,01	35,10	34,27	33,58	33,20	33,27	33,54	33,88	35,02
Var	21,14	18,84	17,29	16,26	15,35	14,70	14,32	14,40	14,78	15,33	16,77
σ	4,60	4,34	4,16	4,03	3,92	3,83	3,78	3,80	3,84	3,92	4,10

Table 3.2 descriptive statistics of implied volatility for options with one week to maturity. Put with delta 50 is ATM and put and call options become increasingly OTM as the delta value decreases. Put 5 indicates a put option with an option delta of 5. Different quantiles measure the level of the implied volatility throughout the data sample.

Table 3.3 Descriptive statistics for options with one year to maturity

	Put 5	*Put 10*	*Put 18*	*Put 25*	*Put 35*	*Put 50*	*Call 35*	*Call 25*	*Call 18*	*Call 10*	*Call 5*
Obs	2994	2994	2994	2994	2994	2994	2994	2994	2994	2994	2994
Mean	12,88	12,05	11,11	10,58	10,05	9,62	9,46	9,50	9,66	10,05	10,41
Min	5,39	5,19	5,01	4,93	4,90	4,97	5,15	5,37	5,61	6,02	6,41
25 %	9,10	8,48	7,89	7,58	7,35	7,23	7,23	7,30	7,45	7,76	8,04
50 %	12,21	11,45	10,58	10,06	9,49	8,95	8,66	8,62	8,67	9,01	9,33
75 %	15,96	14,93	13,81	13,12	12,48	11,91	11,53	11,38	11,39	11,67	12,02
Ma	25,28	23,65	21,69	20,88	20,27	19,91	20,21	20,91	21,79	23,67	25,25
Var	20,90	17,75	14,21	12,40	10,68	9,32	8,70	8,66	9,01	10,18	11,28
σ	4,57	4,21	3,77	3,52	3,27	3,05	2,95	2,94	3,00	3,19	3,36

Table 3.3 descriptive statistics of implied volatility for options with one year to maturity. Put with delta 50 is ATM and put and call options become increasingly OTM as the delta value decreases. Put 5 indicates a put option with an option delta of 5. Different quantiles measure the level of the implied volatility throughout the data sample.

Figure 3.1 Volatility smiles for the different maturities

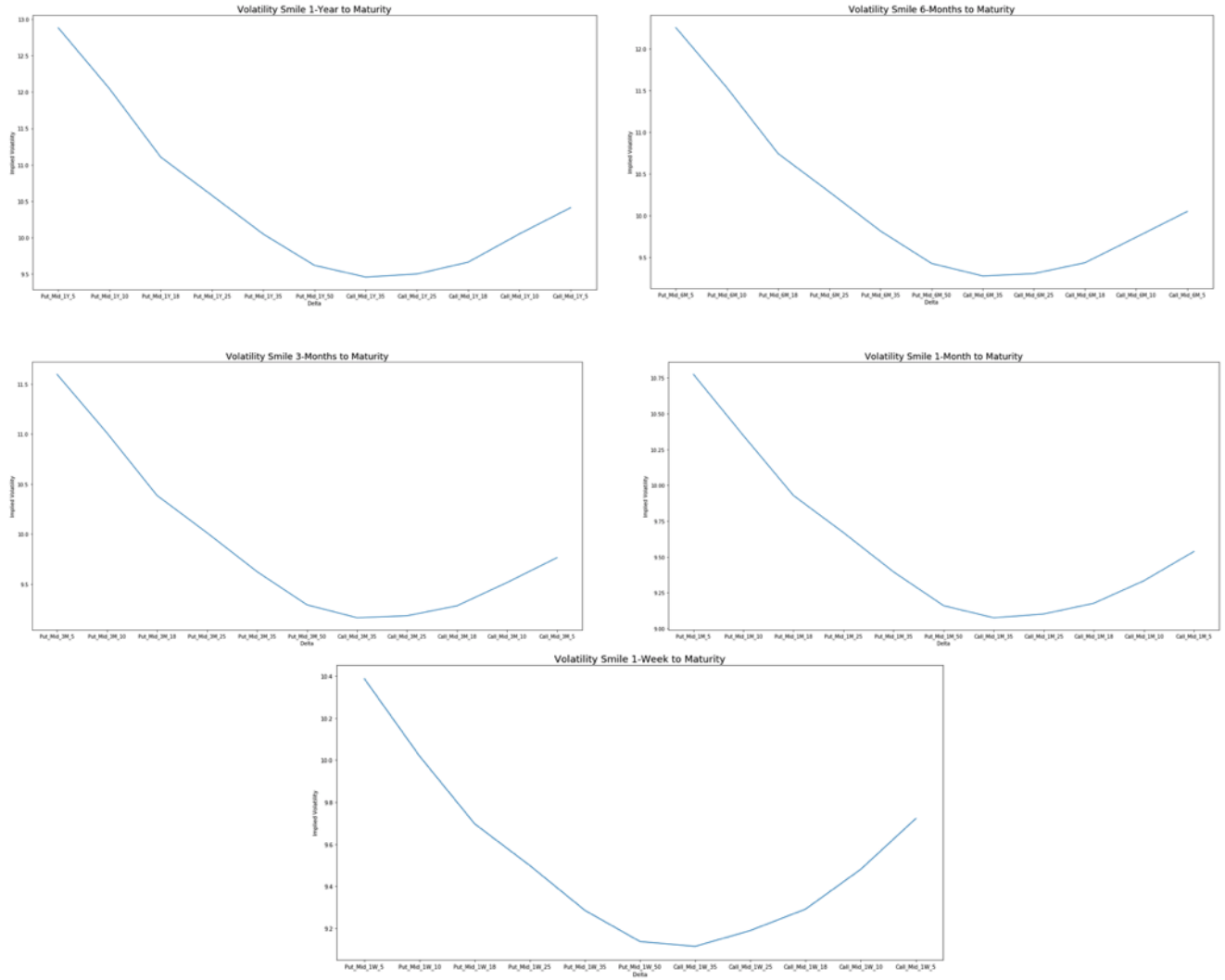


Figure 3.1 Volatility smiles for each distinct maturity, calculated as an average across the data sample. Top left is for one year to maturity, top right six months, mid left is three months, mid right is one month and bottom one week to maturity. Level of implied volatility along the vertical axis, and level of moneyness along the horizontal axis.

When comparing the historical values of the implied volatility for one week and one year options (see Figure 3.2), the variation in implied volatility is vast for the two. Since the one week option has a shorter time to maturity, the implied volatility reacts more to news and small shocks and is more volatile (i.e., more extensive daily changes). The cost of short-term options is smaller than for long-term options. According to financial theory, FX option traders have limited capital, which results in higher demand for short-term options. This causes the longer maturity options to trend more, meaning it recovers slower from massive shocks than the shorter maturities, implying that the time series for the longer maturities is non-stationary (see Appendix A). The shorter maturities tend to return and intersect their mean more often, whilst the longer maturities trend for a longer period before crossing their

mean. When we study the different maturities, we observe a decline in how often the option crosses its mean, as the time to maturity increases. The fact that the amplitude of the daily changes in implied volatility declines will contribute to a change in the tails of the return distribution. Since there is a significant difference in the behavior and distribution for the different maturities and levels of moneyness, we expect that there will be a difference in which models will fit better for the different options. We will come back to this in Section 4.

Figure 3.2 Implied volatility for ATM put options with one week and one year to maturity

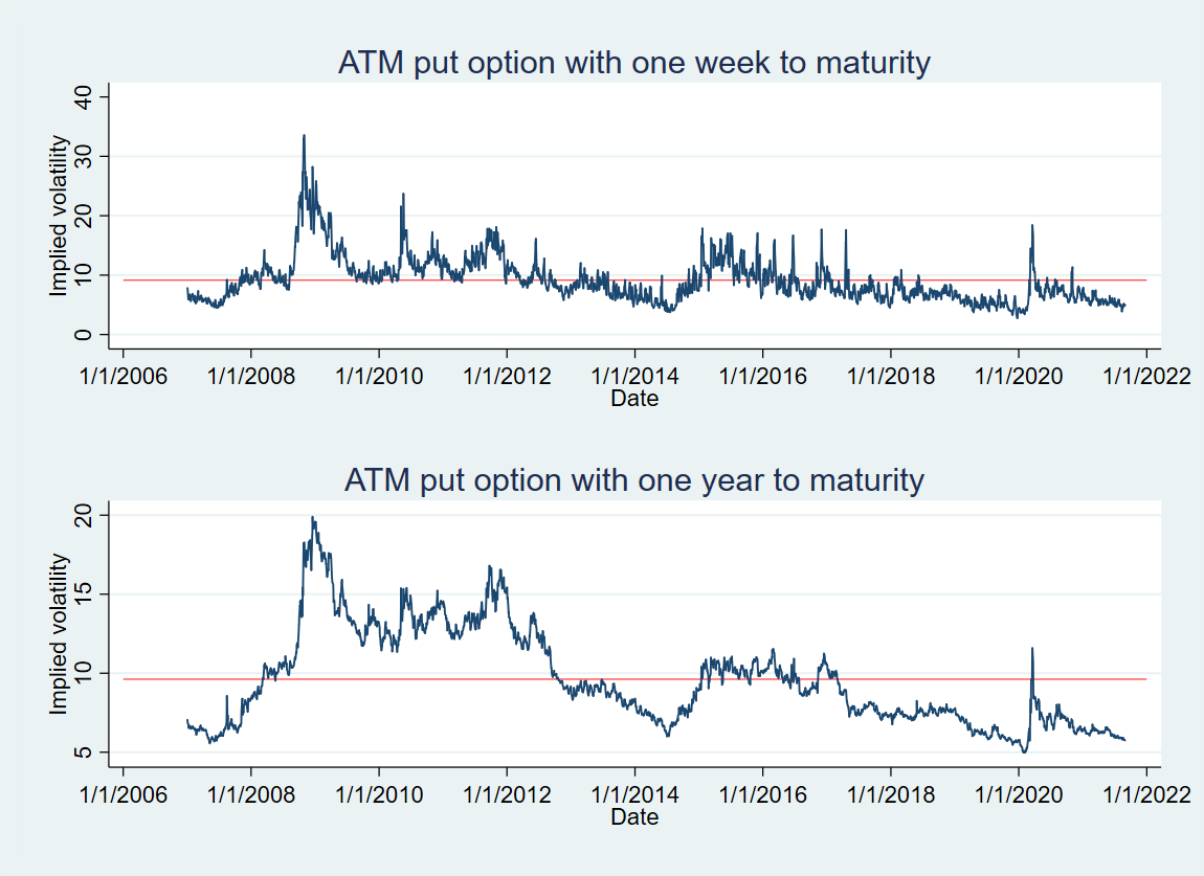


Figure 3.2 Spot rates of implied volatility for ATM put options with one week and one year to maturity from 2. January 2007 until 31. August 2021. The red horizontal line exhibits the options mean value for the sample period

Several macroeconomic factors impact the EUR/USD exchange rate, along with foreign and domestic news for the US and the eurozone, that will directly affect the exchange rate and implied volatility. Our data set stretches from January 2007 to August 2021 and during this time, the financial markets worldwide endured multiple shocks and events that impacted the EUR/USD exchange rate. Figure 3.2 exhibits the implied volatility for the ATM one week and one year option. Especially after the financial crisis in 2008, we see a considerable rise in

implied volatility. Also, shocks such as the European debt crisis, US ceiling debt crisis, and in more recent years, the Covid-19 pandemic had enormous consequences for the implied volatility. These different shocks throughout the data will affect the implied volatility, the daily return of the implied volatility, and the residual distribution of the options when analyzed. Intuitively this can cause challenges for the GARCH-type models, considering the normality assumption for the distribution of the residuals. When looking at the tails of the distribution, the outliers result in fatter tails for the shorter maturities, implying a fatter tail than the normal distribution provides. We will further address these issues regarding our models in Section 5.5. Graphing of daily change distribution and log daily change distribution are found in Appendix B.

Different shocks throughout the data increase the consecutive volatility, resulting in all option maturities having positive skewness, which reflects the right-tailed empirical distribution of the ATM options in figure 3.3. The figure exhibits the ATM option for each maturity, and the distribution widens as the time to maturity increases. The empirical distribution has a double peak for the longest maturities, caused by more extended periods away from their respective mean value, as visualized in Figure 3.2. The distribution pattern corresponds to OTM options, and our findings regarding EUR/USD FX derivatives' statistical and distributional behavior align with earlier literature.

Figure 3.3 Empirical distribution for ATM options for each maturity, Gaussian distribution drawn for each maturity

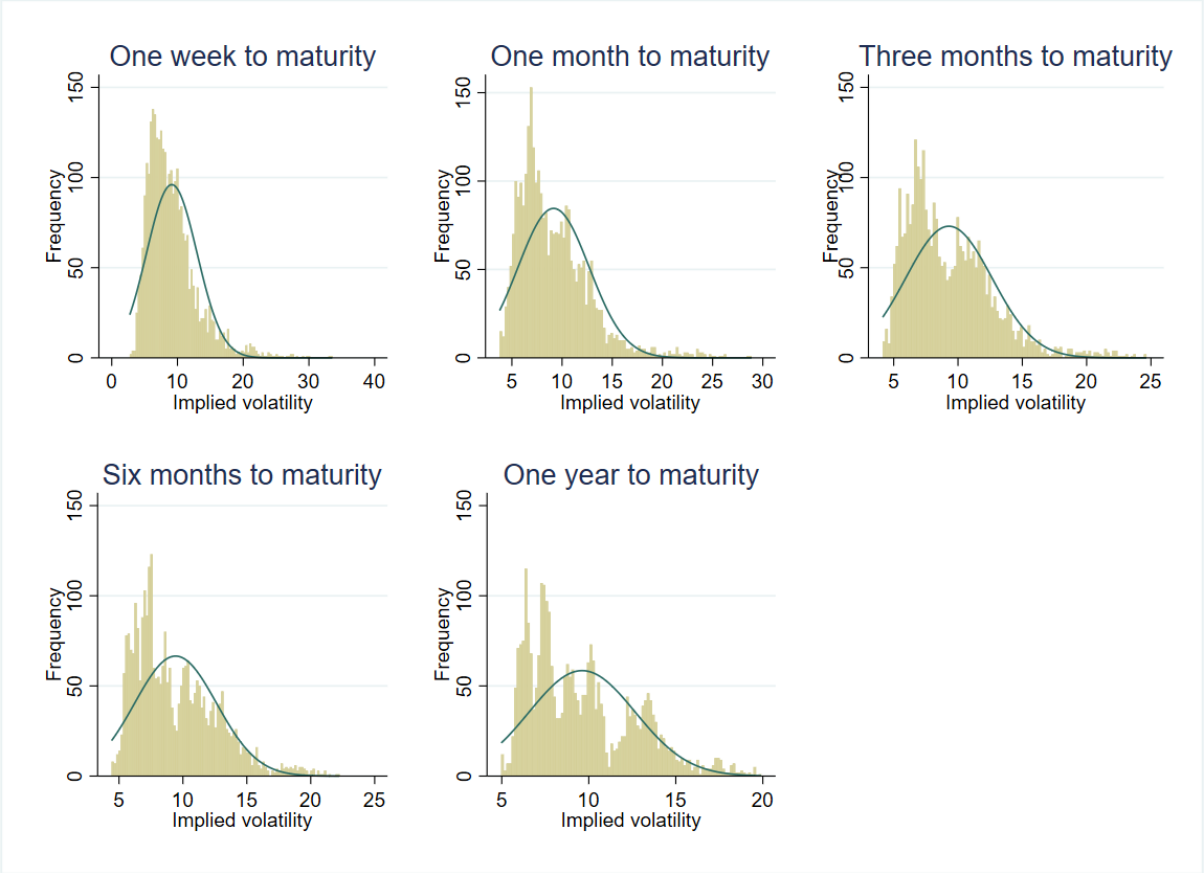


Figure 3.3 Empirical distribution of ATM options for each of the five maturities. The distributions implied volatility along the horizontal axis, and frequency measured in number of observations along the vertical axis. The Gaussian distribution is marked as a curved line for each plot. There are 200 bins and 2994 observations for each maturities plot.

4. Theory and Methodology

In the following Section we will briefly introduce the most important theory behind the models we apply, along with the methodology for the analysis. Further elaborations and theory for the extensions to the models can be found in Appendix A.

4.1 Forecast Evaluation

To evaluate the forecast performance of the different models we use mean squared error (MSE), root mean squared error (RMSE) and mean absolute error (MAE). These statistical measurements are given by the following formulas:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

Where y_t is the implied volatility at time *t,* $\hat{y}_i$ is the forecasted value of implied volatility at time *t,* and *n* is the number of observations. While the mean absolute error measures the average across errors, where the errors are weighted equally, the mean squared error penalizes higher errors. Provided that high forecast errors in the FX market may result in significant losses for market players, we rank the MSE over the MAE.

In addition, we implement a Diebold-Mariano (DM) test to check whether the forecasts are statistically significantly different from each other. The DM test we conduct is based on the MSE for each forecasted value, and test results and methodology are exhibited in Appendix A and E.

4.2 Benchmark Models

We use GARCH and supervised Random Forest as our benchmark model. Common practice is to use 80% of the dataset as a training set and 20% as an out of sample test set. This split results into 2396 observations in the training set and 599 in the out-of-sample test set.

Figure 4.1. 80:20 data split for training and test set, option exhibited is ATM put option with six months to maturity.

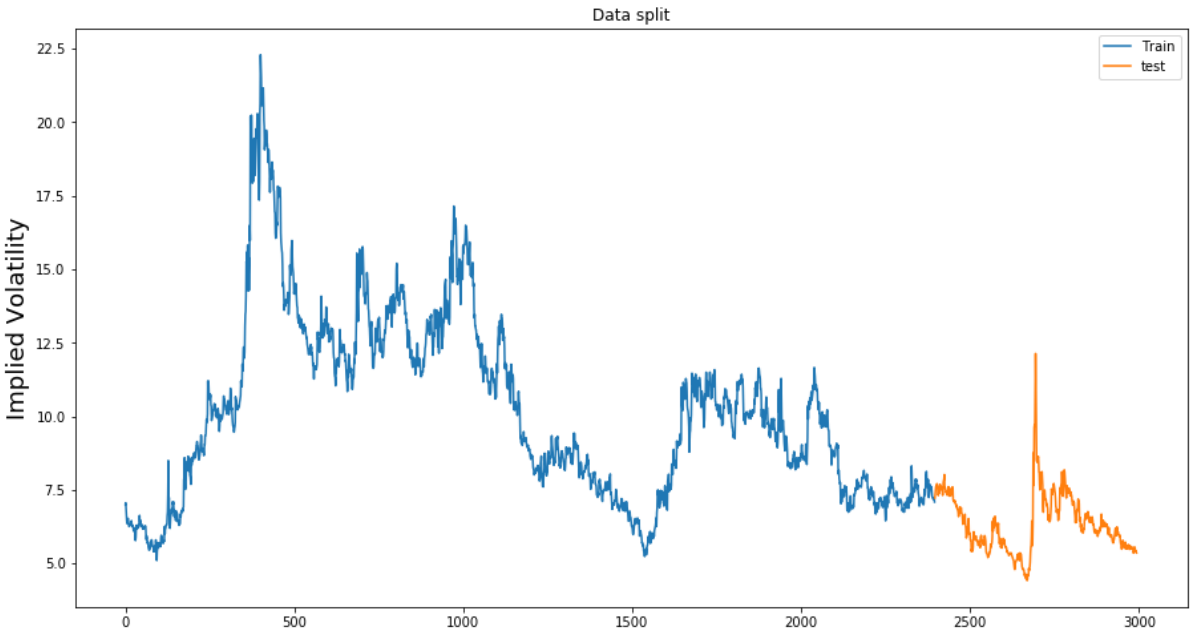


Figure 4.1 80:20 Data Split For an ATM Put Option with Six Months to Maturity

4.2.1 Econometric Model

In traditional economics, there is an assumption that the variance of the residuals is homoscedastic, i.e., they have a constant variance. For time series econometrics, this proves to be difficult, considering different events and shocks yield different levels of volatility over time. This is volatility clustering, where large changes tend to be followed by large changes – of either sign – and small changes tend to be followed by small changes (Mandelbrot, 1963). In 1986, Tim Bollerslev introduced the general heteroskedastic conditional variance model, known as the GARCH model, a much-applied model for modeling and forecasting financial time series. The GARCH is a generalized extension of Robert F. Engles ARCH model from

1982 (see Engle, 1982). The GARCH model allows for the conditional variance to change over time due to past errors leaving the unconditional variance constant (Bollerslev 1986).

We impose an AR(1)-GARCH(1,1) model as the benchmark econometric model. The AR term in the mean equation accounts for autocorrelation in the level of implied volatility and GARCH-terms for the impact of changes in the conditional variance. The GARCH model framework in this study models volatility of implied volatility. The benchmark AR(1)-GARCH(1,1) can be written as:

$$y_t = \mu + \theta y_{t-1} + \varepsilon_t \tag{4}$$

$$\epsilon_t \sim N(0, \sigma_{t-1}^2) \tag{5}$$

$$\sigma_t^2 = \omega + \alpha_1 u_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \tag{6}$$

where equation (4) denotes the mean equation, and equation (6) is the conditional variance equation. θ is the autoregressive coefficient, μ is a constant term, and ε_t is a white noise disturbance term. A critical condition for an AR(p) model is that the model coefficients are stationary. If the models' coefficients are *non*-stationary, the error terms will have a non-declining effect on the value of y_t (see Appendix A). We interpret the fitted variance in equation (6) as a weighted function of information of the volatility from the last period, $\alpha_1 u_{t-1}^2$, the fitted variance from the model for the last period, $\beta_1 \sigma_{t-1}^2$, and a long-term average value which is dependent on the coefficient, ω (Brooks, 2014). All coefficients ω, α_1 and β_1 are non-negative, and the stationary condition states that $\alpha + \beta < 1$ to ensure that the time series process is *weakly* stationary. It is possible to extend the model to the GARCH(m,s) model. However, according to Brooks (2014) a GARCH(1,1) model is sufficient to capture the volatility clustering in the data, and researchers rarely entertain higher-order specifications of GARCH in academic finance literature, particularly not for forecasting purposes. The simpler specifications are less likely to suffer from overfitting to the training data.

When modeling financial data, the distribution of returns appears to have mean clustering and fatter tails than the normal distribution, i.e., more weight in the tails. A distribution that better

fit a variable with clustering around the mean and fat tails is the student t-distribution, which we derive as:

$$\varepsilon_t \sim T_v\left(0, \sigma_t^2\right) \tag{7}$$

where $T_v\left(0, \sigma_t^2\right)$ denotes the Student t-distribution with mean 0, variance σ_t^2 and v degrees of freedom (Miazhynskaia & Dorffner, 2006).

When modeling the volatility, such as within the GARCH framework, the subsequent period volatility reacts to shocks in the data. One of the models' restrictions is that it assumes symmetry in the response to these shocks, i.e., the volatility responds to positive and negative news alike. In financial data, adverse shocks will have a more significant impact on the subsequent period volatility compared to positive shocks. Glosten et al. (1993) formulated a model extension to the GARCH that accounts for asymmetry in response to shocks. This model, known as GJR-GARCH, can be written as an extension of the GARCH(1,1) model:

$$\sigma_t^2 = \omega + \alpha_1 u_{t-1}^2 + \beta_1 \sigma_{t-1}^2 + \gamma u_{t-1}^2 I_{t-1} \tag{8}$$

where $I_{t-1} = 1 \; if \; u_{t-1} < 0$, and otherwise = 0.
Where the γ term captures the asymmetric impact causing a leverage effect when positive, and when $\gamma = 0$, we are back to the standard GARCH(1,1) model (Enders, 2004). The non-negative restrictions of the coefficients ω, α_1 and $\beta_1$still apply.

Econometric Model Methodology

To confirm the presence of autoregressive conditional heteroskedasticity in the data, we perform Engle's Lagrange multiplier test, also known as an ARCH-LM test. The results from the ARCH-LM test can only be interpreted as an indication to investigate if ARCH effects are present or not, and according to Sjölander (2010), the test is biased in finite samples. It does not consider whether the stationarity constraints are met or not. Test results can be found in Appendix E. Test results show that for all options, we can at any significance level reject the null hypothesis of no ARCH effects. We conclude that there is proof of autoregressive conditional heteroskedasticity in the squared residuals for all options.

In addition to the benchmark AR(1)-GARCH(1,1), we extend the analysis with a GJR-GARCH specification to analyze for asymmetric behavior of shocks in the conditional variance. We compare both in-sample goodness of fit and out-of-sample forecast error for all models, focusing on the out-of-sample forecast performance. Further, we test the data with additional lags of the AR(p) and moving average (MA(q)) process, and lastly, we perform the same regressions with a Student t-distribution from equation (7). Additional lags beyond an ARMA(1,1) process do not improve the model's forecasting accuracy and will not be included further in this study.

When modeling and forecasting econometric time series, the variable in question must be stationary. The main problems with non-stationary time series, are that non-stationary variables can produce spurious regressions, meaning the regression has a high R^2 and t-statistics that appear to be significant but without any economic meaning (Enders, 2015). To test for stationarity we use the Augmented Dickey-Fuller test. Results can be found in Appendix E.

Based on the Augmented Dickey-Fuller test and the Phillips-Perron test, we cannot, at a 5% significance level, reject the null hypothesis that the options with time to maturity of 3 months and more follow a unit root process and therefore are non-stationary. To avoid spurious regressions, we apply first difference to all options with three months or more to maturity. After differencing these options, the Augmented Dickey-Fuller test and the Phillips-Perron test rejects the null hypothesis of unit-root, and all options are now stationary. When applying the first difference for the options with one week and one month to maturity, the forecast error measured by MSE, RMSE and MAE was reduced, along with the in-sample goodness of fit. Therefore, the benchmark AR(1)-GARCH(1,1) model is not differentiated for the shorter maturities, i.e., one week and one month to maturity.

The in-sample goodness of fit for the models are compared by information criteria Log-Likelihood, AIC and BIC. According to these information criteria, the in-sample model is considerably improved for all distinct options when adding a moving average (MA) term and other autoregressive term lags. When looking at the autocorrelation and partial autocorrelation, there are individual preferences of which lag of AR(p) and MA(q) terms should be included and show significance for the different options. When testing for asymmetries in the conditional variance, the threshold term improves the goodness of fit and is statistically significant for all options across the level of moneyness and maturities. Based

on these findings, we can say that there is an asymmetry in the volatility shocks, i.e., the volatility reacts differently to positive and negative shocks. When comparing the out-of-sample forecast accuracy measured in MSE, RMSE and MAE, the simple AR(1)-GARCH(1,1) proved to outperform the better in-sample specified models on shorter maturities. Therefore, the in-sample models are *not* the best-fitted models in terms of goodness of fit to the data. Interestingly, the better the in-sample model specification measured in goodness of fit is, the poorer the out-of-sample forecasting accuracy is for the short maturity options. For longer maturities, the findings vary. However, the AR(1)-GARCH(1,1), as described in Equations (3), (4) and (5), is used as a benchmark model for simplicity. The forecast performance of all extensions to the AR(1)-GARCH(1,1) are exhibited in Appendix D.

4.2.2. Random Forest and Tree-Based Model

An ensemble method is an approach that combines many simple models in order to create a single and powerful model (Brownlee, 2021). The simple models are known as weak learners since they may lead to mediocre predictions independently. In this Section we derive regression trees, bagging, Random Forest and our model architecture for the Random Forest model.

Regression Trees

Tree-based regression methods are powerful models to address a regression problem. Classification and regression trees (CART), introduced by Breiman et al. (1984), offer a flexible way to analyze the non-linear relationship between the dependent variable and a set of predictors. Initially, we group all training set records in the same partition, and the algorithm begins allocating the data into the first two partitions, using every possible binary split. The algorithm chooses the splits that minimize an error statistic, such as mean squared error or absolute mean error in the two partitions. Further, we apply the same splitting rule to a new set of partitions, and the same procedure continues until each node reaches a user-specified minimum node size and becomes a terminal node (Vrontos et al., 2021).

Bagging

The issue with the regression trees is that they suffer from high variance. Bootstrap aggregation (Breiman, 1996), widely known as bagging, is a solution to this issue. A Bagging estimator is an ensemble meta-estimator that fits each base regressor on a random subset of the dataset and aggregates their individual predictions by averaging to form a final prediction (Pedregosa *et al.*, 2011). Such a meta estimator can typically reduce the variance of a simple model (e.g. decision trees) by introducing randomization into its construction procedure and then making an ensemble. The algorithm creates B bootstrap samples $B_1, B_2, \ldots, B_B$, and from each bootstrap sample, $B_i,\ i = 1, \ldots, B$, each predictor $\hat{f}_i$ is estimated based on the same learning procedure. Further, we find the bagged predictor, $\hat{f}_{bag}$ by aggregating all the bootstrap predictors.

Random Forest

Bagging is an effective ensemble algorithm as each decision tree fits on different training sets and has a slightly different performance. Random forest, introduced by Breiman (2001), is an extension of bagging. Random forest has the same procedure as bagging, where a specific tree, T_i^{rf}, is created for each bootstrap and the predicator $\hat{f}_i^{rf},\ i = 1, \ldots, B$, is estimated based on the same learning procedure. We obtain the aggregated random forest estimator, $\hat{f}_{rf}$, by averaging all tree-specific estimators. The modeling techniques of Random Forest and bagging are similar. However, there are significant modifications in creating trees in random forest compared to bagging. We chose the splitting variable to be the best among a random subset of *m* candidate variables taken from the complete set of the *p* predictive variables. Further, we take a new random sample of *m* candidate splitting variables from each splitting node of the tree. The use of different bootstrap samples and the introduction of the randomness at each node splitting results in several uncorrelated trees.

Figure 4.2 Random Forest algorithm.

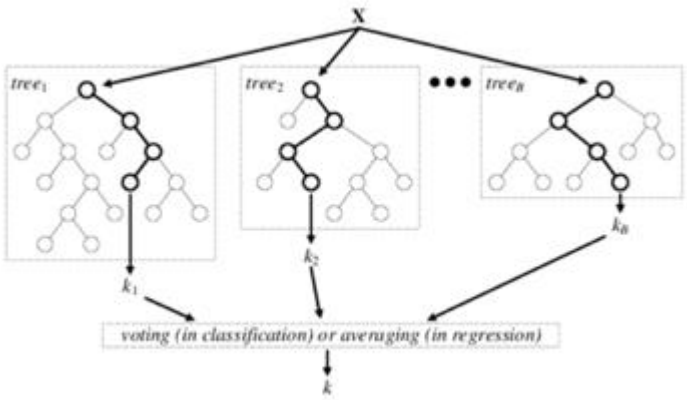


Figure 4.2 Depiction of the Random Forest Algorithm. Source: Vaiciukynas, (2016)

The algorithm in figure 4.2 has the following steps:

step 1: Creation of subsets from the original data.

step 2: Creation of individual decision trees for each subset.

step 3: Output from each decision tree.

step 4: Final output is considered based on the average from all outputs from step 3.

Random Forest Model Architecture

We start our model architecture by implementing the Random Forest regressor from scikit learn (Pedregosa *et al.*, 2011). The sklearn.ensemble.RandomForestRegressor has several parameters that can be tuned. There has been uncertainty in the literature related to many features, the ratio of *m* to *p*, to include in the Random Forest regressor. Breiman (2001) argued that the optimal number of features should be the square root of *p*. Hastie et al. (2008) argued that *p / 3* is the set of features best suitable for the Random Forest regressor. However, Geurts et al. (2006) researched the ratio empirically. They concluded that the optimal set of features is simply *m* = *p*. After running our Random Forest model for different sets of m on different options, we conclude that our model's optimal set of features is *m* = *p*.

Table 4.1 Search for optimal features RF model for ATM put option with one year to maturity.

Features	m = p	Log2 (p)	Sqrt (p)	p / 3
One Year ATM Put	0,0404	0,0448	0,0448	0.0444

Table 4.1 Search for Optimal Features

The next parameters we investigate are the number of trees and the length of the window size. Our initial model was equipped with ten trees and a window size of two. However, increasing the number of trees to the sklearn default of one hundred trees decreased the mean squared error, significantly so for options with a shorter time to maturity. The downside of increasing the number of decision trees is an increase in run-time. We also ran the model with a window size architecture from two to forty. However, this was too computationally expensive to do for each option. Table 4.2 illustrates the mean squared error for an ATM option with one year to maturity. The table indicates that increments in window size do not improve the mean squared error.

Table 4.2 Search for optimal window size RF model for ATM put option with one year to maturity.

	Window size				
	2	**10**	**20**	**30**	**40**
One Year ATM Put	0,0404	0,0421	0,0412	0,0426	0,0412

Table 4.2 Search for Optimal Window Size

4.3 Artificial Neural Networks

Artificial Neural networks are software implementations of the network of neurons present in the human brain. The neurons in the human brain can be thought of as organic switches as the neurons, depending on the strength of their electrical or chemical input, can change their

output state. The neurons have millions of connections with other neighboring neurons (Neural Networks Tutorial – A Pathway to Deep Learning, 2017). This highly complex network allows the human brain to carry out its learning function by activating particular neural connections. The learning process includes feedback resulting in strengthened neural connection when the expected outcome occurs.

Recurrent Neural Networks

A recurrent neural network (RNN) is a neural network where the objective is to predict the next step in the sequence, based on the previous steps observed. The idea of RNNs is to learn from earlier stages to forecast future trends. The earlier data stages need to be remembered when predicting the next step. In RNNs the hidden layers act as internal storage for storing the information captured in earlier stages of reading sequential data (Namin & Namini, 2018). A substantial challenge with vanilla RNNs is that the networks only remember a few earlier steps in the sequence. Ideally, the network's memory horizon should be of the magnitude/length that the network understands real-world coherences, such as trends in financial time series. The more time steps we feed the network, the higher the chance of backpropagation gradients either accumulating, exploding, or vanishing (Brownlee, 2017). Therefore, the gradient becomes close to zero, the weights will not adjust, and the network does not learn relationships separated by time periods.

Long Short-Term Memory (LSTM)

LSTM is an adaptation of the Recurrent Neural Network (RNNs) with features adjusting for the memory challenges in traditional RNNs. The memorization of earlier trends and shocks of the data is possible through gates and a memory line incorporated in LSTM networks. Figure 4.3 demonstrates the internal structure of an LSTM cell.

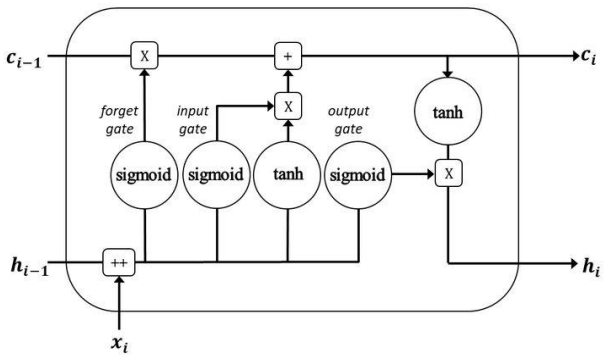


Figure 4.3. LSTM Cell. Source: De Palma, (2019).

Each LSTM is a set of cells, or system modules, where the data streams are captured and stored. The cells resemble a transport line (the upper line in each cell) that connects out of one module to another one conveying data from past modules and gathering them for the present one. The number of cells in an LSTM model equals the size of the chosen window size. The gates in each cell decide if the data passing will be disposed, filtered, or added for the next cells (Namin & Namini, 2018). Hence, the gates which are based on sigmoidal neural network layer, enable the cell to optionally let data pass through or be disposed. Each sigmoid layer yields a number in the range of zero to one, where an estimation of zero implies that no data passes through and an estimation of one implies that all data passes through. There are three types of gates involved of controlling the state of each cell:

Forget Gate generates a number between 0 and 1, where 1 indicates keep all data and zero indicates forget all data.

Memory Gate chooses which new data needs to be stored in the cell. First, a sigmoid layer, called the "input door layer" chooses which values will be modified. Next, a tanh layer makes a vector of new candidate values that could be added to the state

Output Gate decides the output of each cell. The output value is based on the cell state along with the filtered and newly added data.

LSTM Data Split

First, we split our dataset into training, validation and test sets. This is important as the model will be near perfect if we feed the model with the test data. A common practice is to use 80% of the dataset as a training set and 20% as a test set. We settle on a 60:20:20 split, where 60% is used as the training set, 20% is used as the validation set and 20% is used as a test set. In other words, the first 1797 of the first observations in the dataset are used to train the model, the next 599 observations are used to improve the model and fine-tune hyperparameters and the last 599 observations are used for out-of-sample forecasts.

The date intervals are as follows:

- Training set: January 2007 – October 2015
- Validation set: October 2015 – September 2018
- Test set: September 2018 – August 2021

Figure 4.4 Data split for training, validation and test set for LSTM model

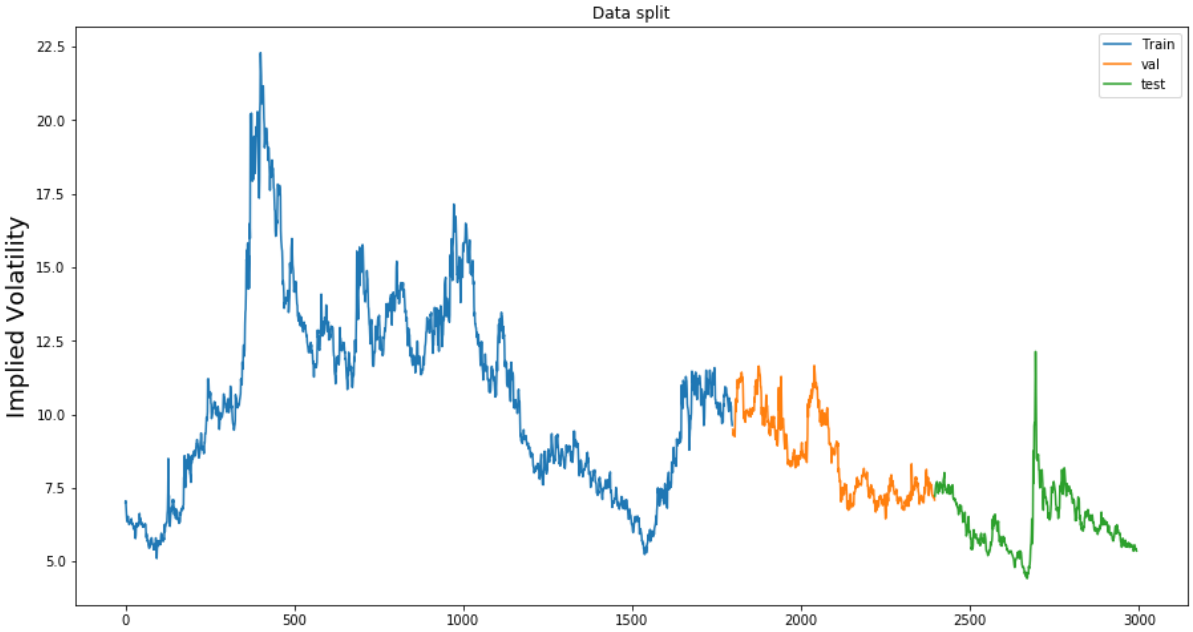


Figure 4.4 LSTM Data Split

LSTM Model Architecture and Hyperparameter Search

Neural network algorithms are stochastic, i.e., they make use of randomness, such as initializing random weights, which will yield different results for a network that is trained on the same data. To improve the LSTM model, we use a random seed, which generates a long sequence of numbers, which will function as weights in the stochastic algorithm, and ensure that the same result occurs when we run the same model twice.

Several parameters require tuning to optimize the LSTM model. The common practice is to evaluate every possible combination of parameters on the validation set and choose the varieties that minimize the statistical properties. However, this approach becomes computationally expensive, especially with an increased window size. For this reason, we develop an architecture that starts by combining smaller sets of hyperparameters, and for each iteration, the hyperparameters increase by 10.

We start our architecture by implementing an LSTM model from the Keras functional API (Chollet, F., & others., 2015) with two hidden layers searching for hyperparameters. We tried using different stacks of LSTM layers, which was computationally expensive and did not

improve the model. Even though the activation function ReLU has risen in popularity because of its computability efficiency (James, 2018), we choose to use the Keras LSTM built-in activation function tanh, as this function seems to work better for our datasets. We use Sigmoid for the recurrent activation function, and Adam as our optimizer. Adam is a variant of the mini-batch gradient descent that adjusts the learning rate at each iteration for each model parameter (Chollet, F., & others., 2015). Our model is specified to minimize the mean squared error. Initially, we construct our model architecture with 300 epochs, a batch size of 64, a window size of 50 and 50 hidden neurons. Another issue with LSTM is overfitting. With excessive training, the model will learn the statistical noise in the training set, predicting the next value based on memory. To avoid overfitting, we implement early stopping, which stops the network as the learning rate stops improving[1].

The hyperparameters that are left to tune are the following:

- Batch size
- Hidden Neurons
- Window Size

Batch Size

We use the previously stated model architecture to speed up the hyperparameter search to locate the optimal batch size. Our goal is to find a batch size that minimizes the mean squared error, and the common practice is to increase the batch size by the power of two because of computational efficiencies (Kandel & Castelli, 2020). Figure 4.5 shows the results for the batch sizes and indicates an optimal batch size of 16.

[1] In addition to early stopping, we tried implementing Keras dropout (Chollet, F., & others., 2015) and Keras Gaussian noise (Chollet, F., & others., 2015), both techniques aimed at preventing overfitting. However, both features either increased - or unaffected the error statistics.

Figure 4.5 Optimal Batch size for LSTM network on ATM six months option

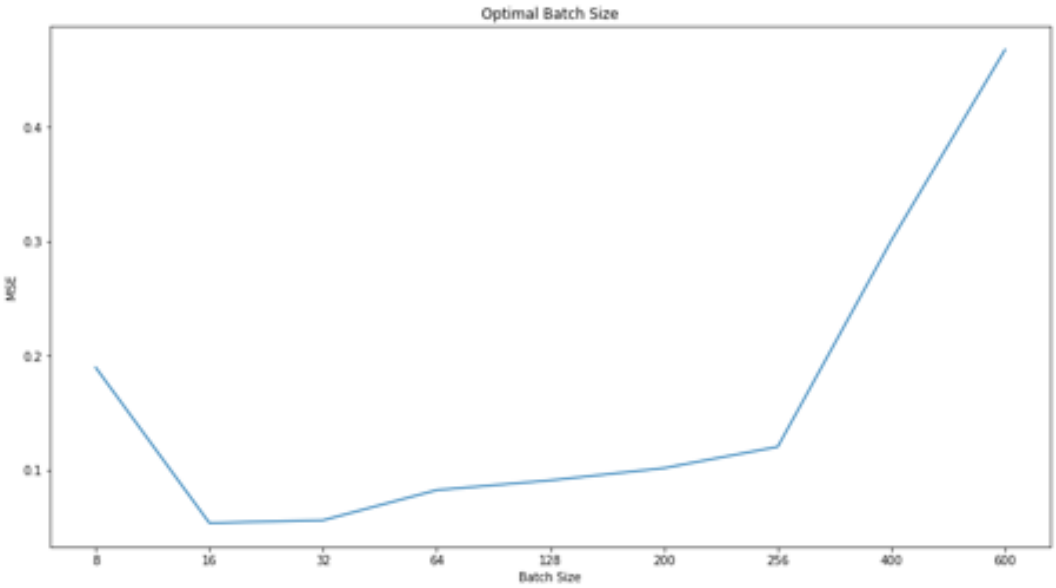


Figure 4.5 Optimal batch size for a Delta 50 Put Option with Six Months to Maturity

Combinations of Window Sizes and Neurons

Further, we investigate the combinations of window sizes and hidden neurons, which are the parameters that largely affect the model's ability to learn. We implement the optimal batch size of 16 to our initial model and develop an architecture that combines the different neurons and window sizes from two to fifty. Table 4.3 reports the mean squared errors for each combination for an ATM option with six months to maturity. The hyperparameter search indicates that a simple model of two lags and twenty hidden neurons minimizes the mean squared error for this specific option.

Table 4.3 Reports the 36 Different Mean Squared Error Estimates for a Delta 50 Put Option with Six Months to Maturity.

	Window size					
Neurons	**2**	**10**	**20**	**30**	**40**	**50**
2	0,2471	0,0566	0,0831	0,0604	0,0880	0,0597
10	0,0933	0,1121	0,1127	0,2031	0,0609	0,0769
20	**0,0550**	0,1008	0,0800	0,5671	0,0721	0,0589
30	0,1035	0,0824	0,0637	0,0556	0,0611	0,0565
40	0,0801	0,0740	0,0652	0,0579	0,0625	0,0565
50	0,0765	0,0682	0,0639	0,0555	0,0594	0,0555

Table 4.3. Reported error estimates for different choices of neurons and window size. The number of neurons along the vertical axis and window size along the horizontal axis. Optimal combination of neurons and window size is highlighted, and for this particular option the best combination of window size and neurons are 20 neurons and a window size of 2.

4.4 Model Expectations

There are different expectations for the forecasting performance of the models when considering the changes in distribution and properties of the various option maturities addressed in Section 3. The AR-GARCH model are modeling the conditional variance of the options, whilst the machine learning models use complicated algorithms to forecast one-day-ahead. As the shorter maturities are more volatile and experience more extensive changes in the implied volatility, the more complex machine learning models expect to perform better than the GARCH-type models. We expect the models to perform more equally on the longer maturities, as the day-to-day changes in implied volatility are minuscule. Previous literature disagrees on which one of RF and LSTM is the best forecaster for securities and volatility. LSTM has proven better than ARIMA models, notably do LSTMs perform well in times of

market turmoil with extreme values. Therefore, we expect LSTM to perform well for volatile options on shorter maturities and OTM options.

5. Forecasting Results

For the out-of-sample forecast, the accuracy is significantly lower for ATM options or options close to ATM, and decreases as the option becomes increasingly OTM. As the time to maturity increases, the forecast accuracy significantly improves. From one-week to maturity to one-year to maturity the average RMSE decreases from 0,7357 to 0,2417 for the GARCH model, 0,7303 to 0,2521 for LSTM and 0,8224 to 0,2473 for Random Forest. A decline of respectively 67,16%, 65,47% and 69,92%. The daily change in the implied volatility, computed as the absolute value of the average daily change for each maturity, declines by 76,21% when the maturity increases from one week to one year. The reduction in RMSE is therefore declining with longer maturities as expected beforehand. The daily change in the implied volatility is also lower for options ATM and options close to ATM than OTM options. It is also lower for call options than put options with the same option delta (negative risk-reversal).

5.1 ATM and OTM Options Summary

The results for an ATM put and an OTM put and call for each of the five specific times to maturity are presented in table 5.1. The first column indicates the options level of moneyness and time to maturity, the three following columns the forecasting accuracy of the AR(1)-GARCH(1,1) model, the three mid columns the results for the LSTM model, and the three columns to the right the results for the Random Forest model.

Table 5.1. Forecast performance for OTM put/call options and ATM put options

	AR(1)-GARCH(1,1)			*LSTM*			*RANDOM FOREST*		
	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE
1 week put 5	**0,6262**	**0,7913**	0,5056	0,6362	0,7976	**0,4974**	0,8277	0,9098	0,5973
1 week put 50	0,4895	0,6997	0,4632	**0,4763**	**0,6901**	**0,4475**	0,6944	0,8333	0,5492
1 week call 5	0,5947	0,7712	0,4813	**0,5917**	**0,7692**	**0,2833**	0,7820	0,8843	0,5583
1 month put 5	**0,2561**	**0,5061**	**0,2729**	0,2645	0,5143	0,2767	0,2670	0,5167	0,3055
1 month put 50	**0,1500**	**0,3873**	0,2709	0,1537	0,3920	**0,2308**	0,1842	0,4292	0,2640
1 month call 5	**0,2519**	**0,5019**	0,2518	**0,2519**	**0,5019**	**0,2513**	0,2806	0,5297	0,2782
3 months put 5	**0,1706**	**0,4131**	**0,2002**	0,1792	0,4233	0,2056	0,1777	0,4215	0,2346
3 months put 50	**0,0773**	**0,2781**	**0,1540**	0,0850	0,2915	0,1610	0,0888	0,2980	0,1757
3 months call 5	**0,1387**	**0,3725**	**0,1772**	0,1413	0,3759	0,2278	0,1642	0,4052	0,1999
6 months put 5	**0,1324**	**0,3639**	**0,1641**	0,1392	0,3731	0,1679	0,1443	0,3799	0,1963
6 months put 50	**0,0506**	**0,2248**	**0,1191**	0,0550	0,2345	0,1269	0,0634	0,2518	0,1445
6 months call 5	**0,0975**	**0,3123**	**0,1433**	0,1022	0,3197	0,1449	0,1137	0,3372	0,1649
1 year put 5	**0,1135**	**0,3369**	**0,1421**	0,1215	0,3486	0,1436	0,1399	0,3740	0,1714
1 year put 50	**0,0367**	**0,1916**	**0,0972**	0,0398	0,1995	0,0992	0,0404	0,2010	0,1124
1 year call 5	**0,0770**	**0,2776**	**0,1195**	0,0815	0,2855	0,1232	0,0841	0,2900	0,1359

Table 5.1 First column indicates the level of moneyness measured in delta for the different maturities. Delta 50 is the ATM option, and delta 5 is the OTM put and call option. The highlighted value indicates the best fitted value for that particular option for MSE, RMSE and MAE, respectively.

The benchmark AR-GARCH model is performing superior for both put and call options compared to the machine learning methods for options with longer maturities. For all options with maturities of three months or more, the AR-GARCH model outperforms both of its more advanced competitors. The Random Forest model has the lowest forecast performance of the three models for all options across the different maturities. On shorter maturities, i.e., for options with one week and one month to maturity, the LSTM and AR-GARCH model are the best-fitted models. For an OTM put option with one week to maturity, the GARCH model is the best fitted with an RMSE of 0,7913 whilst the LSTM model has an RMSE of 0,7976.

The Random Forest model performs significantly poorer with an RMSE of 0,9098, an increase of 14,97% compared to the GARCH model. It is somewhat surprising that the GARCH outperforms the LSTM model for this particular option, considering this is the most volatile of the fifty-five options. The LSTM performs better in terms of MAE, meaning it is not as robust to outliers in the test set as the GARCH model. When we perform a DM test for this option we see that there are no significant differences between LSTM and GARCH or LSTM or RF. However, the AR-GARCH is significantly better than the RF. According to the DM test, AR-GARCH is significantly better than RF for one week options. AR-GARCH is significantly better than LSTM for the ATM option, but not for OTM options when time to expiration is one week.

5.2 One Week to Maturity

For all other options with a maturity of one week, the LSTM model outperforms the benchmark models on RMSE and MAE. An exception is a put option with a delta of 35, where the benchmark GARCH model has an MAE 1,96% lower than the LSTM, and a put option with a delta of 10, in which both models have an RMSE of 0,7635, a forecast accuracy 7,85% better than the Random Forest model. On average, for all options with one week to maturity, the LSTM outperforms the GARCH model with 0,75% in RMSE and 2,77 % measured by MAE, whilst LSTM is 11,07% and 14,99% lower than the Random Forest in terms of RMSE and MAE, respectively. These findings show that the benchmark AR-GARCH model is not significantly poorer than the LSTM model at shorter maturities, whilst the LSTM outperforms the Random Forest model considerably. For the one week to maturity there is no significant difference between LSTM and RF, according to the DM test.

In Figures 5.1, 5.2 and 5.3, the forecasted values for the ATM put option with one week to maturity are plotted against the actual spot of implied volatility for the forecasting period for LSTM, RF and benchmark GARCH, respectively.

Figure 5.1. Forecast results for LSTM model on ATM one week to maturity option.

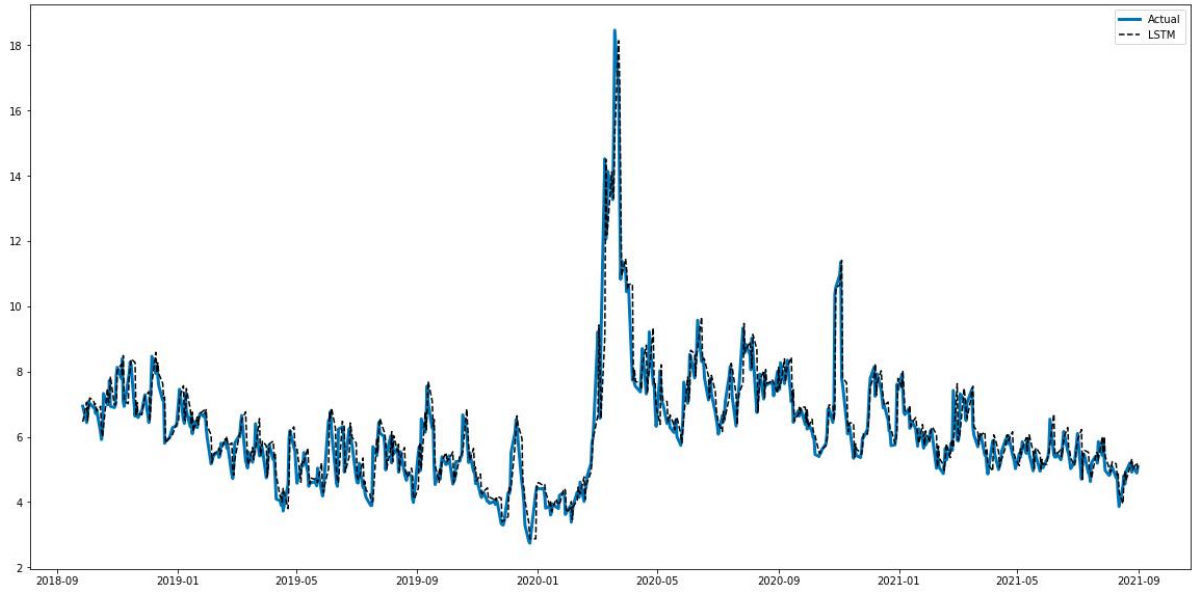


Figure 5.1. Forecast results for LSTM model on ATM one week to maturity option plotted against the actual spot rate for the implied volatility.

Figure 5.2. Forecast results for RF model on ATM one week to maturity option.

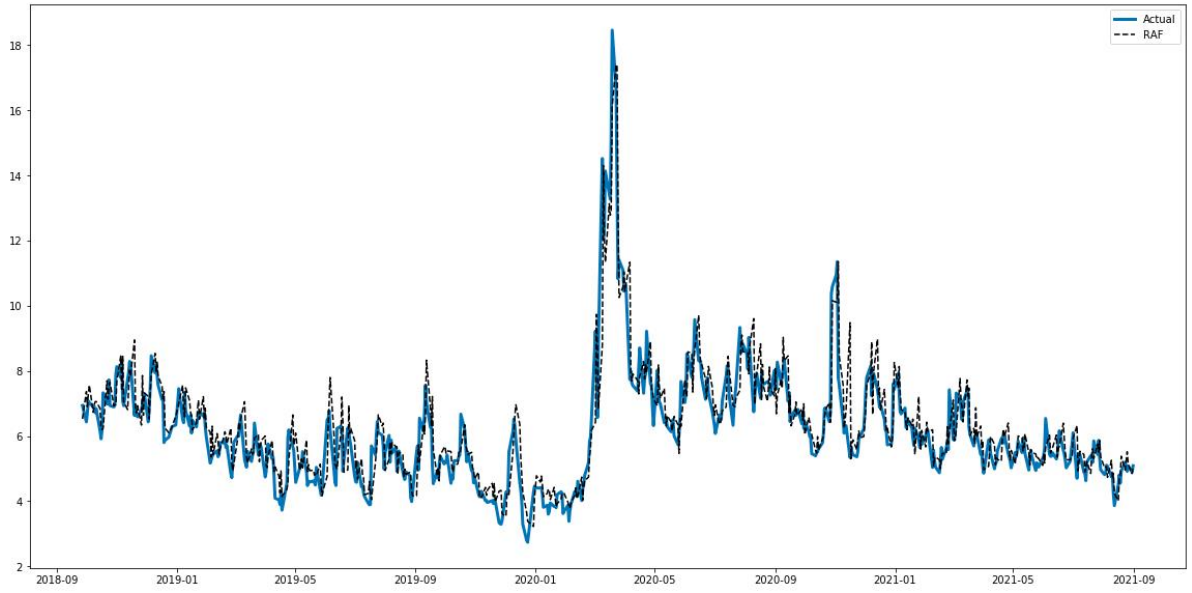


Figure 5.2. Forecast results for Random Forest model on ATM one week to maturity option plotted against the actual spot rate for the implied volatility.

Figure 5.3. Forecast results for benchmark AR-GARCH model on ATM one week to maturity option.

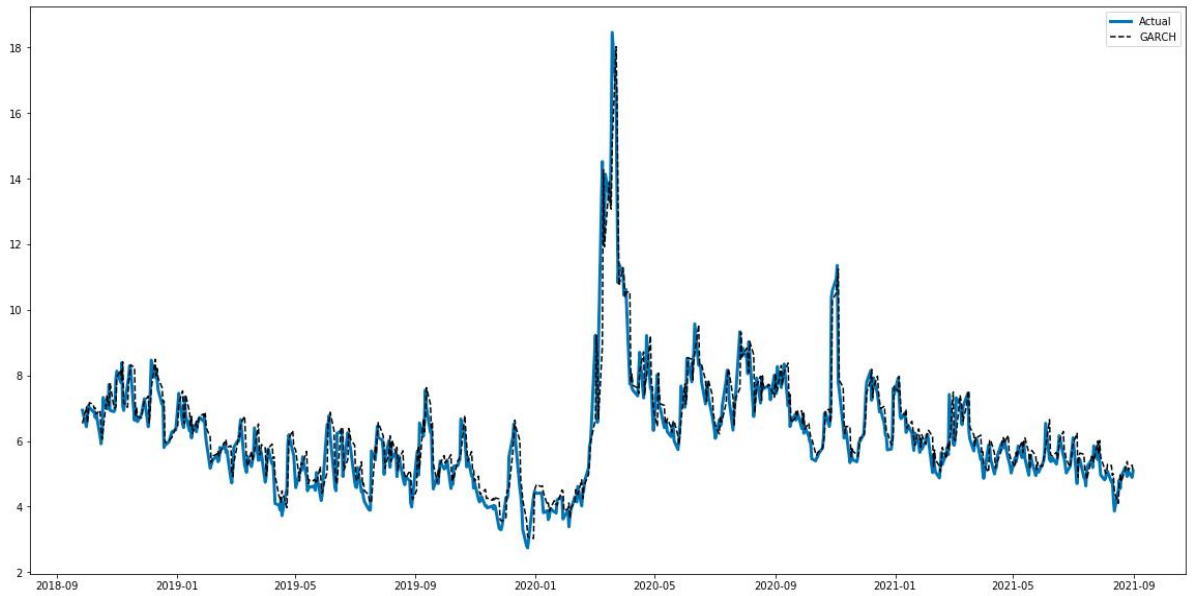


Figure 5.3. Forecast results for the benchmark AR-GARCH model on ATM one week to maturity option plotted against the actual spot rate for the implied volatility.

5.3 One Month to Maturity

When the time to maturity increases to one month, the results fluctuate more. Comparing the OTM and ATM put and call options depicted in Table 5.1, the benchmark AR-GARCH model seems to deliver the best forecast accuracy of the three models measured by RMSE. The LSTM performs equally well for the OTM call option and beats the benchmark AR-GARCH at MAE for the OTM put option. When comparing RMSE for all levels of moneyness, the benchmark AR-GARCH model performs on average 0,78% better than the LSTM model. However, the LSTM is, on average, 1,56% more accurately measured by MAE. The benchmark AR-GARCH model captures the outliers, i.e., significant sudden changes in the volatility, better than the LSTM model. It is also interesting that the LSTM model outperforms both benchmark models in terms of RMSE and MAE for OTM call options, i.e., options with a delta of 25 and lower. Comparing the LSTM to the Random Forest model, the RMSE and MAE are 4,97% and 10,82% lower for the LSTM model. The LSTM model outperforms the Random Forest model more for call options than for put options.

According to the DM test, LSTM is significantly better than the RF for OTM call option, but not significantly better for ATM and OTM put options. Comparing LSTM to AR-GARCH model, the same applies here where the LSTM is significantly better for OTM call option, and there is no significant difference for ATM and OTM put options. When comparing AR-GARCH to RF, RF are significantly poorer for ATM and OTM put options, and there are no significant differences between OTM call options.

Figures 5.4, 5.5 and 5.6, plot the forecasted values for ATM one month to maturity put options for the LSTM, RF and AR-GARCH model. The plot shows that the RF model has problems with the extensive shocks in implied volatility, especially around March 2020, when the COVID-19 pandemic had its outbreak worldwide. The RF overestimates the peaks from COVID-19 shocks, whereas the LSTM model underestimates these shocks. All through the test period, which stretches from the end of September 2018 to august 2021, the AR-GARCH fits the rapid changes in implied volatility better than the machine learning models, especially around the extensive shocks, whereas the implied volatility rises significantly.

Figure 5.4. Forecasting results for LSTM model on ATM one month option.

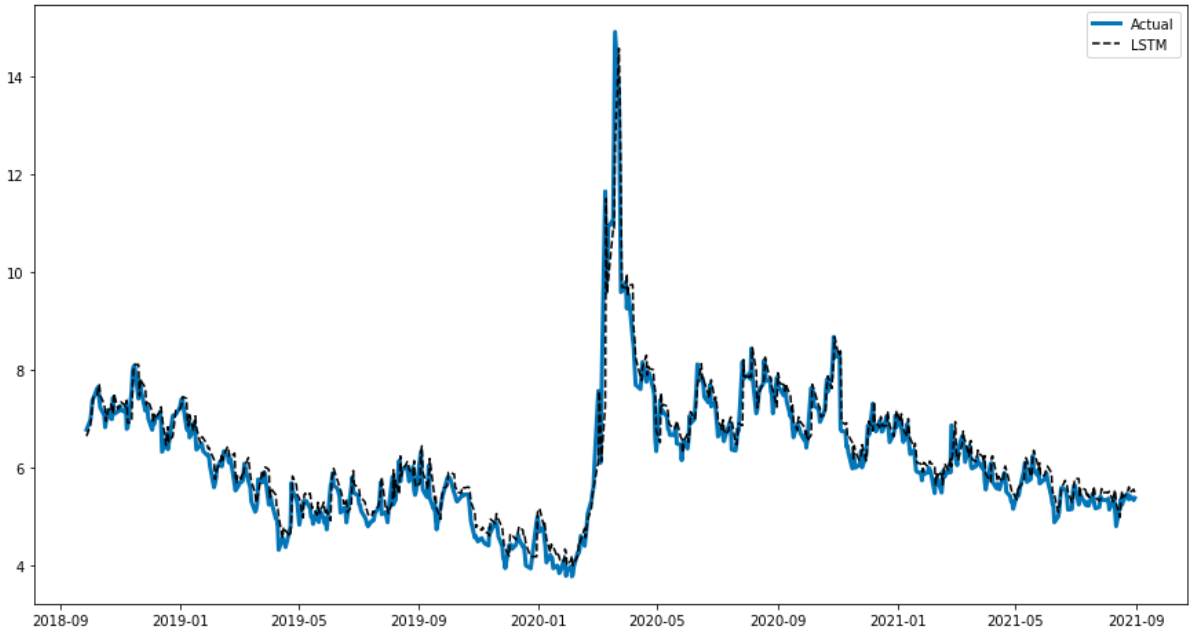


Figure 5.4. Forecast results for LSTM model on ATM one month to maturity option plotted against the actual spot rate for the implied volatility.

Figure 5.5. Forecasting results for Random Forest model.

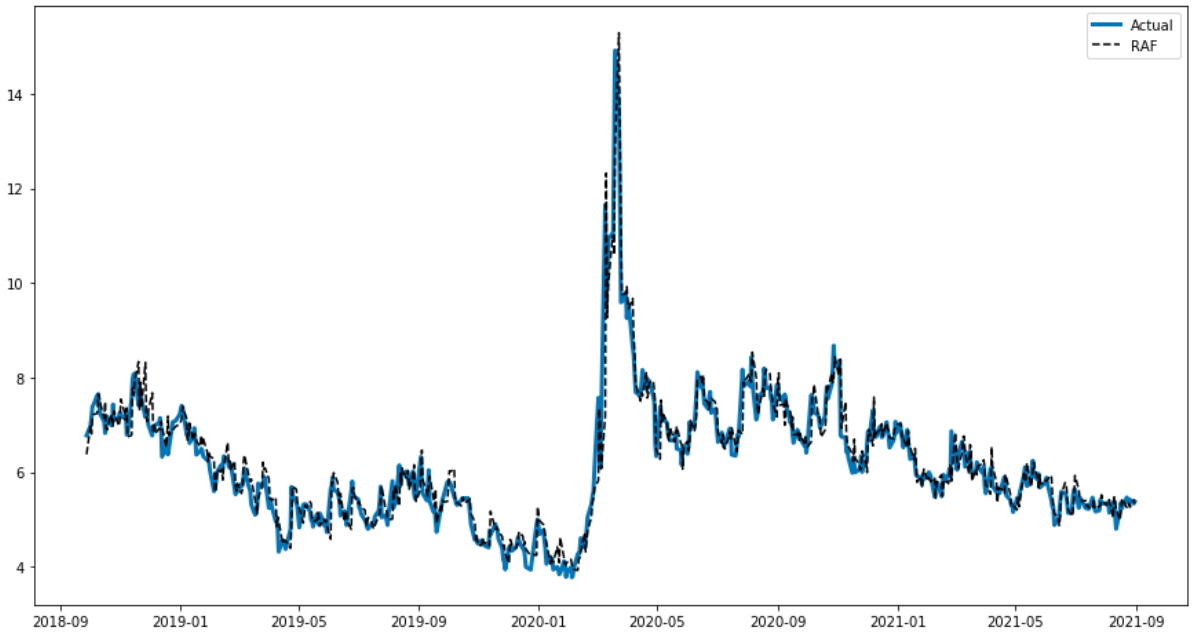


Figure 5.5. Forecast results for Random Forest model on ATM one month to maturity option plotted against the actual spot rate for the implied volatility.

Figure 5.6. Forecasting results for benchmark AR-GARCH model.

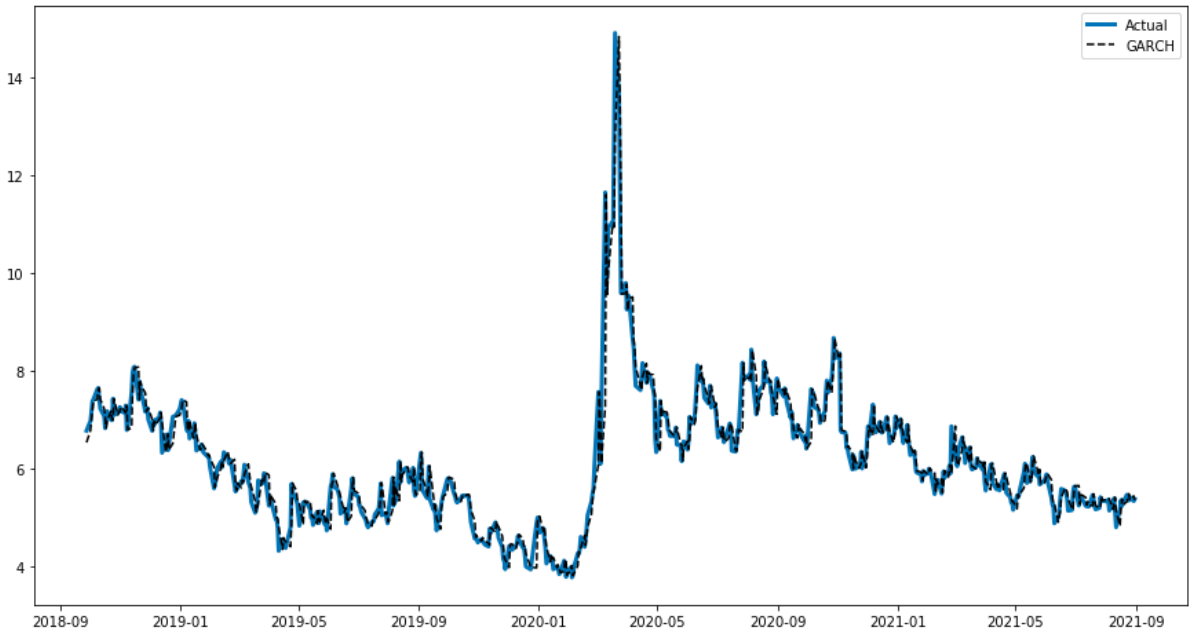


Figure 5.6. Forecast results for AR-GARCH model on ATM one month to maturity option plotted against the actual spot rate for the implied volatility.

5.4 Longer Maturities

For all options with a time to maturity of three months and longer, the simple benchmark AR(1)-GARCH(1,1) model proved superior to the more complicated machine learning models across all moneyness levels. Due to non-stationarity at a 5% significance level, the first difference is applied for all options when the maturity surpasses three months. The benchmark AR-GARCH model is better than the LSTM model with increasing maturity. At the same time, the Random Forest comes closer to the LSTM with increasing maturity, measured in average RMSE. However, LSTM still outperforms the Random Forest for all moneyness levels except for a three-month put option with a delta of five and a call option with a delta value of 35. On average, the difference in RMSE between the Random Forest and LSTM declines from 11,07% for one-week options to 2,80% for a one-year option. When comparing the MAE for the LSTM and Random Forest model, the differences are more significant, varying from the lowest for the three-month option at 8,24% to 14,99% for the one-week option. Interestingly, the MSE increases between the two models as the time to maturity increases with respectively 10,08% at six months and 11,25% when the time to maturity increases to one year, while the difference in RMSE decreases. When conducting the DM test on the longer maturities, the results indicate no significant difference between the forecasts, statistically speaking. This result is expected as the day-to-day changes in implied volatility decrease as maturity increases.

5.5 Other Findings

The distribution for the changes in implied volatility has high peaks and fat tails. As mentioned in Section 3, the distribution changes with time to maturity. When regressing in-sample, we assume that the residuals follow a normal distribution. This precondition is applied for the in-sample regressions for the benchmark AR(1)-GARCH(1,1) model. In the same regressions with Student t-distribution, the average RMSE declined by 1,36% for one-week to maturity options, making the Student t-distributed model the superior model for forecasting compared to the benchmark AR-GARCH model. For the options with one month to maturity, the Student t-distributed model performs 0,24% better than the benchmark AR-GARCH model. The t-distribution fits better for data with mean clustering and fat tails than the normal distribution. Our findings support that the t-distributed models fit the data better for *shorter* maturities. The benchmark AR(1)-GARCH(1,1) model with normal distribution is

better than the t-distribution for the first-order integrated options with a maturity of three months and more. It is essential to state that the in-sample goodness of fit decreases for the Student t-distributed model compared to the normal distribution model used as the benchmark. According to the DM test, the t-distributed AR(1)-GARCH(1,1) is significantly better than the LSTM and RF model for one week OTM options. The differences between other options are not significant.

When the extensions to the AR(1)-GARCH(1,1) as explained in Appendix D are imposed, we see that other econometric model specifications outperform the benchmark model. Further, when all econometric specifications are considered, the machine learning models are outperformed for all options. The results of these models are exhibited in Appendix D.

5.6 Results Discussion

Previous literature has proven that LSTM models are forecasting better than ARIMA models for financial time series. However, the market expectation of risk changes is quickly absorbed when modeling implied volatility. Combining the ARMA and GARCH framework as we impose in this study makes the econometric framework more accurate when different specifications are considered. The LSTM was, compared to the econometric models, more accurate for shorter option maturities, which are the most volatile. When studying the time series properties as discussed in Section 3, the shorter maturities cross their mean more often than the longer maturities. The LSTM model seems to pick up these rapid changes in implied volatility better than the benchmark, which is an interesting finding as transitions from low to high market volatility regimes can be abrupt and short lived. However, when we extend the AR-GARCH model with a moving average term and t-distribution in the residuals are accounted for, the ARMA-GARCH models capture most of the features in the implied moments of the data—resulting in more accurate forecasts for the implied volatility.

6 Conclusion

The main objective of this thesis is to analyze the forecasting performance of an LSTM model compared to two types of benchmark models, i.e. the Random Forest model and Gussian distributed AR(1)-GARCH(1,1) model, for daily observations of the implied volatility for FX options. All regressions and analyses are conducted on daily observations of the spot rate of implied volatility for EUR/USD FX options. Implied volatility is of interest to market participants for hedging and trading purposes. The models applied in this analysis are widely used on financial data such as return rates, volatility and indexes, but literature regarding the implied volatility of FX options are scarce.

We find that the AR-GARCH model outperforms the LSTM model for longer maturities, and the RF model was the poorest overall forecaster. LSTM is the better model for shorter maturities. Shorter maturity options are more volatile than the longer maturities. The LSTM seems to capture rapid changes better than the benchmark models, which is consistent with the findings in previous literature. The LSTM model provides a stable framework for when immense and immediate changes in implied volatility occur, particularly essential for hedging and trading against significant shifts in FX rates to avoid big losses.

Overall, the Random Forest model is a poorer forecaster of implied volatility than the LSTM and AR-GARCH model for all moneyness levels and time to maturity. When we conduct the benchmark AR(1)-GARCH(1,1) with t-distribution, the LSTM model is also outperformed for shorter maturities. According to our findings, the LSTM model is outperformed by traditional econometric models with respect to forecasting accuracy of the implied volatility of EUR/USD FX options when different specifications of the ARMA-GARCH framework are taken into consideration. As discussed in Section 5 we find that for some options the AR-GARCH is significantly better than the LSTM, but for most options the forecasting difference between the two is not statistically significant, according to the Diebold-Mariano test measured by MSE. RF is for most options significantly poorer than the AR-GARCH model. RF is poorer than LSTM, but the difference is not statistically significant for most options.

References

Beckers, S. (1981), Standard deviations implied in option prices as predictors of future stock price variability. *Journal of Banking Finance*, *5(3)*, 363-381. https://doi.org/10.1016/0378-4266(81)90032-7

Bharadia, M. A. J., Christofides, N. & Salkin, G. R. (1996). A Quadratic Method for the Calculation of Implied Volatility Using the Garman-Kohlhagen Model. *Financial Analysts Journal*, *52(2)*. 61–64. http://www.jstor.org/stable/4479908

Black, F. & Scholes, M. (1973) The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, *81*: 637-659.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics, 31(3)*, 307-327. https://doi.org/10.1016/0304-4076(86)90063-1

Buitinck *et al.*, (2013). API design for machine learning software: experiences from the scikit-learn project. *https://scikit-learn.org/*

Breiman, L. (2001). Random Forests. *Machine Learning 45, 5–32* . https://doi.org/10.1023/A:1010933404324

Breiman, L. (1996), Bagging predictors. *Machine Learning 24, 123–140*. https://doi.org/10.1007/BF00058655

Breiman, L., Friedman, J., Olshen, R. & Stone, C. (1984). *Classification and Regression Trees*. Wadsworth.

Brooks, C. (2014). *Introductory econometrics for finance.* 3rd ed. Cambridge: Cambridge University Press.

Brownlee J. (2016). Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras. https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/

Brownlee J. (2017). A Gentle Introduction to Exploding Gradients in Neural Networks. https://machinelearningmastery.com/exploding-gradients-in-neural-networks/

Carr, P., Wu, L. & Zhang, Z. (2020). Using Machine Learning to Predict Realized Variance, *Journal of Investment Management, 18(2)*. 1-16. https://doi.org/10.48550/arXiv.1909.10035

Chollet, F., & others. (2015). Keras. GitHub. https://github.com/fchollet/keras

De Palma, P. (2019). Language-Agnostic Syllabification with Neural Sequence Labeling. https://www.researchgate.net/figure/Diagram-of-the-LSTM-cell-c-i-and-h-i-are-the-cell-states-and-hidden-states-that_fig2_336147416

Enders, W. (2015). *Applied econometrics time series*. 4th ed. New York: John Wiley & Sons Inc.

Engle, F. R., (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica, 50(4)*. 987-1007. https://doi.org/10.2307/1912773

Geurts, P., Ernst, D. & Wehenkel, L. Extremely randomized trees. *Mach Learn 63, 3–42* (2006). https://doi.org/10.1007/s10994-006-6226-1

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*: 1779-180. https://doi.org/10.1111/j.1540-6261.1993.tb05128.x

Haug, E. G., Frydenberg, S. & Westgaard, S. (2010). Distribution and Statistical Behavior of Implied Volatilities. *Business Valuation Review, 29(4).* 186-199. https://doi.org/10.5791/0897-1781-29.4.186

Hastie, T., Tibshirani, R. & Friedman, J. (2008) R. L. (2014). *The Elements of Statistical Learning* (2nd ed). Springer.

James, G., Witten, D., Hastie, T. & Tibshirani. (2021). *An introduction to statistical learning: With applications in R* (2nd ed). Springer

Javed, F. & Mantalos, P. (2013). GARCH-Type Models and Performance of Information Criteria. *Communications in Statistics - Simulation and Computation, 42(8)*. 1917-1933, https://doi.org/10.1080/03610918.2012.683924

Kandel, I,. Castelli M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. *ICT Express, 6(4)*, 312-315. https://doi.org/10.1016/j.icte.2020.04.010

Kryzanowski, L., Galler M. (1993). Using Artificial Neural Networks To Pick Stocks. *Financial Analyst Journal, 49(4),* 21-27. https://www.jstor.org/stable/4479664

Krauss, C., Xuan M. & Huck, N., (2017). Deep Neural Networks, Gradient Boosted Trees, Random Forests: Statistical Arbitrage on the S&P 500. *European Journal of Operational Research, 259(2)*, 689-702. https://doi.org/10.1016/j.ejor.2016.10.031

Latané, H. A. & Rendleman, R. J. (1976). Standard Deviations of Stock Price Ratios Implied in Option Prices. *The Journal of Finance, 31(2)*, 369–381. https://doi.org/10.2307/2326608

Lim, C. M. & Sek, S. K., (2013). Comparing the performances of GARCH-type models in capturing the stock market volatility in Malaysia. *Procedia Economics and Finance, 5.* 478-487.

Mandelbrot, B. (1963). The Variation of Certain Speculative Prices. *Journal of Business, 36(4)*, 394-419. http://www.jstor.org/stable/2350970

McDonald, R. L. (2014). *Derivatives Markets*. 3rd ed. London: Pearson Education Limited.

Miazhynskaia, T. & Dorffner, *G.* (2006). A comparison of Bayesian model selection based on MCMC with an application to GARCH-type models. *Statistical Papers 47***,** 525–549.

Namin, A. & Namini S. (2018). Forecasting Economics and Financial Time Series: Arima vs. LSTM. htttps://doi.org/10.48550/arXiv.1803.06386

Neural Networks Tutorial – A Pathway to Deep Learning. (2017). https://adventuresinmachinelearning.com/neural-networks-tutorial

Ornelas, J. R. H. & Mauad, R. B. (2019). Implied volatility term structure and exchange rate predictability. *International Journal of Forecasting. 35*, 1800-1813.

Pedregosa et al., (2011). Sckit-learn: Machine Learning in Python*, JMLR 12*, 2825-2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

Poom, S. & Granger, C. (2001). *Forecasting financial market volatility: A review.* Department of Econometrics. http://doi.org/10.2139/ssrn.268866

Schmidt, Ludwig. (2021). *Volatility Forecasting Performance of GARCH Models: A Study on Nordic Indices During COVID-19.* [Master thesis] Umeå University.

Ramasamy, R. & Munisamy, S. (2012). Predictive Accuracy of GARCH, GJR and EGARCH Models Select Exchange Rates Application. *Global Journal of Management and Business Research.12(15)*, 89-100.

Sjölander, P. (2010). A Stationary Unbiased Finite Sample ARCH-LM Test Procedure. Applied Econometrics. *Taylor & Francis (Routledge) 43(8), pp.1019.* 10.1080/00036840802600046.

Vaichiukynas, E. (2016). Electromyographic Patterns during Golf Swing: Activation Sequence Profiling and Prediction of Shot Effectiveness. *MDPI.* https://www.researchgate.net/figure/Architecture-of-the-random-forest-model_fig1_301638643

Vrontos, Spyridon & Galakis, John & Vrontos, Ioannis. (2021). Implied volatility directional forecasting: a machine learning approach. *Quantitative Finance 21(4), 1-20.* https://doi/10.1080/14697688.2021.1905869

Zaiontz, C. (2022). *Diebold-Mariano Test.* https://www.real-statistics.com/time-series-analysis/forecasting-accuracy/diebold-mariano-test/ (Hentet 20.05.2020).

7 Appendix

7.1 Appendix A: Theory

Appendix A elaborate theory that is relevant for understanding the processes, analysis and tests conducted in this thesis, but were considered excess information for a knowledgeable reader.

Volatility Theory

Volatility measures the fluctuated deviation of the underlying asset (in our case, the price of a currency) from its mean over a defined period, given by the formula:

$$\sigma = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(x_t - \bar{x})^2} \tag{A.1}$$

Where x_t is the exchange rate at time *t*, $\bar{x}$ is the mean of the exchange rate over the given period, and n is the number of observations. Equation (A.1) is often called the realized volatility, and realized volatility is computed from historical data of the exchange rate (or the underlying asset) spot prices. In the stock market, the implied volatility is derived from option prices observed in the market and interpreted as the market's expectations for future volatility. Thus, implied volatility measures expected future fluctuations in the option price. In the FX market on the other hand, implied volatility is directly quotas and you do not have to calculate the option price to obtain it, and it is a market-based measure of future risk. A widely applied model for option pricing is the Black-Scholes model, and for a European call option on a stock is given by:(Black & Scholes, 1973):

$$C(S, K, \sigma, r, T, \delta) = Se^{-\delta T}N(d_1) - Ke^{-eT}(d_2) \tag{A.2}$$

where:

$$d_1 = \frac{ln(\frac{S}{K}) + (r - \delta + \frac{1}{2}\sigma^2)}{\sigma\sqrt{T}}$$
$$d_2 = d_1 - \sigma\sqrt{T}$$

Where S is the stock's current price, K is the option's strike price, σ is the volatility of the stock, r is the continuous compounded risk-free interest rate, T is time to maturity, and δ is

the dividend yield on the stock. The N(x) function is the cumulative normal distribution function. When applying the Black-Scholes formula to other underlying assets, we need to modify the formula. By replacing the dividend yield with foreign risk-free interest rate r_f and the stock price with the spot exchange rate x_0 we define the prepaid forward price for the currency as:

$$F_{0,T}^{P} = x_0 e^{-r_f T} \tag{A.3}$$

Using equation A.3 above, we can rewrite the Black-Scholes model to a European call option on foreign exchange spot rates as:

$$C(x, K, \sigma, r, T, r_f) = xe^{-r_f T} N(d_1) - Ke^{-rT}(d_2) \tag{A.4}$$

where:

$$d_1 = \frac{ln(\frac{x}{K}) + (r - r_f - \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T}$$

Equation A.4 is known as the Garman-Kohlhagen model (McDonald 2014), after Garman and Kohlhagen (1983). Using the put-call-parity we can derive the price of a European put:

$$P(x, K, \sigma, r, T, r_f) = C(x, K, \sigma, r, T, r_f) + Ke^{-rT} - xe^{-r_f T} \tag{A.5}$$

From equation A.5, it is impossible to rearrange the Black-Scholes model for the volatility alone (McDonald 2014). Instead, it is possible when all other variables are known to observe the volatility from the model. This observed volatility is the implied volatility and for FX options implied volatility is a market-based measure of expected risk.

Econometric Models Theory - ARMA

The autoregressive process is a time series where the current value of a variable depends on its previous or lagged values. The equation for the autoregressive process of order p, the AR(p) process, can be written as a linear difference equation:

$$y_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-i} + \varepsilon_t \tag{A.6}$$

where:

$$\varepsilon_t \sim N(0, \sigma_t^2) \tag{A.7}$$

Which means that the residuals follow a normal distribution with a mean of 0 and variance σ_t^2, Another form of a time series process is the moving average process of order p, the MA(q) process, which is a linear combination of white noise processes so that y_t depends on current and previous values of a white noise disturbance term (Books, 2014). We derive the MA(q) process as:

$$y_t = \mu + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t \tag{A.8}$$

where μ is a constant, θ_i is the moving average coefficient, and ε_t is a white noise disturbance term from equation A.7. The moving average term accounts for the effect of a sudden shock in the mean equation. A time series can be a combination of an AR(p) and MA(q) process by letting the ε_t in equation A.7 be written as an MA(q) process. We obtain the model by combining the expressions for y_t from equations A.6 and A.8, which gives:

$$y_t = \mu + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t \tag{A.9}$$

This is known as an ARMA(p,q) process and is the expression for the mean equation for our model. When the values for p and q equal zero, we are back to the AR(p) and MA(q) processes, respectively. When the ARMA(p,q) model described in equation A.9 is used to forecast, we rewrite the equation like this:

$$f_{t,s} = \sum_{i=1}^{p} \phi_i f_{t,s-1} + \sum_{i=1}^{q} \theta_i \varepsilon_{t+s-j} \tag{A.10}$$

The ARMA model can be extended to an ARIMA model, when the I stand for integrated order. When a time series is integrated of order one, i.e. I(1), the variable in question is differenced one time. The difference of a variable is the change from y_{t-1} to y_t. The lagged value of a variable can be written as $y_{t-1} = Ly_t$. Using this we can formulate the difference of the variable in three ways: $\Delta y_t = (1-L)y_t = y_t - y_{t-1}$.

ARCH-LM

The ARCH-LM test can be expressed as an auxiliary test regression on the squared residuals of the ARCH(q) (See Engle 1982) and is examined by running the regression:

$$e_t^2 = \hat{\delta}_0 + \sum_{s=1}^{q} \hat{\delta}_s e_{t-s}^2 + v_t \tag{A11}$$

Then the regression R^2 is multiplied by the number of observations to obtain the test statistic. A Lagrange multiplier interpretation can be given the test statistic and is asymptotically distributed as a χ^2(p) random variable (Sjölander, 2010). Test results show that for all options, we can at any significance level reject the null hypothesis of no ARCH effects. We conclude that there is proof of autoregressive conditional heteroskedasticity in the squared residuals for all options. The test estimated a 0,0001 for a put option with one week to maturity and a delta of 10, and 0,0004 for the one-week option with a delta of 5. For all other options, the estimator was 0,0000, and the test firmly states the presence of ARCH effects, and GARCH models are the proper econometric framework for modeling the data.

White noise

A white noise time series process can be written as a stochastic difference equation given by:

$$x_t = u_t \sim IID\ (0, \sigma_t^2) \tag{A.12}$$

Where IID stands for independent and identically distributed, and the process is stationary in mean with a E(x_t) = 0 and a variance var(x_t) = σ_t^2.

Stationary

When modeling and forecasting econometric time series, the variable in question must be stationary. A variable need to fulfill three requirements to be weakly stationary, i.e., have a constant mean, a constant variance and a constant autocovariance structure (Brooks, 2014). Weak stationarity essentially means that the time series will return to its mean after a shock, and the effect of the shock will disappear with time. If the effect of shocks affects the time series in the long run, the series is non-stationary and could be trending away from its mean. The white noise series from equation A.12 is an example of a stationary time series.

Augmented Dickey-Fuller (ADF) Test

The augmented Dickey-Fuller test for unit root is a test to determine if the time series process follows a unit root, and therefore is non-stationary. A time series process needs to be stationary to avoid spurious regression, meaning the regression has a high R^2 and t-statistics that appear to be significant, but without any economic meaning (Enders, 2015). Consider the simple autoregressive process of order one, an AR(1) process, given by the equation:

$$y_t = \phi y_{t-1} + u_t \tag{A.13}$$

where u_t is required to be a white noise process. To determine if the process is stationary, the test $\phi = 1$ is tested against the alternative hypotheses that $\phi < 1$, formally expressed:

H_0: $\phi = 1$, a unit root process

H_1: $\phi < 1$, a stationary process

where the test is conducted on the first difference of equation XX, expressed as:

$$\Delta y_t = (\phi - 1) y_{t-1} + u_t \tag{A.14}$$

where $\phi - 1 = \psi$, which entails equation XX to be expressed as:

$$\Delta y_t = \psi y_{t-1} + u_t \tag{A.15}$$

with the test hypotheses:

H_0: $\psi = 0$, a unit root process

H_1: $\psi < 0$, a stationary process

The test statistic DF = $\frac{\hat{\psi}}{se(\hat{\psi})}$ where $\hat{\psi}$ is the estimated coefficient of ψ and se($\hat{\psi}$) = $\widehat{\sigma_{\psi}}$. Given a significance level α, H_0 is rejected if DF < $C_\alpha$where C_α is the critical value. The test is conducted as a t-test, and since the null hypothesis is unit root (non-stationarity), the test statistic does not follow a standard t-distribution. This causes the critical values to be interpreted more strictly, meaning they need to be sufficiently more negative than normal test statistics. This is known as the Dickey-Fuller test. To determine that the residuals follow a unit root process, the test is conducted at equation XX, that includes lags of the process to test for autocorrelations in the error term. The augmented Dickey-Fuller test is therefore the same test as above, applied on the model given by:

$$\Delta y_t = \mu + \psi y_{t-1} + \sum_{i=1}^{p} \alpha_i \Delta y_{t-i} + \lambda t + u_t \tag{A.16}$$

Phillips-Perron Test

The Phillips-Perron, introduced by Phillips and Perron in 1988, test for unit root are a conducted as a regression of y_t as expressed in the following equation:

$$y_t = \alpha + \rho y_{t-1} + \varepsilon_t \tag{A.17}$$

Where ρ works as a correction term for serial correlation and heteroskedasticity. The correction term ρ is non-parametric which makes it robust to the presence of serial correlation and heteroskedasticity. The test is done by a t-test, where the hypothesis is, like the ADF, expressed:

H_0: $\rho = 1$

H_1: $\rho < 1$

The main differences between the ADF and PP test are that the PP test does not need to be specified with the number of lags in the regression, and according to Wang and Tomek

(2004) the estimated coefficients from the regression are modified to Z statistics, which are referred to as Dickey-Fuller critical values.

Diebold-Mariano Test

The Diebold-Mariano test measures if a forecasted time series is significantly better than another, measured from the forecasted MSE.

First compute the loss-differential: $d_i = e_i^2 - r_i^2$ where e_i^2 is the error measure from the test model and ri2 from the reference model. Calculate average loss-differential $\bar{d} = \frac{1}{n}\sum_{i=1}^{n} d_i$, μ=E[d_i]. For each n > k ≥ 1, we define: $\gamma_k = \frac{1}{n}\sum_{i=k+1}^{n}(d_i - \bar{d})(d_{i-k} - \bar{d})$

For each h ≥ 1 the DM test statistic is:

DM = $\frac{\bar{d}}{\sqrt{\gamma_{0+2}\sum_{k=1}^{h-1}\gamma_k / n}}$

Generally, it is sufficient that h = $n^{1/3} + 1$. Under the null hypothesis μ = 0 the DM test statistic follow a normal distribution DM ~N(0,1) where there is a significant difference between the two forecasts if DM > Z_{crit} where Z_{crit} is the two-tailed critical value for the normal distribution (Zaiontz, 2022).

7.2 Appendix B: Descriptive Statistics

In this section we present descriptive statistics that isn't included but referred to in the main paper.

Table B.1 Descriptive statistics of implied volatility for options with one month to maturity

	Put 5	*Put 10*	*Put 18*	*Put 25*	*Put 35*	*Put 50*	*Call 35*	*Call 25*	*Call 18*	*Call 10*	*Call 5*
Obs	2994	2994	2994	2994	2994	2994	2994	2994	2994	2994	2994
Mean	10,78	10,35	9,93	9,67	9,40	9,16	9,08	9,10	9,18	9,33	9,54
Min	3,74	3,69	3,67	3,66	3,69	3,77	3,89	4,01	4,12	4,20	4,29
25 %	7,39	7,18	6,99	6,87	6,73	6,64	6,66	6,71	6,78	6,87	7,02
50 %	9,70	9,36	9,04	8,82	8,57	8,38	8,28	8,32	8,39	8,52	8,68
75 %	13,07	12,56	12,00	11,68	11,28	10,96	10,79	10,76	10,78	10,88	11,05
Maks	35,12	33,51	32,01	31,02	29,94	28,88	28,25	28,19	28,57	29,38	31,04
Var	21,04	18,48	16,23	14,89	13,63	12,58	12,03	12,02	12,32	12,94	14,24
σ	4,59	4,30	4,03	3,86	3,69	3,55	3,47	3,47	3,51	3,60	3,77

Table B.1 *descriptive statistics of implied volatility for options with one month to maturity. Put with delta 50 is ATM and put and call options become increasingly OTM as the delta value decreases. Put 5 indicates a put option with an option delta of 5. Different quantiles measure the level of the implied volatility throughout the data sample.*

Table B.2 Descriptive statistics of implied volatility for options with three months to maturity

	Put 5	*Put 10*	*Put 18*	*Put 25*	*Put 35*	*Put 50*	*Call 35*	*Call 25*	*Call 18*	*Call 10*	*Call 5*
Obs	2994	2994	2994	2994	2994	2994	2994	2994	2994	2994	2994
Mean	11,60	11,01	10,39	10,01	9,62	9,29	9,16	9,18	9,28	9,52	9,76
Min	4,18	4,10	4,05	4,03	4,05	4,14	4,33	4,50	4,64	4,72	4,81
25 %	7,90	7,57	7,23	7,05	6,89	6,75	6,77	6,88	6,97	7,14	7,30
50 %	10,74	10,20	9,63	9,27	8,88	8,51	8,37	8,37	8,48	8,66	8,87
75 %	14,36	13,61	12,82	12,30	11,80	11,24	10,98	10,87	10,89	11,02	11,24
Maks	30,28	28,43	26,82	25,76	24,76	24,65	25,05	25,74	26,58	28,23	29,82
Var	21,61	18,69	15,69	14,04	12,51	11,26	10,69	10,68	11,05	12,03	13,30
σ	4,65	4,32	3,96	3,75	3,54	3,36	3,27	3,27	3,32	3,47	3,65

Table B.2 *Descriptive statistics of implied volatility for options with three months to maturity. Put with delta 50 is ATM and put and call options become increasingly OTM as the delta value decreases. Put 5 indicates a put option with an option delta of 5. Different quantiles measure the level of the implied volatility throughout the data sample.*

Table B.3 Descriptive statistics of implied volatility for options with six months to maturity

	Put 5	*Put 10*	*Put 18*	*Put 25*	*Put 35*	*Put 50*	*Call 35*	*Call 25*	*Call 18*	*Call 10*
Obs	2994	2994	2994	2994	2994	2994	2994	2994	2994	2994
Mean	12,25	11,53	10,74	10,29	9,82	9,43	9,28	9,30	9,44	9,74
Min	4,60	4,48	4,38	4,34	4,34	4,42	4,63	4,84	5,02	5,32
25 %	8,48	8,02	7,57	7,31	7,11	6,96	6,94	7,05	7,19	7,40
50 %	11,44	10,72	10,00	9,56	9,07	8,63	8,42	8,39	8,50	8,75
75 %	15,35	14,44	13,41	12,77	12,09	11,44	11,04	10,89	11,02	11,28
Maks	26,92	25,50	23,83	23,08	22,53	22,29	22,63	23,31	24,17	25,86
Var	21,48	18,45	15,04	13,28	11,61	10,28	9,69	9,67	10,04	11,16
σ	4,63	4,30	3,88	3,64	3,41	3,21	3,11	3,11	3,17	3,34

Table B.3 *Descriptive statistics of implied volatility for options with six months to maturity. Put with delta 50 is ATM and put and call options become increasingly OTM as the delta value decreases. Put 5 indicates a put option with an option delta of 5. Different quantiles measure the level of the implied volatility throughout the data sample.*

Figure B.1 Implied volatility for one month, three and six months

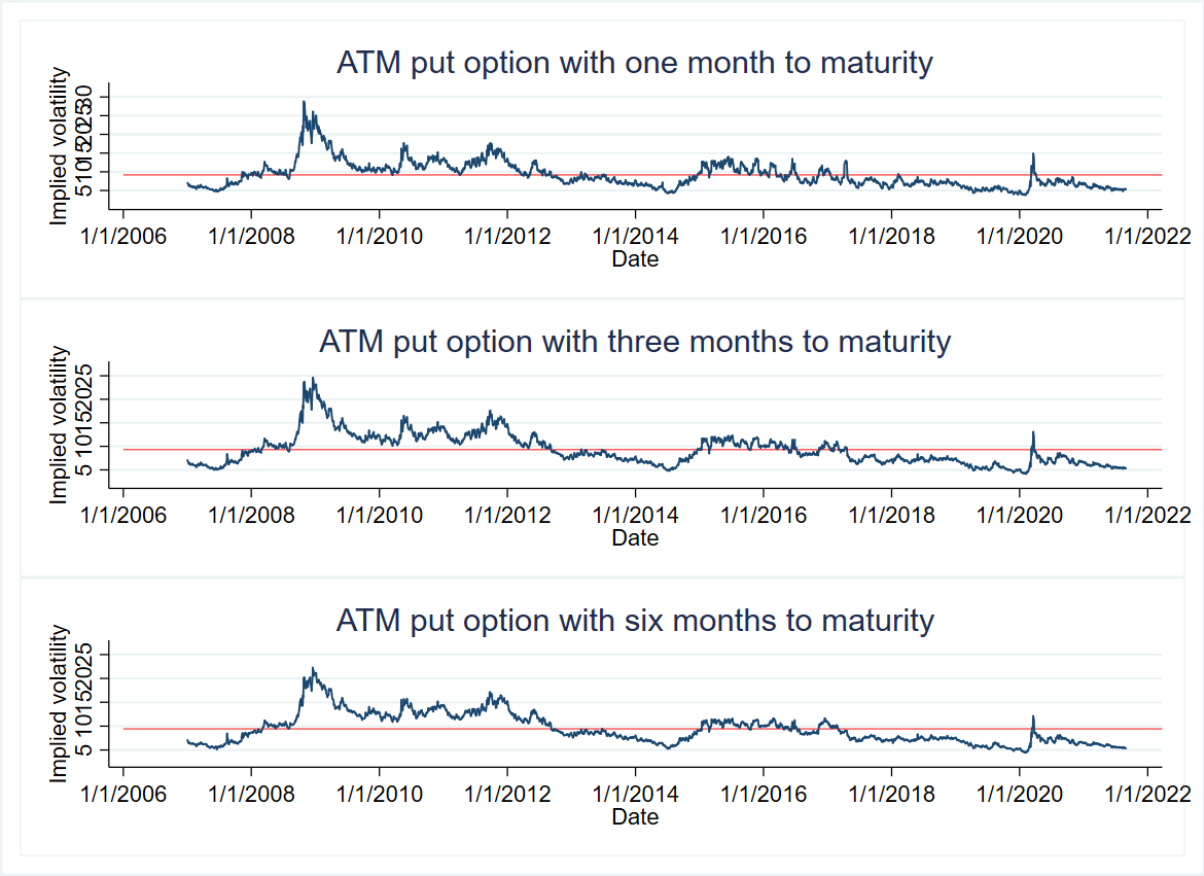


Figure B.1 Time series for the spot rates of implied volatility for ATM put options with one, three and six months to maturity from 2. January 2007 until 31. August 2021. The red horizontal line exhibits the options mean value for the sample period

Figure B.2 Plot of the 25 delta risk reversal for all maturities, an illustration of implied skewness over the sample.

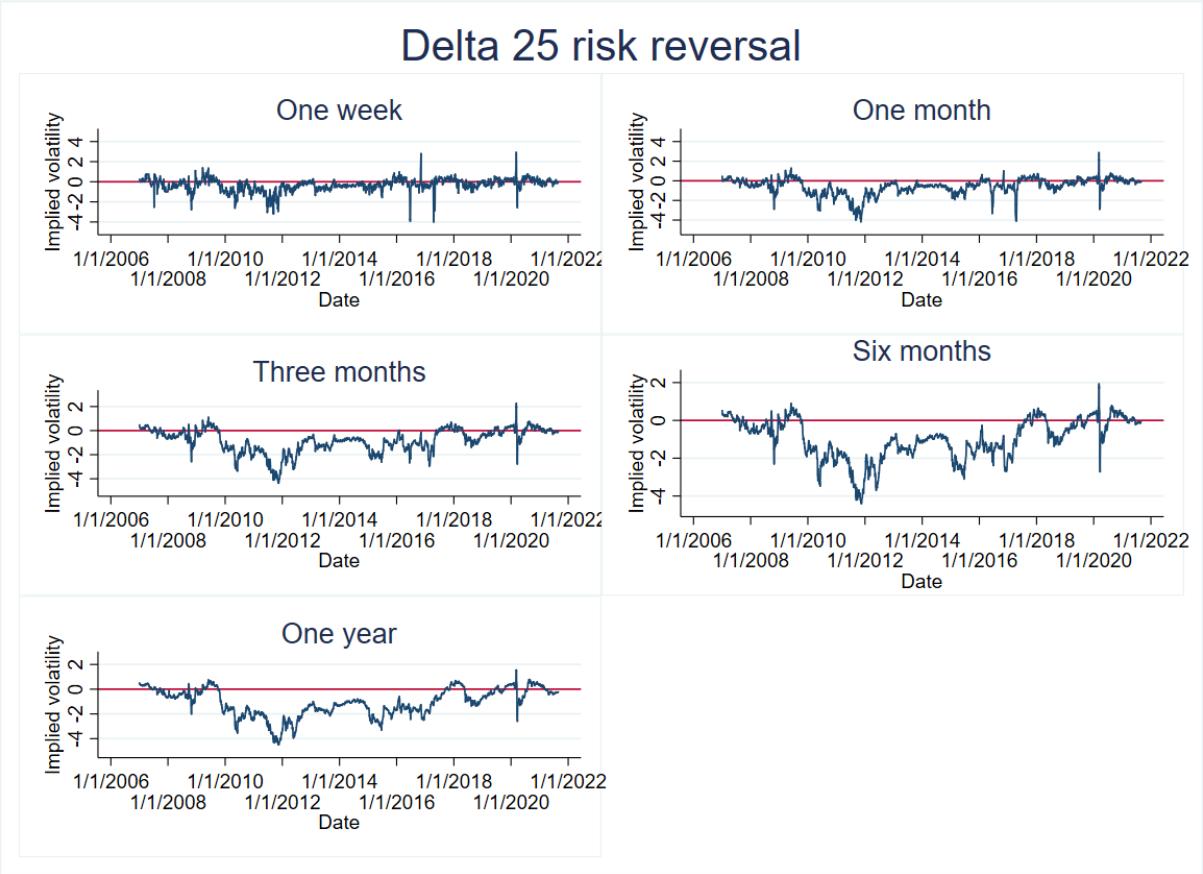


Figure B.2 Plot of the risk reversal for the 25 delta option for each maturity. The red line highlights zero, where implied volatility is equal for OTM call and put options. Recall that risk reversal is RR = IV_{call} - IV_{put} and indicate that put options mostly have a higher implied volatility compared to the equivalent call option.

Figure B.3 Daily returns and log daily returns for ATM options for each maturity

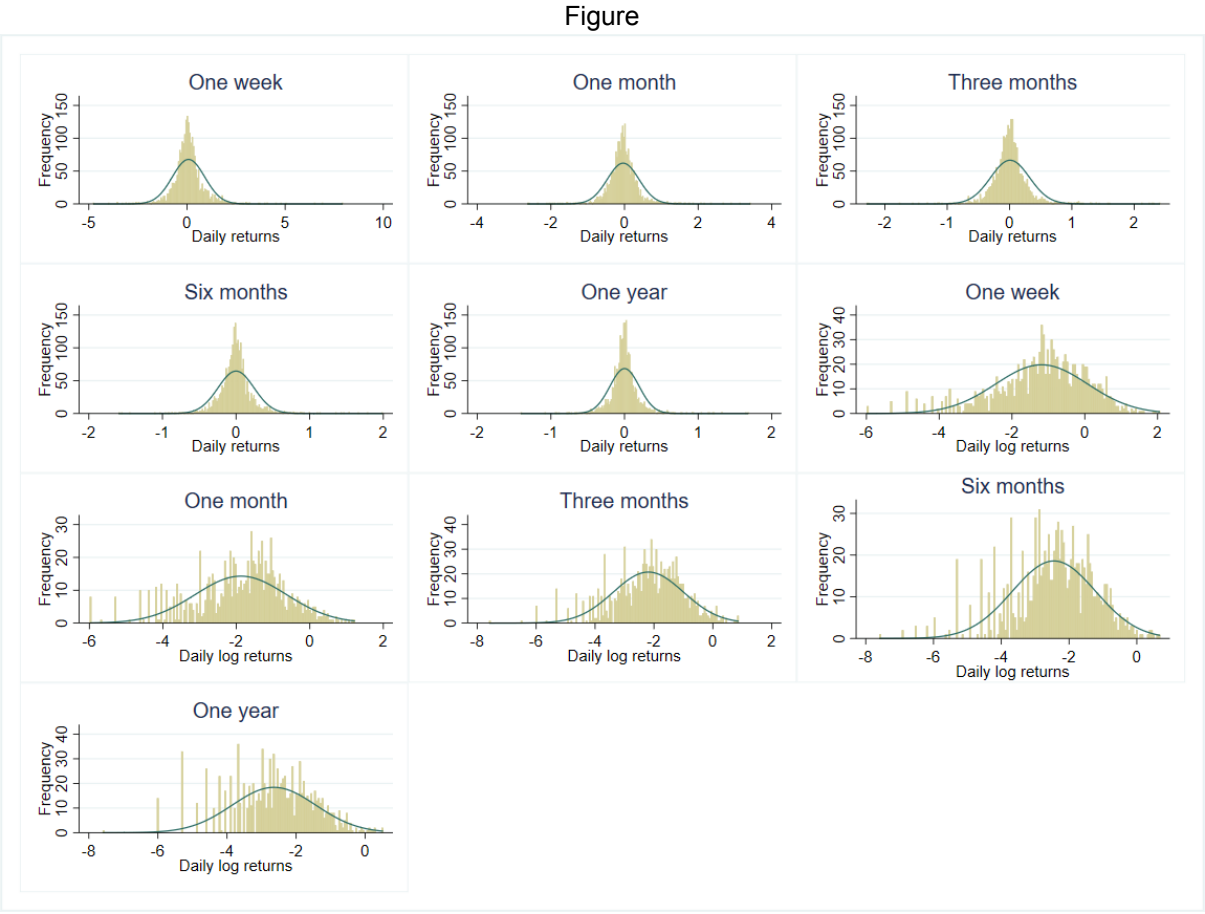


Figure B.3 Daily returns and daily log returns for the ATM option for each maturity. The label indicates option maturity, vertical axis the frequency of returns and horizontal axis amplitude of returns.

7.3 Appendix C: Test Results LSTM and Benchmark Models

In this section the forecast accuracy for LSTM compared to the benchmark RF model and benchmark AR(1)-GARCH(1,1) is reported, measured in MSE, RMSE and MAE.

Table C.1 Forecast accuracy for one week to maturity options.

	LSTM			**AR(1)-GARCH(1,1)**			**Random forest**		
	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE
Put 5	0,6362	0,7976	**0,4974**	**0,6262**	0,7913	0,5056	0,8277	0,9098	0,5973
Put 10	**0,5830**	**0,7635**	**0,4852**	**0,5830**	0,7635	0,4933	0,6864	0,8285	0,5628
Put 18	**0,5332**	**0,7302**	**0,4666**	0,5422	0,7364	0,4826	0,7395	0,8599	0,5509
Put 25	**0,5073**	**0,7122**	**0,4604**	0,5211	0,7218	0,4767	0,6811	0,8253	0,5497
Put 35	**0,4852**	**0,6966**	0,4786	0,4995	0,7067	**0,4692**	0,5813	0,7624	0,5144
Put 50	**0,4763**	**0,6901**	**0,4475**	0,4895	0,6997	0,4632	0,6944	0,8333	0,5492
Call 35	**0,4869**	**0,6978**	**0,4490**	0,4980	0,7057	0,4609	0,6128	0,7828	0,5249
Call 25	**0,5079**	**0,7127**	**0,4579**	0,5155	0,7180	0,4632	0,6269	0,7918	0,5221
Call 18	**0,5209**	**0,7217**	**0,4327**	0,5322	0,7295	0,4652	0,5660	0,7523	0,5158
Call 10	**0,5500**	**0,7416**	**0,4423**	0,5605	0,7486	0,4714	0,6655	0,8158	0,5445
Call 5	**0,5917**	**0,7692**	**0,4707**	0,5947	0,7712	0,4813	0,7820	0,8843	0,5583

Table C.1 One week to maturity forecast accuracy for LSTM and the two benchmark models. The first column indicates option level of moneyness. Put 50 is ATM option and put/call 5 is furthest OTM. The three next columns is reported accuracy for LSTM model, the mid columns the AR(1)-GARCH(1,1) and the three right columns is reported forecast accuracy for RF model.

Table C.2 Forecast accuracy for one month to maturity options.

	LSTM			**AR(1)-GARCH(1,1)**			**Random forest**		
	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE
Put 5	0,2645	0,5143	0,2767	**0,2561**	**0,5060**	**0,2729**	0,2670	0,5167	0,3055
Put 10	0,2310	0,4806	**0,2606**	**0,2236**	**0,4728**	0,2610	0,2557	0,5057	0,2876
Put 18	0,1971	0,4440	**0,2479**	**0,1893**	**0,4351**	**0,2479**	0,1961	0,4428	0,2782
Put 25	0,1771	0,4208	**0,2394**	**0,1709**	**0,4134**	0,2397	0,1978	0,4447	0,2694
Put 35	0,1584	0,3980	0,2316	**0,1555**	**0,3944**	0,2324	0,1683	0,4102	0,2558
Put 50	0,1537	0,3920	**0,2308**	**0,1500**	**0,3873**	0,2709	0,1842	0,4292	0,2640
Call 35	0,1595	0,3994	0,2300	**0,1582**	**0,3978**	**0,2270**	0,1730	0,4159	0,2498
Call 25	**0,1736**	**0,4167**	**0,2292**	***0,1737***	***0,4167***	0,2300	0,1996	0,4468	0,2602
Call 18	**0,1899**	**0,4358**	**0,2302**	0,1909	0,4369	0,2340	0,2272	0,4767	0,2650
Call 10	**0,2181**	**0,4670**	**0,2361**	0,2215	0,4706	0,2415	0,2576	0,5075	0,2731
Call 5	***0,2519***	***0,5019***	**0,2513**	***0,2519***	***0,5019***	0,2518	0,2806	0,5297	0,2782

Table C.2 One month to maturity forecast accuracy for LSTM and the two benchmark models. The first column indicates option level of moneyness. Put 50 is ATM option and put/call 5 is furthest OTM. The three next columns is reported accuracy for LSTM model, the mid columns the AR(1)-GARCH(1,1) and the three right columns is reported forecast accuracy for RF model.

Table C.3 Forecast accuracy for three months to maturity options.

	LSTM			AR(1)-GARCH(1,1)			Random forest		
	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE
Put 5	0,1792	0,4233	0,2056	**0,1706**	0,4131	**0,2002**	0,1777	0,4215	0,2346
Put 10	0,1484	0,3852	0,1906	**0,1421**	0,3769	**0,1888**	0,1626	0,4032	0,2116
Put 18	0,1177	0,3431	0,1800	**0,1127**	0,3357	**0,1758**	0,1284	0,3583	0,2016
Put 25	0,1045	0,3233	0,1802	**0,0975**	0,3123	**0,1685**	0,1159	0,3404	0,1921
Put 35	0,0903	0,3005	0,1660	**0,0845**	0,2907	**0,1604**	0,1120	0,3347	0,1889
Put 50	0,0850	0,2915	0,1610	**0,0773**	0,2781	**0,1540**	0,0888	0,2980	0,1757
Call 35	0,0825	0,2872	0,1805	**0,0795**	0,2820	**0,1532**	0,0900	0,3000	0,1727
Call 25	0,0921	0,3035	0,1688	**0,0872**	0,2953	**0,1556**	0,1049	0,3239	0,1838
Call 18	0,0991	0,3148	0,1601	**0,0974**	0,3122	**0,1595**	0,1111	0,3333	0,1815
Call 10	0,1240	0,3521	0,1746	**0,1185**	0,3442	**0,1679**	0,1370	0,3701	0,1938
Call 5	0,1413	0,3759	0,1886	**0,1387**	0,3725	**0,1772**	0,1642	0,4052	0,1999

Table C.3 Three months to maturity forecast accuracy for LSTM and the two benchmark models. The first column indicates option level of moneyness. Put 50 is ATM option and put/call 5 is furthest OTM. The three next columns is reported accuracy for LSTM model, the mid columns the AR(1)-GARCH(1,1) and the three right columns is reported forecast accuracy for RF model.

Table C.4 Forecast accuracy for six months to maturity options.

	LSTM			AR(1)-GARCH(1,1)			Random forest		
	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE
Put 5	0,1392	0,3731	0,1679	**0,1324**	0,3639	**0,1641**	0,1443	0,3799	0,1963
Put 10	0,1169	0,3419	0,1585	**0,1072**	0,3274	**0,1523**	0,1282	0,3581	0,1814
Put 18	0,0887	0,2978	0,1493	**0,0805**	0,2837	**0,1392**	0,0932	0,3053	0,1673
Put 25	0,0738	0,2717	0,1436	**0,0674**	0,2596	**0,1317**	0,0810	0,2846	0,1571
Put 35	0,0620	0,2490	0,1337	**0,0564**	0,2375	**0,1244**	0,0675	0,2598	0,1490
Put 50	0,0550	0,2345	0,1269	**0,0506**	0,2248	**0,1191**	0,0634	0,2518	0,1445
Call 35	0,0557	0,2360	0,1249	**0,0518**	0,2276	**0,1182**	0,0597	0,2443	0,1365
Call 25	0,0608	0,2466	0,1350	**0,0572**	0,2392	**0,1206**	0,0656	0,2561	0,1429
Call 18	0,0699	0,2644	0,1358	**0,0649**	0,2548	**0,1243**	0,0747	0,2733	0,1422
Call 10	0,0860	0,2933	0,1361	**0,0815**	0,2855	**0,1337**	0,0925	0,3041	0,1534
Call 5	0,1022	0,3197	0,1449	**0,0975**	0,3123	**0,1433**	0,1137	0,3372	0,1649

Table C.4 Six months to maturity forecast accuracy for LSTM and the two benchmark models. The first column indicates option level of moneyness. Put 50 is ATM option and put/call 5 is furthest OTM. The three next columns is reported accuracy for LSTM model, the mid columns the AR(1)-GARCH(1,1) and the three right columns is reported forecast accuracy for RF model.

Table C.5 Forecast accuracy for one year to maturity options.

	LSTM			AR(1)-GARCH(1,1)			Random forest		
	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE
Put 5	0,1215	0,3486	0,1436	**0,1135**	0,3369	**0,1421**	0,1399	0,3740	0,1714
Put 10	0,1006	0,3172	0,1443	**0,0899**	0,2998	**0,1302**	0,1012	0,3181	0,1578
Put 18	0,0722	0,2687	0,1260	**0,0641**	0,2531	**0,1160**	0,0813	0,2851	0,1429
Put 25	0,0582	0,2412	0,1201	**0,0520**	0,2281	**0,1087**	0,0676	0,2600	0,1334
Put 35	0,0468	0,2163	0,1083	**0,0421**	0,2052	**0,1021**	0,0473	0,2175	0,1211
Put 50	0,0398	0,1995	0,0992	**0,0367**	0,1916	**0,0972**	0,0404	0,2010	0,1124
Call 35	0,0390	0,1975	0,0969	**0,0372**	0,1928	**0,0961**	0,0420	0,2049	0,1105
Call 25	0,0451	0,2124	0,1032	**0,0413**	0,2031	**0,0973**	0,0454	0,2131	0,1155
Call 18	0,0516	0,2272	0,1038	**0,0477**	0,2183	**0,1008**	0,0524	0,2289	0,1148
Call 10	0,0674	0,2596	0,1144	**0,0631**	0,2512	**0,1104**	0,0719	0,2681	0,1315
Call 5	0,0815	0,2855	0,1232	**0,0770**	0,2776	**0,1195**	0,0841	0,2900	0,1359

Table C.5 One year to maturity forecast accuracy for LSTM and the two benchmark models. The first column indicates option level of moneyness. Put 50 is ATM option and put/call 5 is furthest OTM. The three next columns is reported accuracy for LSTM model, the mid columns the AR(1)-GARCH(1,1) and the three right columns is reported forecast accuracy for RF model.

7.4 Appendix D: Test Results Extensions to GARCH-type Models

Appendix D exhibits a summery and estimating results for the different ARMA-GARCH extensions, GJR-extensions, and t-distributed residual models. The other econometric model imposed in this study in addition to the benchmark AR(1)-GARCH(1,1) is:

- AR(1)-GARCH(1,1) - t-distributed residuals
- ARIMA(1,1,1)-GARCH(1.1)
- ARIMA(1,1,1)-GARCH(1,1) t-distributed residuals
- ARIMA(1,1,1)-GJR-GARCH(1,1)
- ARIMA(1,1,1)-GJR-GARCH(1,1) - t-distributed residuals

Table D.1 Forecast accuracy for ARIMA(1,1,1)-GARCH(1,1), one week to maturity options. The model specification is labelled above all tables. The three columns to the left show the forecast accuracy for the same model using t-distributed residuals.

		ARMA(1,1)-GARCH(1,1)		**ARMA(1,1)-GARCH(1,1)t-dist**		
	MSE	RMSE	MAE	MSE	RMSE	MAE
Put delta 5	0,6471	0,8044	0,4891	0,6126	0,7827	0,4904
Put delta 10	0,5922	0,7695	0,7489	0,5677	0,7535	0,4780
Put delta 18	0,5418	0,7361	0,4624	0,5271	0,7260	0,4657
Put delta 25	0,5148	0,7175	0,4545	0,5056	0,7111	0,4595
Put delta 35	0,4911	0,7008	0,4463	0,4830	0,6950	0,4507
Put delta 50	0,4796	0,6926	0,4401	0,4741	0,6885	0,4452
Call delta 35	0,4897	0,6998	0,4392	0,4843	0,6959	0,4443
Call delta 25	0,5097	0,7140	0,4427	0,5025	0,7088	0,4475
Call delta 18	0,5303	0,7282	0,4477	0,5207	0,7216	0,4513
Call delta 10	0,5677	0,7534	0,4583	0,5506	0,7420	0,4591
Call delta 5	0,6078	0,7796	0,4694	0,5835	0,7638	0,4694

Table D.2 Forecast accuracy for ARIMA(1,1,1)-GARCH(1,1), one month to maturity options. The model specification is labelled above all tables. The three columns to the left show the forecast accuracy for the same model using t-distributed residuals.

	ARMA(1,1)-GARCH(1,1)			ARMA(1,1)-GARCH(1,1)t-dist		
	MSE	RMSE	MAE	MSE	RMSE	MAE
Put delta 5	0,2559	0,5059	0,2701	0,2541	0,5041	0,2726
Put delta 10	0,2266	0,4760	0,2589	0,2225	0,4717	0,2604
Put delta 18	0,1892	0,4350	0,2451	0,1887	0,4344	0,2477
Put delta 25	0,1724	0,4152	0,2380	0,1705	0,4130	0,2397
Put delta 35	0,1564	0,3954	0,2305	0,1552	0,3939	0,2318
Put delta 50	0,1497	0,3869	0,2248	0,1490	0,3860	0,2262
Call delta 35	0,1576	0,3969	0,2238	0,1571	0,3963	0,2258
Call delta 25	0,1726	0,4154	0,2262	0,1720	0,4147	0,2286
Call delta 18	0,1894	0,4352	0,2301	0,1888	0,4345	0,2326
Call delta 10	0,2196	0,4687	0,2377	0,2176	0,4665	0,2395
Call delta 5	0,2488	0,4988	0,2472	0,2470	0,4970	0,2470

Table D.3 Forecast accuracy for ARIMA(1,1,1)-GARCH(1,1), three months to maturity options. The model specification is labelled above all tables. The three columns to the left show the forecast accuracy for the same model using t-distributed residuals.

	ARMA(1,1)-GARCH(1,1)			ARMA(1,1)-GARCH(1,1)t-dist		
	MSE	RMSE	MAE	MSE	RMSE	MAE
Put delta 5	0,1696	0,4118	0,2002	0,1714	0,4140	0,2005
Put delta 10	0,2495	0,4995	0,3515	0,1892	0,4350	0,2822
Put delta 18	0,1927	0,4390	0,3118	0,1159	0,3405	0,1761
Put delta 25	0,0993	0,3152	0,1683	0,0996	0,3156	0,1682
Put delta 35	0,0855	0,2924	0,1601	0,0858	0,2928	0,1602
Put delta 50	0,0778	0,2788	0,1537	0,0779	0,2792	0,1537
Call delta 35	0,0797	0,2824	0,1526	0,0979	0,3129	0,2040
Call delta 25	0,0874	0,2957	0,1551	0,0875	0,2959	0,1552
Call delta 18	0,0977	0,3126	0,1592	0,0979	0,3128	0,1593
Call delta 10	0,1190	0,3449	0,1677	0,1193	0,3454	0,1682
Call delta 5	0,1393	0,3732	0,1767	0,1397	0,3737	0,1773

Table D.4 Forecast accuracy for ARIMA(1,1,1)-GARCH(1,1), six months to maturity options. The model specification is labelled above all tables. The three columns to the left show the forecast accuracy for the same model using t-distributed residuals.

	ARMA(1,1)-GARCH(1,1)			ARMA(1,1)-GARCH(1,1)t-dist		
	MSE	RMSE	MAE	MSE	RMSE	MAE
Put delta 5	0,1327	0,3642	0,1640	0,1361	0,3689	0,1656
Put delta 10	0,1075	0,3278	0,1522	0,1098	0,3314	0,1533
Put delta 18	0,1319	0,3632	0,2496	0,0818	0,2859	0,1397
Put delta 25	0,0675	0,2598	0,1317	0,0694	0,2634	0,1325
Put delta 35	0,0944	0,3073	0,2178	0,0577	0,2402	0,1250
Put delta 50	0,0510	0,2259	0,1195	0,0512	0,2264	0,1195
Call delta 35	0,0693	0,2632	0,1736	0,0522	0,2285	0,1183
Call delta 25	0,0725	0,2693	0,1708	0,0576	0,2401	0,1207
Call delta 18	0,0652	0,2553	0,1243	0,0654	0,2558	0,1244
Call delta 10	0,0955	0,3090	0,1804	0,0851	0,2918	0,1520
Call delta 5	0,0974	0,3121	0,1434	0,0981	0,3132	0,1440

Table D.5 Forecast accuracy for ARIMA(1,1,1)-GARCH(1,1), one year to maturity options. The model specification is labelled above all tables. The three columns to the left show the forecast accuracy for the same model using t-distributed residuals.

	ARMA(1,1)-GARCH(1,1)			ARMA(1,1)-GARCH(1,1)t-dist		
	MSE	RMSE	MAE	MSE	RMSE	ABE
Put delta 5	0,1136	0,3371	0,1421	0,1166	0,3415	0,1433
Put delta 10	0,0902	0,3004	0,1300	0,0921	0,3034	0,1312
Put delta 18	0,0642	0,2534	0,1160	0,0652	0,2553	0,1168
Put delta 25	0,0522	0,2284	0,1087	0,0525	0,2292	0,1093
Put delta 35	0,0421	0,2052	0,1021	0,0423	0,2056	0,1021
Put delta 50	0,0366	0,1913	0,0972	0,0367	0,1915	0,0968
Call delta 35	0,0368	0,1919	0,0959	0,0376	0,1938	0,0958
Call delta 25	0,0410	0,2024	0,0972	0,0417	0,2041	0,0974
Call delta 18	0,0474	0,2177	0,1007	0,0481	0,2194	0,1005
Call delta 10	0,0630	0,2510	0,1105	0,0633	0,2516	0,1103
Call delta 5	0,0771	0,2776	0,1195	0,0773	0,2781	0,1195

Table D.6 Forecast accuracy for ARIMA(1,1,1)-GJR-GARCH(1,1), one week to maturity options. The model specification is labelled above all tables. The three columns to the left show the forecast accuracy for the same model using t-distributed residuals.

	ARMA(1,1)-GJR-GARCH(1,1)			**ARMA(1,1)-GJR-GARCH(1,1) with t-dist**		
	MSE	RMSE	MAE	MSE	RMSE	MAE
Put delta 5	0,6323	0,7951	0,4880	0,6277	0,7923	0,4853
Put delta 10	0,5798	0,7614	0,4736	0,5776	0,7600	0,4716
Put delta 18	0,5357	0,7319	0,4684	0,5343	0,7310	0,4647
Put delta 25	0,5135	0,7166	0,4567	0,5061	0,7114	0,4514
Put delta 35	0,4895	0,6997	0,4510	0,4837	0,6955	0,4435
Put delta 50	0,4817	0,6940	0,4464	0,4730	0,6878	0,4372
Call delta 35	0,4926	0,7019	0,4455	0,4828	0,6948	0,4357
Call delta 25	0,5031	0,7093	0,4423	0,5013	0,7080	0,4386
Call delta 18	0,5224	0,7228	0,4472	0,5207	0,7216	0,4433
Call delta 10	0,5586	0,7474	0,4575	0,5546	0,7447	0,4525
Call delta 10	0,5985	0,7736	0,4699	0,5916	0,7692	0,4624

Table D.7 Forecast accuracy for ARIMA(1,1,1)-GJR-GARCH(1,1), one month to maturity options. The model specification is labelled above all tables. The three columns to the left show the forecast accuracy for the same model using t-distributed residuals.

	ARMA(1,1)-GJR-GARCH(1,1)			**ARMA(1,1)-GJR-GARCH(1,1) with t-dist**		
	MSE	RMSE	MAE	MSE	RMSE	MAE
Put delta 5	0,2541	0,5041	0,2738	0,2574	0,5073	0,2786
Put delta 10	0,2222	0,4714	0,2618	0,2239	0,4732	0,2595
Put delta 18	0,1897	0,4355	0,2484	0,1904	0,4363	0,2466
Put delta 25	0,1705	0,4129	0,2381	0,1730	0,4160	0,2382
Put delta 35	0,1564	0,3955	0,2317	0,1579	0,3974	0,2400
Put delta 50	0,1497	0,3870	0,2255	0,1502	0,3876	0,2243
Call delta 35	0,1575	0,3969	0,2246	0,1579	0,3973	0,2234
Call delta 25	0,1726	0,4154	0,2273	0,1727	0,4156	0,2261
Call delta 18	0,1894	0,4352	0,2311	0,1885	0,4341	0,2300
Call delta 10	0,2196	0,4686	0,2387	0,2193	0,4683	0,2380
Call delta 10	0,2488	0,4988	0,2484	0,2479	0,4979	0,2470

Table D.8 Forecast accuracy for ARIMA(1,1,1)-GJR-GARCH(1,1), three months to maturity options. The model specification is labelled above all tables. The three columns to the left show the forecast accuracy for the same model using t-distributed residuals.

	ARMA(1,1)-GJR-GARCH(1,1)			**ARMA(1,1)-GJR-GARCH(1,1) with t-dist**		
	MSE	RMSE	MAE	MSE	RMSE	MAE
Put delta 5	0,1679	0,4097	0,2006	0,1707	0,4131	0,2004
Put delta 10	0,1401	0,3742	0,1890	0,1419	0,3767	0,1889
Put delta 18	0,1110	0,3331	0,1760	0,1120	0,3346	0,1757
Put delta 25	0,1333	0,3650	0,2471	0,0967	0,3110	0,1683
Put delta 35	0,0834	0,2887	0,1607	0,0838	0,2895	0,1603
Put delta 50	0,0775	0,2784	0,1540	0,0767	0,2770	0,1540
Call delta 35	0,0795	0,2820	0,1530	0,0791	0,2813	0,1532
Call delta 25	0,0865	0,2941	0,1553	0,0874	0,2956	0,1552
Call delta 18	0,0973	0,3120	0,1603	0,0973	0,3119	0,1597
Call delta 10	0,1186	0,3444	0,1679	0,1174	0,3426	0,1677
Call delta 10	0,1389	0,3727	0,1772	0,1393	0,3732	0,1773

Table D.9 Forecast accuracy for ARIMA(1,1,1)-GJR-GARCH(1,1), six months to maturity options. The model specification is labelled above all tables. The three columns to the left show the forecast accuracy for the same model using t-distributed residuals.

	ARMA(1,1)-GJR-GARCH(1,1)			**ARMA(1,1)-GJR-GARCH(1,1) with t-dist**		
	MSE	RMSE	MAE	MSE	RMSE	MAE
Put delta 5	0,1327	0,3643	0,1646	0,1357	0,3684	0,1656
Put delta 10	0,1074	0,3278	0,1527	0,1095	0,3309	0,1532
Put delta 18	0,0807	0,2840	0,1396	0,0955	0,3090	0,1845
Put delta 25	0,0675	0,2597	0,1319	0,0679	0,2606	0,1320
Put delta 35	0,0560	0,2366	0,1248	0,0566	0,2379	0,1246
Put delta 50	0,0510	0,2258	0,1196	0,0506	0,2249	0,1192
Call delta 35	0,0644	0,2537	0,1620	0,0628	0,2506	0,1586
Call delta 25	0,0571	0,2389	0,1218	0,0571	0,2389	0,1205
Call delta 18	0,0650	0,2550	0,1248	0,0725	0,2693	0,1557
Call delta 10	0,0813	0,2851	0,1342	0,0847	0,2911	0,1505
Call delta 10	0,0973	0,3120	0,1437	0,0981	0,3132	0,1441

Table D.10 Forecast accuracy for ARIMA(1,1,1)-GJR-GARCH(1,1), one year to maturity options. The model specification is labelled above all tables. The three columns to the left show the forecast accuracy for the same model using t-distributed residuals.

	ARMA(1,1)-GJR-GARCH(1,1)			**ARMA(1,1)-GJR-GARCH(1,1) with t-dist**		
	MSE	RMSE	MAE	MSE	RMSE	MAE
Put delta 5	0,1138	0,3374	0,1423	0,1166	0,3414	0,1434
Put delta 10	0,0902	0,3004	0,1302	0,0920	0,3033	0,1312
Put delta 18	0,0643	0,2537	0,1162	0,0675	0,2598	0,1174
Put delta 25	0,0522	0,2285	0,1089	0,0525	0,2292	0,1093
Put delta 35	0,0416	0,2040	0,1023	0,0423	0,2056	0,1021
Put delta 50	0,0362	0,1902	0,0972	0,0366	0,1914	0,0968
Call delta 35	0,0367	0,1917	0,0960	0,0370	0,1925	0,0958
Call delta 25	0,0409	0,2022	0,0974	0,0416	0,2040	0,0974
Call delta 18	0,0473	0,2175	0,1010	0,0481	0,2193	0,1005
Call delta 10	0,0631	0,2512	0,1107	0,0632	0,2515	0,1104
Call delta 10	0,0770	0,2775	0,1197	0,0773	0,2780	0,1196

Table D.11 Forecast accuracy for AR(1)-GARCH(1,1) with t-distributed residuals for the three shortest maturities.

	One week to maturity			**One month to maturity**			**Three months to maturity**		
	MSE	RMSE	MAE	MSE	RMSE	MAE	MSE	RMSE	MAE
Put delta 5	0,6221	0,7888	0,4899	0,2559	0,5059	0,2726	0,1706	0,4131	0,2002
Put delta 10	0,5766	0,7593	0,4775	0,2239	0,4732	0,2605	0,1421	0,3769	0,1888
Put delta 18	0,5358	0,7320	0,4642	0,1907	0,4367	0,2474	0,1127	0,3357	0,1758
Put delta 25	0,5129	0,7162	0,4566	0,1725	0,4154	0,2393	0,0975	0,3123	0,1685
Put delta 35	0,4918	0,7013	0,4491	0,1570	0,3963	0,2314	0,0845	0,2907	0,1604
Put delta 50	0,4819	0,6942	0,4441	0,1507	0,3882	0,2254	0,0773	0,2781	0,1540
Call delta 35	0,4915	0,7011	0,4430	0,1585	0,3982	0,2245	0,0795	0,2820	0,1532
Call delta 25	0,5091	0,7135	0,4457	0,1735	0,4165	0,2271	0,0872	0,2953	0,1556
Call delta 18	0,5271	0,7260	0,4499	0,1900	0,4359	0,2306	0,0974	0,3122	0,1595
Call delta 10	0,5572	0,7465	0,4567	0,2196	0,4686	0,2384	0,1185	0,3442	0,1679
Call delta 10	0,5904	0,7683	0,4658	0,2480	0,4980	0,2473	0,1387	0,3725	0,1772

Table D.12 Forecast accuracy for AR(1)-GARCH(1,1) with t-distributed residuals for the two longest maturities

	Six months to maturity			One year to maturity		
	MSE	RMSE	MAE	MSE	RMSE	MAE
Put delta 5	0,1324	0,3639	0,1641	0,1135	0,3369	0,1421
Put delta 10	0,1072	0,3274	0,1523	0,0899	0,2998	0,1302
Put delta 18	0,0805	0,2837	0,1392	0,0641	0,2531	0,1160
Put delta 25	0,0674	0,2596	0,1317	0,0520	0,2281	0,1087
Put delta 35	0,0564	0,2375	0,1244	0,0421	0,2052	0,1021
Put delta 50	0,0506	0,2248	0,1191	0,0367	0,1916	0,0972
Call delta 35	0,0518	0,2276	0,1182	0,0372	0,1928	0,0961
Call delta 25	0,0572	0,2392	0,1206	0,0413	0,2031	0,0973
Call delta 18	0,0649	0,2548	0,1243	0,0477	0,2183	0,1008
Call delta 10	0,0815	0,2855	0,1337	0,0631	0,2512	0,1104
Call delta 10	0,0975	0,3123	0,1433	0,0770	0,2776	0,1195

7.5 Appendix E: ADF, PP and Diebold-Mariano Test Results

In Appendix E test results from augmented Dickey-Fuller test, Phillips-Perron and Diebold-Mariano tests are reported.

Table E.1 ADF and PP test for options with one week to maturity.

	ADF Test statistic		**Phillips-Perron** Test statistic		
	Z(t)	p-value	Z(rho)	Z(t)	p-value
Put delta 5	-6,968	0,0000	-64,197	-5,711	0,0000
Put delta 10	-7,158	0,0000	-67,347	-5,853	0,0000
Put delta 18	-7,231	0,0000	-68,025	-5,885	0,0000
Put delta 25	-7,307	0,0000	-69,023	-5,930	0,0000
Put delta 35	-7,353	0,0000	-69,442	-5,950	0,0000
Put delta 50	-7,326	0,0000	-68,608	-5,913	0,0000
Call delta 35	-7,287	0,0000	-67,974	-5,884	0,0000
Call delta 25	-7,214	0,0000	-66,779	-5,829	0,0000
Call delta 18	-7,084	0,0000	-64,685	-5,733	0,0000
Call delta 10	-6,945	0,0000	-62,622	-5,637	0,0000
Call delta 5	-6,684	0,0000	-58,145	-5,425	0,0000

Table E.2 ADF and PP test for options with one month to maturity.

	ADF Test statistic		**Phillips-Perron** Test statistic		
	Z(t)	p-value	Z(rho)	Z(t)	p-value
Put delta 5	-3,754	0,0034	-26,242	-3,606	0,0056
Put delta 10	-3,848	0,0025	-26,481	-3,625	0,0053
Put delta 18	-3,906	0,0020	-25,865	-3,580	0,0062
Put delta 25	-3,951	0,0017	-25,476	-3,553	0,0067
Put delta 35	-3,973	0,0016	-24,820	-3,505	0,0079
Put delta 50	-3,970	0,0016	-23,981	-3,443	0,0096
Call delta 35	-3,968	0,0016	-23,518	-3,408	0,0107
Call delta 25	-3,929	0,0018	-23,151	-3,379	0,0117
Call delta 18	-3,871	0,0023	-22,827	-3,353	0,0127
Call delta 10	-3,803	0,0029	-22,692	-3,343	0,0130
Call delta 5	-3,672	0,0045	-21,913	-3,282	0,0157

Table E.3 ADF and PP test for options with three months to maturity.

	ADF		**Phillips-Perron**		
	Test statistic		Test statistic		
	Z(t)	p-value	Z(rho)	Z(t)	p-value
Put delta 5	-2,745	0,0666	-15,473	-2,753	0,0653
Put delta 10	-2,798	0,0586	-15,243	-2,732	0,0687
Put delta 18	-2,862	0,0500	-14,752	-2,683	0,0770
Put delta 25	-2,907	0,0445	-14,446	-2,654	0,0824
Put delta 35	-2,933	0,0417	-14,051	-2,613	0,0904
Put delta 50	-2,943	0,0406	-13,706	-2,576	0,0980
Call delta 35	-2,933	0,0416	-13,620	-2,565	0,1005
Call delta 25	-2,905	0,0448	-13,659	-2,566	0,1002
Call delta 18	-2,854	0,0510	-13,747	-2,573	0,0987
Call delta 10	-2,780	0,0612	-13,894	-2,588	0,0955
Call delta 5	-2,703	0,0736	-14,034	-2,601	0,0928

Table E4 ADF and PP test for options with six months to maturity.

	ADF		**Phillips-Perron**		
	Test statistic		Test statistic		
	Z(t)	p-value	Z(rho)	Z(t)	p-value
Put delta 5	-2,247	0,1898	-11,655	-2,386	0,1458
Put delta 10	-2,253	0,1875	-11,186	-2,333	0,1616
Put delta 18	-2,305	0,1704	-10,607	-2,264	0,1838
Put delta 25	-2,332	0,1618	-10,220	-2,218	0,1998
Put delta 35	-2,351	0,1561	-9,851	-2,170	0,2172
Put delta 50	-2,363	0,1525	-9,614	-2,137	0,2299
Call delta 35	-2,362	0,1527	-9,627	-2,135	0,2307
Call delta 25	-2,338	0,1599	-9,757	-2,147	0,2261
Call delta 18	-2,305	0,1702	-9,993	-2,174	0,2160
Call delta 10	-2,227	0,1965	-10,243	-2,205	0,2046
Call delta 5	-2,211	0,2024	-10,737	-2,261	0,1849

Table E.5 ADF and PP test for options with one year to maturity.

	ADF		**Phillips-Perron**		
	Test statistic		Test statistic		
	Z(t)	p-value	Z(rho)	Z(t)	p-value
Put delta 5	-1,998	0,2874	-9,618	-2,188	0,2107
Put delta 10	-1,964	0,3027	-9,040	-2,108	0,2413
Put delta 18	-1,982	0,2945	8,347	-2,016	0,2794
Put delta 25	-1,990	0,2911	-7,913	-1,954	0,3069
Put delta 35	-1,996	0,2882	-7,567	-1,901	0,3318
Put delta 50	-1,995	0,2888	-7,347	-1,861	0,3504
Call delta 35	-1,997	0,2880	-7,400	-1,862	0,3501
Call delta 25	-1,982	0,2947	-7,577	-1,882	0,3407
Call delta 18	-1,956	0,3062	-7,867	-1,920	0,3229
Call delta 10	-1,904	0,3302	-8,235	-1,969	0,3005
Call delta 5	-1,928	0,3191	-8,857	-2,052	0,2643

Table E6 Critical values for Z(t) augmented Dickey-Fuller test for stationarity.

Significant level	*1 %*	*5 %*	*10 %*
Z(t)	-3,43	-2,86	-2,57

Table E.7 Critical values for Phillips-Perron test for stationarity.

Significant level	*1 %*	*5 %*	*10 %*
R(rho)	-20,700	-14,100	-11,300
Z(t)	-3,430	-2,860	-2,570

Table E.8 Test statistics for the Diebold-Mariano test for benchmark models

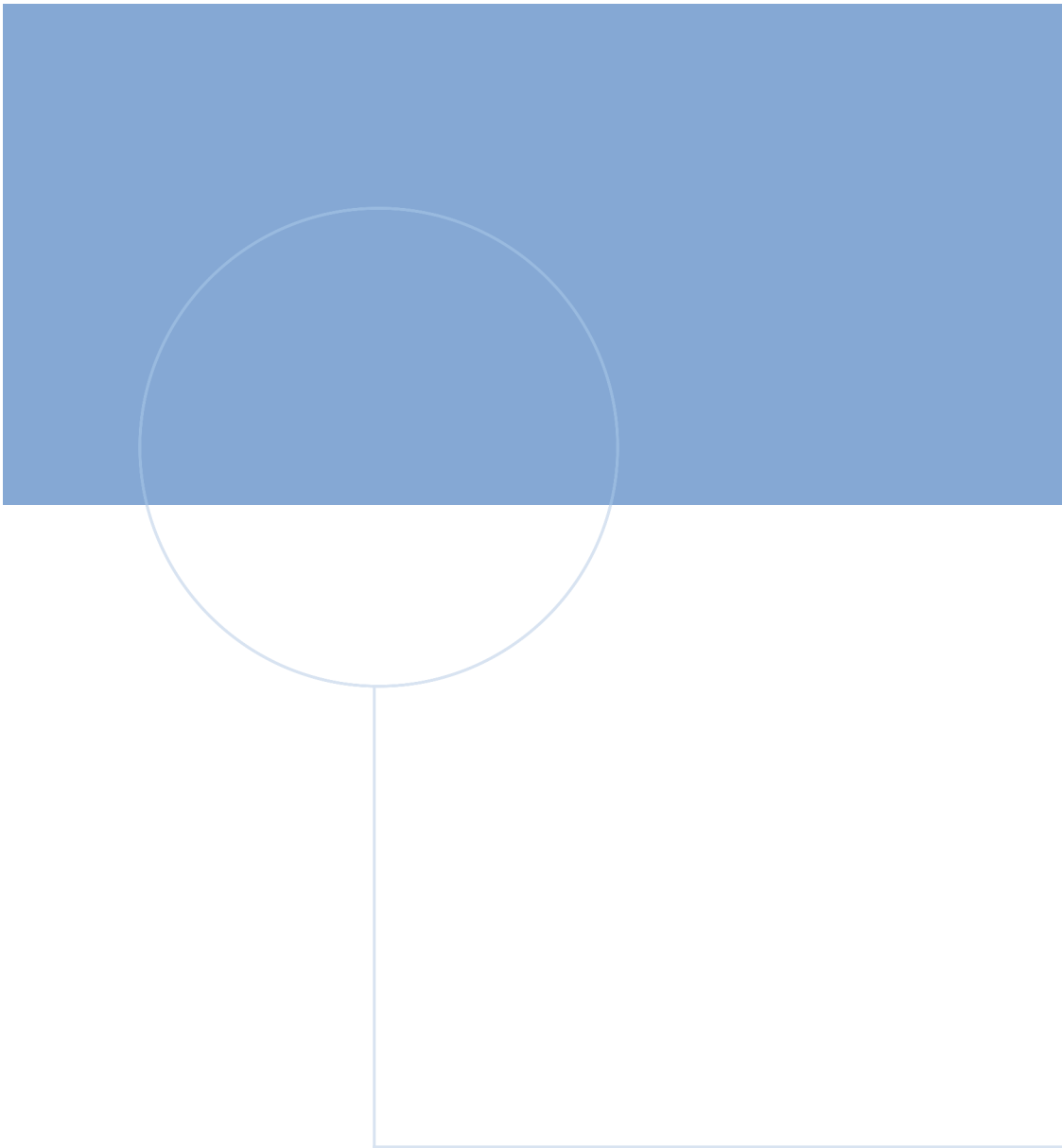
Option	*LSTM / RF*	*LSTM / AR-GARCH*	*AR-GARCH / RF*
One week OTM put	0,1993	0,1515	0,0012
One week ATM put	0,1539	0,0208	0,0008
One week OTM call	0,1290	0,1038	0,0000
One month OTM put	0,0667	0,0622	0,0410
One month ATM put	0,2135	8,0319	0,0017
One month OTM call	0,0276	0,0202	0,706
Three months OTM put	0,1378	0,1586	0,1238
Three months ATM put	0,3411	0,2007	0,2824
Three months OTM call	0,000	0,000	0,3121
Six months OTM put	0,0721	0,0744	0,5098
Six months ATM put	0,4053	0,1204	0,1111
Six months OTM call	0,3084	0,2885	0,2899
One year OTM put	0,3465	0,2947	0,2571
One year ATM put	0,0717	0,0606	0,0630
One year OTM call	0,2363	0,2413	0,3675

Table E.8 Test statistic to reject null hypothesis is 0,05. Numbers <0,05 reject H_0 and the difference in the forecasts are statistically significant from each other.

Table E.9 Test statistics for the Diebold-Mariano test for t-distributed AR-GARCH against benchmark models for one week and one month.

Option	*LSTM/RF*	*LSTM/AR-GARCH*	*LSTM/AR-GARCH t-dist*	*GARCH/t dist*	*RF/GARCH*	*RF/t-dist*
One week OTM put	0,2193	0,3177	0	0,3179	0,3177	0
One week ATM put	0,2127	0,4786	0,4291	0,2208	0,2428	0,2508
One week OTM call	0,0882	0,5677	0,496	0,054	0,107	0,1177
One month OTM put	0,0471	0,0702	0,1585	0,5737	0,1468	0,1581
One month ATM put	0,1378	0,8831	0,9217	0,9269	0,0985	0,0952
One month OTM call	0,3814	0,307	0,3048	0,4314	0,137	0,1255

Table E.9 Test statistic to reject null hypothesis is 0,05. Numbers <0,05 reject H_0 and the difference in the forecasts are statistically significant from each other.



NTNU

Kunnskap for en bedre verden