Norwegian School of Economics
Bergen, Fall, 2022

Stock Market Volatility Forecasting Using Ensemble Models

Tore Kamsvåg and Mark Willard

Supervisor: Jonas Andersson

Master thesis, MSc in Economics and Business Administration,
Business Analytics

NORWEGIAN SCHOOL OF ECONOMICS

Foreword

This master thesis is a part of our master's degree in economy and administration, majoring in Business Analytics at the Norwegian School of Economics (NHH). We would like to thank our supervisor Jonas Andersson for guiding us throughout this paper and contributing with valuable input and feedback.

Abstract

Extensive research has been done within the field of finance to better predict future volatility and anticipate changes in financial market uncertainty. The advent of more advanced machine learning methods, such as artificial neural networks, has led to ground-breaking improvements to modeling capabilities across many fields and industries, including finance and volatility forecasting. These advances have led to rendering some of the previous state of the art models obsolete. Even though it has been established that artificial neural networks are capable of outperforming traditional finance forecasting models when it comes to volatility forecasting, it remains an open question whether a more advanced machine learning algorithm can benefit from incorporating the strengths of specialized volatility forecasting models. In this study, we seek to uncover whether traditional finance volatility forecasting models, such as GARCH type models, contain unique information that when combined with artificial neural networks can lead to more capable models and improved prediction accuracy.
We will explore these effects by looking into S&P 500 one-day-ahead volatility using GARCH type models to generate volatility forecasts and include those into different artificial neural networks to measure improvements in forecasting capabilities. GARCH forecasts will be added into the different artificial neural networks in the form of two different types of ensemble models. One approach being a stacked ensemble, and the other an averaging ensemble. We find evidence to suggest that even though the GARCH type models consistently underperform compared to artificial neural networks, there is sufficient grounds to conclude that there is great potential in combining different volatility forecasting models to attain better volatility predictions.

Table of Contents

List of Figures

List of Tables

1. Introduction

The volatility of stock market returns is an important topic in finance literature. Volatility is regarded as an indicator for the risk and uncertainty in the stock market and can have a crucial function in many investment decisions. This can for example be illustrated through Modern Portfolio Theory (MPT), which was first introduced in 1952 by Harry Markowitz. The essence of MPT is that an investor would always like to choose the portfolio that provides the highest expected return, while having the lowest amount of risk (Bodie et al., 2021). However, it's not always simple to decide on which investment portfolio to choose when the risk is increasing along with the expected return. In MPT, it is assumed that each investor has a utility function that is increasing with higher expected returns and decreasing with higher volatility. This allows investors with a varying degree of risk aversion, to select portfolios based on their calculated utility score. Good estimations of volatility are needed for investors to make informed investment decisions when selecting a portfolio.

Volatility is also a major component for the pricing of some derivative instruments, as for example, in option pricing theory. To price an option, you must know the current volatility estimate of the underlying asset up until the option expires. This results in a need to accurately estimate and make good forecasts of volatility.

There exists a vast number of ways to forecast volatility, and new models are still being introduced, especially due to the increase in the use of machine learning methods (Ge et al., 2022). It has been widely established that modern machine learning methods, such as neural networks, are able to consistently outperform GARCH type models when predicting financial market volatility (Charef & Ayachi, 2016). However, can we still benefit from incorporating the properties of GARCH type models into more advanced machine learning methods in order to achieve improved accuracy? To test this concept, we will make use of two different types of ensemble models to explore whether or not we can leverage GARCH outputs to further improve the results of a neural network model.

We will start by giving a definition of volatility, where we also explain the different ways volatility can be estimated. Then we will introduce some statistical models for time series data that are commonly used in volatility forecasting. Some of the standard tools we are going to use are the ARCH/GARCH models as well as ordinary least squares (OLS). We will then

select a benchmark model that we will use for comparison between different models. Next, we will present some machine learning models, in the form of artificial neural networks (ANN) used for volatility forecasting and explore whether these outperform GARCH type models. Finally, we will attempt to combine the benefits of GARCH type models and artificial neural networks in order to see whether the combination yields an even better model. Essentially, we are testing whether GARCH type models contains some additional information that a more advanced neural network model would not be able to identify on its own.

2. Volatility

2.1 Defining volatility

Volatility is commonly defined as a statistical measure of the dispersion of returns for a given security or market index over a specified period of time. For example, the higher the price fluctuates from its average, the higher the volatility. This measure of dispersion can be calculated in different ways, but most often it is calculated using standard deviation (denoted by σ) of logarithmic returns (Poon & Granger, 2003). This method of calculating volatility is known as historical volatility and it can be formulated as

$$\sigma = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(r_t - \mu)^2}, \tag{2.1}$$

where r_t is the log return in time period *t*. The parameter μ represents the mean (expected) return, and *n* is the number of observations that are being used. The logarithmic returns are used instead of daily closing prices because the former follows a normal distribution. Let P_t denote the stock price at the end of time period *t*. Assuming that there are no dividends paid between time *t* and time *t-1* (Tsay, 2013). Then, the simple net return of a given stock can be defined as

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} = \%\Delta P_t. \tag{2.2}$$

The log return, which is known as the continuously compounded return, can then be calculated as

$$r_t = \log(1 + R_t) = \log\left(\frac{P_t}{P_{t-1}}\right) = \log(P_t) - \log(P_{t-1}). \tag{2.3}$$

The log return can be obtained by taking the first difference of the logarithmic prices.

2.2 Realized volatility

One of the main problems in volatility forecasting is that volatility is not directly observable, even ex-post (Patton, 2011). This particular aspect of volatility complicates the evaluation and comparison of forecasting models. A common solution to this problem is to use volatility proxies as a measurement of the ex-post volatility. This could be implemented by incorporating range-based volatility estimators, using high frequency data, or simply using the squared returns as a volatility proxy. It is important to note that there are some potential drawbacks in using these volatility proxies, which we will highlight.

High-frequency data has been increasingly accessible over the years. However, it can in some cases be quite costly and time consuming to collect this type of data. Like historical volatility, realized volatility is a backward-looking measure, which depends on the past price history. However, realized volatility is utilizing high frequency intraday return data as a measurement of volatility. This is proven to provide accurate forecasts, especially for the one-day-ahead horizon (Andersen & Bollerslev, 1998). Barndorff-Nielsen and Shephard (2002), show that realized volatility is a more efficient estimator of the conditional variance in comparison with daily squared returns. Let t denote the time measured in days, and N is the total number of intervals within a day, where (i = 1, 2, …, N). The intraday return can be calculated as the difference between the log prices at the i-th interval and the previous interval, i-1 within day t. Thus, it can be written as

$$r_{i,t} = p_{i,t} - p_{i-1,t}. \tag{2.4}$$

The realized variance for a given day t, can be calculated as the sum of the squared intraday returns, and is given by

$$RV_t = \sum_{i=1}^{N} r_{i,t}^2. \tag{2.5}$$

We can then obtain the realized volatility, by taking the square root of the realized variance.

$$\sigma_{realized\ vol} = \sqrt{\sum_{i=1}^{N} r_{i,t}^2} \tag{2.6}$$

The sampling frequency for the intraday observations are often set between 5 to 30-minute intervals. One should be aware of the trade-off between using a few observations a day and having high frequency return data (e.g., every minute). By increasing the sampling frequency,

it has been proven to give more accurate ex-post volatility measurements (Andersen & Bollerslev, 1998). However, one drawback from using too frequently sampled returns is potential market microstructure noise (Aït-Sahalia & Yu, 2009). This noise component captures a lot of the trading frictions in the market, such as bid-ask bounce, price jumps, and non-synchronous trading.

2.3 Squared returns

The squared daily returns is a commonly used proxy for the true conditional variance, as it is quite simple to implement. The squared returns is considered a conditionally unbiased estimator of the true unobserved conditional variance, under the assumption that the mean return is zero (Patton, 2011). However, as noted by Andersen & Bollerslev (1998) and Hansen & Lunde (2005), the squared return is considered a noisy proxy for the true conditional variance.

While the daily returns itself show little signs of serial correlation, the squared returns, on the other hand, exhibit positive signs of serial correlation (Triacca, 2007). These positive signs indicate the presence of volatility clustering, which refers to the observation that "large changes tend to be followed by large changes, of either sign, and small changes tend to be followed by small changes" (Mandelbrot, 1963).

2.4 Annualized volatility

As illustrated above, volatility can be measured in various ways. However, volatility is often expressed in annual terms and will therefore need to be annualized. This can be done as follows:

$$\sigma_{annualy} = \frac{\sigma_T}{\sqrt{T}} \tag{2.7}$$

Where $\sigma_{annualy}$ is the annualized volatility, which can be expressed as the standard deviation of yearly logarithmic returns. The notation, σ_t is the standard deviation over a single time period, while T denotes the number of periods in a year for a specified unit of time. For

instance, if we have calculated the daily volatility and want to express this in annual terms, we can do the following:

$$\sigma_{annualy} = \sigma_{daily} \cdot \sqrt{252} \tag{2.8}$$

In the formula above we use 252 daily time periods, which is a common assumption to make since the average number of trading days within a year is close to this number.

2.5 Implied volatility

Implied volatility (IV) is estimated based on current market prices of options as opposed to historical and realized volatility, which depends on historical data (Danielsson, 2011). Therefore, this estimation method is said to be forward-looking. Implied volatility is not directly observable, and it is based on the market's expectations of how the price will fluctuate over a given time period. Thus, it is important to note that the IV is just an estimate of what the volatility will be in the future. IV can be determined from an option pricing model, such as the Black and Scholes (1973) option pricing model. By using the observed transaction price of a European option, and then applying the Black-Scholes formula, you are able to back out the implied volatility. An important assumption of the Black and Scholes theorem is that there are no dividends paid out during the life of the option (Bodie et al., 2021).

The Black and Scholes pricing formula for a European call option is given by

$$C_0 = S_0 N(d_1) - Xe^{-rT} N(d_2) \tag{2.9}$$

where

$$d_1 = \frac{\ln\left(\frac{S_0}{X}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}$$

$$d_2 = d_1 - \sigma\sqrt{T} \tag{2.10}$$

and

C_0 = Call option price

S_0 = Current stock price

N(d) = Risk-adjusted probabilities that the call option will expire in the money

X = Exercise price (or strike price)

r = Continuously compounded risk-free interest rate

T = Time to expiration of the option

σ = Standard deviation of log returns (volatility).

All of the inputs except for the implied volatility are observable, either from the option contract itself, the stock market or from using some form of proxy. For example, as a proxy for the risk-free interest rate it is common to use the money market rate for a maturity equal to that of the option (Bodie et al. 2021). By performing some algebraic operations, we are then able to get an estimate of the implied volatility.

The main drawback of implied volatility estimations is that the estimates are highly dependent on the accuracy of the Black-Scholes model, which relies on the assumption that the volatility remains constant over the option's life (Danielsson, 2011). This is not necessarily true in reality, because volatility can fluctuate over different time periods. This is related to a phenomenon known as volatility smile, which can be described as the pattern of implied volatility for a series of options that have the same underlying asset, the same expiration date, but different strike price. When the IV is plotted against the strike price we get a line that slopes upward at either end, hence the term volatility smile. Volatility smiles should not occur according to standard Black-Scholes option price theory, which requires a straight horizontal line.

2.6 The CBOE Volatility Index

The CBOE volatility index (VIX) is one of the most recognized volatility measures of stock market volatility. The index was created by the Chicago Board Options Exchange in 1993 (CBOE, 2021). The VIX is commonly referred to as the "fear gauge" because the volatility index tends to go up when the stock prices are falling. The VIX index is constructed based on

the 30-day expected volatility of the S&P 500 Index (SPX). Only SPX option contracts with more than 23 days and less than 37 days to expiration are used to calculate the VIX index. The generalized formula to calculate the VIX index is given by:

$$\sigma^2 = \frac{2}{T}\sum_i \frac{\Delta K_i}{K_2^i} e^{RT} Q(K_i) - \frac{1}{T}\left[\frac{F}{K_0} - 1\right]^2 \quad (2.11)$$

Where T indicates the time to expiration, F denotes the forward index level derived from index option prices, K_0 is the first strike price below the forward index level, K_i is the strike price of the i-th out-of-the-money option, ΔK_i is the interval between strike prices, R is the risk-free interest rate to expiration and $Q(K_i)$ is the midpoint of the bid-ask spread for each option with strike K_i.

2.7 Characteristics of volatility

There have been confirmed in countless studies that volatility in financial time series exhibits various characteristics (Knight & Satchell, 2007). In this section, we will present some of these features and discuss its relevance in forecasting.

A common assumption in financial theory is that the distribution of stock returns are following a normal distribution. However, as first documented by Mandelbrot in 1963, the distribution of stock returns exhibit fatter tails than the normal distribution. What this means in practice is that we are more likely to observe extreme outlier values (Tsay, 2013). This can be described as a leptokurtic distribution which has kurtosis greater than three. Kurtosis is the fourth standardized moment, which describes to what extent a distribution is heavy-tailed or light-tailed relative to a normal distribution. A normal distribution has kurtosis equal to three, thus a leptokurtic distribution which has kurtosis greater than three is said to have excess kurtosis.

Next, we have volatility clustering, which is a well-known concept in financial time-series and was first introduced by Mandelbrot in 1963 and later documented by Fama (1965). Volatility clustering can be described as the tendency that volatility changes persists over time. Thus, if volatility is considered high (low) today, then it is more likely that volatility will be high (low) tomorrow.

There has also been evidence showing signs that volatility exhibits mean-reverting behavior (Knight & Satchell, 2007). Eventually, there will be a time where volatility goes back to its long-term average level of volatility, and this behavior is known as mean-reversion. In the long run it is assumed that forecasts will converge to this average level of volatility, independently of the starting point of the forecast.

The leverage effect is another stylized fact about volatility which refers to the negative relationship between stock returns and future volatility. High (low) levels of volatility are typically followed by decreasing (increasing) returns. Explanations of this negative relationship have been proposed and documented by Black (1976) and Christie (1982). As the price of a stock is decreasing, the equity value of a company goes down, but the value of the debt remains the same. This means that the company will have a higher debt-to-equity ratio, which will make the stock become riskier, which also implies that the volatility should increase. The asymmetric structure of volatility also refers to the fact that a volatility increase due to a price increase tends to have a greater impact than a price increase of the same magnitude (Francq & Zakoian, 2010).

Lastly, there has also been evidence of volatility co-movements among different stock markets. López-García et al. (2021) found that stocks that are more similar in terms of volatility show a tendency to have greater co-movement than stocks of different volatility. The authors also show results confirming that the co-movement of volatility is greater during periods of crisis

3. Econometric models

3.1 The ARCH Model

The ARCH model stands for autoregressive conditional heteroskedasticity, and it is one of the most popular tools in the literature of volatility forecasting. The model was first developed by Robert F. Engle in 1982. The autoregressive (AR) term of the ARCH process indicates that current values are dependent on past values. The ARCH model is conditionally heteroskedastic, which refers to the time-varying aspect of the conditional variance. The traditional econometric models, such as linear regression, assume that the residuals have constant variance for all values of the independent variables (Engle, 2001). This is also known as homoskedasticity. However, as noted by many researchers, the presence of heteroskedasticity in time series is not uncommon. As previously mentioned, volatility clustering is a phenomenon that occurs in financial time series, and it indicates that there are some time periods that are riskier than others. Heteroskedasticity does not cause ordinary least squares (OLS) coefficient estimates to be biased. However, it can cause the estimated variances of the regression coefficients to be biased, which leads to lower precision. The ARCH model has the property of time-varying conditional variance and is therefore designed to deal with this issue.

To understand how the ARCH model works, we will start by formulating and explaining the first order ARCH(1) process:

$$r_t = \mu_t + \epsilon_t, \tag{3.1}$$

$$\epsilon_t = \sigma_t z_t, \;\; z_t \sim iid\; N(0,1) \tag{3.2}$$

$$\sigma_t^2 = \omega + \alpha_1 \epsilon_{t-1}^2 \tag{3.3}$$

$$\omega > 0, \;\; \alpha_1 \geq 0, \tag{3.4}$$

where $\in_t$ denotes the error term at the present time period, which is expressed in equation (3.2) as a sequence of independent and identically distributed (iid) random variables, z_t, which has zero mean and variance equal to 1, multiplied by the conditional time dependent

volatility, σ_t. In practice, z_t is commonly assumed to follow the standard normal distribution or a standardized student-t distribution (Tsay, 2013). It is also possible that the observed asset returns can be skewed, thus in some cases it can be relevant to assume that we have a skewed distribution. The conditional mean of the model is represented by μ, which is often assumed to be constant and equal to zero. However, in some applications it can be useful to determine the conditional mean by an autoregressive-moving-average (ARMA) model.

The time varying conditional variance in equation (3.3) is a linear function of the squared error term at time t – 1. The omega parameter, ω, denotes the variance intercept. The persistence of the autocorrelations are expressed by the parameter alpha. This parameter can also be interpreted as how the volatility reacts to market movements. For instance, large values of α, indicates that market movements have a significant effect on future volatility. Lastly, the constraint in equation (3.4) is required to make sure that the variance cannot be negative.

The omega parameter in equation (3.3) can also be decomposed as a constant, γ (gamma) multiplied with the long-run variance (V_L), which is also referred to as the unconditional variance. By doing so, we will get the following equation for the conditional variance:

$$\sigma_t^2 = \gamma V_L + \alpha_1 \epsilon_{t-1}^2. \tag{3.5}$$

The conditional variance is generated by the history of past errors, denoted by $\in_{t-1}$. The long-run variance can then be derived from the following equations:

$$\sigma_t^2 = E(\epsilon_t^2 | \epsilon_{t-1}) \tag{3.6}$$

$$V_L = E(\epsilon_t^2) = E[E(\epsilon_t^2 | \epsilon_{t-1})] \tag{3.7}$$

$$= E(\omega + \alpha_1 + \epsilon_{t-1}^2) \tag{3.8}$$

$$= \omega + \alpha_1 V_L. \tag{3.9}$$

The ARCH process is said to be stationary if the sum of the positive autoregressive parameters is less than one (Bollerslev et al., 1994). In the case of an ARCH(1) process, the long-run variance is given by:

$$V_L = \frac{\omega}{1-\alpha_1} \quad (\text{if } 0 < \alpha_1 < 1). \tag{3.10}$$

Then we have the ARCH(q) model which extends the autocorrelation structure of the ARCH(1) model, and it takes the following form:

$$\sigma_t^2 = \omega + \alpha_1\epsilon_{t-1}^2 + \alpha_2\epsilon_{t-2}^2 + \cdots + \alpha_q\epsilon_{t-q}^2 , \tag{3.11}$$

where q is the length of the ARCH lags. By including more lags in the model, we allow changes in the variance to occur more slowly (Knight & Satchell, 2007). To ensure that the conditional variance remains positive, we need to include the following constraint:

$$\omega > 0,\ \alpha_1 \geq 0, \alpha_2 \geq 0, \ldots, \alpha_q \geq 0. \tag{3.12}$$

As mentioned above, the ARCH model is simple to use and is able to capture some of the important features of financial time series, such as volatility clustering and mean reversion. However, the model still has some drawbacks. When forecasting volatility using the ARCH model it will often require a large number of lags to be included in the model. This leads to a large number of parameters that need to be estimated, which in turn makes the process more complex.

3.2 The GARCH Model

The generalized autoregressive conditional heteroskedasticity (GARCH) model was proposed by Bollerslev (1986). The model differs from the ARCH model by allowing lagged values of the conditional variances to be included in the equation for the current conditional variance. This generalization allows for a more parsimonious representation of the conditional variance than the ARCH model. The GARCH model also has a more flexible lag structure, which allows changes in the variance to occur at a slower rate (Knight & Satchell, 2007).

The GARCH(p,q) model can be formulated as

$$r_t = \mu_t + \sigma_t z_t, \quad z_t \sim iid\ N(0,1)$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{p} \beta_j \sigma_{t-j}^2$$

$$\omega > 0,\ \alpha_i \geq 0, i = 1, \dots, q,\ \beta_j \geq 0, j = 1, \dots, p\ , \tag{3.13}$$

where q refers to the number of autoregressive lags or ARCH terms, while p denotes the number of past conditional variances, which often is referred to as GARCH terms (Engle, 2001). The coefficient β can be interpreted as the degree of volatility persistence. For instance, a large β value indicates that the conditional variance decays slowly. The GARCH process is also assumed to be wide-sense stationary (WSS) if the sum of the two parameters, α and $\beta < 1$ (Bollerslev, 1986). The term WSS implies that the mean and the autocorrelation functions are time invariant. The value of the three parameters ω, α, and β can be obtained using Maximum Likelihood Estimation (MLE), which we will describe in section 3.4.

The GARCH(1,1) model is a popular specification of the GARCH(p,q) model, due to its simplicity and robustness. In comparison with other volatility models there has been evidence that the GARCH(1,1) model is quite hard to beat, as examined by Hansen and Lunde (2005). The GARCH(1,1) model can be expressed as:

$$\sigma_t^2 = \omega + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2. \tag{3.14}$$

However, it is worth noting that the standard GARCH models have some drawbacks. If we consider the GARCH(p,q) model shown in equation (2.7), we can see that the conditional variance is a function of squared innovations, which means that it disregards the sign of the returns (Knight & Satchell, 2007). However, it is quite unlikely that positive and negative shocks in the price of a given security will have the same effect on volatility. Thus, we can conclude that the standard GARCH models are not able to capture the asymmetry and leverage effects that are observed in stock returns.

There exist many different extensions of the standard GARCH model, all of which have different properties. For example, we have regime switching GARCH models, which can take into account the sudden changes in the state of the market. Other popular extensions include the asymmetric GARCH models, such as the GJR-GARCH and exponential GARCH (EGARCH) model.

3.3 Exponential GARCH Model

The exponential GARCH model was first proposed by Daniel B. Nelson in 1991. Recall that we had to impose a non-negative constraint on the standard ARCH and GARCH model to ensure that the conditional variance remains positive. The EGARCH model specifies the conditional variance in logarithmic form, and we therefore do not need to include the non-negative constraint (Poon & Granger, 2003). The standard GARCH models only consider the magnitude of a shock, while disregarding if the shock is positive or negative (Nelson, 1991).

The EGARCH(p,q) model can be written in the following manner:

$$\begin{cases} \epsilon_t = \sigma_t z_t \\ log\ \sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i\, g(z_{t-i}) + \sum_{j=1}^{p} \beta_j\, log\ \sigma_{t-j}^2 \end{cases}, \tag{3.15}$$

where

$$g(z_t) = \theta\epsilon_t + \gamma(|z_t| - E|z_t|). \tag{3.16}$$

The weighted innovation, described by the function $g(z_t)$ allows the model to respond asymmetrically to positive and negative values of asset returns. (Tsay, 2013). In equation (3.16) the parameters, θ (theta) and γ (gamma) are real constants. Both the absolute residuals and the expectation of the absolute residuals are zero-mean iid sequences with continuous distributions.

3.4 Estimation of ARCH and GARCH Models

In this section we will explain how to estimate the parameters in the ARCH and GARCH models. The simplest way to estimate an ARCH model is by using ordinary least squares (OLS). Although this is a simple estimation method, the OLS estimators are inefficient in the presence of heteroskedasticity (Francq & Zakoian, 2010). There exists an alternative estimation method which performs better in the presence of heteroskedasticity and non-linearity, and this method is known as Maximum Likelihood Estimation (MLE). This estimation method is typically used by both ARCH and GARCH models. In MLE, the parameter values are obtained by maximizing a likelihood function in such a way that the observed data is most probable. When performing MLE we need to make an important assumption which states that the data needs to be identically and independently distributed. In other terms, it means that a sample of *n* random variables must share the same probability distribution and that all of the samples are independent events. As previously mentioned, it is commonly assumed that the returns follow a normal distribution. However, the returns will in many cases have fatter tails than suggested by the normal distribution. This indicates that an ARCH or GARCH model can be improved by instead assuming a heavy-tailed distribution, such as the student-t distribution.

The return data is assumed to be generated from a known density function, as follows:

$$z \sim f(r_t | r_{t-1}, r_{t-2}, \dots; \theta) \tag{3.17}$$

Where r_t is the sample data of returns, which depends on the parameter known as theta. The parameter theta, denoted by θ, is a parameter vector which contains a set of unknown parameters that needs to be estimated.

Given the fact that our data is assumed to be iid, we can construct the following likelihood and joint density function:

$$L(\theta|z) = \prod_{t=1}^{T} f(z_1, \dots, z_t|\theta) \tag{3.18}$$

The likelihood function denoted by $L(\theta|z)$ is a function of the parameter θ, and it must not be confused with the probability density function, which is a function of each observation of z_t with the parameter θ fixed. In practice, it is often more convenient to use the log-likelihood

instead of the original likelihood function. This is because it is usually easier to work with a sum rather than the products of densities. The natural logarithm is a monotonically increasing function which ensures that the maximum value of the log-likelihood function occurs at the same point as the original likelihood. We can then express the log-likelihood function as following:

$$\log L(\theta|z) = \sum_{t=1}^{T} \log f(z_1, \dots, z_t|\theta) \tag{3.19}$$

3.4.1 Normal Distribution

By assuming that our data follows a normal distribution we can express equation (3.18) as:

$$L(\theta|z) = \prod_{t=1}^{T} f(z_1, \dots, z_t|\theta) = \prod_{t=1}^{T} \left[\frac{1}{\sqrt{2\pi\sigma_t^2}} \, exp\left(-\frac{z_t^2}{2\sigma_t^2}\right) \right] \tag{3.20}$$

The log-likelihood function is then given by:

$$L(\theta|z) = \sum_{t=1}^{T} \left[-\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\sigma_t^2) - \frac{1}{2}\frac{z_t^2}{\sigma_t^2} \right] \tag{3.21}$$

We can then simplify the equation by ignoring the constant values since they do not impact the solution (Danielsson, 2011). The equation can then be written as:

$$L(\theta|z) = \sum_{t=1}^{T} \left[-\frac{1}{2}\log(\sigma_t^2) - \frac{1}{2}\frac{z_t^2}{\sigma_t^2} \right] = -\frac{1}{2}\sum_{t=1}^{T} \left[\log(\sigma_t^2) + \frac{z_t^2}{\sigma_t^2} \right] \tag{3.22}$$

To specify the log-likelihood function for the ARCH and GARCH models, we can simply substitute the conditional variance with the terms expressed in equation (3.11) and (3.13). For example, if we want to estimate the parameters of a GARCH(1,1) model we obtain the following equation:

$$L(\theta|z) = -\frac{1}{2}\sum_{t=1}^{T} \left[\log(\omega + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2) + \frac{z_t^2}{\omega + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2} \right] \tag{3.23}$$

3.4.2 Student-t Distribution

As noted previously, observed returns will often have fatter tails than implied by the normal distribution. Hence, a conditionally fat distribution such as the student-t distribution can lead to a better fit. We assume that the innovation in returns, denoted z_t takes the following form:

$$z_t \sim t_{(v)} \tag{3.24}$$

Where, the degrees of freedom are denoted by *v* and $t_{(v)}$ represents a student-t distribution. The density function for the student-t distribution is given by

$$f_t(z_1, \dots, z_t|\theta) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{((v-2)\pi)^{\frac{1}{2}}\Gamma\left(\frac{v}{2}\right)}\left(1 + \frac{z_t^2}{v-1}\right)^{-\frac{v+1}{2}}, \quad v > 2. \tag{3.25}$$

where (r) denotes the gamma function. As the degrees of freedom are increasing towards infinity, the student-t distribution approaches the normal distribution with mean zero and variance equal to one (Danielsson, 2011).

The log-likelihood function for the student-t distribution is then given by

$$L_{Student} = T\left[log\ \Gamma\left(\frac{v+1}{2}\right) - log\ \Gamma\left(\frac{v}{2}\right) - \frac{1}{2}log[\pi(v-2)]\right]$$

$$= -\frac{1}{2}\sum_{t=1}^{T}\left[\log(\sigma_t^2) + (v+1)\log\left(1 + \frac{z_t^2}{v-2}\right)\right]. \tag{3.26}$$

4. Machine Learning Models

4.1 Artificial Neural Networks

Neural networks have been gaining increasing popularity over the years, especially due to successful applications in many areas of industry, such as robotics, automotive industry, power plants, aircraft control, medical systems, and others (Schumann et al. 2010). Other typical tasks performed by neural networks include classification, prediction, clustering, and pattern recognition. There exists a large variety of neural network structures, and in this chapter, we are going to present some of them, emphasizing on volatility forecasting.

Artificial neural networks (ANNs), commonly just referred to as neural networks (NNs), are computational models inspired by the biological nervous system. Neural networks were first proposed by McCulloch and Pitts (1943), where they created a model that simulates how the neurons function in the human brain. An Artificial neural network is typically made up of multiple layers, where the neurons from one layer connect with the neurons in the preceding and following layers. A neural network with a single layer is called a perceptron and a neural network with multiple layers is called an artificial neural network. Most linear relationships can be accurately modeled by a perceptron, but more complex artificial neural networks are required for more difficult problems.

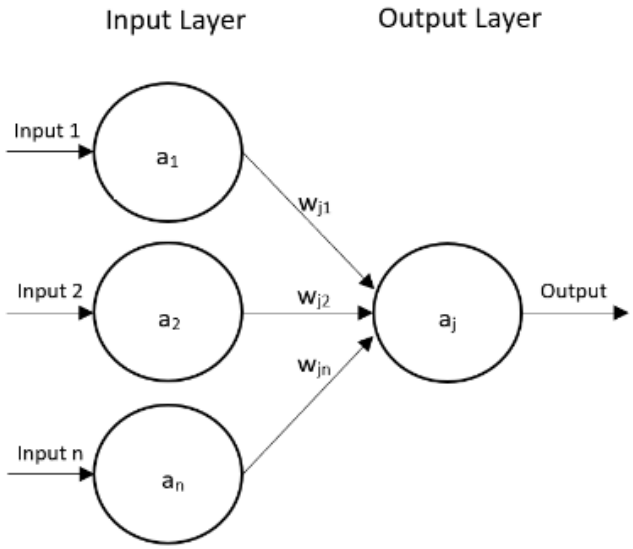


Figure 4.1: Artificial neural network without any hidden layers.

Figure 4.1 shows one of the simplest versions of an artificial neural network, one without any hidden layers. Each unit's output is the result of applying the chosen activation function to a summing node shown in equation (4.1). Where a_j is the incoming sum for processing unit j after applying the activation value for unit j, and w_{ji} is the weight from unit i to unit j.

$$a_j = f(\sum_{i=0}^{n} w_{ji}\, a_n) \tag{4.1}$$

In artificial neural networks, the coefficients attached to each predictor are instead called weights and the algorithm uses randomness to find a good enough set of weights for the specific mapping function from the inputs to outputs (Dayhoff & DeLeo, 2001). However, using a simple artificial neural network, like the one shown in figure 4.1 can be the equivalent of a very inefficient approach to achieving the same results as ordinary least squares or a logistic regression.

A common approach for artificial networks is to make use of a deep neural network (DNN). Deep neural networks often start with the input layer which receives the input variables and passes them to the rest of the network. The next type of layer is referred to as the hidden layer and is located between the input and output layer. The number of hidden layers can be zero or more. Lastly, we have the output layer which is providing the estimated dependent variable. In figure 3.1 which is presented below, you can see a typical representation of a fully connected deep neural network with a single hidden layer. The purpose of the hidden layer is to grant flexibility to the model to find relationships that are not based on just the input values and are instead limited to the weights established by the input layer. The hidden layer can come up with entirely new weights based solely on weights from the previous layer, a hidden layer is the main reason why neural networks are able to fit very complex problems with multiple layers of information, while other models would require more supervision to achieve a good fit.

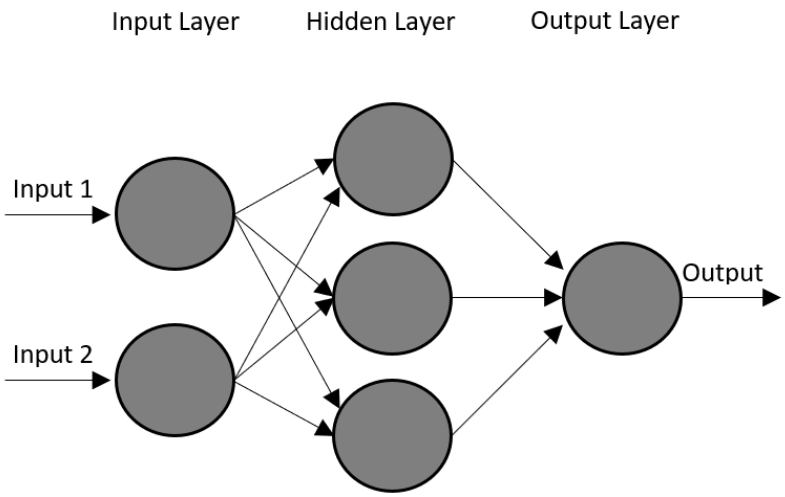


Figure 4.2: Single hidden layer deep neural network

This type of neural network consists of two phases: forward propagation and back propagation. Forward propagation entails multiplying feature values with assigned weights and applying activation functions to each neuron. Back propagation on the other hand updates the weights by partially differentiating gradients of the loss function. An optimization function is required to perform back propagation. The purpose of an optimizer is to establish a relationship between the loss function and the model weights. The optimizer dictates how the learning and improvements take place and at what rate, this entails the different optimizers and learning rates must be tested to find the best specifications to achieve the best out-of-sample performance. We will mainly work with Stochastic gradient descent (SGD), since it has been proven to be a reliable optimizer, superior to more modern approaches such as Adam (Wilson et al., 2017).

Stochastic gradient descent (SGD) is a convex function whose output is the partial derivative of the weights of its inputs (Bottou & Bousquet, 2008). The formulation of the SGD can be seen in equation 4.2. Where w denotes the weight, while t stands for the current iteration, the parameter η refers to the learning rate, and ℓ stands for the current loss.

$$w(t+1) = w(t) - \frac{\eta}{t}\frac{\partial \ell}{\partial w} \tag{4.2}$$

The greater the gradient, the steeper the slope and learning rate. Gradient descent is run iteratively to find the optimal values of the weights to lead to the minimum possible value of the given loss function. The SGD convergence is essentially limited by the stochastic noise induced by the random choice of one example at each iteration. For simple regression problems, a faster learning rate would be ideal to reach the global minima quicker, while more complex problems may require a slower fitting process to detect subtle improvements at the expense of speed.

4.2 Activation functions

The purpose of an activation function is to introduce non-linearity to the data. Introducing non-linearity helps to identify more complex underlying patterns within the data. It is also used to scale the value to a particular interval. For example, the sigmoid activation function, shown in equation (4.3), scales the value between 0 and 1. If an activation function is not applied, the output signal becomes a simple linear function, which in some cases is the ideal approach.

Given that realized volatility is a continuous value, we will instead be using activation ReLU, as well as linear activation. ReLU stands for rectified linear activation unit, while linear activation is essentially multiplying coefficients with weights.

$$\text{Sigmoid:} \qquad f(x) = \frac{1}{1+e^{-x}} \tag{4.3}$$

$$\text{ReLU:} \qquad f(x) = \begin{cases} 0 & for\ x < 0 \\ x & for\ x \geq 0 \end{cases}$$

$$= max\{0, x\} \tag{4.4}$$

$$\text{Linear:} \qquad f(x) = x \tag{4.5}$$

The main advantage of making use of neural networks is the fact that they are able to model complex non-linear relationships, as well as variable interactions to identify relationships that would be impossible for a human or would require extensive feature engineering for

traditional machine learning methods to detect and model. The downside however is that even though artificial neural networks assign interpretable weights/coefficients to each variable just like a regular regression model, artificial neural networks rely on sets of relationships within each neuron that lead to very complex, in some cases unnecessarily complex, large sets of weights that often times are far too many for it to be interpretable beyond its final answer, which leads to the issue of Blackbox solutions which are very difficult to interpret (Dayhoff & DeLeo, 2001).

4.3 Recurrent Neural Networks

Recurrent neural networks (RNN) are a type of neural network which has the additional feature of being able to take into consideration previous output of the model as additional input to search for relationships between past predictions and the next prediction in the time series. Recurrent neural networks are especially useful when dealing with time series data or natural language processing data because of its inherent ability to incorporate previous predictions to forecast based on the regressors, as well as establish a relationship between previous predictions and future predictions (Gers et al., 2002).

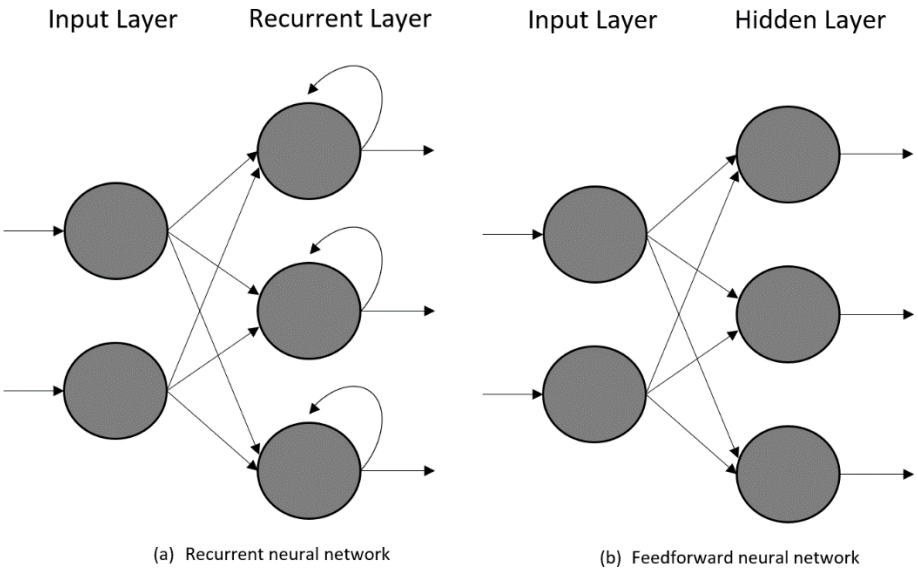


Figure 4.3: Illustration of a recurrent neural network and a feedforward neural network.

For example, if we were to train a model to predict the dietary patterns of an individual, the main difference between a neural network and a recurrent neural network would be that the

standard NN would only be able to predict based on the available variables, while the RNN would also be able to incorporate previous predictions as part of its input if deemed useful. This additional feature of recurrent neural networks allows us to access additional information generated by the model, similar to GARCH type models. This added feature of recurrent neural networks makes them specially qualified for working with time series data.

4.4 Ensemble Models

In traditional machine learning, models use cleaned and sometimes normalized inputs, as well as engineered features to generate the best fit to reduce the chosen loss function. Ensemble modeling on the other hand is a process where multiple different models are used to predict an outcome. The motivation for using ensemble models is to reduce the generalization error of the prediction as well as reduce overfitting. Going beyond relying on one single model has proven to further improve model forecasting performance (Hölldobler et al., 2017). incorporating predictions from different models into a larger model can add additional information beyond the data itself. For instance, an example of how capable the combination of models can be is the case of AlphaGo, which was the first model to be able to win a game of Go at a professional level and was achieved with an ensemble of different neural networks focusing on different tasks

To leverage the information provided by GARCH and improve volatility predictions, we will use two different approaches to combine our models. One approach will be using ensemble averaging, and the second approach will be to use stacked ensemble models.

4.4.1 Ensemble Averaging Models

Ensemble models can reduce the chance of overfitting by combining different model strengths while reducing modeling method biases. However, the decrease in predicting error should only arise if the models used are diverse and independent. An easy way to produce ensemble models is to generate an average estimate from all the models used, which is done by generating out-of-sample predictions for different models and calculating the average prediction. An operation as simple as the average of different model predictions can oftentimes lead to superior models (Disorntetiwat & Dagli, 2000).

4.4.2 Stacked Models

Even though combining predictions is a good option to incorporate different models, we will also be making use of stacked models, which is a different form of ensemble models.
In model stacking, we don't combine multiple outputs to produce a final estimate, we instead forecast separate predictions with several different models, and then use those predictions as features for a higher-level meta model. It can work especially well by combining several varied types of lower-level learners, all contributing their different strengths to the final meta model (Ramos-Pérez et al., 2019). Model stacks can be built in many ways, and there isn't one "correct" way to use stacking. However, if our model achieves better results than one single model, then we can safely say we benefited from the additional information the other models contained.

The main difference between Ensemble Averaging and Stacking is that the Averaging approach calculates an average of the final outputs of different models to generate a combined prediction which could lead to better out of sample performance compared to any of the single models. On the other hand, stacked models would instead incorporate these predictions as additional inputs for a final model that would predict based on the actual data, as well as what other models forecasted.

5. Preliminary data analysis

5.1 Data description

In this study we use daily logarithmic returns of the S&P 500 Index. The S&P 500 Index is a stock market index, which consists of 500 large-cap companies, all of which are listed on stock exchanges in the United States. The reason why we use the returns instead of the closing prices has to do with the common assumption of stationarity in time series. The observed prices of a stock or an index are typically non-stationary, and thus have to be transformed into a stationary time series. This can be achieved by applying a method known as differencing, which computes the difference between consecutive observations (Hyndman & Athanasopoulos, 2018). The reason why we do this is to eliminate trends and seasonality in our time series. Taking the logarithm of the returns is another technique used to stabilize the variance of our time series. Let P_t be the closing price of a financial asset at a specified time period given by the subscript *t*. The logarithmic return of an asset can then be formulated as:

$$r_t = \log\left(\frac{P_t}{P_{t-1}}\right) = \log(P_t) - \log(P_{t-1}) \qquad (5.1)$$

These returns are calculated based on adjusted closing prices, which is considered to be more accurate and reliable than using the raw closing prices. The data is obtained from Yahoo Finance, and it includes 10 years of historical data. The stock markets are not open during weekends or holidays. Thus, our data only contains the number of trading days over the 10-year period. Our sample period starts from 01.03.2012 to 01.03.2022, and the total sample size is equal to 2505 observations.

5.1.1 Volatility proxy

The volatility proxy we are going to use in this study is realized volatility, which is computed based on squared intraday returns. The data is obtained from the Oxford-Man Institute's realized library. On their website, we could choose between using a sampling frequency of either 5 or 10 minutes for the realized variance. We decided to use a sampling frequency of 10 minutes, to reduce some of the potential market microstructure noise. Since we are interested in the realized volatility, we had to take the square root of the realized variance estimates. The

realized variance data for our selected 10-year period were missing 12 daily observations, therefore we decided to exclude these observations from our data set. Afterwards, we merged the realized variance data frame with the S&P 500 Index data in order to make the data points match.

5.1.2 Descriptive statistics

A summary of the descriptive statistics of the S&P 500 Index and the daily realized volatility is presented in Table 5.1. As shown in the table, we can see that the S&P 500 Index had a positive average return of around 0.05% per day. The standard deviation of daily return is about 1.05%, which corresponds to an average annualized volatility of around 0.167%. By observing the skewness and kurtosis we can assess the distribution of our data. As presented in the table, the daily returns have a skewness of around -0.96 which means that we have slight negative skewness in our data. In other terms, this suggests that the tail of the distribution is longer on the left side, than what is assumed by a normal distribution. Next, we can see that the kurtosis coefficient is very large, which indicates that our distribution is heavy-tailed and that it has a high peak at the mean. As a reference point, the normal distribution has a kurtosis value of 3 and a skewness of zero, thus our results suggest that a gaussian distribution is unlikely. These results are further confirmed by the Jarque-Bera test, which is a goodness-of-fit test to find out whether the skewness and excess kurtosis is significantly different from zero. The value of the Jarque-Bera test statistic is around 46050, and the null hypothesis is rejected at a 1% significance level. This also indicates that our data does not fit a normal (gaussian) distribution.

The Augmented Dickey-Fuller test (ADF) is being used to test whether our time series of returns is stationary or not. A stationary process is characterized by having a constant mean, a constant variance and a covariance structure that is stable over time. The null hypothesis of the ADF test is that the time series is non-stationary. As we can observe in the table below, the null hypothesis of the ADF test is rejected at a 1% significance level, which suggest that our data is indeed stationary.

Table 5.1: Descriptive statistics of S&P 500 Index and RV for the past 10 years.

	Close price	Daily return	Realized volatility
Sample size	2505	2505	2505
Mean	2518	0.04 %	7.75 %
Std	849.96	1.05 %	2.39 %
Min	1278	-12.77 %	3.38 %
25 %	1937	-0.34 %	6.13 %
50 %	2342	0.06 %	7.29 %
75 %	2914	0.52 %	8.84 %
Max	4797	8.97 %	26.00 %
Skewness	0.88	-0.96	1.89
Kurtosis	0.12	20.92	7.34
Jarque Bera test	316,68*	46050*	7116*
ADF test	-2.09	-13,45*	-6,602*

Note: *, ** and *** indicate significance at 1%, 5%, and 10% level, respectively.

Figure 5.1 displays the daily returns and price level of the S&P 500 Index over a 10-year period. The graph on the left shows evidence that the volatility of returns varies over time. Upon further inspection, we can also observe that our time series has a mean-reverting behavior, where the returns tend to stay around zero in the long run.

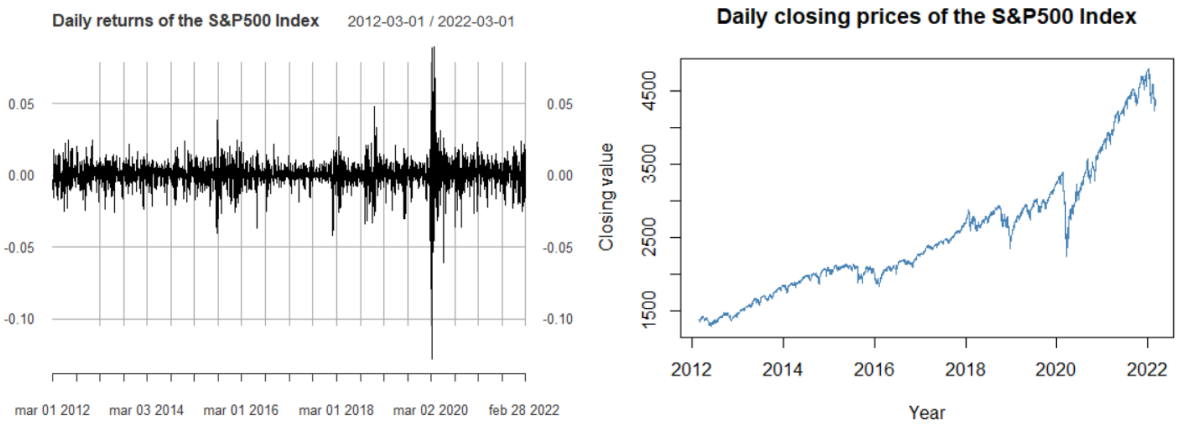


Figure 5.1: S&P 500 Index of returns and closing prices from 01.03.2012 to 01.03.2022.

5.1.3 Autocorrelation

The autocorrelation function (ACF) is a useful tool when analyzing time series, as it can display the serial correlation between the returns and their lagged values (Hyndman &

Athanasopoulos, 2018). If none of the lags in a time series display significant autocorrelation, then the series is considered to be white noise (WN).

The sample ACF plot and partial ACF plot of returns can be used to determine the order of autoregressive (AR) and moving average (MA) lags in our time series. By observing the correlograms in figure 5.2 we can observe that both the sample ACF plot and partial ACF have several significant spikes outside of the confidence interval, which suggests that our series is unlikely to be white noise (Hyndman & Athanasopoulos, 2018). We can use the information displayed in table 5.2 to identify which type of model is suggested by the ACF and PACF plots. Both the sample ACF and partial ACF plots display a pattern of gradual decay towards zero, as the lags are increasing. This suggests that an ARMA(p,q) model would be an appropriate structure for our time series. Determining the order of an ARMA(p,q) model can be difficult from just inspecting the ACF and PACF plots. As seen in Brockwell & Davis (1991), a common approach among practitioners is to try out different variations of ARMA models and see which one provides the smallest value for the Akaike information criterion (AIC). These results can be found in Appendix B.

Table 5.2: Identifying the order of an ARMA model.

	AR(p)	MA(q)	ARMA(p,q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

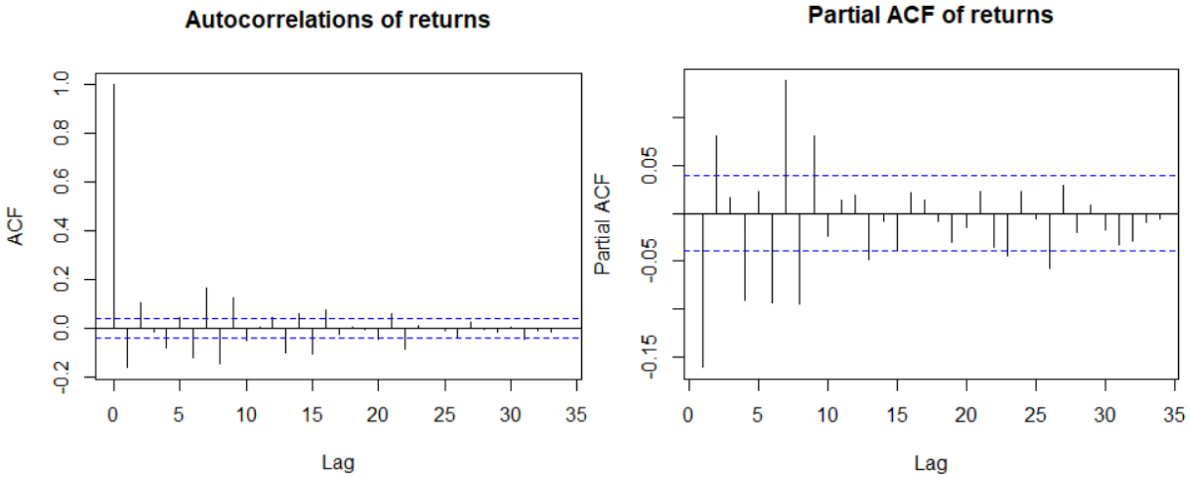


Figure 5.2: Sample ACF and Partial ACF plot of daily S&P 500 returns. The dashed blue lines indicate whether the correlations are significantly different from zero.

The ACF plots of squared and absolute returns can be used to identify if there are any signs of volatility clustering in our data. From observing the two plots in figure 5.3, we can clearly see that most of the lags are outside of the 95% confidence interval. This indicates that we do have volatility clustering in our time series, which motivates the use of the GARCH class of models.

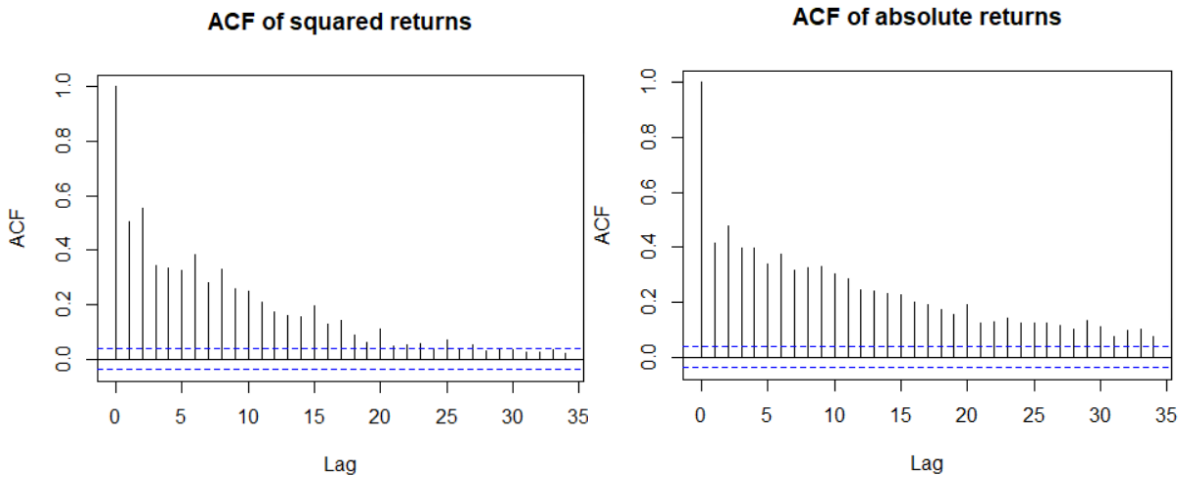


Figure 5.3: ACF of squared and absolute returns for the S&P 500 Index.

5.1.4 Normal Q-Q plot and histogram

Figure 5.4 displays a quantile-to-quantile (Q-Q) plot and a histogram of the S&P 500 Index returns. The normal Q-Q plot is being used to test our assumption that the returns follow a normal distribution. From observing the Q-Q plot, we can see that many data points are deviating in the tails from the reference line. Moreover, we can see that the Q-Q plot yields an inverted S shape, which indicates a heavy-tailed distribution. This indicates that the returns do not seem to follow a normal distribution. The same conclusion can be drawn from inspecting the histogram of the returns from the S&P 500 Index. We can clearly see that the daily returns have a much higher peak at the mean than suggested by the normal distribution.

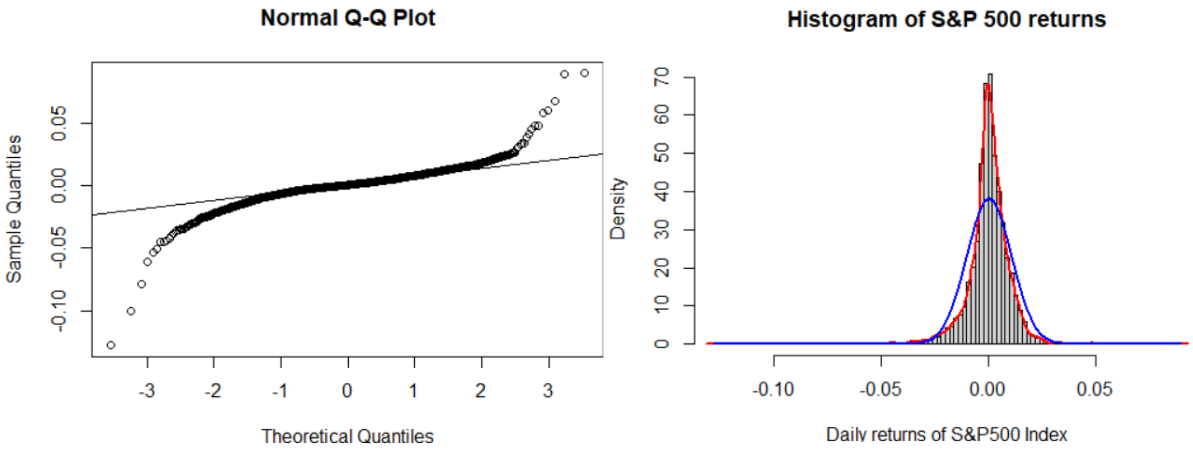


Figure 5.4*:* Normal Q-Q plot and histogram of the S&P 500 Index returns. The red line in the histogram indicates an estimated kernel density function of the returns, while the blue line is a fitted normal distribution with the mean and standard deviation of the returns.

6. Methodology

To explore the potential to improve the already established time series predictive properties of neural networks, we will incorporate GARCH type models into our neural networks by making use of ensemble models to better predict the one-day ahead volatility. We will be using Ordinary least squares (OLS) as our benchmark, and study different ways of combining GARCH type models with artificial neural networks (ANN) and Recurrent Neural networks (RNN). Other methods, such as support vector machines (SVM) and Gradient Boosting, could have also been incorporated into the study. However, neural networks were chosen as the method to study the combination of models due to its inherent ability to adapt to complex data to achieve high performance models without much feature engineering.

6.1 Forecast horizon

Volatility forecasting is extremely important to professional as well as individual portfolio managers, getting an accurate understanding of what the future volatility will be could mean the difference between a successfully hedged portfolio during uncertain times and a wasteful arbitrary purchase of derivatives based on market swings. There are multiple time horizon options for predicting volatility, both long-term and short-term offer different advantages and disadvantages. For this study, we will focus on one-day-ahead realized volatility to avoid exposure to macroeconomic events or events that are not contained within the time series data, and instead focus entirely on how previous returns and volatility can be used to predict future volatility. Given that short-term forecasting has shown to be more reliable than long-term forecasting due to reduced chance of unquantifiable events that are not reflected in the data, yet can affect the dependent variable (Nissi et al., 2020).

Additionally, it used to be very difficult and costly to access meaningfully large intraday price datasets to study short term market behavior. Today, there is more and more data made available to the general public by institutions as well as companies. This will likely lead to more accurate forecasts of future volatility.

6.2 Model implementation

All of the models used in this thesis are implemented in the open-source programming language known as R. We will start with the implementation of an ARMA model, to determine the number of lags we should use in the conditional mean of the GARCH models. The ARMA model is fitted to the realized volatility proxy, and we test with different orders for the AR and MA lags. We would prefer to use a parsimonious model to avoid overfitting, therefore we will start with a low order of lags. We examine the residual ACF plots to check if the residuals look like white noise. A time series is considered to be white noise if more than 95% of the lags in the ACF lie within $\pm 2/\sqrt{T}$, where T is the length of the series (Hyndman & Athanasopoulos, 2018). Additional AR and MA terms are added if the residual ACF plots display large significant spikes outside of the confidence interval, or if more than 5% of the lags are outside these bounds. We end up with selecting an ARMA(1,1) model, and the residual ACF plot for that model is shown in figure 6.1 below. The autocorrelation of lag zero will always be equal to 1 and can therefore be ignored. From the figure we can observe that there are three spikes that are slightly outside of the confidence bands. Based on these observations, we conclude that our series can be approximated as white noise.

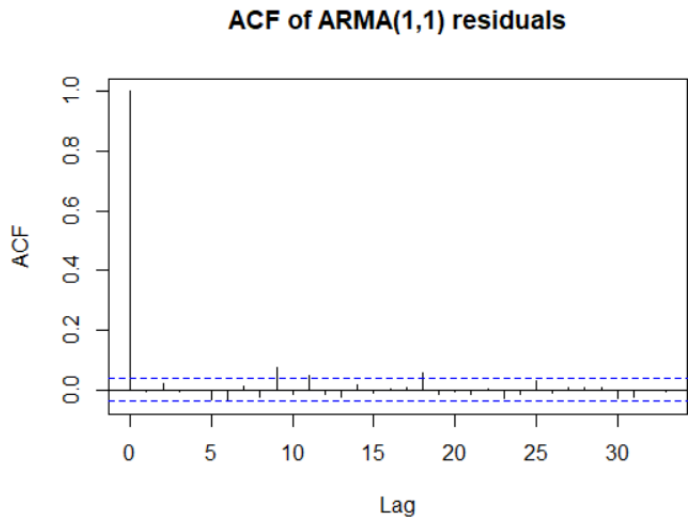


Figure 6.1: ACF plot of ARMA(1,1) residuals.

6.2.1 Implementation of GARCH models without external regressor

All of our GARCH models are built using the "rugarch" package by Ghalanos (2022), which is an open-source software created for univariate GARCH modelling. We are going to implement a standard GARCH model and an exponential GARCH model under different distributional assumptions. Then we will examine which of these model specifications has the best in-sample fit and later see which model performs the best at forecasting the volatility of the daily returns in the S&P 500 stock index. At first, we are going to see how the models perform without the inclusion of an external regressor in the variance equation. During our initial analysis of the conditional variance, we find that the models performed well with a low order of lags. As mentioned previously, we would prefer to select the more parsimonious model, thus we specify the number of GARCH and ARCH terms (p,q) to be equal to (1,1). Usually, the GARCH models do not benefit from additional lag terms, unless you have data that extends over a long period of time, such as several decades of daily data (Engle, 2001).

When we fit the models, we specify that we want to use the "hybrid" solver in order to avoid situations where the solver would fail to converge. The data is split into both training and test data to prevent overfitting. It is important that we have sufficient training data. A common way of dividing the data is by selecting 80% of the data for training and the remaining 20% as test data. In our study we specify that the last 488 observations in our data will be held out for out-of-sample forecasting. Thus, the remaining 2017 observations in the data set are selected as training data, which is being used for fitting the models.

Before we can move on to forecasting, we make sure that the standardized residuals resemble white noise. Figure 6.2 presents the correlograms of the standardized residuals and the squared standardized residuals of our standard GARCH(1,1) model. From inspecting the ACF plot of the standardized residuals we observe that the 17th lag is slightly outside of the confidence interval, but overall, we conclude that the series behave like white noise. We can also examine the correlogram of the squared standardized residuals, to see if the volatility model has sufficiently captured all of the persistence in the variance of the returns (Knight & Satchell, 2007). From the plot of the squared standardized residuals, we can observe a small spike at the 10th lag, but the majority of the lags are not significantly different from zero. Thus, we can conclude that the residuals behave like white noise and that our chosen model is

adequate. We also observe similar results when we check the ACF of the residuals for the EGARCH(1,1) model. Results from the EGARCH model can be found in Appendix C.

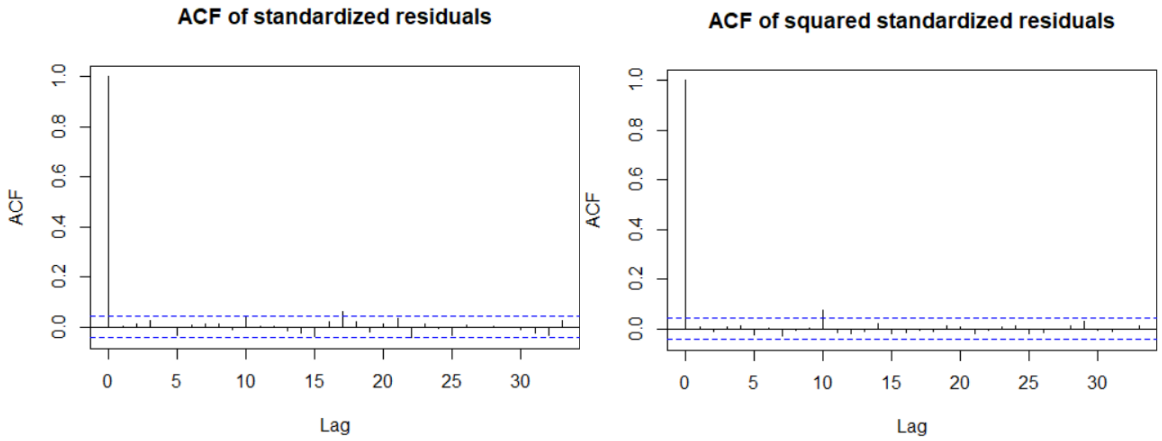


Figure 6.2: Correlograms of standardized residuals and the standardized squared residuals of an ARMA(1,1)-sGARCH(1,1) model.

The Ljung-Box test is another statistical tool we use to test for autocorrelation in the standardized residuals of our GARCH models (Tsay, 2013). The null hypothesis of the Ljung-Box test is that the residuals are independently distributed. The alternative hypothesis is that the residuals are not independently distributed and that they exhibit serial correlation. Table 6.1 presents the results from the Weighted Ljung-Box test, which is tested on various lags. From the table we can see that the p-values are not statistically significant for any of the lags tested. Thus, we fail to reject the null hypothesis and cannot conclude that the autocorrelations are significantly different from zero.

Table 6.1: Weighted Ljung-Box Test: (a) Standardized residuals, (b) Standardized squared residuals.

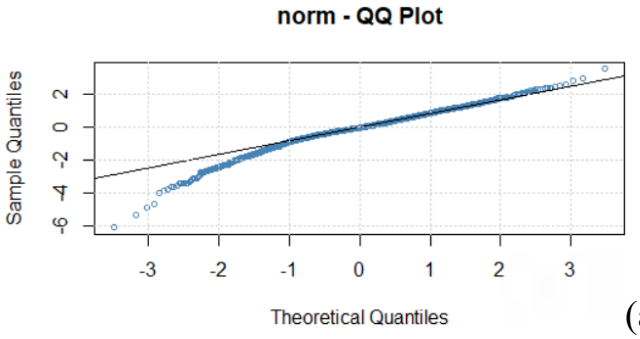
Lag	sGARCH		EGARCH	
	Statistic	p-value	Statistic	p-value
Lag[1]	0.0239	0.8771	0.2684	0.6044
Lag[5]	1.3438	0.9995	1.4610	0.9985
Lag[9]	2.5102	0.9507	2.5706	0.9448

(a)

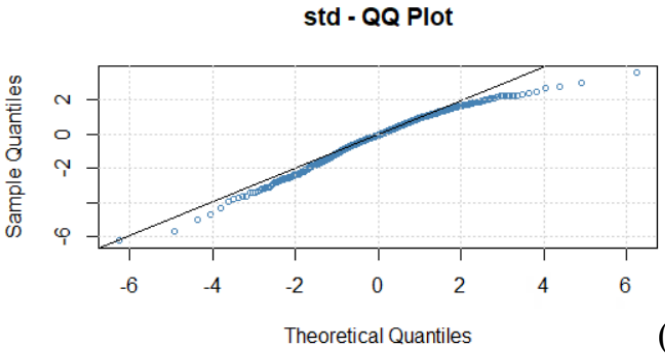
Lag	sGARCH		EGARCH	
	Statistic	p-value	Statistic	p-value
Lag[1]	0.1537	0.6951	0.5287	0.4672
Lag[5]	0.8290	0.8970	0.9872	0.8626
Lag[9]	1.9093	0.9155	1.6582	0.9416

(b)

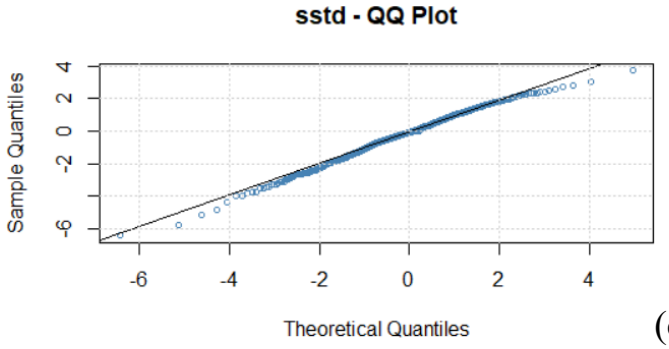
To assess the validity of the distributional assumptions in our models we can examine the quantile-to-quantile (Q-Q) plots of the standardized residuals (Tsay, 2013). If the sample quantiles and theoretical quantiles lie on a straight line, then we can conclude that the assumed distribution is correct. The first plot (a) in figure 6.3 displays a normal Q-Q plot, which is negatively skewed to the left. This clearly indicates that the standardized residuals are not normally distributed. In the second plot (b) we instead assume that the innovations follow a student-t distribution. In this case, we also see some deviations in the tails of the distribution. In the final plot (c) we assume that the residuals follow a skewed student-t distribution. We can observe that the majority of the data points lie close to the line, but there are still some outliers in the tails. Overall, the skewed student-t distribution seems to be the best fit. Similar observations were found for the EGARCH model, and these can be found in Appendix D.



(a)



(b)



(c)

Figure 6.3: Quantile-to-quantile plots for the standardized residuals. An ARMA(1,1)-sGARCH(1,1) model with different innovation distributions: (a) Gaussian, (b) student-t, and (c) skewed student-t.

6.2.2 Implementation of Artificial Neural Networks

Artificial Neural Networks were chosen as the representative model to test whether a combination of both GARCH type models and more advanced machine learning tools would lead to improved performance. Both standard ANN as well as RNN were included in the experiment.

The packages used to implement the artificial neural networks were TensorFlow and Keras, both libraries were originally developed for python, but have been ported into R packages. Each model with different inputs is hyperparameter tuned independently in order to find the best model parameters for each situation using the options shown in table 6.2, as opposed to using the same model specifications for all artificial neural networks. This is done because every different input could require a slightly different model.

Table 6.2: NN tuning options

Parameters	Tested options
Number of lags	2,5,10,20
Dense layer units	1,4,8,16,32,64,128
Activation	Linear, ReLu, Sigmoid
optimize learning rate	0.001, 0.01, 0.05, 0.01
callback patience	200, 400
loss monitor	MSE, MAE

No regularization techniques will be implemented in order to keep the models relatively simple. All models involved three layers, one of which is a hidden layer, but the option of only using two layers is also included in event that the problem did not benefit from having a hidden layer. Each model is given 1000 epochs and the best validation performance is chosen as the superior model to be used for the test set RMSE measurement.

In order to incorporate GARCH forecasts into the model, an additional GARCH forecast variable is added into a stacked model input to allow the model to take into consideration what GARCH predicts. Additionally, an ensemble model in the form of calculating the average prediction between the neural network without GARCH as input and GARCH predictions are used as a different way to combine the two different models.

6.2.3 Adding realized volatility as an external regressor

In this study, we are interested to know if the intraday returns can provide additional information that could benefit our existing forecasting models. We are going to investigate the effect of incorporating realized volatility as an external regressor in our models. The external regressor is essentially an explanatory variable that will be added to the variance equation of our GARCH models and will therefore have no impact on the conditional mean. Several previous studies, such as Zhang and Hu (2013), have explored this topic, however their findings are quite mixed. They find that for some stocks in the Chinese stock market, their models can benefit from the additional information contained in realized volatility, but for other stocks there seems to be no gain from incorporating the realized volatility measure.

The realized volatility we are going to use is calculated based on the 10-minute realized variance of the S&P 500 Index. When we add the external regressor to our models, we make sure that it is a time lag of one period. In this study we will compare the models with and without the realized volatility as an additional explanatory variable. We will first investigate the estimated fit of the models, and later compare the forecast performance of the various models.

6.3 Cross validation

To prevent overfitting, we use the hold-out cross validation approach. The data is split into 60% train data, 20% validation data, and 20% test data. The training data is used to find the best model fit for our dependent variable realized volatility, the validation set is used to evaluate and select which model performed best outside of the training data to avoid overfitting, and the final test set is reserved for evaluating our overall out-of-sample performance. The data will not be split randomly and instead based on chronological order, the training data being the first 60%, validation the next 20%, and the test set being the last 20% data points. Splitting the data in this fashion also allowed us to take advantage of models that incorporate recent predictions, such as RNN, to make a more accurate estimate of the future.

6.4 Model estimation

For all GARCH models, an expanding window is used in order to allow GARCH to use all data available until the current one-step ahead forecast. Neural networks on the other hand were given a choice between different moving windows from table 6.1. However, the predominant number of NN lags for all inputs is 5 lags per each different input for every moving window.

Both GRACH and neural network model estimation will be done only using training data to then compute a one-step ahead forecast on the test set without re-estimation. In other words, the model is given new lag windows to produce out-of-sample forecasts using the best performing models from the training and validation sets. For all ensemble and stacked neural network models which include GARCH forecasts as input, a moving window of one-step ahead forecasts is used instead of incorporating the fitted values as input (Ramos-Pérez et al., 2019).

6.5 Benchmark Model

The Benchmark model at all times is the equivalent neural network or recurrent neural network without GARCH forecasts as input. If a model that included GARCH forecast as input is able to outperform a neural network without GARCH, then this would be an important sign that GARCH models contain useful information unavailable to the neural networks. Additionally, to the non-GARCH neural networks an OLS regression is incorporated as a reference point of what a simple model would achieve under these circumstances. OLS is given all the same inputs as the Neural Networks and compared on equal terms. Even though it is likely that the OLS will fail to keep up with the machine learning models, it is important to track what could be achieved with one of the simplest models. All OLS regressions were also implemented using Keras.

6.6 Forecast evaluation

As previously stated, our data is divided into a training and test set. Forecast accuracy can be evaluated both in-sample and out-of-sample, but in general we are more interested in the out-of-sample accuracy. The reason is that we would like to consider how well a model performs when it is applied to the previously unseen test data that were not used when fitting the model.

6.6.1 Evaluation metrics

There exists many different statistical measures to calculate the accuracy of a model. The root mean squared error (RMSE) and mean absolute error (MAE) are two of the most commonly used metrics for evaluating forecasting models. These two measures can be formulated as the following:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{\sigma}_t - \sigma_t)^2} \tag{6.1}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|\hat{\sigma}_t - \sigma_t| \tag{6.2}$$

6.6.2 Diebold-Mariano test

We are also going to employ the Diebold-Mariano (DM) test to compare the forecast accuracy between some of our models. The DM tests whether two competing forecasts have equal forecasting accuracy (Diebold & Mariano, 1995). The null hypothesis is that the two forecasts have equal predictive accuracy. If the p-value from the pairwise DM test is significant, we can reject the null hypothesis, and conclude that the two forecasts have different forecast accuracy.

7. Results and discussion

7.1 In-sample results

Table 7.1 presents the estimated GARCH models without the addition of an external regressor. The estimation period that was used to obtain the in-sample results is from 01.03.2012 to 12.03.2020. The GARCH type models are estimated under different distributional assumptions, such as the normal distribution (norm), student-t distribution (std) and the skewed student-t distribution (sstd). It is also important to note that the p-values presented in the two tables below correspond to the robust (white) standard errors.

By inspecting the sGARCH models we can see that almost all of the estimated coefficients are significant on a 5% level, except for the omega parameter with the assumption of a student-t distribution or a skewed student-t distribution. We can also observe that the combined value of the parameters alpha and beta vary between 0.945 to 0.980. This implies that the volatility is highly persistent.

If we instead look at the EGARCH models we can see that all of the estimated parameters are significant on a 5% level. In the table we also display the akaike information criterion (AIC) for each model, and the models that perform the best are marked in bold. We can observe that the assumption of a student-t distribution leads to a substantial improvement over the normal distribution in terms of model fit. A smaller improvement can be seen when we move from the student-t distribution to the skewed student-t distribution. Both the sGARCH and EGARCH models perform the best with the skewed student-t distribution. However, the AIC values seem to be generally lower for the EGARCH models, which suggest that the models can benefit from capturing the asymmetric effects in the S&P 500 stock returns.

Table 7.1: Estimated coefficients for the S&P 500 return series, associated p-values and Akaike information criteria. GARCH model specifications without an external regressor.

Model	mu	ar1	ma1	omega	alpha1	beta1	gamma1	skew	shape	AIC
$sGARCH(1,1)_{norm}$	0,00068	0,95520	-0,98217	0,00001	0,21902	0,72593				
	(0,00000)	(0,00000)	(0,00000)	(0,000208)	(0,00000)	(0,00000)				-6,9614
$sGARCH(1,1)_{std}$	0,00074	0,93920	-0,97584	0,00000	0,21719	0,76258			4,92709	
	(0,00000)	(0,00000)	(0,00000)	(0,49937)	(0,00000)	(0,00000)			(0,00000)	-7,0330
$sGARCH(1,1)_{sstd}$	0,00066	0,92683	-0,97147	0,00000	0,19357	0,77255		0,87095	5,67247	
	(0,00000)	(0,00000)	(0,00000)	(0,46923)	(0,00000)	(0,00000)		(0,00000)	(0,00000)	**-7,0406**
$EGARCH(1,1)_{norm}$	0,00042	0,51836	-0,55825	-0,70571	-0,24796	0,92794	0,18142			
	(0,00003)	(0,00000)	(0,00000)	(0,00000)	(0,00000)	(0,00000)	(0,00000)			-7,0357
$EGARCH(1,1)_{std}$	0,00053	0,31054	-0,35313	-0,62037	-0,25819	0,93784	0,17907		6,15923	
	(0,00000)	(0,00000)	(0,00000)	(0,00000)	(0,00000)	(0,00000)	(0,00000)		(0,00000)	-7,0840
$EGARCH(1,1)_{sstd}$	0,00032	0,30149	-0,34851	-0,62980	-0,25262	0,93599	0,17401	0,84426	6,90588	
	(0,00519)	(0,00000)	(0,00000)	(0,00000)	(0,00000)	(0,00000)	(0,00000)	(0,00000)	(0,00000)	**-7,0973**

In table 7.2 we display the in-sample results of our GARCH models with the addition of realized volatility as an external regressor in the variance equation. The external regressor parameter is displayed as vxreg1 in the table, and as we can see the associated p-values are non-significant for all the standard GARCH models.

Let us first compare the estimated coefficients and p-values of the sGARCH models with the previous results from table 7.1. With the addition of an external regressor the coefficient estimates for the alpha and beta parameters tend to be lower, except for the sGARCH model with a student-t distribution as the p-value of the vxreg1 parameter is close to being 1. If we look at the associated AIC values, we can see some minor improvements for the sGARCH models. There is a bigger improvement in terms of AIC value when the external regressor is added to the sGARCH model with a skewed student-t distribution. However, based on the insignificant p-values for the external regressor we do not have much evidence that the external regressor adds any value to the sGARCH models.

Moving to the EGARCH models we can observe much higher coefficient values for the external regressor, and the corresponding p-values are highly significant (P<.001). The AIC values are also substantially lower compared to the EGARCH models which excludes the information contained in the realized volatility. This indicates that realized volatility provides

additional information to the estimation process of the EGARCH models. Again, we can observe that the best fit is obtained under the assumption of a skewed student-t distribution.

Table 7.2: Estimated coefficients for the S&P 500 return series, associated p-values and Akaike information criteria. GARCH model specifications with the inclusion of realized volatility as an external regressor.

Model	mu	ar1	ma1	omega	alpha1	beta1	gamma1	vxreg1	skew	shape	AIC
$sGARCH(1,1)_{norm}$	0,00057	0,96378	-0,98345	0,00000	0,16634	0,49540		0,00360			
	(0,00000)	(0,00000)	(0,00000)	(1,00000)	(0,00010)	(0,17829)		(0,30048)			-7,0080
$sGARCH(1,1)_{std}$	0,00074	0,93917	-0,97585	0,00000	0,21716	0,76236		0,00000		4,92465	
	(0,00000)	(0,00000)	(0,00000)	(0,00000)	(0,00000)	(0,00000)		(0,99996)		(0,00000)	-7,0320
$sGARCH(1,1)_{sstd}$	0,00060	0,92505	-0,96775	0,00000	0,14792	0,58839		0,00276	0,84887	6,54725	
	(0,00010)	(0,00000)	(0,00000)	(1,00000)	(0,23945)	(0,61044)		(0,78699)	(0,00000)	(0,10935)	**-7,0689**
$EGARCH(1,1)_{norm}$	0,00028	-0,17156	0,14063	-1,96028	-0,24186	0,82310	0,05291	35,09878			
	(0,04435)	(0,00000)	(0,00000)	(0,00000)	(0,00000)	(0,00000)	(0,01650)	(0,00000)			-7,0631
$EGARCH(1,1)_{std}$	0,00048	-0,11539	0,08252	-2,05456	-0,26612	0,81692	0,03134	38,72127		6,75952	
	(0,00003)	(0,78292)	(0,84230)	(0,00000)	(0,00000)	(0,00000)	(0,11101)	(0,00000)		(0,00000)	-7,1091
$EGARCH(1,1)_{sstd}$	0,00027	-0,01651	-0,01614	-1,97199	-0,25708	0,82284	0,03448	36,16123	0,84058	7,86797	
	(0,05578)	(0,21057)	(0,15851)	(0,00000)	(0,00000)	(0,00000)	(0,11733)	(0,00000)	(0,00000)	(0,00000)	**-7,1222**

7.2 Out-of-sample evaluation

7.2.1 Forecast evaluation for GARCH models

In this section, we will present the results of the out-of-sample forecasts, in which we will compare the different GARCH models. The best performing GARCH specification with and without an external regressor will be selected to be included into both the standard ANN and RNN models. We compute the RMSE and MAE of the different GARCH specifications, and the results are shown in table 7.3. The models that perform the best out-of-sample are highlighted using bold numbers. Let us first consider the GARCH models without the realized volatility as an explanatory variable. We can observe that the EGARCH(1,1) model with a normal distribution provides the lowest values in terms of both RMSE and MAE.

When the realized volatility is introduced in the variance equation of our GARCH models, we obtain even lower RMSE and MAE measures for some of the models. The model that gives the lowest error metrics is the sGARCH(1,1) model with a normal distribution. We find that the models with the lowest out-of-sample performance do not correspond to the models which

had the best in-sample fit. However, a good fit does not necessarily lead to good out-of-sample forecasting performance (Hyndman & Athanasopoulos, 2018). A possible explanation for this behavior could be the bias-variance tradeoff (James et al, 2013). The EGARCH model with a skewed student-t distribution is a more flexible model, as it has a greater number of parameters that needs to be estimated. As the flexibility of a model increases, the variance will increase and the bias decreases. This could potentially lead to the risk of overfitting the data, which means that a model fits too closely to the training data, while performing poorly on the test data.

Table 7.3: Out-of-sample forecast performance for the different GARCH models.

	Excl. external regressor		Incl. external regressor	
Model	RMSE	MAE	RMSE	MAE
$sGARCH(1,1)_{norm}$	0.00569	0.00396	**0.00368**	**0.00292**
$sGARCH(1,1)_{std}$	0.00650	0.00440	0.00649	0.00440
$sGARCH(1,1)_{sstd}$	0.00610	0.00417	0.00370	0.00297
$EGARCH(1,1)_{norm}$	**0.00474**	**0.00333**	0.03938	0.00946
$EGARCH(1,1)_{std}$	0.00549	0.00362	0.05205	0.01167
$EGARCH(1,1)_{sstd}$	0.00511	0.00349	0.04492	0.01048

A visual representation of the best performing GARCH models against the realized volatility can be shown in figure 7.1. We can observe that both the EGARCH(1,1) model and the sGARCH(1,1) model with the addition of realized volatility seem to overestimate the future volatility. We can also clearly see some delay in the predictions for the GARCH models, which is to be expected. Upon further inspection of the graphs, we can also observe that the GARCH models fail to properly capture the large spikes in future volatility.

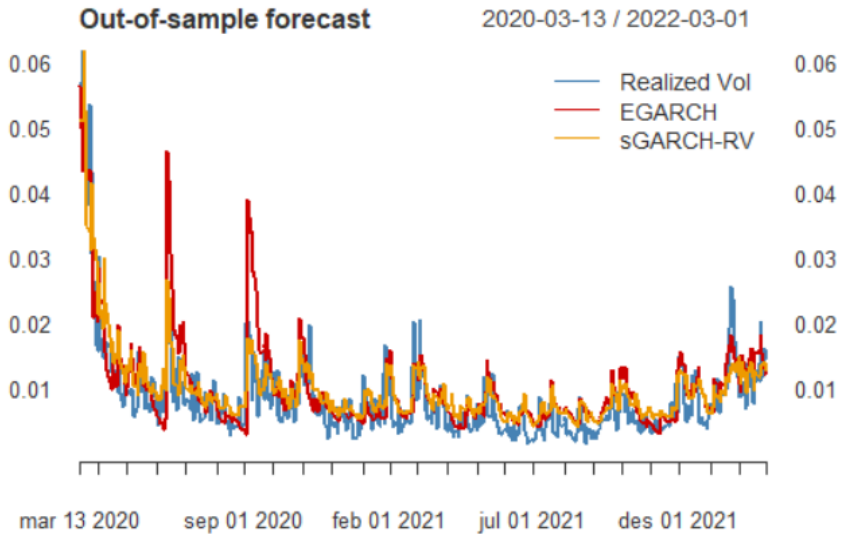


Figure 7.1: Comparing the out-of-sample forecasts of the best performing GARCH models against the realized volatility. sGARCH-RV is an abbreviation for the best performing sGARCH model with the realized volatility included.

7.2.2 Forecast evaluation excluding external regressor

Table 7.4 contains the best performing models calculated exclusively using log returns as input. The table also includes the best performing neural networks, as well as the neural network GARCH ensemble models. The GARCH model incorporated into the neural network models is the EGARCH(1,1) normal distribution from table 7.3, as it is the best performing model between all the GARCH type models that did not include the external regressor.

Table 7.4: Out-of-sample results excluding external regressor.

	Excl. external regressor	
Model	RMSE	MAE
OLS	0.00696	0.00417
$EGARCH(1,1)_{norm}$	0.00474	0.00333
NN	0.00383	0.00255
RNN	**0.00375**	**0.00250**
Stacked GARCH NN	0.00396	**0.00250**
Stacked GARCH RNN	0.0038	0.00251
Averaging GARCH NN	0.00376	0.00264
Averaging GARCH RNN	0.00377	0.00267

We can see in table 7.4 that the best performing model is the RNN that does not include GARCH as part of its input. However, the RNN results are only slightly better than the other models. Based on these findings, we can conclude that artificial neural networks do perform better than GARCH models when it comes to calculating daily realized volatility. Regarding our study on whether incorporating GARCH output into an artificial neural network, there is no evidence to conclude that GARCH models are able to add any additional value. The artificial neural network models were able to capture the information in the S&P 500 log returns on their own. It is important to note that even though the best performing model is a regular RNN without any additional data, the second and third best performing models in regard to RMSE were the averaging NN GARCH, followed by the averaging RNN GARCH. Additionally, when looking at the MAE, the best performing models were instead the stacked GARCH NN and the RNN, followed by the Stacked GARCH RNN.

Even though the ensemble approach did not produce the best results and did not show any signs of containing information that is unavailable to the regular artificial neural networks, the ensemble models did consistently produce some of the best performances.

7.2.3 Forecast evaluation including external regressor

To make use of all data available within the time series and achieve the best possible realized volatility predictions, we repeat the previous exercise and include the lags of realized volatility itself as input into our models.

Table 7.5: Out-of-sample results including external regressor.

	Incl. external regressor	
Model	RMSE	MAE
OLS	0.00369	0.00250
$sGARCH(1,1)_{norm}$	0.00368	0.00292
NN	0.00362	0.00240
RNN	0.00352	0.00247
Stacked GARCH NN	0.00362	**0.00239**
Stacked GARCH RNN	0.00352	0.00248
Averaging GARCH NN	**0.00324**	0.00240
Averaging GARCH RNN	0.00343	0.00260

Table 7.5 shows that every model performed better when including realized volatility lags as part of the input. The results also reveal that when we use all information available within the time series, we are able to obtain better forecasting accuracy by using ensemble models. The best performing RMSE model is the ensemble model which combines the outputs of GARCH and a deep neural network, while the best MAE performance is achieved by the stacked NN GARCH model. In both cases, the best results were achieved by an ensemble model. In the case of the RMSE results, the second-best model is the ensemble model which combines the outputs of GARCH and a recurrent neural network.

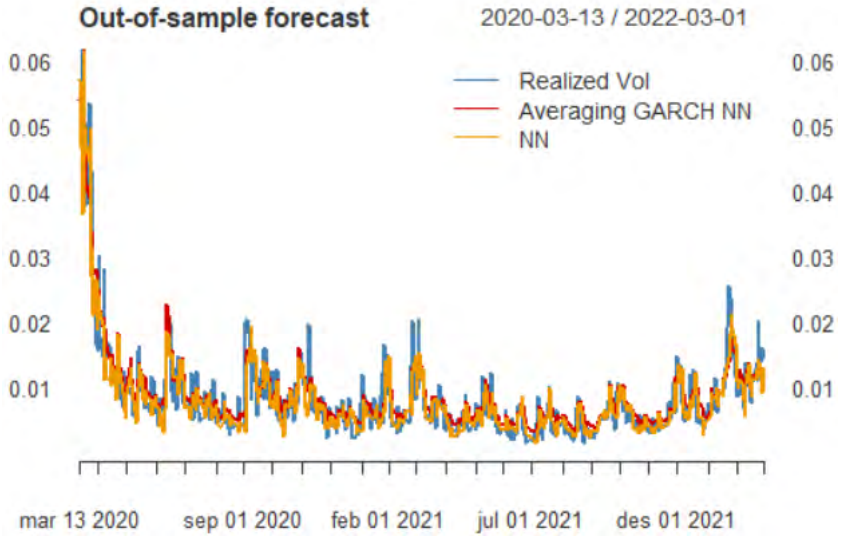


Figure 7.2: Comparison of out-of-sample forecasts for the NN model and averaging GARCH NN model including the external regressor, against the realized volatility proxy.

We can see from figure 7.2 that our best performing model, the Averaging GARCH NN, tends to overestimate volatility when compared with the regular NN. This makes sense because the Averaging GARCH NN is the average forecast between the NN, and the sGARCH-RV shown in figure 7.1. sGARCH-RV has the tendency to overshoot its volatility forecasts when there is a volatility spike. This could be interpreted as a weakness due to the worsened performance. However, when sGARCH-RV is combined with a regular NN, which has the tendency to underestimate volatility spikes, we end up with a better model which can more accurately predict volatility spikes, as seen in table 7.2.

Table 7.6: Pairwise Diebold-Mariano tests**.**

Pairwise DM tests	Coef.	p-value
NN vs Stacked GARCH NN	-1.3075	0.1917
RNN vs Stacked GARCH RNN	-1.7312	0.0912
NN vs Averaging GARCH NN	4.2656	0.0000
RNN vs Averaging GARCH RNN	4.6654	0.0000

In order to get a better understanding of what our forecasts are actually doing, we employ the Diebold-Mariano test to determine if the forecast accuracy of the best ensemble models are significantly different from the forecast accuracy of the NN and RNN models. In table 7.6 we can see the results from the pairwise DM tests. From the table we can observe that the p-value for the stacked models are insignificant on a 5% level. This means that we accept the null hypothesis of equal forecasting accuracy. As for the ensemble models, the p-values are significant on a 5% level, hence we reject the null hypothesis that the pairwise forecasts have equal accuracy. This result suggest that the averaging ensemble models are performing better than the neural network models. However, we do not have enough evidence to say that the stacked models are better than the neural network models.

8. Conclusion

We set out to explore if combining GARCH type models and artificial neural networks can lead to better performance than when the different models working independently and relying exclusively on the data within the time series. Two different ensemble methods were used to combine the different models' predictive capabilities. One method was a stacked artificial neural network which included GARCH forecasts as part of its input, and the other was an averaging ensemble of the outputs of the best performing GARCH model and the outputs of an artificial neural network.

Regarding forecasting realized volatility relying exclusively on log return lags, we were unable to find any evidence that an ensemble model could take advantage of the additional information introduced by incorporating GARCH forecasts into an artificial neural network. In other words, the artificial neural networks could, without any additional information besides past log returns, achieve better results than any of the GARCH type models. However, it is important to note that our best performing model for both RMSE and MAE was the recurrent neural network, which is known to be capable of outperforming regular artificial neural networks due to its additional attribute of being able to incorporate the previous prediction as additional information to produce better forecasts, although the better forecasting performance of the RNN was only slightly better than the second-best performing model.

When it comes to incorporating realized volatility lags as input for our models, we found that all out-of-sample forecasting performance for models including external regressor was better than when not including the external regressor. Additionally, we found evidence to suggest that ensemble artificial neural networks have the potential to outperform artificial neural networks and GARCH type. Overall, the ensemble models consistently outperformed the regular artificial neural networks regarding out-of-sample forecast RMSE, and somewhat consistently for the out-of-sample forecast MAE. More specifically, the best RMSE model was the Averaging GARCH NN, and the best MAE model was the Stacked GARCH NN. Moreover, we were able to, with the use of the Diebold-Mariano test, reject the null hypothesis and conclude that the ensemble models were making statistically different forecasts than the original models from which they were derived.

Regarding whether ensemble models were superior to regular artificial neural networks, we did not find conclusive evidence that ensemble models were the ideal approach. We did however identify a tendency for ensemble models to be among the top performing models. This does however make sense because any subtle improvement to volatility forecasting should be extremely difficult and would require a much more powerful model than simply adding a single GARCH type model to a machine learning model. It is important to note however that, when including the external regressor, we encountered consistent evidence that ensemble models, with just a simple ensemble, on average performed better than artificial neural network models.

9. Further research

Based on our findings, we were able to conclude that there is potential for more accurate estimations of future realized volatility with the use of ensemble models. Given that we only used GARCH forecasts are additional regressors into our models, and we only worked with S&P 500 daily returns, there is an immense potential for different approaches and objectives.

One important addition would be to include additional machine learning models, such as XGBoost, and support vector machines. This could be done both parallel to neural networks and in the form of a larger ensemble model that seeks to take advantage of the strengths of different machine learning techniques, as well as reduce the effect of any individual model.

Furthermore, exploring different indexes, commodities, and individual stocks could lead to completely different results and perhaps even more interesting findings that would not be present in a highly liquid and not so volatile indexes, such as the S&P500.

References

Aït-Sahalia, Y., Yu, J. (2009). *High Frequency Market Microstructure Noise Estimates and Liquidity Measures*. The Annals of Applied Statistics. Vol. 3, No. 1, 422-457.

Andersen, T. G., Bollerslev, T. *Answering the Skeptics: Yes, Standard Volatility Models Do Provide Accurate Forecasts*. Int. Econ. Rev. 1998, 39, 885–905.

Andersen, T. G., Benzoni, L. (2008). *Realized Volatility*. FRB of Chicago, Working Paper No. 2008-14.

Barndorff-Nielsen, O. E., Shephard, N. (2002). *Estimating quadratic variation using realized variance*. Journal of applied econometrics. Vol. 17(5), 457-477.

Black, F., Scholes, M. (1973). *The Pricing of Options and Corporate Liabilities*. Journal of Political Economy, 8, 637-654.

Black, F. (1976). *Studies of stock price volatility changes. In: Proceedings of the 1976 Meetings of the American Statistical Association*. 171-181.

Bodie, Z., Kane, A., Markus A. J. (2021). *Investments* (12th ed.). McGraw-Hill Education, 159-212.

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

Bollerslev, T., Engle, R. F., Nelson, D. B. (1994). *ARCH Models*. Handbook of Econometrics, 4, 2959-3038.

Bottou, L., Bousquet, O. (2008). "*The tradeoffs of large scale learning*". Advances in neural information processing systems, 161–168

Brockwell, P. J., Davis, R. A. (1991). *Time Series: Theory and Methods*. Springer-Verlag. 273-329.

Bucci, A., 2020, "*Realized volatility forecasting with neural networks*," Journal of Financial Econometrics, 18(3), 502–531.

Carr, P., Wu, L., Zhang, Z. (2020). *Using machine learning to predict realized variance. Journal of investment management*, vol 18, 1-16.

CBOE. (2021). *Cboe Volatility Index*. [White paper]. Cboe Exchange, Inc. https://cdn.cboe.com/resources/vix/vixwhite.pdf

Christie, A., A. (1982). *The stochastic behavior of common stock variances: Value, leverage and interest rate effects*. Journal of financial economics, vol. 10, 407-432.

Charef, F., Ayachi, F. (2016). *A comparison between neural networks and GARCH models in exchange rate forecasting*. International Journal of Academic Research in Accounting, Finance and Management Sciences, 6(1). https://doi.org/10.6007/ijarafms/v6-i1/1996

Corsi. (2009). "*A Simple Approximate Long-Memory Model of Realized Volatility*," Journal of Financial Econometrics 7(2), 174-196.

Danielsson, J. (2011). *Financial Risk Forecasting: The Theory and Practice of Forecasting Market Risk, with Implementation in R and Matlab*. John Wiley & Sons, inc.

Dayhoff, J. E., DeLeo, J. M. (2001). *Artificial neural networks: Opening the Black Box*. Cancer, 91(S8), 1615-1635. https://doi.org/10.1002/1097-0142(20010415)91:8+<1615::aid-cncr1175>3.0.co;2-l

Diebold, F. X., Mariano, R. (1995). *Comparing Predictive Accuracy.* Journal of Business and Economic Statistics, vol. 13, 253-265.

Disorntetiwat, P., Dagli, C. (2000). *Simple ensemble-averaging model based on generalized regression neural network in financial forecasting problems.* Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373). https://doi.org/10.1109/asspcc.2000.882522

Engle, R. F. (1982). *Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation*. Evanston: The Econometric Society, 987–1007.

Engle, R. F. (2001). *GARCH 101: An Introduction to the Use of ARCH/GARCH models in Applied Econometrics*. https://web-static.stern.nyu.edu/rengle/GARCH101.PDF

Engle, R. F., Patton, A. (2001). *What Good is a Volatility Model?* NYU Working Paper No. S-DRP-01-03, Available at SSRN: https://ssrn.com/abstract=1296430

Fama, E. (1965). *"The Behavior of Stock Market Prices,"* Journal of Business 38, 34-105.

Francq, C., Zakoian, J. (2010). *GARCH Models: Structure, Statistical Inference and Financial Applications* (2nd ed.) John Wiley & Sons, inc.

Ge, W., Lalbakhsh, P., Isai, L., Lenskiy, A., Suominen, H. (2022). Neural Network–Based Financial Volatility Forecasting: A Systematic Review. ACM Computing survey, vol. 55, No. 14, 1-30.

Gers, F. A., Eck, D., Schmidhuber, J. (2002). Applying LSTM to time series predictable through time-window approaches. Perspectives in Neural Computing, 193-200. https://doi.org/10.1007/978-1-4471-0219-9_20

Ghalanos, A. (2022). *rugarch: Univariate GARCH Models*. Version 1.4-8. https://cran.r-project.org/web/packages/rugarch/rugarch.pdf

Ghysels, E., Santa-Clara, P., Valkanov, R. (2006). Predicting volatility: Getting the most out of return data sampled at different frequencies. Journal of Econometrics, 131(1-2), 59-95. https://doi.org/10.1016/j.jeconom.2005.01.004

Harvey, D., Leybourne, S., Newbold, P. (1998). *Tests for Forecast Encompassing*. Journal of Business & Economic Statistics, vol. 16, No. 2, 254-259.

Hansen, P., Lunde, A. (2005). *A forecast comparison of volatility models: does anything beat a GARCH(1,1)?* Journal of Applied Econometrics, vol. 20, 873-889.

Hölldobler, S., Möhle, S., Tigunova, A. (2017). Lessons Learned from AlphaGo. YSIP 92-101.

Hyndman, R. J., Athanasopoulos, G. (2018). *Forecasting principles and practice* (2nd ed.). OTexts.

James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. Springer Texts in Statistics.

Kingma, D. P., Ba, J. (2014). *Adam: A method for stochastic optimization*. https://arxiv.org/abs/1412.6980

Kim, H. Y., Won, C. H. (2018). *Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models*. https://www.sciencedirect.com/science/article/pii/S0957417418301416

Knight, L. J., Satchell, S. (2007). *Forecasting Volatility in the Financial Markets* (3rd ed.). Elsevier, Science & Technology. https://ebookcentral-proquest-com.ezproxy.nhh.no/lib/nhh-ebooks/reader.action?docID=287974

López-García, M., Sánchez-Granero, M., Trinidad-Segovia, J., Puertas, A., De las Nieves F. (2021). *Volatility Co-Movement in Stock Markets*. Mathematics, 9, 598.

Mandelbrot, B. (1963). "*The Variation of Certain Speculative Prices*," Journal of Business 36, 394-419.

McCulloch, W. S., Pitts, W. (1943). *A logical calculus of the ideas immanent in nervous activity*. The bulletin of mathematical biophysics, vol. 5, 115–133.

Nelson, D. B. (1991). *Conditional Heteroskedasticity in Asset Returns: A New Approach*, Econometrica, vol. 59, 347-370.

Nissi, J., Småros, J., Ylinen, T., Ala-Risku, T. (2020). Measuring forecast accuracy: The complete guide. RELEX Solutions. https://www.relexsolutions.com/resources/measuring-forecast-accuracy/

Parkinson, M. (1980). *The Extreme Value Method for Estimating the Variance of the Rate of Return*. The Journal of Business, vol. 53, 61-65.

Patton, A. J. (2011). *Volatility Forecast Comparison Using Imperfect Volatility Proxies*. Journal of Econometrics, vol. 160, 246-256.

Poon, S., Granger, C., W. (2003). *Forecasting Volatility in Financial Markets*: A Review. https://faculty.washington.edu/ezivot/econ589/PoonGrangerJELsurvey.pdf

Ramos-Pérez, E., Alonso-González, P. J., & Núñez-Velázquez, J. J. (2019). *Forecasting volatility with a stacked model based on a hybridized artificial neural network*. Expert Systems with Applications, 129, 1-9. https://doi.org/10.1016/j.eswa.2019.03.046

Rogers, L. C. G., Satchell, S. E. (1991). *Estimating Variance From High, Low and Closing Prices*. The Annals of applied probability, Vol. 4, 504-512.

Schumann, J., Gupta, P., Liu, Y. (2010). *Application of Neural Networks in High Assurance Systems: A Survey*. Springer. 1-19.

Spyridon D. Vrontos, John Galakis & Ioannis D. Vrontos (2021) *Implied volatility directional forecasting: a machine learning approach, Quantitative Finance*, 21:10, 1687-1706, DOI: 10.1080/14697688.2021.1905869.

Triacca, U. (2007). *On the variance of the error associated to the squared return as proxy of volatility*. Applied financial economics letters, vol. 3, 255-257.

Tsay, R. S. (2013). *An introduction to analysis of financial data with R.* Wiley.

Vortelinos, D., (2017). *Forecasting realized volatility: HAR against principal components combining, neural networks and GARCH*. Research in international Business and Finance, Elsevier, vol. 39 (PB), 824-839.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., Recht, B. (2017). *The Marginal Value of Adaptive Gradient Methods in Machine Learning*. arXiv:1705.08292v2

Zhang, J., Hu, W. (2013). *Does realized volatility provide additional information?* International journal of managerial finance, Vol. 9. Bradford: Emerald Group Publishing Limited.

Appendix

Appendix A: General process of forecasting using an ARIMA model (Hyndman & Athanapoloulos, 2018).

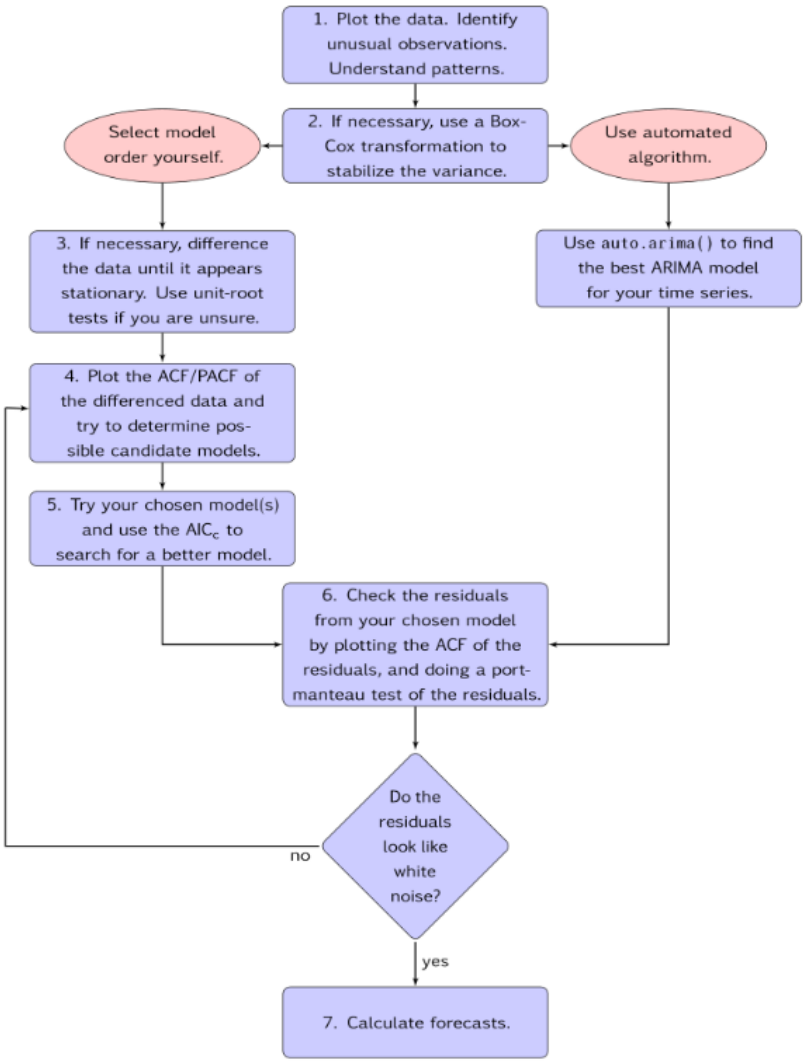


Figure A.1: General process of forecasting

Appendix B: AIC values for sGARCH(1,1) model

Normal distribution

Variance model	ArmaOrder	AIC	Log-likelihood
sGARCH(1,1)	0,0	-6,956	7019,28
sGARCH(1,1)	1,0	-6,957	7020,62
sGARCH(1,1)	0,1	-6,957	7020,66
sGARCH(1,1)	1,1	-6,961	7026,60
sGARCH(1,1)	2,1	-6,961	7027,02
sGARCH(1,1)	1,2	-6,961	7027,11
sGARCH(1,1)	2,2	-6,958	7025,21

Student-t distribution

Variance model	ArmaOrder	AIC	Log-likelihood
sGARCH(1,1)	0,0	-7,024	7088,43
sGARCH(1,1)	1,0	-7,025	7091,08
sGARCH(1,1)	0,1	-7,026	7091,18
sGARCH(1,1)	1,1	-7,033	7099,78
sGARCH(1,1)	2,1	-7,033	7100,57
sGARCH(1,1)	1,2	-7,033	7100,75
sGARCH(1,1)	2,2	-7,032	7100,58

Skewed student-t distribution

Variance model	ArmaOrder	AIC	Log-likelihood
sGARCH(1,1)	0,0	-7,028	7093,88
sGARCH(1,1)	1,0	-7,031	7098,09
sGARCH(1,1)	0,1	-7,032	7098,43
sGARCH(1,1)	1,1	-7,041	7108,45
sGARCH(1,1)	2,1	-7,041	7110,11
sGARCH(1,1)	1,2	-7,042	7110,43
sGARCH(1,1)	2,2	-7,041	7110,47

Appendix C: ACF plot of standardized residuals and squared standardized residuals for the EGARCH model.

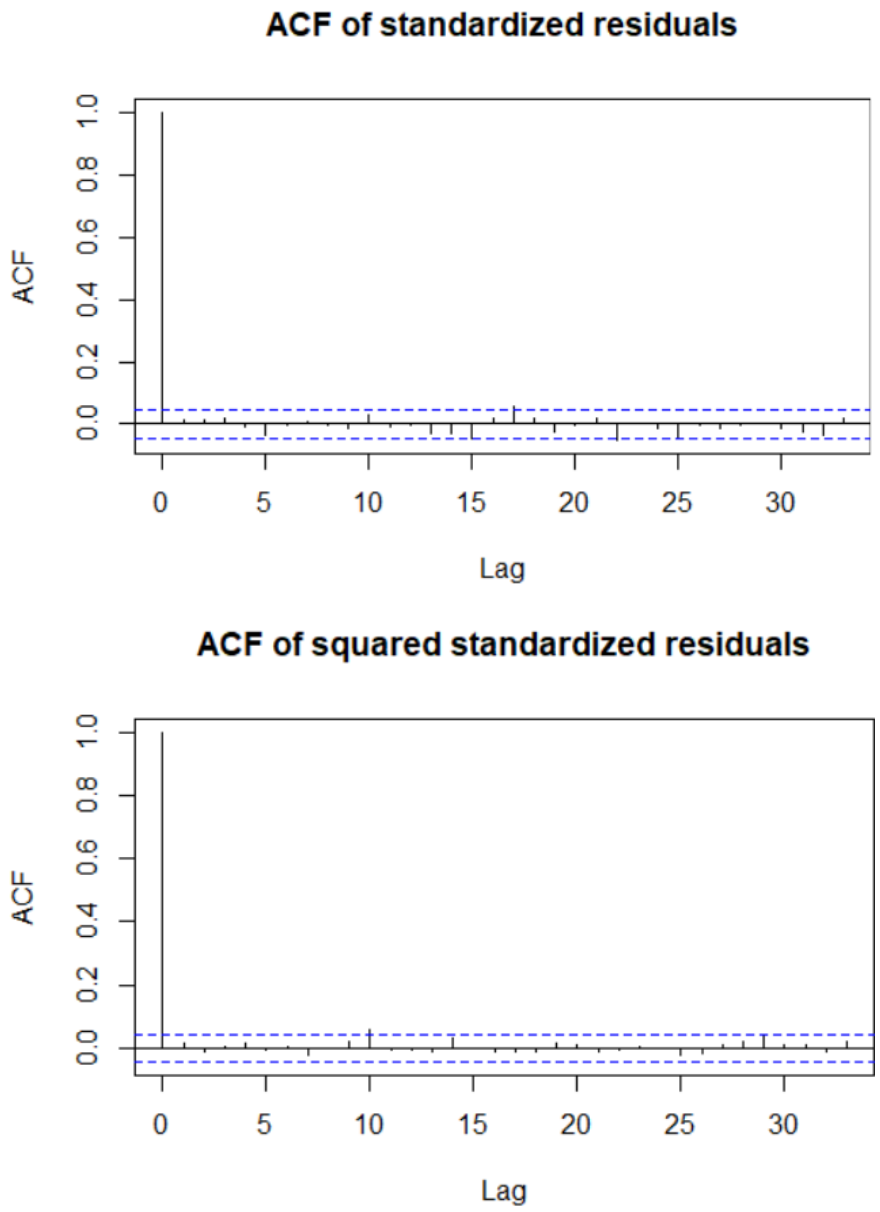
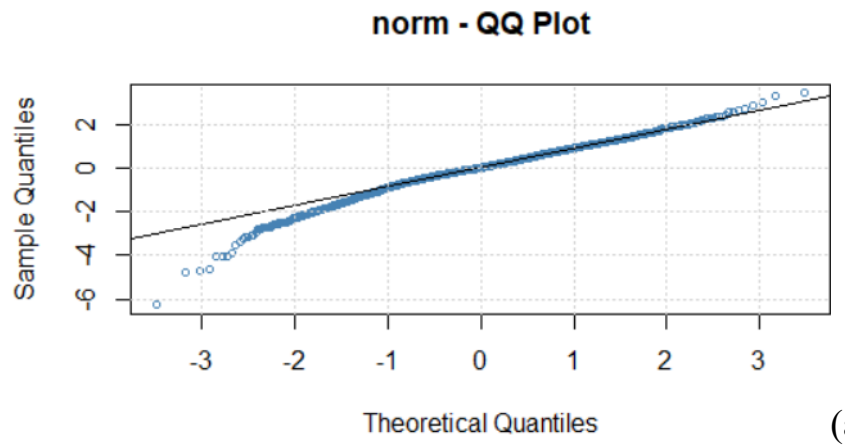
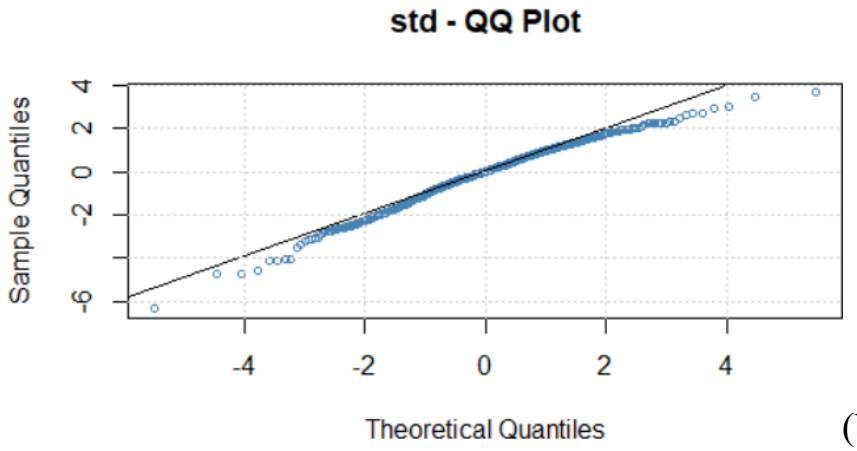


Figure C.1: Correlograms of standardized residuals and the squared standardized residuals of an ARMA(1,1)-EGARCH(1,1) model.

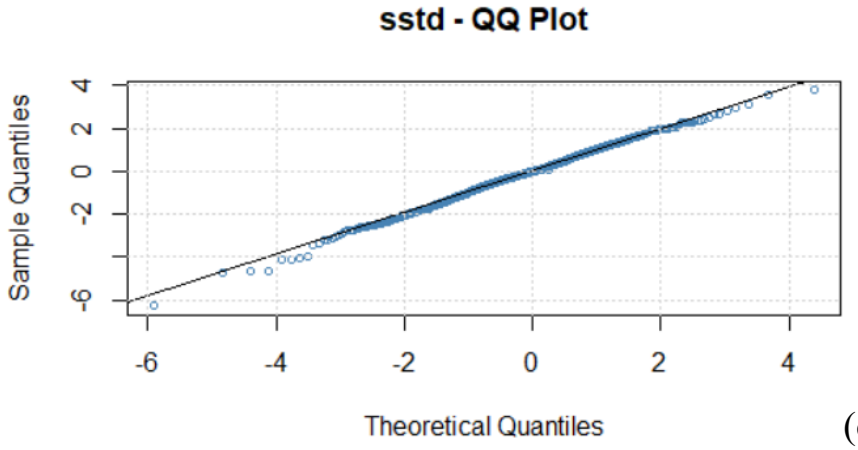
Appendix D: Q-Q plots for the EGARCH model.



(a)



(b)



(c)

Figure D.1: Quantile-to-quantile plots for the standardized residuals. An ARMA(1,1)-EGARCH(1,1) model with different innovation distributions: (a) Gaussian, (b) student-t, and (c) skewed student-t.