LUND UNIVERSITY

LUNDS UNIVERSITY - DEPARTMENT OF ECONOMICS

MASTER THESIS

Evaluating VaR and ES for commodities - both conventionally and with neural networks

Måns EILE

David FANG

Supervisor: Birger Nilsson

NEKH02 - Master in Finance: Master Thesis

June 22, 2020

Abstract

As commodities are becoming more popular and accessible assets for speculative and hedging purposes, the limited research regarding risk management for said asset-class justifies further contribution to the deficient output. Many previous studies have highlighted the extraordinary high volatility, with non-linear and clustering characteristics associated with commodities. Hence, incorporating volatility forecasts in risk management seems warranted. As the standard risk measurements for market risk in the last decades have been Value at Risk (VaR) and Expected Shortfall (ES), these metrics are evaluated based on a Volatility Weighted Historical Simulation with volatility forecasts provided by a GARCH(1,1) approach and a Recurrent Neural Network (LSTM) approach for oil, gold and soybean. The data period spans from 1990 to 2019 and the results indicate that both approaches work remarkably well in estimating both VaR and ES. In general, the GARCH(1,1) approach displays somewhat more accurate VaR estimates according to Kupiecs test. However, The LSTM approach does spread the violations more adequate according to Christoffersen's test. Both approaches display very well specified ES estimates, with somewhat better test statistics for the GARCH(1,1) approach according to the "*Testing ES directly*"-test proposed by Acerbi & Szekely.

Keywords: Value-at-Risk, Expected Shortfall, Commodities, GARCH(1,1), ANN, LSTM, Volatility forecasting, VWHS

Acknowledgement

Birger Nilsson - We would like to express our sincerest appreciation to our thesis supervisor for providing guidance and thorough feedback throughout the thesis.

Anders Wilhelmsson - For giving us the inspiration and idea for our thesis subject.

Ella Klynning, Johan Lindberg and Malin Wahlberg - We would also like to thank fellow student colleagues at Lund University for their continuous support and encouragement.

Contents

Abbreviations

- **ANN** - Artificial neural network(s)
- **AR** - Autoregressive
- **ARCH** - Autoregressive conditional heteroscedasticity
- **ES** - Expected shortfall
- **FNN** - Feedforward neural network(s)
- **FRTB** - Fundamental review of the trading book
- **GARCH** - Generalized autoregressive conditional heteroscedasticity
- **LSTM** - Long short-term memory
- **MAD** - Mean absolute deviation
- **MAE** - Mean absolute error
- **MSE** - Mean squared error
- **ML** - Maximum likelihood
- **RNN** - Recurrent neural network(s)
- **SGD** - Stochastic gradient decent
- **SPGSCI** - Standard & Poor's Goldman Sachs commodity index
- **VaR** - Value at risk
- **VWHS** - Volatility weighted historical simulation

Terminology

There are many technical terms used in conjunction with neural networks. Some keywords are presented here that will be frequently used throughout the essay.

- **(Artificial) neural network** - A system or collection of artificial neurons, which are loosely based on a human neuron, that learn to perform certain tasks without being programmed with task specific rules.
- **Input/Visible layer** - Refers to the input variables.
- **Hidden node/Neuron** - Each artificial neuron is often referred to as a hidden node or simply a neuron. They are called 'hidden' as they are not directly observable.
- **Hidden layers** - A layer or multiple layers in between input layer and output layer, where neurons take in a set of inputs and generate an output. The input can come from the input layer or from a previous layer of hidden nodes. A graphical representation of two hidden layers is shown in Figure 3 on page 12.
- **Output layer** - The last layer of the network, consisting of the output variables.
- **Weight(s)** - All inputs to an artificial neuron carries an associated weight, which can be thought of as a way for the model to determine the importance of different input variables. These weights represent the influence of each parameter and are optimized when training a neural network.
- **Hyperparameter** - Parameter whose value is set before model training commences, such as number of hidden layers, activation function and drop out rate. Input weights are *not* hyperparameters as they are continuously fine tuned when training the model.

1 Introduction

The importance of risk management became fundamentally clear in the financial meltdown of 2008. Value at Risk (VaR), the most widely adopted risk measurement at the time, had experienced great criticism for its deficiency in accounting for tail events and its lack of subaditivity[1]. Following the evident shortcomings, the Basel committee proposed a transition from VaR to Expected Shortfall (ES) in their consultative document "*A Fundamental Review of the Trading Book*" (BIS, 2013). While there are stipulated criteria that financial institutions have to meet, there are some discretion in the methodology adopted when estimating the risk measures. This has opened up the scene for academics to investigate different methodical approaches in estimating VaR and ES. In this area of research, the most commonly investigated financial assets are stocks, bonds and exchange rates.

Another investable asset class that has seen an immense increase in investing activity lately is commodities. This has, according to Buyuksahin, Haigh, and Robe (2009), come with a desire to investigate the characteristics of commodities in comparison to that of the traditional financial assets. Utilizing dynamic correlation and recursive cointegration methods on SP500 and SPGSCI (the first major investable commodity index, introduced by Goldman Sachs), the authors find that the asset classes seem dis-synchronized. Further, the relationship does not seem to have changed significantly in the past 15 years, despite the increased trading activity and availability of instruments. Hence, they conclude that commodities continue to offer benefits in regards to portfolio diversification. In addition to the seemingly absent correlation between commodities and stocks, Deaton and Laroque (1992) propose in their paper "*On the Behaviour of Commodity Prices*" evidence of non-linear characteristics and "*extreme volatility*" in the price mechanics of commodities. Many studies[2] have since affirmed these findings of exceptionally high volatility and further highlighted the well known phenomenon of volatility clustering to be present for commodities.

One conventional way to deal with clustering and non-linear volatility is the Generalized Autoregressive Conditional Heteroscedasticity (GARCH) model introduced independently by Bollerslev (1986) and Taylor (1986). There are several different types of volatility models within the ARCH-family. Hansen and Lunde (2005) found that GARCH(1,1) in general[3] has the best performance out of all the different ARCH-models. The main drawback in utilizing GARCH to model volatility with non-linear characteristics is that it is not model free in that it forces an explicit relationship onto the data. This comes with

[1] Subadditivity is a condition for Coherency, see Appendix A for the definition of Coherency.

[2] See for instance OECD (1993), Giot and Laurent (2003), Vivian and Wohar (2012).

[3] While GARCH(1,1) was not the most optimal model for IBM returns, overall it was not outperformed by more sophisticated models.

great risk of mis-specifying the underlying volatility relationship. While Andersen and Bollerslev (1998a) proved that GARCH(1,1) shows tremendous in-sample performance, question marks have been raised regarding its deficient out of sample forecasting ability. Andersen and Bollerslev (1998a) refuted the scepticism by showing that well specified volatility models do in fact provide accurate forecasts.

Despite this, the fact that GARCH is not model-free encourages researchers to direct their attention to more unconventional, yet proven approaches in mapping complex non-linear relationships, such as Artificial Neural Networks (ANN).

ANN is a machine-leaning method inspired by the human brain in that it mimics the organic neuron. These mirrored neurons can then be connected in different ways, allowing the collection of neurons to find complex, non-linear patterns. The great benefit of neural networks in comparison to GARCH is that they do not impose any restricting relationship onto the data. However, this comes at a cost of drastically increased computational burden and often with the requirement of a vast dataset to train on in order to be effective.

ANNs can be divided into two main categories, namely Feed Forward Neural Networks (FNN) and Recurrent Neural Networks (RNN). The difference lies in how the data reallocates within the network. RNN allows for loops of the data within the network, meaning that the predictions at earlier stages impact decisions at later stages of the network. This can be likened to an Autoregressive (AR) process. However, AR processes are linear in nature and have a finite dynamic response, while RNNs are unbounded and have an infinite dynamic response.

The Long short-term memory (LSTM) unit is a special type of RNN that has the ability to remember long term patterns. The inherent characteristics of LSTM advocates that it should be superior to FNN in predicting time series data. Nevertheless, FNN is by far the most adopted model in financial studies. One reason behind this might be the computational difficulties of LSTM, requiring more computational power than FNN.

Despite the growing interest in commodity investing and the proven ability of neural networks in handling non-linearity exceptionally well, no previous paper has to our knowledge investigated a neural network approach to volatility forecasting when estimating VaR and ES for commodities.

Therefore, the purpose of this thesis is to evaluate and compare VaR and ES estimates for different commodities based on a volatility weighted historical simulation, with volatility forecasts provided by;

1. A conventional GARCH(1,1) approach

2. A neural network (LSTM) approach

The results of the thesis indicate that overall, both approaches manage to estimate VaR and ES adequately. GARCH(1,1) deliver somewhat more accurate VaR estimates in terms of the amount of violations, LSTM on the other hand seems better at spreading the violations according to Christoffersen's test. Finally, both models estimate well specified ES estimates, with slightly better test statistics for the GARCH(1,1) approach.

The remainder of the thesis is structured as following: Section 2 presents a theoretical overview of key risk measurement concepts and techniques as well as an introductory section to neural networks. Section 3 introduces a historical background including previous research relevant to the thesis. Section 4 motivated the choice of data and methodological framework adopted in the thesis. Section 5 conducts the results which is then followed up by a discussion of the results in Section 6. Section 7 then finally outlines the conclusions of the thesis.

2 Theory

The theory section aims to give a theoretical background to the reader, with definitions of the methods and models that will be used in the remainder of the thesis. It starts with definitions of key risk measurements and different methods to estimating the measurements, then it covers the approach to backtesting the models and lastly an introduction to neural networks is conducted.

2.1 Losses

A loss 'L' (stochastic) or 'ℓ' (realized) is simply defined as a negative gain. It is expressed in monetary terms, and hence depend on the nominal currency and size of the investigated asset or portfolio. The distribution of losses is then simply the reversed distribution of gains, with the largest losses in the right tail of the distribution. With an initial investment of X (set equal to 100 throughout this thesis) at the start of each trading day, the loss for day t is defined as

$$\ell_t = -\frac{P_t - P_{t-1}}{P_{t-1}} X \tag{1}$$

2.2 Value At Risk

VaR is a widely used risk measure, largely due to its simplicity and applicability to all types of asset classes. It is defined as the smallest monetary loss, such that the probability of a larger loss is less than or equal to $1-\alpha$, over some pre-specified holding period, usually one or ten days, as following

$$VaR_\alpha = \min\{\ell : \Pr(L > \ell) \leq 1 - \alpha\} \tag{2}$$

Hence, VaR takes a holistic perspective on risk by focusing directly on the distribution of the losses, and can be thought of as the cut-off point that leaves the $(1-\alpha)$ worst losses in the tail. If the loss distribution is continuous, every loss is a VaR for *some* confidence level and similarly every VaR is a loss, hence, the definition of VaR becomes equivalent to the probability of a larger loss than VaR being *exactly* $1-\alpha$.

Following from the definition of VaR, it can also be interpreted as the α-quantile of the loss distribution as following

$$VaR_\alpha = q_\alpha \tag{3}$$

The main drawback of VaR is that it does not reveal anything about the potential magnitude of a loss ending up in the tail. Hence it fails to capture extreme losses that occur with very small probabilities, so called 'Black Swans' (Taleb, 2007). Therefore VaR

lacks the subadditive property of a coherent risk measure[4], *i.e.* does not always encourage diversification (Artzner et al., 1999).

2.3 Expected shortfall

ES builds on VaR but incorporates the entire tail of the loss distribution by taking the average of the VaR's (losses) larger than VaR_α, and is defined as

$$ES_\alpha = \frac{1}{1-\alpha}\int_\alpha^1 VaR_x dx \tag{4}$$

Again, because there is a VaR for every loss, and a loss for every VaR if the distribution is continuous, ES can instead be defined as the conditional expected loss exceeding VaR_α as following

$$ES_\alpha = E[L \mid L > VaR_\alpha] \tag{5}$$

The great advantage of ES is that it incorporates the entire tail of the loss distribution and hence has all the desirable properties of a coherent risk measure[5].

2.4 Holding period & confidence level

Both VaR and ES are functions of a pre-specified holding period and a confidence level, α. The holding period is simply the number of days that losses are measured over. The usual case, which is also adopted throughout this thesis, is to set the holding period to one day, which is convenient in order to incorporate as many observations as possible (Hull, 2015). The confidence level is the certainty level (probability) that for any given holding period there will not be a loss larger than VaR. It should be clear that a higher confidence level implies fewer but greater losses when they occur. Hence VaR and ES are increasing functions in both the holding period and the confidence level.

2.5 Parametric and non-parametric approaches

The approaches in which to estimate VaR and ES from a dataset can be categorized into either being parametric or non parametric. Parametric models imposes a distribution to the losses, such as the normal distribution or the student t-distribution, while non-parametric methods do not. The non-parametric approach rather 'lets the data speak' and utilizes the empirical distribution portrayed by the data. The great advantage of non-parametric approaches is that they do not impose any distributional restrictions such as normality, which is often an unreasonable assumption for financial instruments. Even when excess kurtosis are imposed in the form of a student t-distribution, it might not

[4]See Appendix A for definition of coherency.
[5]See Footnote 4.

portray the actual distribution very well, leading to miss-specified VaR and ES estimates. On the other hand, non-parametric approaches are generally highly dependant on the data at hand and could be misleading if a too calm or too volatile period is examined. Additionally, it might react slowly to changing market conditions, however, volatility- or age-weighting the data can somewhat mitigate this problem.

2.5.1 Basic historical simulation

The Basic historical simulation (BHS) is a non-parametric approach to estimating VaR and ES from a sample of losses of an asset or portfolio. The losses are sorted in descending order, where VaR is simply the $N(1-\alpha)+1$ largest loss and ES is the average of the $N(1-\alpha)$ largest losses. Since losses are discrete, $N(1-\alpha)+1$ might not be an integer, in which case one can either interpolate between the two losses capturing $N(1-\alpha)+1$ to get the VaR estimate or pick the immediate smaller loss than $N(1-\alpha)+1$, as the probability of a larger loss in the sample will then be less than $1-\alpha$.

The usual procedure used to get individual estimates of VaR and ES for the next day out of sample ($\widehat{VaR_{\alpha,t+1}}$, $\widehat{ES_{\alpha,t+1}}$) is to conduct a so called rolling window. The rolling window is a fixed amount of loss-observations used to estimate VaR and ES: ℓ_1, ℓ_2, ..., ℓ_t. For every new day $(t+1)$ the oldest loss observation (ℓ_1) is discarded and the most recent one (ℓ_{t+1}) is incorporated in the *new* window that is then used to estimate $\widehat{VaR_{\alpha,t+2}}$ and $\widehat{ES_{\alpha,t+2}}$, and so on.

2.5.2 Volatility Weighted Historical Simulation

One drawback of the BHS approach is that it reacts slowly to changing market conditions such as changing volatility. Hull and White (1998) suggests a so called volatility weighted historical simulation (VWHS) to mitigate this problem. In VWHS the expectation of the volatility in the next period affects today's estimations of VaR and ES for the next period. All the losses in the sample are rescaled by the estimation of tomorrows volatility (σ_{T+1}) according to the following formula

$$\ell_t^R = \frac{\sigma_{T+1}}{\sigma_t}\ell_t \tag{6}$$

As can be seen from the formula above, a high expected volatility in the next period will increase the magnitude of all the losses in sample which will give a higher VaR and ES estimate, and vice versa. After rescaling all the losses, BHS is applied to the rescaled losses as described in the previous section. The only remaining question then becomes how to estimate the volatility for all periods: $\sigma_1, \sigma_2, ..., \sigma_T, \sigma_{T+1}$. As touched upon, one conventional time series model used when estimating clustering volatility is the Generalized Autoregressive Conditional Hetroscedasticity (GARCH) model.

2.5.3 GARCH(1,1) and Maximum Likelihood

As the name suggests, GARCH is used to model time series when allowing for clustering and time varying (heteroscedastic) volatility. The term volatility clustering comes from the fact that asset returns tend to see periods of higher- and lower volatility (Lux and Marchesi, 2000). To model this, GARCH assumes that asset returns have an expected part, μ_t, and an unexpected part, η_t, that captures the varying volatility as following:

$$r_t = \mu_t + \eta_t \tag{7}$$

Further, $\eta_t|\Omega_{t-1} \sim N(0, \sigma_t^2)$ which tells us that conditional on all the information of past returns that are available at $t-1$ (Ω_{t-1}), η_t has a zero mean and conditional (time varying) volatility σ_t^2. In a GARCH(1,1), η_t is defined as

$$\eta_t = \epsilon_t \sqrt{(\omega + \alpha\eta_{t-1}^2 + \beta\sigma_{t-1}^2)} \tag{8}$$

where ω, α, β are parameters and $\epsilon_t \sim N(0,1)$ is a random shock. Here it becomes evident where the (1,1) comes from in the GARCH(1,1). The current unexpected return depends on *one* lagged unexpected return as well as *one* periods lagged conditional variance of the return. By utilizing the fact that $E[\eta_t]^2 = 0$, $E[\epsilon_t]^2 = 0$ and that the formula for variance can be written as $E[X^2] - E[X]^2$, the conditional variance of η_t becomes

$$E_{t-1}[\eta_t^2] = \sigma_t^2 = \omega + \alpha\eta_{t-1}^2 + \beta\sigma_{t-1}^2 \tag{9}$$

The parameters μ, ω, α, β can then be estimated with Maximum Likelihood (ML). ML finds the most probable values for the parameters ω, α and β given the data that is observed. It can be though of as finding the parameters that have the highest probability of resulting in the observed data. Technically, this is done by maximizing the following likelihood function

$$ln\, L(\mu, \omega, \alpha, \beta) = \sum_{t=1}^{T} \left(-\frac{1}{2} ln(2\pi) - \frac{1}{2} ln(\sigma_t^2) - \frac{\eta_t^2}{2\sigma_t^2} \right) \tag{10}$$

Finally, it should be noted that the initial values of η_0 and σ_0 are often set to zero and the standard deviation of the returns (losses) in sample, respectively.

2.6 Backtesting VaR and ES

2.6.1 Kupiec test

The Kupiec (1995) test is the conventional test for backtesting the performance of VaR estimates. It is a binomial test that follows from the definition of VaR by comparing the observed number of violations (losses exceeding VaR) to the expected number of violations given that the VaR-estimation is correct. The probability of obtaining *less than or equal* to the amount of violations observed (if the model is correctly specified) is

$$\Pr(X \leq x) = \sum_{i=0}^{x} \binom{N}{k} p^i (1-p)^{N-i} \tag{11}$$

Where x is the amount of violations observed, p is the probability of a violation if the model is correct and N is the sample size. Since the expected amount of violations is $N(1-\alpha)$, the formula can be used directly to calculate the probability of the amount of violations or an even more extreme outcome if less than expected violations are observed. If the observed amount of violations *exceeds* the expected, $1-\Pr(X \leq x-1)$ is used to calculate the probability of receiving the outcome received or an even more extreme outcome. This probability is then compared to the confidence level chosen and the VaR estimate is rejected if the probability is less than the confidence level. Finally, a two-sided test can be conducted by creating a confidence interval with the confidence level split equally for the two tails.

2.6.2 Christoffersen's test

Besides comparing the obtained and expected amounts of violations, one could also test the independence of the violations. One such independence-test is the forecast evaluation test introduced by Christoffersen (1998), known as the Christoffersens's test. By the definition of VaR the probability of a violation should be (1-α) for any given day. However, heavy losses tend to come in clusters or high frequencies. Christoffersen's test builds on the fact that the probability of a violation today should be independent of the outcome the day before. Therefore, loosely speaking, Christoffersen's test could be though of as evaluating how well a VaR model manages to adapt to changing market conditions. The tests builds on a likelihood-ratio as following

$$LR = -2\log\left(\frac{(1-\pi)^{n00+n10}\pi^{n01+n11}}{(1-\pi_0)^{n00}\pi_0^{n01}(1-\pi_1)^{n10}\pi_1^{n11}}\right) \sim \chi^2(1) \tag{12}$$

where

$n00$ - Amount of periods without a violation followed by a period without a violation

$n01$ - Amount of periods without a violation followed by a period with a violation

$n10$ - Amount of periods with a violation followed by a period without a violation

$n11$ - Amount of periods with a violation followed by a period with a violation

π_0 - Probability of a failure in period t, given that no failure occurred in period t-1

π_1 - Probability of a failure in period t, given that a failure occurred in period t-1

π - The unconditional probability of a failure in period t

2.6.3 Backtesting Expected Shortfall

The area of backtesting ES has not been as researched empirically as that of backtesting VaR. Following the decision of the Basel Committee to adopt ES in spite of VaR, Acerbi and Szekely (2014) proposes three model free non-parametric methods to backtesting ES. The second model, named "*Testing ES Directly*", has computational advantages as it only required two estimations for each day; The ES estimate ($\widehat{ES_{\alpha,t}}$) and the magnitude of the loss if a VaR violation occurs ($L_t I_t$) where I_t is an indicator function for a VaR violation as following

$$I_t = \begin{cases} 1 & \text{if } L_t > VaR_t \\ 0 & \text{if } L_t \leq VaR_t \end{cases} \tag{13}$$

The test statistic proposed by Acerbi and Szekely (2014) is then

$$\text{z-value} = -\frac{1}{T(1-\alpha)} \sum_{t=1}^{T} \frac{L_t I_t}{\widehat{ES_{\alpha,t}}} + 1 \tag{14}$$

which can be shown to have an expected value of zero under the null hypothesis that $\widehat{ES_{\alpha,t}}$ is correct for each day t. The test is one-sided in that it only rejects models that underestimate the 'actual' ES. Therefore, the alternative hypothesis states that $\widehat{ES_{\alpha,t}}$ underestimates the actual ES for at least one day t. From the z-value above, it should be clear that if ES is underestimated, then the average ratio in the sum will be greater than one and the total z-value will be negative. Acerbi and Szekely (2014) find that the critical value corresponding to a 5% confidence level displays remarkable stability across different distribution types of around -0.7, and hence suggests using this value when backtesting ES.

2.7 Artificial Neural Networks

ANN is a special form of Machine learning where a collection of algorithms and functions are designed to mimic the human brain (Haykin, 1998). This is achieved by constructing so called *artificial neurons* which are then connected together. Just like a biological neuron, the artificial neuron can communicate with other neurons in the network and if the input signal is strong enough the artificial neuron will be triggered.

ANNs are excellent in finding patterns from large data sets and especially non linear ones which conventional machine learning algorithms, such as support vector machines or decision trees, often are unable to do satisfactory. ANN does this by breaking down a complex problem into smaller and simpler subsets by combining multiple layers[6] of artificial neurons. Each layer performs a simple task and feeds it to the next, resulting in a model capable of tackling complicated tasks.

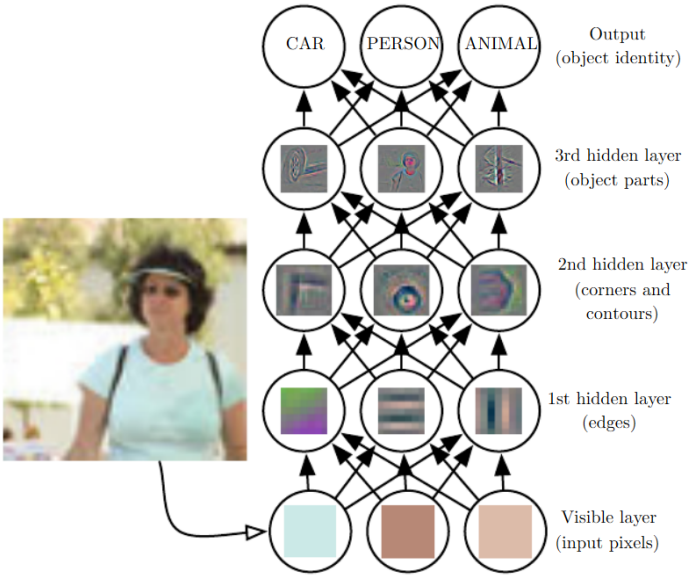


Figure 1: Illustrates how an image is fed into a neural network where each layer performs a simple task which results in a model being able to determine what object is presented on the image. Image retrieved from Goodfellow, Bengio, and Courville (2016, p. 145).

Figure 1 shows how a simple neural network can from an image as an input determine what object the picture portrays. The first hidden layer determines the orientation of the edges, which is then fed to the second layer that outlines the corners and contours. The third and last hidden layer then uses this to identify key object parts which in turn is fed to the output layer which determines what object the image displays.

The greatest drawback of neural networks is that they are computationally expensive compared to conventional machine learning algorithms and require a large set of data to be useful. With computational power becoming cheaper and with the abundance of data in

[6]Refer to the Terminology section for some introductory terminology used.

recent times, the disadvantages of neural networks are slowly disappearing (Goodfellow, Bengio, and Courville, 2016). However, neural networks are, in general, much more difficult to interpret. It is often difficult to assert what each hidden layer determines or contributes. Most often, it is impossible to establish which parameters or patterns the neural network attaches importance to.

2.7.1 The artificial neuron

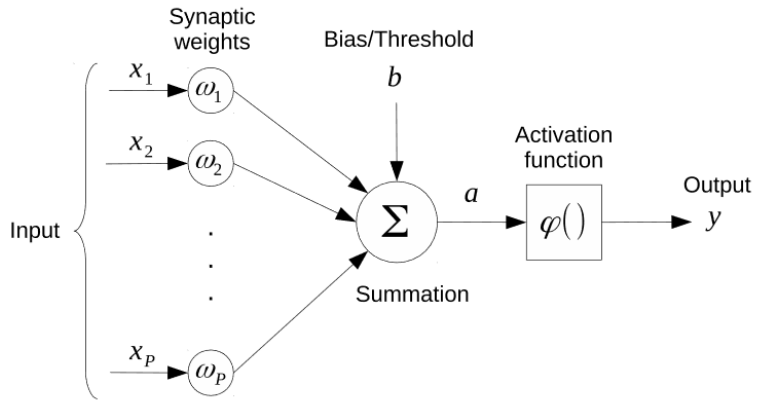


Figure 2: The basic element of ANN, the artificial neuron. Each input variable, x, is fed to the neuron with associated weights, ω. These are then summed up and a bias, b, is added in order to fit the model to the desired outcome. The weighted summation, a, is then fed through an activation function, φ, with the final output of the neuron being y. Image retrieved from Ohlsson and Edén (2019).

Unlike human neurons which are limited to the five senses as input variables, an artificial neuron can have as many input variables as desired as long as they can be numerically represented. Figure 2 illustrates how a single neuron, takes multiple numerical inputs and converts it to a potential output. Each input variable, x_1 to x_P, is fed to the artificial neuron and multiplied with its associated weights ω_1 to ω_P and then summed up. The neuron also has a so called bias, b, which is a value that either deflates or inflates the summation, with the weighted summation of the neuron, a, then calculated as

$$a = \sum_{k=1}^{P} \omega_k x_k + b \tag{15}$$

a is then fed to the activation function $\varphi()$ where

$$y = \varphi(a) \tag{16}$$

In the most simplest form of a neural network consisting only of one single neuron, the output, y, can be the final output of the network. However, neurons are often connected

together to construct a larger network. The output from an artificial neuron is then often fed to another artificial neuron as an input. The next artificial neuron conducts the same operation, but with inputs now originating from other artificial neurons, as shown in Figure 3 below.

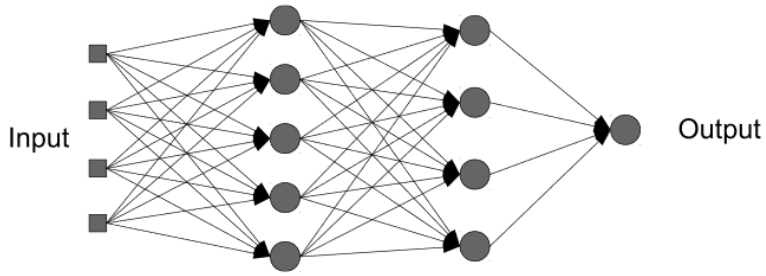


Figure 3: An FNN illustrated with 4 input nodes in the input layer. There are 2 hidden layers, with 5 hidden nodes in the first layer, 4 hidden nodes in the second layer with a single output node in the output layer. Image retrieved from Ohlsson and Edén (2019).

The activation function is perhaps one of the most important aspects of the neural network. It allows for the network to turn a linear function into a non-linear output in a simple and convenient manner. There are multiple activation functions to choose from, each with different advantages and disadvantages. The six most common ones are presented below along with a graphical illustration (see Figure 4).

1. Linear:

$$\varphi(x) = x$$

2. Threshold function:

$$\varphi(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

3. Logistic function:

$$\varphi(x) = \tfrac{1}{1+e^x}$$

4. Hyperbolic tangent:

$$\varphi(x) = \tanh x$$

5. Rectified linnear unit (ReLU):

$$\varphi(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

6. Softplus:

$$\varphi(x) = \log(1 + e^x)$$

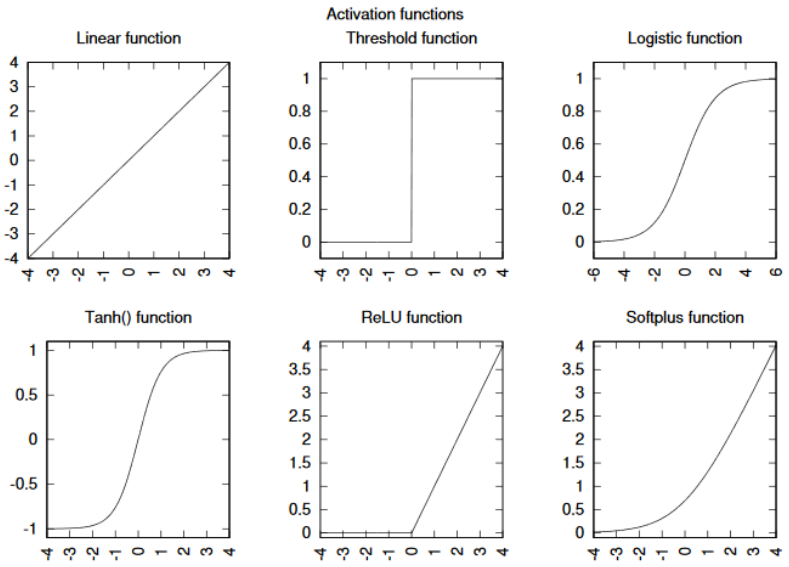


Figure 4: The activation functions illustrated. Image retrieved from Ohlsson and Edén (2019).

The most appropriate activation function depends heavily on the task the neural network or node is assigned to perform and the complexity of the network. More complex networks often require more primitive activation functions to reduce computational burden. Activation functions where the output range is limited, such as the threshold, logistic and hyperbolic tangent functions, are most suitable for classification tasks while Softplus, ReLU and linear are often used for regression type tasks.

The weights, bias and activation function will all play a role in optimizing the network. The goal is for the output of the activation function to be as close to the true correct value as possible, which will be further explained in section 2.7.3.

2.7.2 Recurrent neural network

The single artificial neuron can be linked in different ways with other artificial neurons to produce a certain type of architecture. The quality of the output is closely linked to the architecture of the network. RNN have feedback connections which are used to capture long term and short term temporal dependencies in the data (see Figure 5). This makes RNNs particularly useful when analyzing sequential data[7]. However, as noted by Bengio, Simard, and Frasconi (1994), ordinary RNNs have a hard time remembering long term dependencies in data as they suffer from the vanishing gradients problem[8]. While one may choose to develop a more intricate learning algorithm to capture long term dependencies, a more common approach is to use a more sophisticated architecture.

[7]In the interest of this paper, only recurrent networks will be discussed and their mechanisms. Goodfellow, Bengio, and Courville (2016) gives detailed and in depth descriptions of multiple different architectures such as feedforward and convolutional networks.

[8]Vanishing and exploding gradients problem will be further discussed in section 2.7.3.

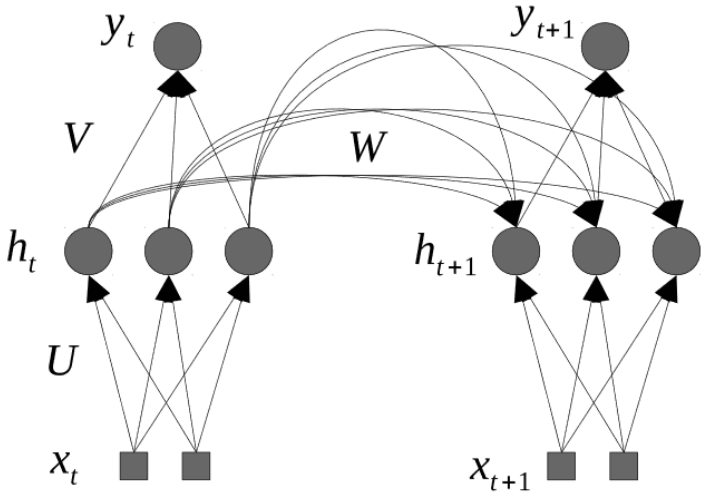


Figure 5: An RNN illustrated with 2 input nodes in the input layer. There is 1 hidden layer with 3 nodes and 1 output node. Each hidden node has a feedback connection to all other nodes in the hidden layer, including itself. Each input at time t (left) gets fed to each hidden neuron where U is the product of the input values and the weights for the first input layer, V is the product of the output from the hidden layer and the weights for the output layer. W takes the same output of the hidden layer but multiplies it with another set of weights which is then fed back to the hidden layer at the next time step $t + 1$ (right). Image retrieved from Ohlsson and Edén (2019).

2.7.2.1 Long short-term memory

The LSTM is a special type of RNN architecture which was initially proposed by Hochreiter and Schmidhuber (1997). Since then, minor changes and improvements have been made over the years with the implementation that Graves, Mohamed, and G. Hinton (2013) used being the one most widely recognized and the one utilized in this paper.

Unlike a regular RNN which simply calculates the weighted sum of each input, an LSTM network conduct a more sophisticated operation. Each LSTM-unit has a so called 'memory cell' that can maintain information over long time periods with the help of 'gates'. Each LSTM-unit has multiple gates called output-, forget- and input gates. These gates can be seen as regulators which dictate the flow of information inside the LSTM-unit. The output gate modulates the amount of memory content exposure, the input and forget gates updates the memory cell by partially forgetting and adding new memory content as shown in Figure 6. Unlike the simple RNN unit which overwrites its content at every time step, an LSTM unit decides whether to keep the existing memory or partially overwrite the memory using the introduced gates. This gives LSTM networks the ability to detect important features and carry them over a long period of time[9]. The long term memory

[9]This paper only aims to give the reader an intuitive understanding of LSTM networks. The output of the LSTM unit is more complicated than the simple neuron shown in section 2.7.1. A full mathematical description of an LSTM is provided by Graves, Mohamed, and G. Hinton (2013).

characteristics have made LSTM one of the most popular architectures in deep learning, with many of the famous artificial intelligence systems having an LSTM architecture, such as Alphastar (Stanford, 2019) and OpenAI (Rodriguez, 2018).

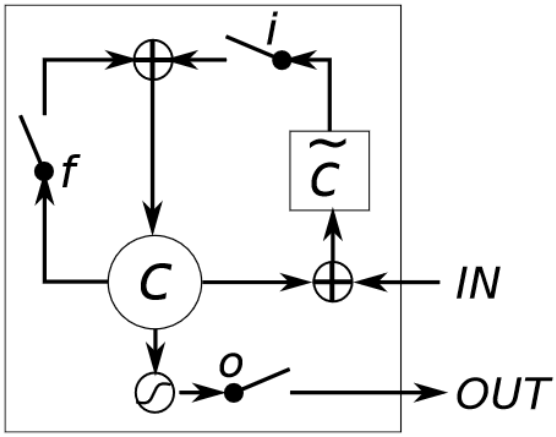


Figure 6: Illustration of an LSTM unit. i, f and o are input-, forget- and output gates respectively. c and $\tilde{c}$ denote the memory cell and the new memory cell content. Figure retrieved from Chung et al. (2014).

2.7.3 Optimizing a neural network

Goodfellow, Bengio, and Courville (2016) refers to three ingredients that are quintessential for neural networks and other machine learning algorithms, with the three being: experience, task and the performance measure. The task can be anything from classification, to regressions or imputation of missing values. The experience is often referred to the data set available. The performance measure is the evaluation of how well the neural network performs the task with the available experience. However, evaluating how well a model performs is often difficult and subjective. It is therefore easier to measure how incorrect the model is by using an error measure, often referred to as *cost function*.

When training a neural network, the algorithms attempt to find the optimal weights attached to each input variable. To identify the most optimal weights the model must know the magnitude of the errors. Therefore it is necessary to construct a relevant error function depending on what data is being dealt with. For regressions the two most common error function utilized are the mean squared error (MSE) and mean absolute error (MAE) given as

$$E_{MSE}(\omega) = \frac{1}{2N}\sum_{n=1}^{N}(d_n - y(x_n))^2 \tag{17}$$

$$E_{MAE}(\omega) = \frac{1}{2N}\sum_{n=1}^{N}|d_n - y(x_n)| \tag{18}$$

where N is the number of data points, d_n is the target output value and $y(x_n)$ is the output of the model. While the two cost functions are very similar, MSE puts more weight on large deviations due to the squared term. This means that MSE should be preferred if large errors are particularly undesirable.

The aim is to find a vector consisting of weights ω for each input that minimizes the error function. This can be achieved by using an iterative procedure called *gradient descent learning*. Gradient descent achieves this by the following steps:

1. Initiate all weights, ω_k, with small random numbers
2. Define a *learning rate*, η.
3. Compute the output, $y_n = y(x_n)$ and the difference, $\delta_n = \frac{1}{N}(d_n - y_n)$ for each data point, n.
4. Update the weights according to: $\omega_k \to \omega_k + \eta \sum_n \delta_n x_{nk}$
5. repeat step 3 and 4 until convergence.

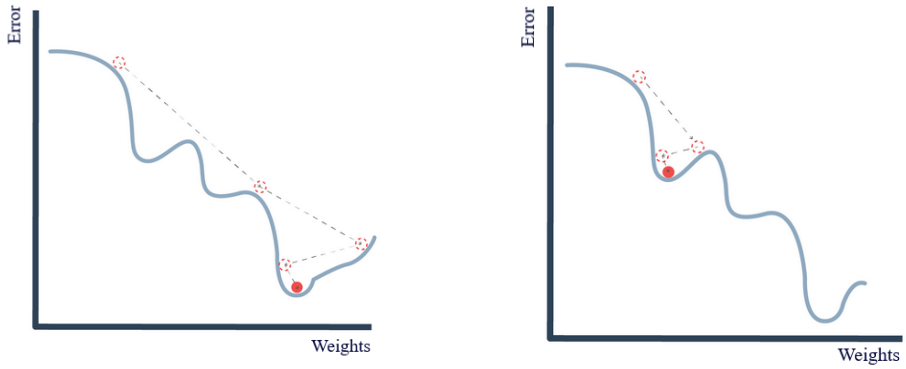
This means that the change in weights can be expressed as[10]:

$$\Delta\omega_k = -\eta \frac{1}{N} \sum_{n=1}^{N} \frac{\partial E_n}{\partial \omega_k} \tag{19}$$

meaning that the change in weights occurs in the opposite direction of the gradient and the size of the change is proportional to the partial derivative and the learning rate. A smaller learning rate gives a smoother trajectory towards the minimum point but may come at the cost of excessive computational time. On the other hand, if the learning rate is set too high the model might overshoot the minimum point (Haykin, 1998). However, it is impossible to know if the minimum point the model identifies is the local or the global minimum. It is therefore sensible to intentionally overshoot the initial minimum in effort to determine whether or not the minimum identified is the global minimum or not (see Figure 7).

In addition, gradient descent has to run through the entire training set for a parameter in a particular iteration. Thus, if the training sample is large, which it has to be for ANN to be effective, then gradient descent is extremely time inefficient as running one single iteration will take too long. Therefore, *stochastic gradient descent* (SGD) is often utilized instead. Unlike regular gradient descent, SGD only iterates over a subset of the training data to update the parameter. Typically, the training set is divided into 10-50 smaller subsets, usually referred to as *minibatches*. Every time a minibatch has been used to update the weights, an iteration has been performed. After all minibatches have been

[10]See Appendix B for derivation.



(a) Larger learning rate (b) Small learning rate

Figure 7: Figure 7a shows a model that intentionally overshoots the minima in order to find the global minimum. Figure 7b shows when a learning rate is set too small and the model gets 'stuck' at the first local minimum. Image retrieved from Peltarion (2020).

iterated, a so called *epoch* has been completed. After an epoch, new random partitioning usually occurs until convergence is reached. This drastically reduces the computational burden at the cost of a slightly lower convergence rate.

While SGD is an improvement upon regular gradient descent, it imposes a common learning rate for all parameters. For models with large number of parameters, this can be inefficient. The *Adam* optimizer[11] (the name is derived from ***ada****ptive* ***m****oments*) builds on SGD but also incorporates a dynamic learning rate and momentum. In layman terms, if the optimizer realises that the current weights are far off from the optimal weights, it will move more aggressively towards the optimum weights (Kingma and Ba, 2014).

If a neural network has a complex structure or is very large, then one often has to deal with the problem of *vanishing* or *exploding gradients*. This occurs when the weight updates either approaches zero or becomes very large in equation 19, which can happen if for example the utilized activation function is hyperbolic tangent. In an n-layered network, the small or very large gradients[12] gets multiplied, meaning the gradient will decrease or increase exponentially with n. While there may lack any direct solutions one can incorporate other activation functions that suffer far less from the vanishing/exploding gradient problem. Activation functions such as linear and ReLU always has a fixed derivative and therefore will neither vanish nor explode.

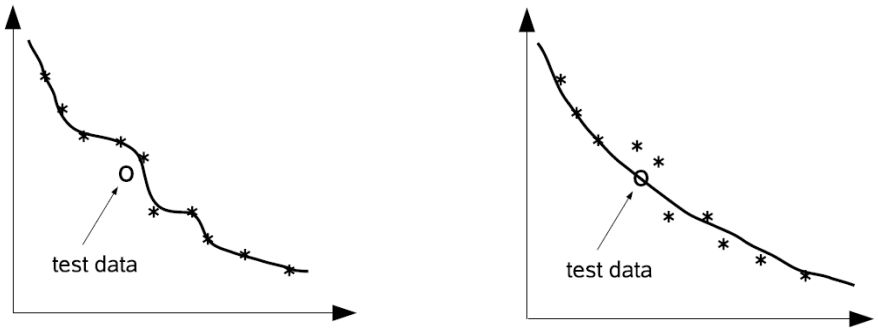
[11]The adam optimizer is far more mathematically advanced compared to SGD since it combines two relatively complex optimization methods, RMSprop and AdaGrad. For the full mathematical derivation and a full theoretical background of the combined methods, see Kingma and Ba (2014).

[12]The gradient is represented by the partial derivative in equation 19. If this partial derivative is very small in a multilayered network, it will decrease exponentially due to the chain rule when differentiating composite functions.

2.7.4 Regularization

A central problem in machine learning is to construct a model that performs well not only on the training data, but also on new, previously unseen data. A model with such ability is said to be able to *generalize*. However, overfitting the model on the training data often prevents models to be able to generalize (see Figure 8). The process of decreasing the errors on new data by tweaking the learning algorithm is referred to as *regularization*.

Most network regularization methods from conventional machine learning algorithms are applicable on neural networks, such as L1 regularization (often referred to as LASSO). However, in deep learning there are also additional approaches to regularization that conventional machine learning algorithms do not have, due to the distinct features of neural networks.



(a) Overfit (b) Generalized

Figure 8: Figure 8a shows a model that fits very well to the trained data but with poor generalization performance. Figure 8b shows a much better generalized model at the cost of increased training errors. Image retrieved from Ohlsson and Edén (2019).

2.7.4.1 L1 and L2 regularization

The two most known regularization methods, commonly referred as L1 and L2, modifies the cost function according to

$$\tilde{E}(\omega) = E(\omega) + \alpha\Omega \tag{20}$$

where $\tilde{E}$ is the adjusted cost function, Ω is the regularization term and α controls the amount of regularization. For both approaches, α needs to be fine tuned which adds an additional undesired computational burden. Nevertheless, L1 and L2 are easy to implement and often increase generalization performance.

L2 often increases generalization performance by setting the regularization term as follows

$$\Omega = \frac{1}{2}\sum_i \frac{(\omega_i/\omega_0)^2}{1 + (\omega_i/\omega_0)^2} \tag{21}$$

where ω_0 is a new parameter introduced, that needs to be fine tuned in the same manner

as α from equation 20. L2 normalization forces large weights towards zero while keeping necessary weights to a non-zero value. However, L2 is unlikely to completely remove parameters as weights are extremely unlikely to ever reach zero[13]. This is something that L1 regularization accomplishes by setting the regularization term to

$$\Omega = \frac{1}{2} \sum_i |\omega_i| \tag{22}$$

The main difference between L1 and L2 is that the gradient of L1 is constant and does not approach zero as the weights approaches zero. Meaning that small weights can be forced to zero and therefore LASSO can be used as a feature selection method (Tibshirani, 1995).

2.7.4.2 Early stopping

Another approach to improving generalization performance is the *early stopping*-approach. This method achieves generalization by restricting the complexity of the model. It does this by dividing data into two sets, a training set which the model gets to observe and a validation set that is unknown to the model which is used to estimate the generalization error. During an iteration the model trains on the training set and fine tunes its parameters then valuates the performance on the validation data. The typical behaviour during the minimization process is shown in Figure 9. The early stop method stops iterating once generalization errors starts to increase (Goodfellow, Bengio, and Courville, 2016). Early stopping is computationally cheap and easy to implement which makes it a popular regularization method. However, the validation error rarely exhibit a nicely uniform u-shape form during the minimization process. Hoffer, Hubara, and Soudry (2017) showed that it is often a good idea to continue training for 20-50 epochs as validation errors tend to increase temporarily before decreasing again.

[13]The gradient of the regularization parameter close to $\omega_i = 0$ is almost zero, suffering from the vanishing gradient problem.

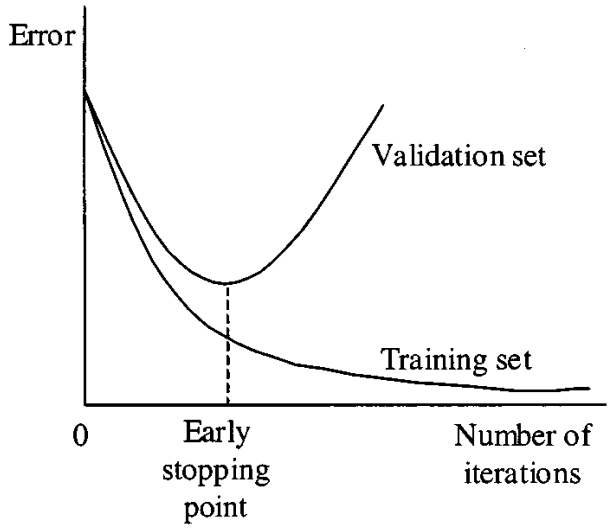


Figure 9: The training error and the validation error during the minimization process. The early stop method halts the training when validation errors has reached a minimum. Image retrieved from Gençay and Qi (2001).

2.7.4.3 Dropout regularization

The dropout method, first introduced by G. E. Hinton et al. (2012), temporarily removes nodes from a neural network to prevent overtraining. Each node, has a probability p of getting *removed* from the model. When a node is removed, all the weights fed in and out from the node are temporarily removed along with the node. The dropout method is an easy and efficient method to generalize a network which works on a broad range of architectures in deep learning. Unlike L1 and L2 regularization, no additional parameter needs to be fine tuned.

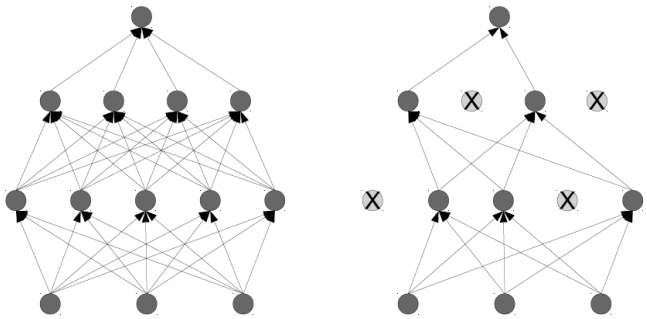


Figure 10: (Left) A network with 2 hidden layers. (Right) The same network but with a few hidden nodes temporarily removed along with the associated weights due to the dropout regularization method. Image retrieved from Ohlsson and Edén (2019).

3 Background and previous research

This section introduces a brief historical background as well as previous research relevant to the thesis. It starts with introductions to commodities and VaR & ES, followed by empirical research in volatility forecasting, and lastly, an overview of the research in the area of ANN is conducted.

3.1 Commodities

Commodities have been traded for thousands of years, as they are natural resources extracted from the earth, which can be refined, used in production or consumed in one way or another. In order to hedge the price risk of a certain commodity, companies can buy futures, which are contracts with pre-specified transaction terms for a future transaction date (Hull, 2015). There are evidence that futures trading can be dated back to transactions on rice in China over 6000 years ago. However, the first institution with the sole purpose of commodities futures trading was the Chicago Board of Trade (CBOT), introduced in 1848 (Nasdaq, 2017).

The digital technology has immensely increased the availability of commodity investments to the broader public, besides futures and other derivatives, there is also a growing appetite for so called exchange traded funds (ETFs) that tracks individual commodities or commodity indices. According to Buyuksahin, Haigh, and Robe (2009), the sums invested in ETFs and other vehicles tracking SPGSCI has gone from 5 billion USD in 1999 to 140 billion USD in 2008. The authors investigate SP500 and SPGSCI with dynamic correlation and recursive cointegration methods, with the results indicating that the asset classes are dis-synchronized. Further, the immensely increased trading activity in commodities does not seem to have altered this relationship significantly in the previous 15 years. Hence, a conclusion is that commodities are still attractive as diversification means for equity investors.

Regardless of the upsurged interest for commodity investments, not too much research has been conducted regarding risk management in the area, especially concerning ES. However, Giot and Laurent (2003) examines VaR on metal- energy- and cocoa-futures with RiskMetrics, skewed student APARCH and skewed student ARCH models. The authors investigate both long and short positions in the futures markets. The results indicate that the skewed student APARCH[14] model performs the best in all cases. Hung, Lee, and Liub (2008) investigated VaR on energy commodities utilizing a GARCH-HT[15]

[14]The skewed Student APARCH approach is a parametric approach. The VaR estimate is the product of the forecasted volatility from an APARCH (Which is an alternation of GARCH but with a leverage effect) and the adequate quantile from the skewed student distribution. For more information, please refer to the work of Giot and Laurent (2003).

[15]Heavy Tail distribution proposed by Politis (2004).

as well as the conventional GARCH-N and GARCH-t. The empirical results suggest that the VaR estimates based on the HT distribution has good forecasting power and further indicates that fat tailed distributions are more suitable than Normal and Student-t for energy commodities. While VaR has been somewhat investigated, especially on energy commodities, ES is yet to be evaluated as a serviceable risk measure for commodities.

Apart from the somewhat deficient output of risk-management research in the area, the general price and volatility characteristics of commodities has been investigated by Deaton and Laroque (1992) in their study "*On the Behaviour of Commodity Prices*". A central finding in the study is non-linearity in predicted commodity prices due to the fact that the market as a whole cannot carry negative inventory. In addition, the authors acknowledge the fact that commodities exhibit extraordinary high volatility. The findings of vast, and clustering, volatility as well as heavy tailed distributions of commodities is also confirmed in the study "*Commodity Price Variability: Its Nature and Cause*" conducted by OECD (1993). Due to the void in risk-management research output, in combination with the volatility characteristics of commodities and the increased investment interest, conducting further research in the area of VaR and ES is warranted.

3.2 Introduction to VaR and ES

Since the introduction of VaR in the late 1980's, the adoption quickly spread across several industries following the financial crisis of 1987 (Linsmeier and Pearson, 1996). However, It was not until 1996 that the Basel Committee introduced a capital charge for market risk based on VaR. The Basel committee, is an international banking supervisory authority, founded in 1973 by the central banks of the so called G10-countries[16]. It has gradually expanded over time and now includes 45 member states with the aim of streamlining banking standards worldwide (BIS, 2020). The 1996-amendments implemented a standardized approach to set capital requirements for banks, which was based on the calculation of VaR over a 10 day horizon on a 99% confidence level (Hull, 2018).

The swift adoption of VaR as a universal risk measure, even outside of regulatory purposes, was largely attributed to its ease of implementation and interpretation (Linsmeier and Pearson, 1996). However, many researchers, such as Embrechts (2000), criticised VaR as a complete risk measure and claimed that the popularity lies to a great extent in its simplicity and applicability to any financial instrument. In addition, Artzner et al. (1999) proves in numerous ways that VaR is in general not subadditive[17] and provides no information regarding tail events and the potential magnitude of such events. As a response to these shortcomings, the authors introduced the idea behind a desirable prop-

[16]The Group of Ten is made up of eleven industrial countries (Belgium, Canada, France, Germany, Italy, Japan, the Netherlands, Sweden, Switzerland, the United Kingdom and the United States) which consult and co-operate on economic, monetary and financial matters (BIS, 2020).

[17]Subadditivity is one of the conditions for Coherency, see Appendix A for definition of Coherency.

erty of a risk measure called 'Coherency'[18]. The lack of subaditivity is concerning since the risk of a portfolio could be far greater than the individual risks of the constituents when VaR is applied. Acerbi and Tasche (2002) claim that subadditivity is crucial to capital adequacy requirements in banking supervision by the analogy of the total risk of a bank being adequate in relation to the summed risks of its branches. Tasche (2002) further criticises VaR for its deficiency in rewarding diversification.

The limitations to the prevailing capital requirement model built upon VaR became painfully evident during the financial crisis of 2007-2008. As the capital held by the banks was far too low in relation to the undertaken risk, it became clear that the current capital requirements were insufficient. In response to the meltdown of 2008, and to ensure the regulation, supervision and risk management of banks, the Basel III accords were released in 2010. Along with the accords, the committee also issued a consultative document known as "*A Fundamental Review of the Trading Book*" in 2012 (BIS, 2020). One of the major changes imposed by FRTB, in order to assure adequate measurement of tail risk, was the adoption of ES on a 97.5% confidence level in spite of the prevailing 99% VaR in measuring the capital charge for market risk (Hull, 2018). The reasoning behind the change of confidence level is that $\text{VaR}_{0.99}$ and $\text{ES}_{0.975}$ are approximately equal if returns are normally distributed (Hull, 2018). However, many empirical studies reject normality for returns of financial assets. Hagerman (1978), Bollerslev (1987) and Fama (1965) display in their respective studies evidence of a non-normal and leptokurtic behaviour of stock returns. Modeling the right tail of the loss distribution becomes crucial as tail events are a threat to the solvency of financial institutions and subsequently the stability of the global economy (BIS, 2013). Even though many studies agreed on the superiority of ES over VaR, some were concerned with the issue of backtesting ES. As a response, Acerbi and Szekely (2014) proposed three approaches, that according to the authors *"introduce no conceptual limitations nor computational difficulties of any sort"*.

While the new accords have introduced stricter rule sets for risk management such as strengthen capital requirements and further liquidity requirements, the financial institutions may still to some extent implement their own approach to volatility forecasting and the estimation of ES.

3.3 Volatility Forecasting

Forecasting the volatility of financial assets is incorporated in many aspects of the economy. The applications range from monetary policy, fiscal policy and the risk management of banks to derivatives pricing, investment strategies and speculation (Poon and Granger, 2003). Like most forecasting practices, volatility forecasting often incorporate previous observations in order to forecast future ones. Empirically, volatility has to a large extent

[18]See Appendix A for the four axioms that a coherent risk measure needs to fulfill.

been forecasted using random walks, simple averages or weighted moving averages (Poon and Granger, 2003). However, these approaches make use of the sample standard deviation, assuming that the volatility within the sample is the same. Mandelbrot (1963) found that volatility tends to cluster and noted in regards to asset prices that "*large changes tend to be followed by large changes, of either sign, and small changes tend to be followed by small changes*". Mandelbrots findings gave rise to a pursuit of more sophisticated models that could account for the volatility clustering.

This eventually gave birth to the family of autoregressive conditional heteroscedasticity (ARCH) models, introduced by Engle (1982). A few years later, Bollerslev (1986) and Taylor (1986) generalized on Engles work in introducing GARCH-models, where the volatility today is explicitly a function of yesterdays deviation from expectations as well as yesterdays volatility. The conditional variance incorporated in these models were considerably better at capturing volatility clustering than simple standard deviation models.

However, Nelson (1991) criticises GARCH for neglecting the well established finding that asset returns are negatively correlated with the volatility. The criticism stems from the fact that negative and positive returns impact the forecasted volatility the same way in GARCH. Different models were then introduced to account for the asymmetric properties of asset returns, including the E-GARCH model introduced by Nelson (1991) and the GJR-GARCH proposed by Glosten, Jagannathan, and Runkle (1993). Despite the emergence of more sophisticated GARCH models, Hansen and Lunde (2005) showed that GARCH(1,1) still performs well on most asset classes.

A fundamental disadvantage with conventional volatility models is that they are not model free, meaning that they impose an explicit relationship onto the data. As stated by Zhang, Patuwo, and M. Hu (1998), formulating a non-linear model to a data set is a challenging task due to vast number of possible non-linear relationships. Therefore, it may cause some heavy misspecification if the underlying conjunction is particularly complex. In addition, X. Chen, Lai, and Yen (2009) have noted that with capital markets being increasingly globalized, it is becoming increasingly difficult to capture and model market risk, which further exacerbates the risk of model mis-specification. These shortcomings incentives the search for a volatility forecasting approach that is model free and adaptable to the most complex non-linear relationships, such as neural networks.

3.4 Artificial Neural Networks

The first early model of the human brain was introduced by McCulloch and Pitts (1943) and is commonly regarded as the inception of artificial neural networks. Not long after, Rosenblatt (1958) published about the perceptron[19], the first model through supervised learning to be able to automatically set weights. Rosenblatt's discovery gave rise to new

[19]The perceptron is a basic artificial neuron which was discussed in section 2.7.1.

architectures such as the multilayered perceptron and other FNNs. However, RNNs would rarely be considered until Hochreiter and Schmidhuber (1997) introduced the LSTM unit.

Consequently, most previous research with an ANN approach on financial data have not been conducted using RNNs. Donaldson and Kamstra (1997) and M. Y. Hu and Tsoukalas (1999) separately compared an ANN approach to forecasting conditional volatility. Both studies combined volatility forecasts made by conventional GARCH approaches with forecasts made by FNNs and found that forecasts that incorporated neural networks had greater predictive power compared to the forecasts that excluded the FNN contribution. However, neither of the studies used an RNN architecture. Within the field of commodities, Kohzadi et al. (1996) compared an ANN approach against conventional time series models for predicting commodity prices and found that ANN outperformed conventional approaches. However, as before, the authors utilized an FNN approach rather than RNN. In more recent times, Kulkarni and Haidar (2009) forecasted the direction of crude oil prices by using oil futures as predictors, with the results showing that the FNN managed to predict both the direction of the price change and the price with great accuracy.

Most established research on financial data implementing ANN seem to have omitted RNN altogether and only just recently has LSTM architectures been implemented. The majority of published articles investigate LSTM networks capabilities in equity price forecasting. In a study by K. Chen, Zhou, and Dai (2015), the authors utilized an LSTM network to forecast stock returns on Chinese stocks and found that LSTM outperformed in terms of forecast accuracy compared to a random prediction approach. Fischer and Krauss (2018) compared the LSTM performance on stock indices against other machine learning algorithms such as random forest, logistical classifier and even an FNN, and found that LSTM again outperformed the other approaches. Min (2020) also found that LSTM compared better than other existing recurring network architectures in forecasting financial data.

Still, there seems to be a research void left on LSTM networks ability to forecast volatility. Recently published papers seem to investigate if volatility forecasts can be improved upon by using a hybrid model[20] which incorporates both GARCH and ANN models. However, none of these utilizes an LSTM network in the hybrid model nor evaluates an LSTM networks ability to forecast volatility. On top of that, there appears to be a complete lack of LSTM network approaches made on commodities, whether it be predicting returns or volatility. General consensus from previous research seem to indicate that neural networks might be better at forecasting both prices and volatility than conventional approaches. However, whether or not this is case with commodities remains to be seen. It is also unclear if the increased forecasting accuracy will translate

[20]See Kristjanpoller, Fadic, and Minutolo (2014), Kristjanpoller and Minutolo (2015) and Tseng et al. (2008)

into improved VaR and ES estimations.

4 Data and Methodology

The data and methodology section outlines and motivates the choise of data as well as the adopted approach in this thesis. It starts with a delimitation section, followed by explaining the choice, and preparation, of the data as well as a short summary of each index. After that, the data preparation for LSTM as well as the motivation behind the hyperparameters are presented. Lastly, sections on estimating the GARCH(1,1) parameters with ML and the approach to estimating and backtesting VaR and ES are conducted.

4.1 Delimitations

Due to the time-frame of this thesis, some delimitations are necessary. The number of commodities investigated is set to three due to long training sessions for LSTM. The data available imposed further limitations. The data is only extracted from 1990 and onwards due to the start date of the oil index. In order to keep consistency across commodities, the other commodities are also sampled for the same time-period. This results in about 7500 daily observations, which might be a sufficient data set for GARCH, but on the small end of the spectrum with respect to LSTM.

A potential remedy to this could be to use more predictors for LSTM in order to increase the robustness of the model which is not an option in this case. This is due to the fact that artificial neural networks requires a substantial amount of computing power. Due to the limited computing and memory resources available[21], the model complexity is needed to be kept low in order to complete training sessions. This means that other than historical volatility, no other predictors are used for forecasting volatility in the LSTM model. However, this does make for a more fair comparison of LSTM and GARCH as they utilize the same set of input-data.

4.2 Data

The data examined in this thesis is daily price data of three indices that track the prices of individual commodities, namely

- **SPGSCLTR** - Crude Oil Total Return Index
- **SPGSGC** - Gold Spot Index
- **SPGSSO** - Soybean Spot Index

The three indices are sub-indices to the Standard & Poor's Goldman Sachs Commodity Index (SPGSCI), which is the first major investable commodity index, and according to

[21] All model training was performed on a Dell XPS 13' 9360 with 8 GB of LPDDR3 2.133 MHz RAM and an Intel I5-7200U CPU with a base clock speed of 2.50 GHz.

Goldman Sachs (2020) "*Provides investors with a representative and realistic picture of realizable returns attainable in the commodities markets*".

The motivation behind the choice of commodities is that they represent three of the most traded commodities within their respective sectors (energy, metals and agricultural) based on futures trading in 2017 with figures from the Futures Industry Association (IG, 2018). While gold is not the most traded metal according to the numbers, it has been considered to be one of the most important hedge assets to the stock market[22] and was the first ever commodity to be securitized in an ETF (Mukul, Kumar, and Ray, 2012).

Three SPGS indices are published: *Excess Return*, *Total Return* and *Spot Return*. The *Excess Return* index measures the returns from investing in uncollateralized[23] commodity futures, the *Total Return* index measures the returns from investing in fully-collateralized[24] commodity futures, and the *Spot* index measures the level of commodity spot prices. In sampling the data, Bloomberg did not provide the same index type for all three commodities. The Crude oil index was not available as *Spot*, the soybean index was not available as *Total Return* and so on. However, the gold index was available as both *Spot* and *Total Return*, and after comparing the two, it was deemed that they were highly similar, and hence, the fact that the indices vary in types should not in any way impact the remainder of the analysis.

The constituents of the indices are weighted by world production and the main requirements for a commodity futures contract to be included can be summarized intro three main categories[25], as following

- **General Eligibility** - Included contracts must have a specified expiration date, be denominated in USD and be traded on a trading facility that has its principal place in an OECD country.
- **Volume and Weight** - Included contracts must meet total dollar value trading requirements and reference percentage dollar weight requirements.
- **Number of Contracts** - Further requirements for the total amount of contracts and the number of contracts for each commodity applies.

The dataset is downloaded via Bloomberg in April 2020 for the period 1990-01-01 to 2019-12-31. The reasoning behind the choice of timespan is that the crude oil index was introduced just a few years prior (1987), and for consistency, the same timespan was kept for all three commodities. Even though 30 years of data (7827 daily observations) per

[22]See for instance "Is Gold a Hedge or a Safe Haven? An Analysis of Stocks, Bonds and Gold" by Baur and Lucey (2010).

[23]Uncollateralized simply means that the collateral is not invested elsewhere, and hence it measures only the return of the futures contracts.

[24]Fully-collateralized refers to the collateral being invested in a risk-free T-bill, the return is hence the excess return plus the risk free return.

[25]For more info: https://us.spindices.com/documents/methodologies/methodology-sp-gsci.pdf.

index should be considered a long enough timespan to evaluate VaR and ES sufficiently for GARCH(1,1), it is on the small side for LSTM, as elaborated on in section 4.1. After downloading the data, it was noted that 260 datapoints had the same index value as the previous day for all the commodities. According to SP Dow Jones Indices[26] there are certain extraordinary events that would lead to 'Unexpected Exchange Closures' resulting in the same index value for two consecutive days. After converting the index to returns, these 'zero-returns' were then excluded in order not to bias the volatility estimates of GARCH(1,1) and LSTM. The descriptive statistics of the loss series is presented in Table 1 below.

Table 1: Descriptive statistics of the loss series for the three commodities.

Loss series	**N**	**Min**	**Max**	**Mean**	**Std**	**Kurtosis**	**Skewness**
Oil	7565	-14.6	31.9	-0.034	2.2	12.2	0.37
Gold	7565	-9.2	9.3	-0.023	1.0	11.0	0.09
Soybean	7565	-6.9	7.1	-0.017	1.4	5.5	0.09

[26]For more info: https://www.spindices.com/documents/index-policies/methodology-sp-options-indices-policies-practices.pdf.

4.2.1 SPGSCLTR - Oil

The oil index SPGSCL was first launched in 1991, with a staring index value on December 31, 1986. According to S&P Global, the index *"provides investors with a reliable and publicly available benchmark for investment performance in the crude oil market."*.

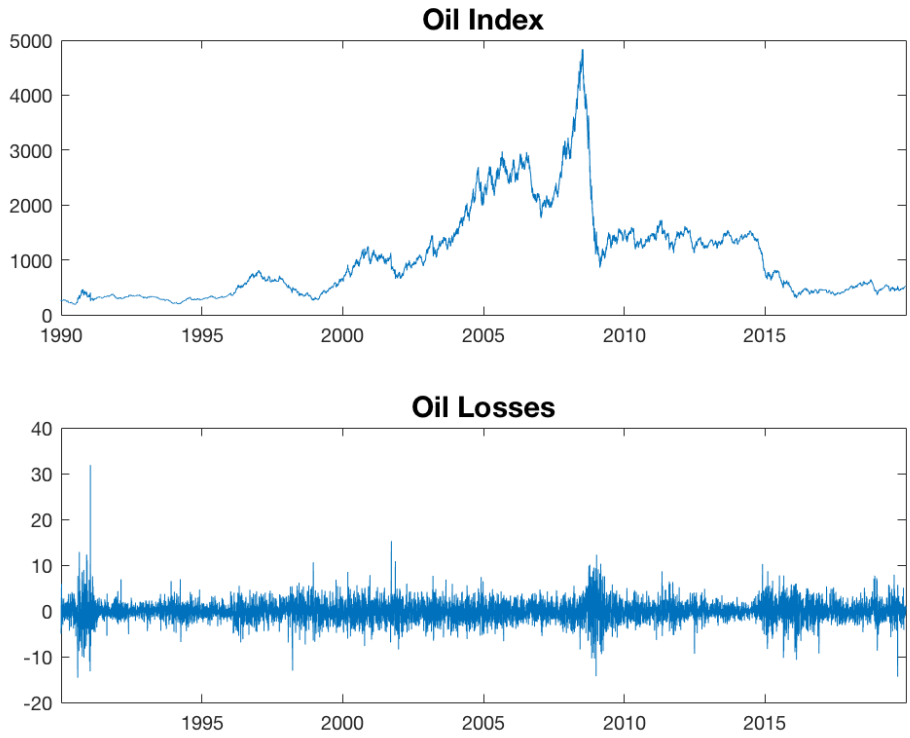


Figure 11: The evolution of the oil total return index SPGSCLTR (upper panel) and the corresponding losses (lower panel). Both plots are over the time period 1990-2019.

As can be seen from Figure 11, the oil price had a strong increasing trend leading up to the financial crisis of 2008. This was largely attributed to soaring tension in the middle east, increased demand in China and India, as well as a deteriorating value of the USD (Huntington, 1998). In the recession of 2008, the demand of oil collapsed and the price went from 147 USD/barrel of crude oil on NYMEX at its peak in July 2008 to a low of 30 USD/barrel in December the same year. Since the collapse, the oil price was consolidating for a few years before having another deep dive in the beginning of 2016 when USA drastically increased its output. Further, oil has experienced the highest volatility out of the examined commodities, as can be seen from the standard deviations (Std) in Table 1. This is in line with previous research that indicates that oil is one of the most volatile commodities (Regnier, 2007). The high volatility that oil has experienced throughout the years is also clear from examining the the loss chart, with a noticeable 32% daily decline on the 16th of January 1991. Furthermore, as can be seen from Table 1, the kurtosis and skewness is higher than for the other commodities, indicating that large

losses occur rather frequently. This is also one of the reasons behind the modest price increase from the ending of 2019 as compared to 30 years prior.

4.2.2 SPGSGC - Gold

The gold index SPGSGC was first launched in 1991 with a starting index value on December 30, 1977. According to S&P Global, the index "*provides investors with a reliable and publicly available benchmark tracking the COMEX gold future*".

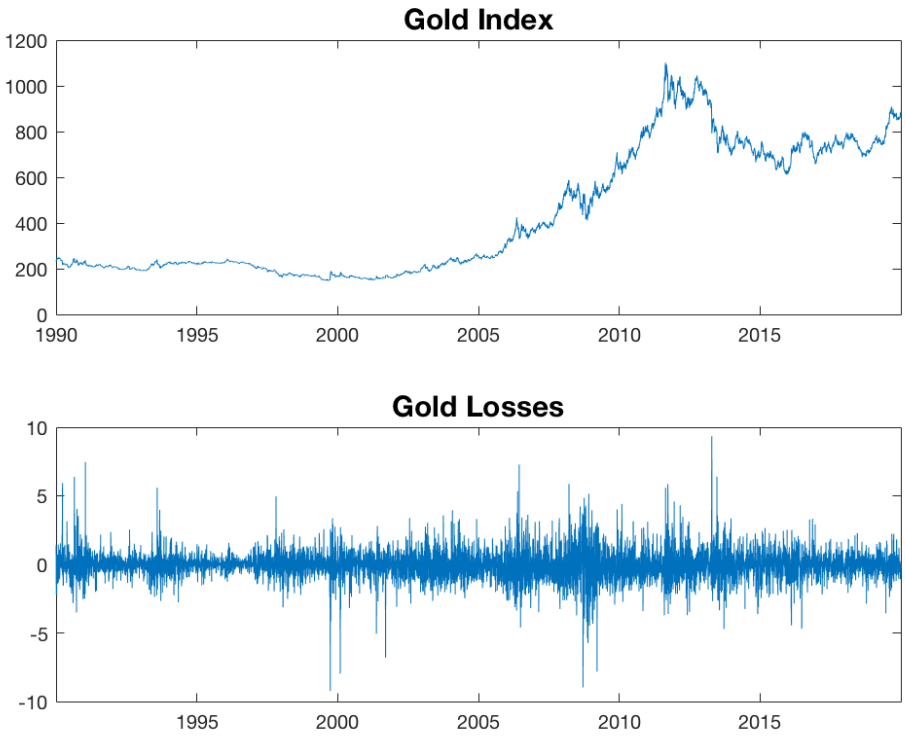


Figure 12: The evolution of the Gold spot index SPGSGC (upper panel) and the corresponding losses (lower panel). Both plots are over the time period 1990-2019.

Figure 12 displays the daily spot price of gold and the daily losses. It is quite evident that gold has had a far less erratic price growth compared to oil, with the largest daily loss being just below 10%. Despite having the lowest standard deviation as can be seen in Table 1, the high kurtosis is attributed to the rather large amount of outliers. Regardless, the price of gold lays almost dormant for the first 15 years with an expansive increase in value between 2005 to 2012 followed up by a period of turmoil. This could to some extent be explained by crises such as the 9/11 attacks, the dot-com-bubble bursting in the early 2000's and this financial crisis of 2008, which would support the idea that investors turn to gold as a mean to hedge themselves against a volatile stock market[27].

[27]Which is the conclusion by Baur and Lucey (2010), in their study "Is Gold a Hedge or a Safe Haven? An Analysis of Stocks, Bonds and Gold".

4.2.3 SPGSSO - Soybean

The soybean index SPGSSO was first launched in 1991 with a starting index value on December 31, 1969. According to S&P Global, the index "*provides investors with a reliable and publicly available benchmark for investment performance in the soybean commodity market*".

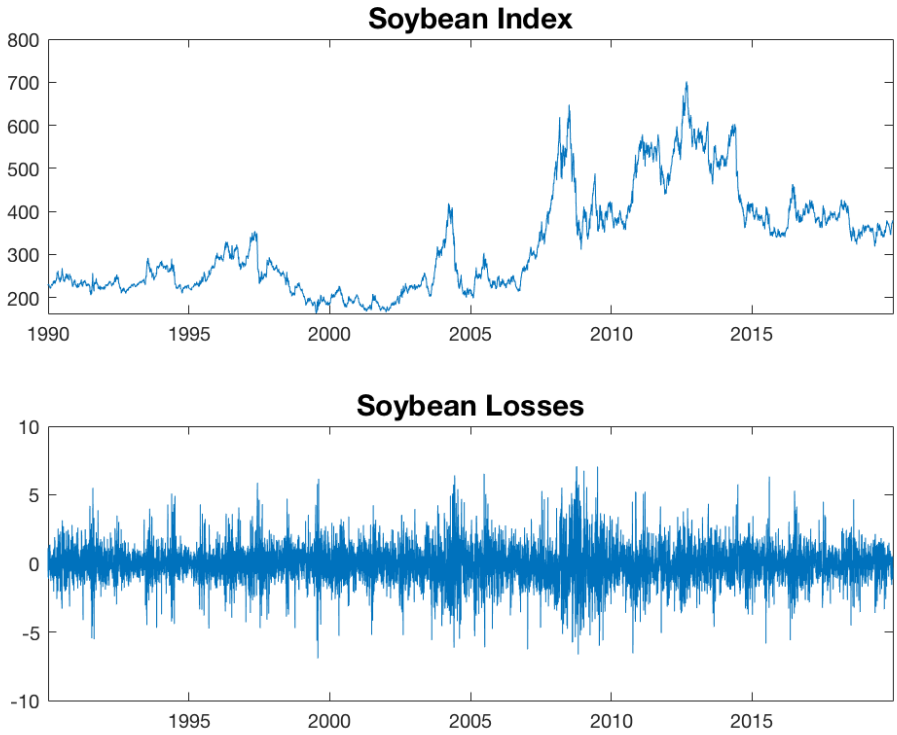


Figure 13: The evolution of the Soybean spot index SPGSSO and the corresponding losses (lower panel). Both plots are over the time period 1990-2019.

The progression of the price of soybean is quite different from that of oil and gold. Soybean has displayed a more consistent volatility pattern throughout the entire time period as can be seen from the losses in Figure 13. The 'thick' pattern of the losses that indicates a period of relatively high volatility is much closer and more evenly spread than for gold and oil. The losses also show a seasonality pattern, with clustering seemingly forming in regular intervals. This could be explained by the fact that the price of soybeans are heavily influenced by the season of the year, weather and harvest yield (*Introduction to Grains and Oilseeds - Understanding Seasonality in Grains*, 2020). However, the individual daily losses and gains are smaller in magnitude in comparison to the other commodities. This is further supported by comparing the standard deviation and kurtosis of Soybean and Gold in Table 1. Soybean displays a higher standard deviation while exhibiting lower kurtosis, which clearly indicates a generally more volatile pattern but with less severe outliers.

A strong upwards trend, much like gold and oil is evident for the first one and a half

decade of the 21th century. This is probably a combination of increased demand for soy produced animal food in China (which is the worlds largest importer accounting for 60% of global imports), increased popularity for soy based human food products as well as increased usage of soybean oil in food products, biodiesel and bioheat.[28]

4.3 LSTM data preparation

In general, most academic studies in machine learning agrees that data should be divided into three sample sets, namely a training set which will be available for the model to train on, a validation set which will be used to prevent overfitting and a testing set which consists of new unseen data to evaluate the model on. However, there seems to be lack of consensus around how to partition the three different sets in the most suitable way. This is due to the fact that each data set being trained on is unique and depending on the task at hand. In general, the data is divided somewhere between 70%/15%/15% to 60%/20%/20% into training, validation and testing set respectively.

The data in this thesis is split 70%/10%/20/% for the categories (see Table 2). The reason for a slightly lowered allocation to the validation set is due to the fact that the size of the data set is on the smaller end of the spectrum. Since the validation set is only there to prevent overtraining, it is often customary to allocate a smaller weight to the validation set.

Table 2: The number of observations in each data subset, with start dates and end dates.

Data subset	**Start date**	**End date**	**Days**	**%**
Training	1990-01-01	2010-12-28	5296	70%
Validation	2010-12-29	2013-12-23	756	10%
Test	2013-12-27	2019-12-31	1514	20%

Before the raw data can be fed to the network it has to be modified. As the task of the LSTM is to forecast future volatility, in combination with the fact that it is a supervised learning[29] approach there has to be a sequence of target volatility for the network to analyze and train on in order to come up with its predictions. Realized volatility (RV) is used as the volatility estimator since it is a model-free, ex-post estimator of volatility[30]. Realized volatility is given as

$$\mathrm{RV}_{i,t} = \sqrt{R_{i,t}^2} \tag{23}$$

[28]Visit https://ncsoy.org/media-resources/uses-of-soybeans/ for more info.

[29]Supervised learning refers to there being a desired outcome where the network has to infer a pattern to match the target outcome. Whereas unsupervised learning allows the network to discover features of the input letting the neural network decide for itself what the outcome should be.

[30]Andersen and Bollerslev (1998b) found that realized volatility, while a noisy indicator, is unbiased for daily volatility given sufficient sampling frequency. Unfortunately the data is only sampled once per day which may give rise to a biased estimation of realized volatility.

where $R_{i,t}$ is the return of commodity i at time t. The input data for LSTM is kept univariate, *i.e.* historical realized volatility is the only predictor utilized in order to predict one step ahead volatility. This is done in order to keep the results from the LSTM-network and GARCH(1,1) comparable and to limit the computational burden as elaborated on in section 4.1.

The sequence of the data is preserved and not randomly partitioned, in order to preserve long term dependencies. The data series is not normalized between [0,1]. As the data fed to the neural network is univariate there is no need to normalize the data, as normalization is often done in order to be able to compare multiple different parameters that might have different scaling. Normalization is also sometimes done in order to reduce the chances of vanishing or exploding gradients. However, after many iterations, it was deemed that normalization did not improve the results.

4.4 Setting hyperparameters for LSTM

There are no optimal settings of hyperparameters as each data set is unique and therefore the most optimal hyperparameters have to be found manually, often by trial error (Goodfellow, Bengio, and Courville, 2016). A few hyperparameters remain unchanged throughout the process due to the very nature of the task with further elaborations presented below. These are:

- Cost function: The cost function chosen is mean squared error (MSE). It is often the preferred error function when used in regression problems as it punishes large deviations. As the goal is to forecast the volatility in order to estimate VaR and ES, large deviations becomes particularly undesirable. This cost function is used to evaluate training, validation and test samples.

- Optimizer function: The optimizer used is the Adam optimizer as it is has been shown to have the best performance for recurring networks (Kingma and Ba, 2014).

- Sequence length: The sequence length refers to the number of observations the network should use in order to forecast the desired outputs. Since LSTM should have long term memory there is little reason to have a long sequence length. Harmon and Klabjan (2018) and Chong, Han, and Park (2017) used a sequence of 10 days while K. Chen, Zhou, and Dai (2015) used 30 days, suggesting that the sequence length could be anywhere between 10-30. In order to reduce computational burden, a sequence length of 10 is chosen.

- Number of hidden layers: Two hidden layers is chosen in order to increase the neural networks capacity.

- Dropout rate: Dropout rate is left at 0.25, *i.e.* a 25% chance that a node and its associated nodes randomly get removed.

- Activation function: Linear activation is preferred due to the nature of the task being a regression problem and that the data is not being normalized.

The remaining hyperparameters are manually found by iterating over multiple different configurations, slightly changing the hyperparameters between iterations to improve the models accuracy. The progress of each hyperparameter is presented below:

- Number of hidden nodes: In effort to simplify the process, the same number of nodes are always present in both hidden layers. First order of business is to determine whether the complexity of the model needs to be increased or decreased by changing the number of hidden nodes by steps of 25 and then gradually lower the step sizes in order to determine the optimal number of hidden nodes.

- Batch size: Initially, a larger batch size of 512 is chosen in order to speed up each epoch and thereby achieving preliminary results quicker. Thereafter, batch sizes are gradually reduced to 256 and lastly 128 in order to get more accurate results. Smaller batch sizes results in weights being updated more frequently for each epoch.

- Epochs: The number of epochs influences the accuracy of the model. Ideally, the model should stop running once the validation errors starts to increase in order to prevent overfitting. However, this is not always possible due to the increasing memory usage of the model, hence, an upper limit of how many epochs to run through must be chosen. Therefore, 500 epochs are performed unless signs of overfitting occurs earlier.

- Learning rate: The learning rate is initially set significantly smaller than usual ($\eta = 0.00001$) due to the non normalized data. It is then slowly decreased for each run in order to refine the model. According to Goodfellow, Bengio, and Courville (2016), the learning rate is arguably the most important hyperparameter and it is crucial to set correctly. Therefore, it is the last hyperparameter adjusted as the other hyperparameters needed to be as well specified as possible before tweaking the learning rate.

The hyperparameters are kept the same for each commodity as there turned out to be little variation in performance between each commodity given the same hyperparameters. The finalized hyperparameters are presented in Table 3. All LSTM-related implementation is performed in Python 3.7 with the Keras library package and a Tensorflow backend.

Table 3: The finalized hyperparameters for the LSTM network. These hyperparameters are used for all commodities.

Hyperparameter	**Value**
Hidden layers	2
Hidden nodes per layer	75
Learning rate	0.000007
Epochs	500

4.5 Estimating GARCH(1,1) parameters with ML

When estimating the volatility series with GARCH(1,1), there are some considerations to the estimation frequency of the ML parameters (μ, ω, α, β). Ideally one would like the parameters to portray the actual relationship of the data which may or may not change over time. Note again, that this may lead to poor forecasts all together if the parameters does in fact not describe the actual relationship well. The two extreme cases would be to either only estimate the parameters once and use the estimations for the entire test-data, or to reestimate the parameters for every step of the rolling window.

The disadvantages of the first approach is that it may be to dependant on the actual observations in the training data and could perform poorly on new data or if the underlying relationships change. The second approach is adaptable to changes in the underlying parameters, however, it is computationally exhaustive and might overinterpret the significance of temporary prevailing market conditions. The approach in this thesis is however the first alternative. This is to keep GARCH and LSTM as comparable as possible, in the sense that both models get the same data to 'train' on (or estimate parameters in the case of GARCH). Since GARCH is not a machine learning approach, there is no concept of overfitting, and therefore the validation set becomes redundant. Hence, the training data and validation data (80% of the total dataset) are combined and used to estimate the parameters.

4.6 Estimating & backtesting VaR and ES

There are different approaches to estimating and backtesting VaR and ES. The methodology in this thesis is to utilize a non-parametric, VWHS approach. The motivation behind the decision is to the extent possible avoid an explicit distributional assumption, as is the case for parametric approaches. Since the major advantage of ANN is its efficiency in mapping complex non-linear functions, it would not allow for its true capacity if distributional assumptions were imposed. Further, one of the most critical issues with historical approaches is that they are slow in reacting to changing market conditions, which is largely mitigated with a volatility weighting. According to Hull and White (1998), rescaling the losses with volatility weighting comes with significant improvement to simply conducting

BHS.

The most common for historical simulations like VWHS is to incorporate a rolling window with a fixed number of observations in order to estimate $\widehat{VaR_{\alpha,t+1}}$ and $\widehat{ES_{\alpha,t+1}}$, *i.e.* for the very next day following the window. All the losses in the rolling window are rescaled according to the procedure in section 2.5.2, and then $\widehat{VaR_{\alpha,t+1}}$ and $\widehat{ES_{\alpha,t+1}}$ are the α-quantile and the average above the α-quantile respectively, according to the procedure in section 2.5.1. After estimating $\widehat{VaR_{\alpha,t+1}}$ and $\widehat{ES_{\alpha,t+1}}$ the window moves one day and the estimates are evaluated against the realized loss of that day (ℓ_{t+1}). The oldest observation in the window is then discarded while the new one (ℓ_{t+1}) is included in the next window. This new window is then used to estimate $\widehat{VaR_{\alpha,t+2}}$ and $\widehat{ES_{\alpha,t+2}}$, and this procedure continues until the window has 'walked' through the entire sample.

The size of the rolling window is another important aspect to consider. A long enough rolling window to incorporate turbulent periods is desirable, but a too long window might include observations that are of minor relevance for the economic state 'tomorrow'. While Hull (2015) promotes the computational convenience of using a window with 501 days (500 returns), a window of 1001 days (1000 returns) is adopted in this thesis to keep a longer memory of approximately 4 trading years.

Since LSTM only produces volatility forecasts for the test set, the same test period is utilized for GARCH. This results in 1514 VaR and ES estimated for GARCH and 1504 estimated for LSTM[31] for the timeperiod 2013-12-27 to 2019-12-31. The estimations are then backtested according to the procedures outlined in section 2.6.1, 2.6.2 and 2.6.3. All tests are evaluated according to the following critical values:

* Significant on a 10% level

** Significant on a 5% level

*** Significant on a 1% level

[31]The difference in number of volatility estimations is down to the fact that the LSTM model set up has a sequence length of 10 days before making a forecast. The difference in number of volatility estimations reflects the sequence length of the LSTM network.

5 Empirical results

Section 5 outlines the results from the thesis. It starts with displaying the results of the GARCH(1,1) approach, followed by the outcome from the LSTM approach, divided by the different commodities.

5.1 Oil

Table 4: Results from running a VWHS with volatility estimates from GARCH(1,1) on oil for 1513 consecutive days (20% of the sample-size) ending on Dec 31, 2019. α is the confidence level for VaR and ES, Violations are the amount of losses exceeding VaR, CI 95% is a confidence interval for the Kupiec test with 2.5% in each tail, Kupiec p-value, Christ. p-value and ES z-value are the test statistic for the three different tests. The optimized GARCH parameters are $\mu = 0.0006$, $\omega = 0.0000$, $\alpha = 0.0620$, $\beta = 0.9322$

GARCH(1,1) Oil

α	Violations	CI 95%	Kupiec p-value	Christ. p-value	ES z-value
95.0%	85	[59 , 93]	0.15	0.13	-0.14
97.5%	42	[26 , 50]	0.27	0.87	-0.14
99.0%	20	[8 , 23]	0.13	0.46	-0.34

As can be seen from Table 4, the GARCH(1,1) approach works well in estimating VaR and ES for oil. Regardless of confidence level, the model delivers a VaR estimate where the violations are within the confidence interval for the Kupiec test on a 95% certainty level. Further, Christoffersen's test shows p-values ranging from 0.13 to 0.87, indicating that the violations are adequately spread and not too clustered together. It should be noted that the model performs the best for the 97.5% confidence level, where the highest p-values for Kupiec and Christoffersen's test are displayed of 0.27 and 0.87 respectively. There is no clear cut difference in which out of the remaining confidence levels the model performes the best on in terms of VaR. The 95% confidence level performs slightly better according to Kupiecs test and the 99% level has a higher p-value for Christoffersen's test.

When it comes to ES, GARCH(1,1) again delivers solid results, with no violations of the proposed z-value of -0.7. However, it should be noted from the negative z-values that it consistently slightly underestimates the actual ES, meaning that it predicts on average the loss exceeding VaR slightly lower than the actual loss exceeding VaR. While the 97.5% and 95% confidence levels are close to a perfect zero z-value (-0.14), it is slightly worse for the 99% confidence level (-0.34).

In conclusion, GARCH(1,1) is not rejected for either VaR nor ES for any of the confidence levels. Additionally, the VaR violations are satisfactory spread over the test

period according to Christoffersen's test. The model seems to be the most suitable for the 97.5% confidence level, with some ambiguity for the remaning confidence levels.

Table 5: Results from running a VWHS with volatility estimates from LSTM on oil for 1504 consecutive days (20% of the sample-size) ending on Dec 31, 2019. α is the confidence level for VaR and ES, Violations are the amount of losses exceeding VaR, CI 95% is a confidence interval for the Kupiec test with 2.5% in each tail, Kupiec p-value, Christ. p-value and ES z-value are the test statistic for the three different tests.

LSTM Oil

α	Violations	CI 95%	Kupiec p-value	Christ. p-value	ES z-value
95.0%	86	[59 , 92]	0.11	0.65	-0.15
97.5%	42	[26 , 50]	0.25	0.87	-0.15
99.0%	20	[8 , 23]	0.13	0.46	-0.35

The results from LSTM in Table 5 reveals somewhat similar results to those obtained by the GARCH(1,1). The model sufficiently produce VaR estimates that are within acceptable ranges for the Kupiec test, again with the confidence interval 97.5% being the most adequate. The amount of violations are very much in line with those obtained by GARCH(1,1), with slightly lower p-values for the 95% and 97.5% confidence levels. In terms of the independence of the violations, Christoffersen's test is passed with a wide margin for all the specifications and reveal highly similar p-values to GARCH(1,1) with the exception of a much higher p-value for the 95% confidence level.

The ES estimates are well within acceptable ranges but similarly to GARCH(1,1) slightly underestimates the actual average VaR violation (i.e ES). The z-values are again very close to the ones obtained by GARCH(1,1).

In conclusion, even though the performance is very similar, GARCH(1,1) seems slightly more accurate in its VaR estimates in terms of violations, while LSTM seems better at spreading them according to the improved p-value for the 95% confidence level. The z-values for ES are almost identical, and the methods seems interchangeable in estimating ES for oil.

5.2 Gold

Table 6: Results from running a VWHS with volatility estimates from a GARCH(1,1) on gold for 1513 consecutive days (20% of the sample-size) ending on Dec 31, 2019. α is the confidence level for VaR and ES, Violations are the amount of losses exceeding VaR, CI 95% is a confidence interval for the Kupiec test with 2.5% in each tail, Kupiec p-value, Christ. p-value and ES z-value are the test statistic for the three different tests. The optimized GARCH(1,1) parameters are $\mu = 0.0000$, $\omega = 0.0000$, $\alpha = 0.0429$, $\beta = 0.9591$

GARCH(1,1) Gold

α	Violations	CI 95%	Kupiec p-value	Christ. p-value	ES z-value
95.0%	73	[59 , 93]	0.41	0.10*	0.09
97.5%	37	[26 , 50]	0.49	0.92	0.11
99.0%	11	[8 , 23]	0.17	0.07*	0.37

From Table 6, it is quite evident that a GARCH(1,1) approach works well on gold for estimating both VaR and ES. The number of violations sits almost perfectly in the middle of the confidence interval with the exception of the 99% confidence level, where the number of violations are close to being too few. For the 95% confidence level and 97.5% confidence level, the amount of violations are very close to the expected amounts of 75.7 and 37.8 respectively. The performance for the Kupiecs test on gold is improved in comparison to oil, with the same pattern for the confidence levels with the 97.5% level being the most accurate, followed by 95% and lastly 99%.

However, GARCH(1,1) drops in performance when it comes to the Christoffersen's test. With p-values of 0.10 and 0.07 for the 95% and 99% confidence levels respectively, resulting in both being rejected at the 10% significance level[32]. This indicates that the violations of the aforementioned confidence levels are not adequately spread. The performance is however inconsistent, and the 97.5% confidence level displays an almost perfect score of 0.92.

When it comes to ES, The positive z-value obtained suggests that the estimated ES is, in general, somewhat overestimated[33] for gold. This could partially be explained by the fact that the test period (2013-12-27 to 2019-12-31) is a somewhat calmer period than the training period. Regardless, the models are not rejected as they are close to zero, and the Acerbi Szekely test only rejects models that underestimate ES.

In conclusion, GARCH(1,1) works well for gold. The VaR estimates are well within the confidence intervals, with extraordinary p-values for the 95% and 97.5% levels. Christoffersens test shows ambiguity as it displays a very high p-value for the 97.5% confidence

[32]The p-value of the 95% confidence level is 0.09938, which rounds to 0.10

[33]It should be mentioned that since there is no suggested critical value for the positive region, it is somewhat problematic to claim an 'overestimation', however the term in this context refers to the positive z-values.

level and much lower values for the remaining confidence levels, resulting in rejections. The ES estimates are satisfactory far from the rejection region, but with somewhat over-estimations of the 'actual' ES.

Table 7: Results from running a VWHS with volatility estimates from LSTM on gold for 1504 consecutive days (20% of the sample-size) ending on Dec 31, 2019. α is the confidence level for VaR and ES, Violations are the amount of losses exceeding VaR, CI 95% is a confidence interval for the Kupiec test with 2.5% in each tail, Kupiec p-value, Christ. p-value and ES z-value are the test statistic for the three different tests.

LSTM Gold

α	Violations	CI 95%	Kupiec p-value	Christ. p-value	ES z-value
95.0%	58	[59 , 92]	0.02**	0.34	0.24
97.5%	33	[26 , 50]	0.25	0.75	0.18
99.0%	10	[8 , 23]	0.12	0.71	0.37

As can be seen from Table 7, LSTM seems to consistently overestimate VaR, with actual number of violations systematically being on the lower end of the confidence interval. The Kupiec test is even failed at the 95% confidence level with a p-value of 0.02. When compared to GARCH(1,1), it is evident that GARCH(1,1) does not overestimate in the same fashion for the 95% and 97.5% confidence levels, as the amount of violations are very close to expectations.

Interestingly, the somewhat lower performing VaR estimates does not translate into poor results for Christoffersen's test. The p-values are higher for both the 95% and 99% confidence level in comparison to GARCH(1,1). However, this fact might be explained by the lower amount of violations, which everything equal should result in a lower probability of violation-clustering.

When it comes to ES, LSTM is very much in line with GARCH(1,1) in that it is far from the rejection region, and instead somewhat overestimates the 'actual' ES. The z-values are however slightly further from zero, indicating that GARCH(1,1) works somewhat better in this regard.

All in all, both GARCH(1,1) and LSTM perform satisfactory results for VaR on gold, with the exception being LSTM on the 95% level and GARCH(1,1) on the 95- and 99% levels where the Kupiecs test and Christoffersens tests respectively are rejected. GARCH(1,1) displays more accurate VaR estimates than LSTM in regards to the high Kupiec p-values. LSTM does however produce more well spread violations than GARCH(1,1) for the 95% and 99% levels, as can be seen from the higher p-values of Christoffersen's test. Lastly, ES is also far from rejection for both models at all confidence levels. However, GARCH(1,1) is somewhat more accurate (with z-values closer to zero), and it should be noted that

opposite to the results of oil, ES is consistently overestimated[34].

5.3 Soybean

Table 8: Results from running a VWHS with volatility estimates from GARCH(1,1) on Soybean for 1513 consecutive days (20% of the sample-size) ending on Dec 31, 2019. α is the confidence level for VaR and ES, Violations are the amount of losses exceeding VaR, CI 95% is a confidence interval for the Kupiec test with 2.5% in each tail, Kupiec p-value, Christ. p-value and ES z-value are the test statistic for the three different tests. The optimized GARCH(1,1) parameters are $\mu = 0.0003$, $\omega = 0.0000$, $\alpha = 0.0687$, $\beta = 0.9191$

GARCH(1,1) Soybean

α	Violations	CI 95%	Kupiec p-value	Christ. p-value	ES z-value
95.0%	69	[59 , 93]	0.24	0.13	0.11
97.5%	32	[26 , 50]	0.19	0.18	0.17
99.0%	11	[8 , 23]	0.17	0.69	0.25

The results from the GARCH(1,1) approach on soybean displayed in Table 8 portrays yet again satisfactory results. The VaR violations are within the confidence intervals, however, VaR is slightly overestimated in comparison to *e.g.* gold, resulting in lower p-values of Kupiec's test. Quite interestingly, the confidence level that portrays the best results in terms of VaR violations (95% level) has the most clustered violations according to Christophersen's test and vice versa for the lowest performer in Kupiecs test (99% level). Regardless, the model is not rejected for either of the VaR tests on any of the confidence levels.

When it comes to the ES estimates, they are on average very close to the average loss exceeding VaR, with z-values close to zero. The same pattern as for gold can be seen with a slight overestimation of ES for all the confidence levels. This time its somewhat more ambiguous as to why since the level of volatility cannot be distinctly separated between the training set and test set from an ocular inspection of 13. Even though the differences in performances on ES are slim for the different confidence levels, it should be noted that the estimates are somewhat better the lower the confidence level.

In conclusion it is evident that GARCH(1,1) manages to estimate VaR and ES well even for soybean. The VaR estimates are somewhat less accurate than the ones delivered for gold in terms of the amount of violations being further from the expected. Christoffersen's test and the z-value are also outside the rejection region and when compared to oil and gold, it performs better for some confidence levels and worse for some.

[34]It should be mentioned that since there is no suggested critical value for the positive region, it is somewhat problematic to claim an 'overestimation'. However, the term in this context refers to the positive z-values.

Table 9: Results from running a VWHS with volatility estimates from LSTM on soybean for 1504 consecutive days (20% of the sample-size) ending on Dec 31, 2019. α is the confidence level for VaR and ES, Violations are the amount of losses exceeding VaR, CI 95% is a confidence interval for the Kupiec test with 2.5% in each tail, Kupiec p-value, Christ. p-value and ES z-value are the test statistic for the three different tests.

LSTM Soybean

α	Violations	CI 95%	Kupiec p-value	Christ. p-value	ES z-value
95.0%	67	[59 , 92]	0.18	0.26	0.13
97.5%	31	[26 , 50]	0.16	0.67	0.18
99.0%	13	[8 , 23]	0.34	0.63	0.12

As can be seen from Table 9, the VaR estimate based on LSTM delivers violations that are quite similar to GARCH(1,1) for the 95% and 97.5% level, with the 99% level being closer to the expected amount of violations, resulting in a high p-Value of 0.34. Christoffersen's test is also passed with a wide margin for all confidence levels, with higher p-values than GARCH(1,1) for the 95% and 97.5% confidence levels.

When it comes to the ES estimates, they are well specified, with p-values close to zero. The p-values for the 95% and 97.5% confidence levels are just very slightly higher than those obtained by GARCH(1,1), while the 99% confidence level is somewhat improved.

In summary, one can see that both approaches perform well for soybean, just like for the other two commodities. LSTM displays improved results for Kupiec's test and ES for the 99% confidence level and Christoffersens test for the remaining two confidence levels. The remaining test/confidence-level combinations are very similar between the models. Therefore it is not clear cut which model works the best overall on soybean, even if the improvements for the 99% confidence level of LSTM are the most compelling.

6 Discussion

The results indicate that VaR and ES estimates based on volatility forecasted by GARCH(1,1) works exceptionally well for the three commodities investigated, passing almost[35] all tests for the examined commodities on all confidence levels. For oil, the number of VaR violations are consequently slightly higher than expected on all confidence levels. The opposite behaviour is found for soybean with the number of VaR violations sitting in the lower part of the confidence interval. For said commodities, the VaR estimates provided by LSTM yield highly similar amount of violations. Furthermore, the same relationship is present in that the number of observations are above expected for oil and below expected for soybean. This strongly indicates that oil is a far more volatile commodity compared to soybean since both approaches results in more violations than expected. This is in line with previous research, which has shown that crude oil is one of the most, if not the most, volatile commodity (Regnier, 2007).

The two approaches diverges in results when it comes to gold, with the VaR estimates based on the LSTM-network systematically generating too few violations, resulting in the Kupiec's test being rejected at a 5% level for the 95% confidence level. The GARCH(1,1) approach on the other hand yields an almost ideal amount of violations for the same confidence level as well as the 97.5% confidence level. ANNs somewhat lower performance in this aspect could perhaps partly be explained by how the training och test samples are divided. From the lower panel in Figure 12 one can observe that gold in general has seen a relatively calm period from 2015 and forward (testing period being between 2013-12-27 to 2019-12-31). Since ANNs are extremely dependent on the training data it is no surprise that the LSTM forecasts yields a much more conservative estimation of VaR, as the training data has been more volatile than the test data.

Quite interestingly, the fact that the GARCH(1,1) approach seems superior to LSTM in estimating VaR, as shown by a higher p-value for eight out of nine of the Kupiec tests, does not translate into better performance for Christoffersen's test. This is most evident by the rejection of the test at the 10% level for the 95% and 99% confidence levels on gold as compared to the LSTM p-values of 0.34 and 0.71 respectively. While one may argue that a rejection at a 10% level is not very significant, the GARCH(1,1) approach to estimating VaR appears to systematically yield lower p-values for Christoffersens test compared to the LSTM approach. This is somewhat worrying as it indicates that the GARCH(1,1) approach to estimating VaR might not be as adequate in handling changing and clustering volatility, which commodities are known to exhibit. Multiple consecutive VaR violations could leave a potential investor or bank insolvent. This means that from a regulators perspective, such as the Basel committee, an LSTM approach to estimating

[35] With the exception of Christoffersen's test on a 10% level for the 95% and 99% confidence levels for gold

VaR could be preferred in this regard.

However, under the current Basel regulations, capital requirements for market risk are no longer set by VaR but is now determined by ES. None of the z-values from the Acerby and Szekely tests suggests that ES is underestimated for any of the commodities, regardless of forecasting approach. For oil, the results are practically identical between the two approaches, gold displays that GARCH(1,1) results in z-values closer to zero for all of the confidence levels and soybean reveals mixed results in which approach yields the most accurate ES estimates. These findings indicate that given the same circumstances, both a GARCH(1,1)- and an LSTM approach works adequately in estimating ES. Consequently, a GARCH(1,1) might be favored as it produces slightly more accurate estimates overall, and does so without the same need of computing power and large data sets that an LSTM approach requires. Nonetheless, the Acerby and Szekely test accounts only for the magnitude of the violation and the total number of violations while completely disregarding the frequency of the violations (how closely the violations occur). Since frequent violations are a serious threat to the solvency of financial institutions, the incorporation by regulators of an independence test like Christoffersen's could be warranted. Keeping this in mind, one should be cautious in claiming that the GARCH(1,1) approach performs better in respect to ES.

The superiority of GARCH(1,1) for Kupiecs test and the Acerby and Szekely test are, to some extent, surprising. Hansen and Lunde (2005) did show that a GARCH(1,1) performs very well. Be that as it may, Table 10 (in Appendix D) shows that an LSTM approach results in both smaller mean squared errors and mean absolute errors compared to GARCH(1,1) for all three commodities investigated. Despite this, the improved accuracy of volatility forecasts fails to translate into superior VaR and ES estimates. Possibly suggesting that a VWHS does not capitalize on the improved accuracy. However, it should be noted that both models delivers satisfactory results with rather similar outputs. Therefore, the potential gain of slightly more accurate volatility forecasts might be limited.

With the results being somewhat similar and not clear cut, further research to commodities seems warranted. The commodities examined, being from different sub-categories, showed different characteristics and varying test results. While GARCH(1,1) and LSTM performed highly similar for oil and soybean, the real divergence is displayed for gold. Here it becomes evident that LSTM overestimates VaR in that it consistently produces less than expected violations. However, Christoffersen's test show high p-values that outperformes GARCH(1,1) for the highest and lowest confidence levels. The overestimation displayed by LSTM is likely to some extent an implication of what data ends up in the training- / test sets as touched upon before. Despite this, one can not rule out the possibility of gold, and other precious metals, having volatility characteristics that may be difficult for GARCH and LSTM to model. These metals are often used to diversify risk

and therefore historical returns are unlikely to be suitable predictors. This could partly explain LSTMs failure to adequately estimate VaR and GARCHs inability to compensate for the volatility clustering.

The volatility pattern of soybean differs from the other commodities investigated, in its periodic pattern, with large losses appearing almost in regular intervals. Since LSTM networks should have the ability to remember long term behaviour, an LSTM approach should be better suited in handling (given sufficient complexity) periodic volatility. This behaviour might also be found in other similar agricultural commodities since most agricultural products should share some type of seasonality in the pricing mechanisms. However, this does not translate into superior results for the LSTM conducted in this thesis. This is majorly attributed to the low complexity of the LSTM network adopted in this thesis.

As aforementioned, the rather primitive architecture is due to the lack of computing power. The LSTM results can hence be seen as a 'floor', *i.e.* with a more sophisticated neural network the volatility forecasts can only be improved upon. While the LSTM-network did show improved volatility forecasts, its failure to translate this into more improved VaR and ES estimates could be down to the improved accuracy not being significant enough, and the fact that GARCH(1,1) approach already performs satisfactory.

7 Conclusion

The purpose of the thesis was to evaluate and compare VaR and ES estimates for three different commodities based on volatility forecast provided by GARCH(1,1) and an LSTM network.

In general, the results indicate that both models deliver satisfactory VaR and ES estimates, and none of the approaches can immediately be deemed superior to the other. The GARCH(1,1) approach displays an amount of VaR violations that are closer to the expected according to the Kupiec test, while LSTM approach provides less clustering of the violations according to Christoffersen's test. Further, both methods deliver well specified ES estimates, without rejections for any of the commodities and confidence levels, with slightly better z-values displayed by the GARCH(1,1) approach.

Due to the fact that current regulations incorporates ES and that GARCH(1,1) performs slightly better in this regard, the computational burden and time consumption of adopting an LSTM approach cannot be justified. Despite this, some concerns are raised in regards to the frequency of the violations displayed by the GARCH(1,1) approach. Whether or not this is a problem that only commodities or a certain type of sub-class of commodities exhibits is unclear. Therefore we encourage researchers to investigate whether or not the behaviour is present in, not only other commodities, but also in larger investment portfolios where commodities are included. If GARCH(1,1) fails to account for clustered losses, more sophisticated neural networks may be warranted despite the increased complexity and computing power required.

References

Acerbi, Carlo and Balazs Szekely (2014). "Backtesting Expected Shortfall". In: *MSCI Inc.* URL: https://www.msci.com/documents/10199/22aa9922-f874-4060-b77a-0f0e267a489b.

Acerbi, Carlo and Dirk Tasche (2002). "Expected Shortfall: A Natural Coherent Alternative to Value at Risk". In: *Economic Notes* 31.2, pp. 379–388.

Andersen, Torben G. and Tim Bollerslev (1998a). "Answering the Skeptics: Yes, Standard Volatility Models do Provide Accurate Forecasts". In: *International Economic Review* 39.4, pp. 885–905. ISSN: 00206598, 14682354. URL: http://www.jstor.org/stable/2527343.

— (1998b). "Deutsche Mark–Dollar Volatility: Intraday Activity Patterns, Macroeconomic Announcements, and Longer Run Dependencies". In: *The Journal of Finance* 53.1, pp. 219–265. DOI: 10.1111/0022-1082.85732.

Artzner, Philippe et al. (1999). "Coherent Measures of Risk". In: *Mathematical Finance*.

Baur, Dirk G. and Brian M. Lucey (2010). "Is Gold a Hedge or a Safe Haven? An Analysis of Stocks, Bonds and Gold". In: *The Financial Review* 45.2, pp. 217–229.

Bengio, Y., P. Simard, and P. Frasconi (1994). "Learning long-term dependencies with gradient descent is difficult". In: *IEEE Transactions on Neural Networks* 5.2, pp. 157–166.

BIS (2013). "Fundamental Review of the Trading Book: A Revised Market Risk Framework". In: *Consultative document*.

— (2020). *History of the Basel Committee*. URL: https://www.bis.org/bcbs/history.htm (visited on 04/02/2020).

Bollerslev, Tim (1986). "Generalized Autoregressive Conditional Heteroskedasticity". In: *Journal of Economics* 31.3, pp. 307–327.

— (1987). "A Conditionally Heteroskedastic Time Series Model for Speculative Prices and Rates of Return". In: *The Review of Economics and Statistics* 69.3, pp. 542–547.

Buyuksahin, Bahattin, Michael Haigh, and Michel Robe (Nov. 2009). "Commodities and Equities: Ever a 'Market of One'?" In: *Journal of Alternative Investments* 12. DOI: 10.3905/JAI.2010.12.3.076.

Chen, K., Y. Zhou, and F. Dai (2015). "A LSTM-based method for stock returns prediction: A case study of China stock market". In: *2015 IEEE International Conference on Big Data (Big Data)*, pp. 2823–2824.

Chen, X., K. K. Lai, and J. Yen (2009). "A Statistical Neural Network Approach for Value-at-Risk Analysis". In: *2009 International Joint Conference on Computational Sciences and Optimization*. Vol. 2, pp. 17–21.

Chong, Eunsuk, Chulwoo Han, and Frank Park (Apr. 2017). "Deep Learning Networks for Stock Market Analysis and Prediction: Methodology, Data Representations, and

Case Studies". In: *Expert Systems with Applications* 83. DOI: 10.1016/j.eswa.2017.04.030.

Christoffersen, Peter F. (1998). "Evaluating Interval Forecasts". In: *International Economic Review* 39.4, pp. 841–862. ISSN: 00206598, 14682354. URL: http://www.jstor.org/stable/2527341.

Chung, Junyoung et al. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. arXiv: 1412.3555 [cs.NE].

Deaton, Angus and Guy Laroque (1992). "On the Behaviour of Commodity Prices". In: *The Review of Economic Studies* 59.1, pp. 1–23. ISSN: 00346527, 1467937X. URL: http://www.jstor.org/stable/2297923.

Donaldson, R.Glen and Mark Kamstra (1997). "An artificial neural network-GARCH model for international stock return volatility". In: *Journal of Empirical Finance* 4.1, pp. 17–46. ISSN: 0927-5398. DOI: https://doi.org/10.1016/S0927-5398(96)00011-4. URL: http://www.sciencedirect.com/science/article/pii/S0927539896000114.

Embrechts, Paul (2000). "Extreme Value Theory: Potential And Limitations As An Integrated Risk Management Tool". In: *Derivatives Use, Trading Regulation* 6.1, pp. 449–456.

Engle, Robert F. (1982). "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation". In: *Econometrica* 50.4, pp. 987–1007. ISSN: 00129682, 14680262. URL: http://www.jstor.org/stable/1912773.

Fama, Eugene (1965). "The Behavior of Stock-Market Prices". In: *The Journal of Business* 38.1, pp. 34–105.

Fischer, Thomas and Christopher Krauss (2018). "Deep learning with long short-term memory networks for financial market predictions". In: *European Journal of Operational Research* 270.2, pp. 654–669. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2017.11.054. URL: http://www.sciencedirect.com/science/article/pii/S0377221717310652.

Gençay, Ramazan and Min Qi (Aug. 2001). "Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging". In: *Neural Networks, IEEE Transactions on* 12, pp. 726–734. DOI: 10.1109/72.935086.

Giot, Pierre and Sébastien Laurent (Sept. 2003). "Market risk in commodity markets: a VaR approach". In: *Energy Economics* 25, pp. 435–457. DOI: https://doi.org/10.1016/S0140-9883(03)00052-5.

Glosten, Lawrence R., Ravi Jagannathan, and David E. Runkle (1993). "On the Relation between the Expected Value and the Volatility of the Nominal Excess Return on Stocks". In: *The Journal of Finance* 48.5, pp. 1779–1801. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2329067.

Goldman Sachs (2020). *SP GSCI COMMODITY INDEX*. URL: https://www.goldmansachs.com/what-we-do/global-markets/business-groups/sts-folder/gsci/ (visited on 04/08/2020).

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. MIT Press.

Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton (2013). *Speech Recognition with Deep Recurrent Neural Networks*. arXiv: 1303.5778 [cs.NE].

Hagerman, Robert (1978). "More evidence on the distribution of security returns". In: *Journal of Finance* 33.4, pp. 1213–1221.

Hansen, Peter R. and Asger Lunde (2005). "A forecast comparison of volatility models: does anything beat a GARCH(1,1)?" In: *Journal of Applied Econometrics* 20.7, pp. 873–889. DOI: 10.1002/jae.800. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jae.800. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.800.

Harmon, Mark and Diego Klabjan (2018). *Dynamic Prediction Length for Time Series with Sequence to Sequence Networks*. arXiv: 1807.00425 [cs.LG].

Haykin, Simon (1998). *Neural Networks: A Comprehensive Foundation*. 2nd. USA: Prentice Hall PTR. ISBN: 0132733501.

Hinton, Geoffrey E. et al. (2012). *Improving neural networks by preventing co-adaptation of feature detectors*. arXiv: 1207.0580 [cs.NE].

Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). "Long Short-Term Memory". In: *Neural Comput.* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

Hoffer, Elad, Itay Hubara, and Daniel Soudry (2017). *Train longer, generalize better: closing the generalization gap in large batch training of neural networks*. arXiv: 1705.08741 [stat.ML].

Hu, Michael Y. and Christos Tsoukalas (1999). "Combining conditional volatility forecasts using neural networks: an application to the EMS exchange rates". In: *Journal of International Financial Markets, Institutions and Money* 9.4, pp. 407–422. ISSN: 1042-4431. DOI: https://doi.org/10.1016/S1042-4431(99)00015-3. URL: http://www.sciencedirect.com/science/article/pii/S1042443199000153.

Hull, John (2015). *Options, Futures and Other Derivatives*. 9th. One Lake Street, Upper Saddle River, New Jersey 07458: Pearson Education, Inc. ISBN: 978-0133456318.

— (2018). *Risk Management and Financial Institutions*. 5th. USA: John Wiley Sons. ISBN: 978-1-119-44809-9.

Hull, John and Alan White (1998). "Incorporating volatility updating into the historical simulation method for value at risk". In: *Journal of Risk* 1, pp. 5–19.

Hung, Jui-Cheng, Ming-Chih Lee, and Hung-Chun Liub (May 2008). "Estimation of value-at-risk for energy commodities via fat-tailed GARCH models". In: *Energy Economics* 30, pp. 1173–1191. DOI: https://doi.org/10.1016/j.eneco.2007.11.004.

Huntington, Hillard (1998). "Crude Oil Prices and U.S. Economic Performance: Where Does the Asymmetry Reside?" In: *The Energy Journal* Volume19.Number 4, pp. 107–132. URL: https://EconPapers.repec.org/RePEc:aen:journl:1998v19-04-a05.

IG (2018). *Top 10 most traded commodities in the world*. URL: https://www.ig.com/en/trading-opportunities/top-10-most-traded-commodities-180905 (visited on 04/21/2020).

Introduction to Grains and Oilseeds - Understanding Seasonality in Grains (2020). URL: https://www.cmegroup.com/education/courses/introduction-to-grains-and-oilseeds/understanding-seasonality-in-grains.html (visited on 04/28/2020).

Kingma, Diederik and Jimmy Ba (Dec. 2014). "Adam: A Method for Stochastic Optimization". In: *International Conference on Learning Representations*.

Kohzadi, Nowrouz et al. (1996). "A comparison of artificial neural network and time series models for forecasting commodity prices". In: *Neurocomputing* 10.2. Financial Applications, Part I, pp. 169–181. ISSN: 0925-2312. DOI: https://doi.org/10.1016/0925-2312(95)00020-8. URL: http://www.sciencedirect.com/science/article/pii/0925231295000208.

Kristjanpoller, Werner, Anton Fadic, and Marcel C. Minutolo (2014). "Volatility forecast using hybrid Neural Network models". In: *Expert Systems with Applications* 41.5, pp. 2437–2442. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2013.09.043. URL: http://www.sciencedirect.com/science/article/pii/S0957417413007975.

Kristjanpoller, Werner and Marcel C. Minutolo (2015). "Gold price volatility: A forecasting approach using the Artificial Neural Network–GARCH model". In: *Expert Systems with Applications* 42.20, pp. 7245–7251. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2015.04.058. URL: http://www.sciencedirect.com/science/article/pii/S0957417415003000.

Kulkarni, Siddhivinayak and Imad Haidar (2009). *Forecasting Model for Crude Oil Price Using Artificial Neural Networks and Commodity Futures Prices*. arXiv: 0906.4838 [cs.NE].

Kupiec, Paul (1995). "Techniques for Verifying the Accuracy of Risk Management Models 3:73-84". In: *Journal of Derivatives*.

Linsmeier, Thomas and Neil Pearson (1996). "Risk Measurement: An Introduction to Value at Risk". In: *Working paper, University of Illinois at Urbana-Champaign*.

Lux, Thomas and Michele Marchesi (2000). "Volatility clustering in financial markets: a microsimulation of interacting agents". In: *International Journal of Theoretical and Applied Finance* 3.4, pp. 675–702.

Mandelbrot, Benoit (1963). "The Variation of Certain Speculative Prices". In: *The Journal of Business* 36.4, pp. 394–419. ISSN: 00219398, 15375374. URL: http://www.jstor.org/stable/2350970.

McCulloch, Warren S. and Walter Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". In: *Bulletin of Mathematical Biophysics* 5, pp. 115–133.

Min, Jonghyeon (2020). *Financial Market Trend Forecasting and Performance Analysis Using LSTM*. arXiv: 2004.01502 [q-fin.ST].

Mukul, Mukesh Kumar, Vikrant Kumar, and Sougata Ray (2012). "Gold ETF Performance: A Comparative Analysis of Monthly Returns". In: *The IUP Journal of Financial Risk Management* IX.2, pp. 59–63.

Nasdaq (2017). *Commodity Trading – Chapter 1: History of Commodity Trading*. URL: https://www.nasdaq.com/articles/commodity-trading-chapter-1-history-commodity-trading-2012-02-02 (visited on 04/08/2020).

Nelson, Daniel B. (1991). "Conditional Heteroskedasticity in Asset Returns: A New Approach". In: *Econometrica* 59.2, pp. 347–370. ISSN: 00129682, 14680262. URL: http://www.jstor.org/stable/2938260.

OECD (1993). *Commodity Price Volatility: Its Nature and Causes*. URL: http://www.oecd.org/officialdocuments/publicdisplaydocumentpdf/?cote=OCDE/GD(93)71&docLanguage=En.

Ohlsson, Mattias and Patrik Edén (Oct. 2019). *Lecture Notes on Introduction to Artificial NeuralNetworks and Deep Learning*.

Peltarion (2020). *Optimizers and compiler options*. URL: https://peltarion.com/knowledge-center/documentation/modeling-view/run-a-model/optimizers-and-compiler-options (visited on 04/16/2020).

Politis, Dimitris (Jan. 2004). "A Heavy-Tailed Distribution for ARCH Residuals with Application to Volatility Prediction". In: *Annals of Economics and Finance* 5, pp. 283–298.

Poon, Ser-Huang and Clive Granger (2003). "Forecasting Volatility in Financial Markets: A Review". In: *Journal of Economic Literature* XLI, pp. 478–539.

Regnier, Eva (2007). "Oil and energy price volatility". In: *Energy Economics* 29.3, pp. 405–427. ISSN: 0140-9883. DOI: https://doi.org/10.1016/j.eneco.2005.11.003. URL: http://www.sciencedirect.com/science/article/pii/S0140988305001118.

Rodriguez, Jesus (2018). *The Science Behind OpenAI Five that just Produced One of the Greatest Breakthrough in the History of AI*. URL: https://towardsdatascience.com/the-science-behind-openai-five-that-just-produced-one-of-the-greatest-breakthrough-in-the-history-b045bcdc2b69 (visited on 04/12/2020).

Rosenblatt, Frank (1958). "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review* 65.5, pp. 386–408.

Stanford, Stacy (2019). *DeepMind's AI, AlphaStar Showcases Significant Progress Towards AGI*. URL: https://medium.com/@stanford__ai/deepminds-ai-alphastar-showcases-significant-progress-towards-agi-93810c94fbe9 (visited on 04/12/2020).

Taleb, Nassim (2007). *The Black Swan: The Impact of the Highly Improbable*. 1st. USA: Random House. ISBN: 978-1400063512.

Tasche, Dirk (2002). "Expected Shortfall and Beyond". In: *Journal of Banking Finance* 26.7, pp. 1519–1533.

Taylor, Stephen (1986). *Modelling Financial Time Series*. John Wiley Sons, Chichester.

Tibshirani, Robert (1995). "Regression Shrinkage and Selection Via the Lasso". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1, pp. 267–288. DOI: 10.1111/j.2517-6161.1996.tb02080.x. eprint: https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.2517-6161.1996.tb02080.x. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1996.tb02080.x.

Tseng, Chih-Hsiung et al. (2008). "Artificial neural network model of the hybrid EGARCH volatility of the Taiwan stock index option prices". In: *Physica A: Statistical Mechanics and its Applications* 387.13, pp. 3192–3200. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2008.01.074. URL: http://www.sciencedirect.com/science/article/pii/S0378437108000320.

Vivian, Andrew and Mark E. Wohar (2012). "Commodity volatility breaks". In: *Journal of International Financial Markets, Institutions and Money* 22.2, pp. 395–422. DOI: 10.1016/j.intfin.2011.12..

Zhang, Peter, Eddy Patuwo, and Michael Hu (Mar. 1998). "Forecasting With Artificial Neural Networks: The State of the Art". In: *International Journal of Forecasting* 14, pp. 35–62. DOI: 10.1016/S0169-2070(97)00044-7.

A Coherency

Coherency is introduced by Artzner et al. (1999) as a desirable property of a risk measure, fullfilling four axioms. ES is always coherent while VaR is not. The properties of coherency are listed below:

1. Monotonic: $L_a \geq L_b \Rightarrow Risk(L_a) \geq Risk(L_b)$
2. Positive Homogeneous: $h > 0 \Rightarrow Risk(hL) = Risk(L) \cdot h$
3. Subadditive: $Risk(L_{a+b}) \leq Risk(L_a) + Risk(L_b)$
4. Translation Invariant: $Risk(L - c) = Risk(L) - c$

B Gradient descent

From Figure 2 it can be established that the signal fed to the activation function φ is

$$a = \sum_{k=1}^{P} \omega_k x_k + \omega_0 \tag{24}$$

where the bias, b, has been denoted as ω_0 instead. This means that the output function can be rewritten into:

$$y(a) = y(x, w) = \varphi(\sum_{k=1}^{K} \omega_k x_k + \omega_0) \tag{25}$$

In a regression task, the error function subject to minimization is

$$E(\omega) = \frac{1}{2N} \sum_{n=1}^{N} (d_n - y(x_n))^2 = \frac{1}{2N} \sum_{n=1}^{N} E(n) \tag{26}$$

Differentiating equation 26 with respect to the weights, ω, gives the following relationship:

$$\frac{\partial E(\omega)}{\partial \omega_k} = \frac{1}{2N} \sum_{n=1}^{N} \frac{\partial E(n)}{\partial y(n)} \frac{\partial y(n)}{\partial \omega_k} \tag{27}$$

where the right hand side has been extended with $\frac{\partial y(n)}{\partial y(n)}$. Solving for each partial differentiation gives

$$\frac{\partial E(n)}{\partial y(n)} = 2(y(x_n) - d_n) \tag{28}$$

$$\frac{\partial y(n)}{\partial \omega_k} = \varphi'(\sum_{k=1}^{K} \omega_k x_k + \omega_0) + x_{nk} \varphi(\sum_{k=1}^{K} \omega_k x_k + \omega_0) \tag{29}$$

Under the assumption that the activation function is a linear activation function and that bias is zero (typical for regression tasks) then equation 29 becomes

$$\frac{\partial y(n)}{\partial \omega_k} = x_{nk} \tag{30}$$

Equation 27 then becomes:

$$\frac{\partial E(\omega)}{\partial \omega_k} = \frac{2}{2N} \sum_{n=1}^{N} (y(x_n) - d_n) x_{nk} = \frac{1}{N} \sum_{n=1}^{N} (-\delta_n) x_{nk} \tag{31}$$

The weight updates after each iteration can therefore be rewritten as:

$$\Delta \omega_k = -\eta \frac{\partial E(\omega)}{\partial \omega_k} \tag{32}$$

C Figures

C.1 Oil

Figure 14: Training error and validation error during the optimization process for oil.

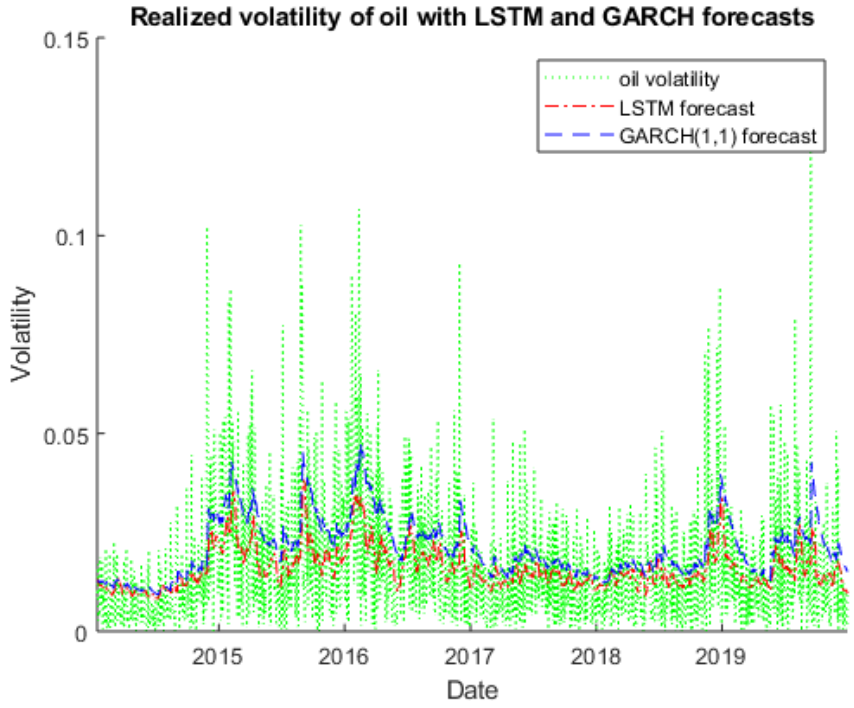


Figure 15: Realized volatility (green dotted line) of oil alongside the LSTM- (red dash dotted line) and GARCH(1,1) (blue dashed line) forecasts.

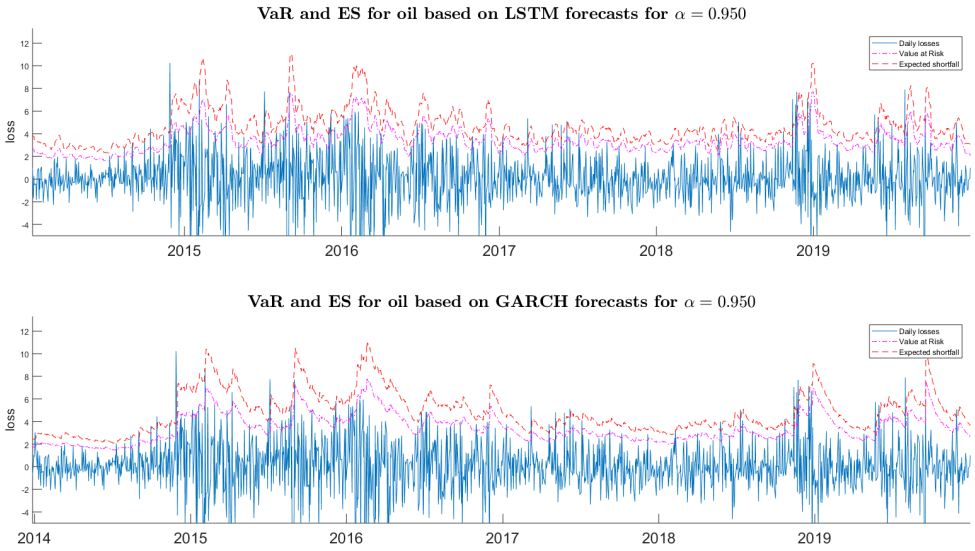


Figure 16: VaR and ES estimations for $\alpha = 0.975$ based on LSTM (top) and GARCH(1,1) (bottom) volatility forecasts for oil. The VaR and ES estimates are based on a VWHS with a rolling window of 1000 losses.

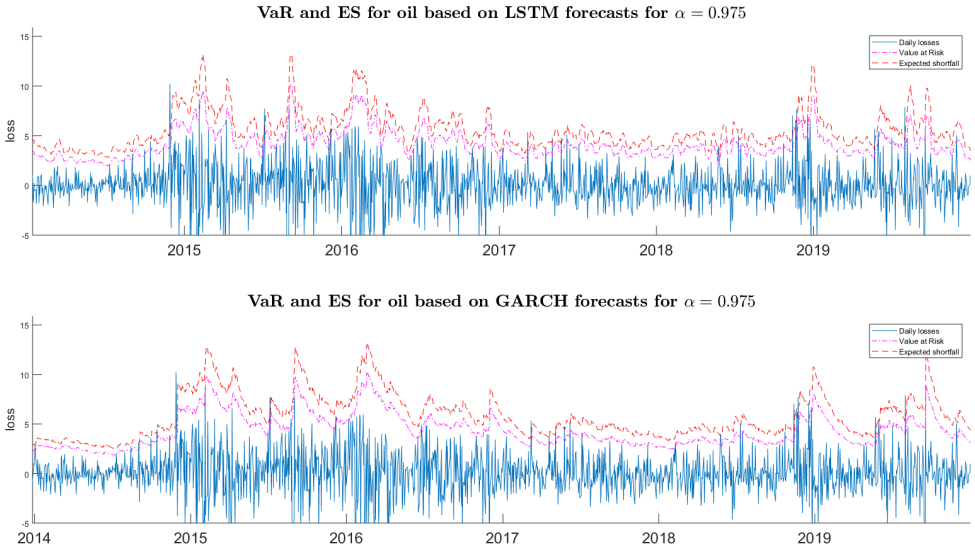


Figure 17: VaR and ES estimations for $\alpha = 0.975$ based on LSTM (top) and GARCH(1,1) (bottom) volatility forecasts for oil. The VaR and ES estimates are based on a VWHS with a rolling window of 1000 losses.

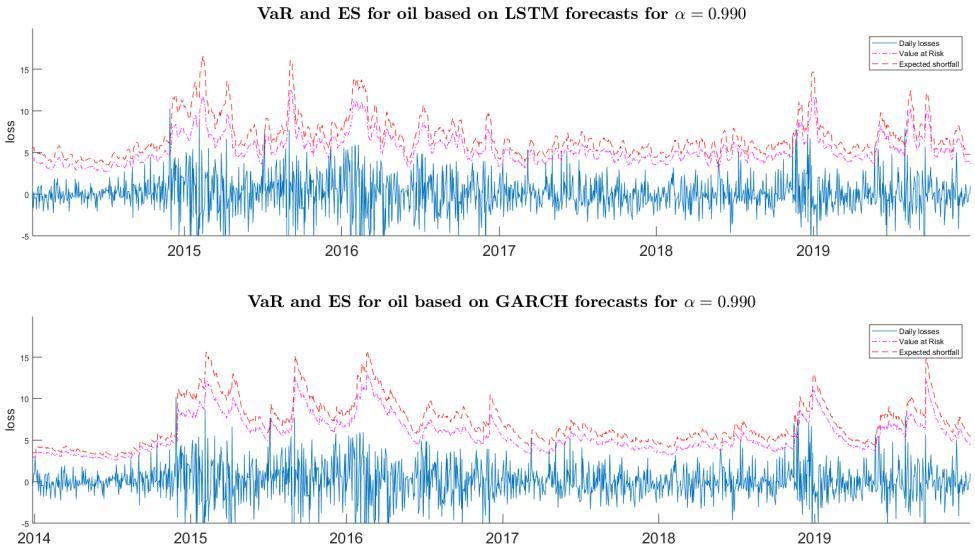


Figure 18: VaR and ES estimations for $\alpha = 0.990$ based on LSTM (top) and GARCH(1,1) (bottom) volatility forecasts for oil. The VaR and ES estimates are based on a VWHS with a rolling window of 1000 losses.

C.2 Gold

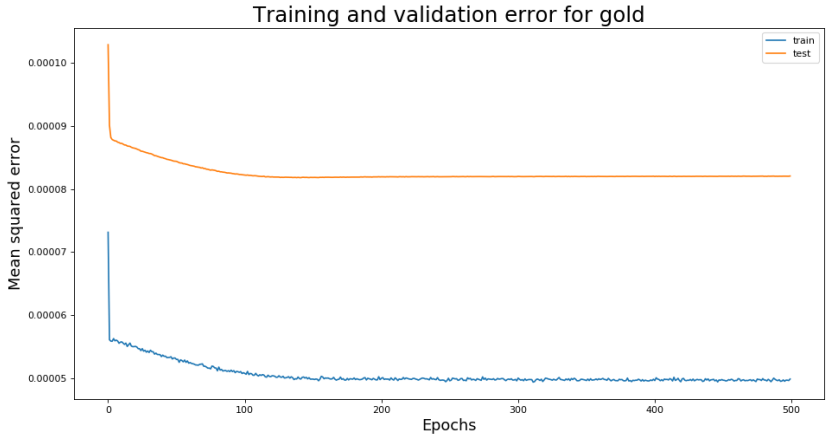


Figure 19: Training error and validation error during the optimization process for gold.

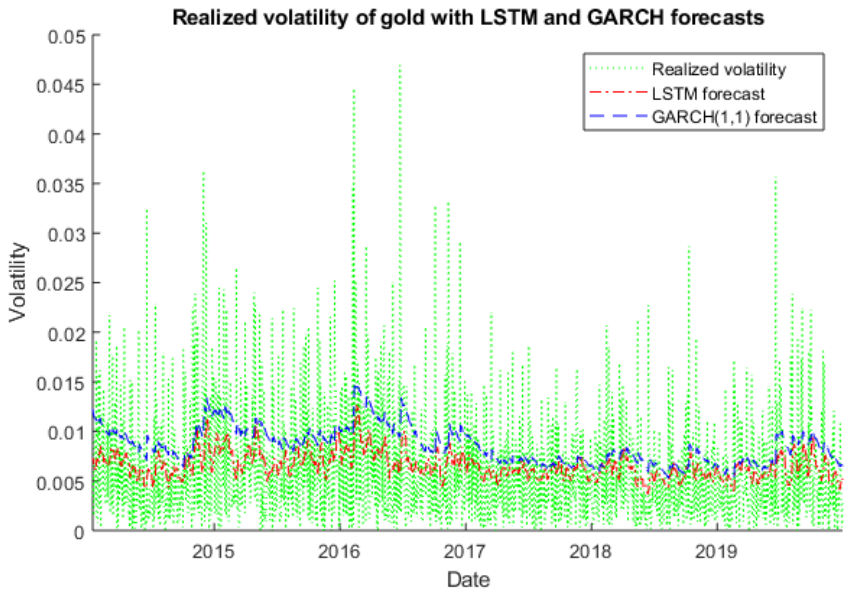


Figure 20: Realized volatility (green dotted line) of gold alongside the LSTM- (red dash dotted line) and GARCH(1,1) (blue dashed line) forecasts.

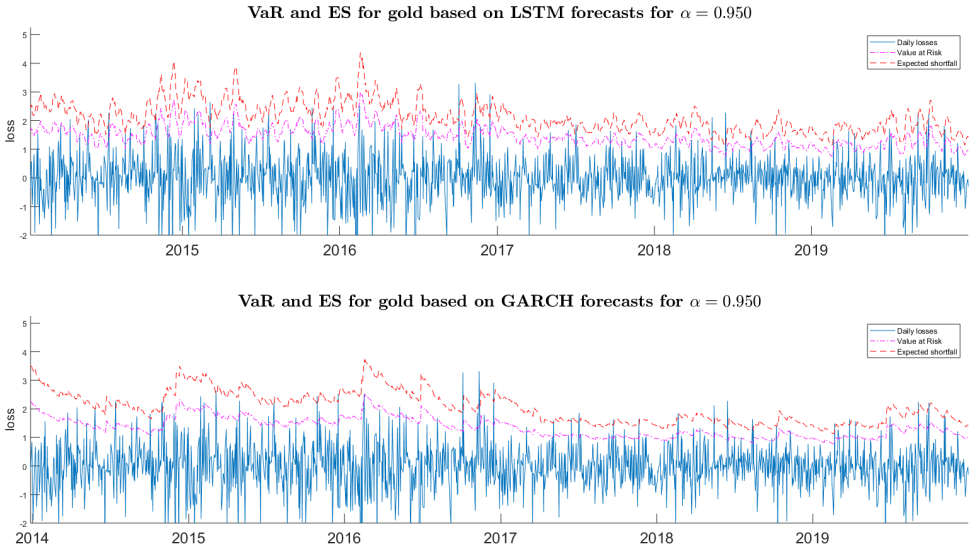


Figure 21: VaR and ES estimations for $\alpha = 0.950$ based on LSTM- (top) and GARCH(1,1) (bottom) volatility forecasts for gold. The VaR and ES estimates are based on a VWHS with a rolling window of 1000 losses.

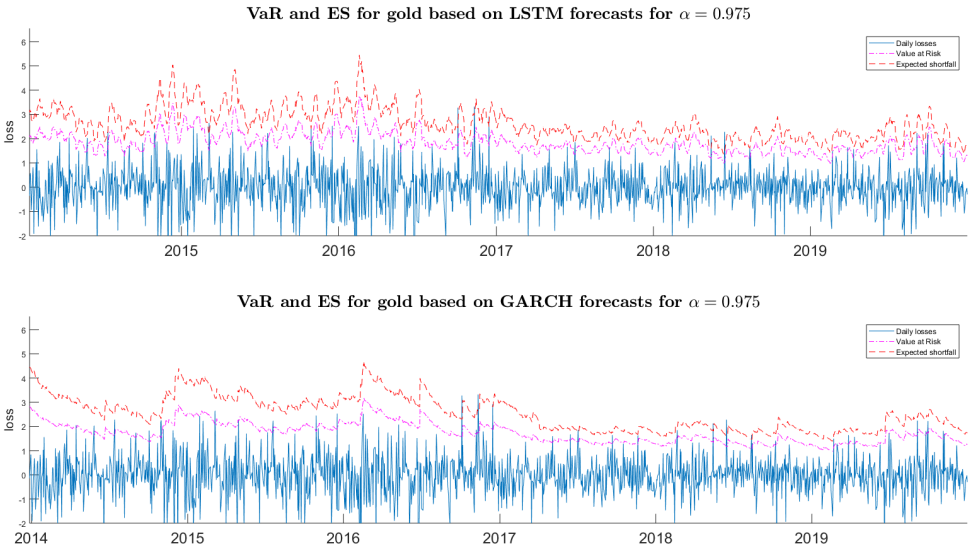


Figure 22: VaR and ES estimations for $\alpha = 0.975$ based on LSTM- (top) and GARCH(1,1) (bottom) volatility forecasts for gold. The VaR and ES estimates are based on a VWHS with a rolling window of 1000 losses.

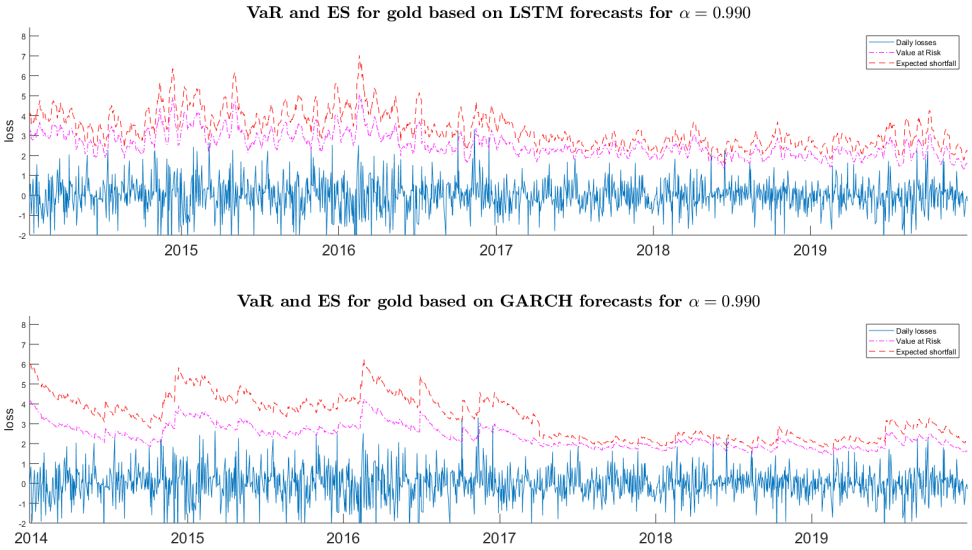


Figure 23: VaR and ES estimations for $\alpha = 0.990$ based on LSTM- (top) and GARCH(1,1) (bottom) volatility forecasts for gold. The VaR and ES estimates are based on a VWHS with a rolling window of 1000 losses.

C.3 Soybean

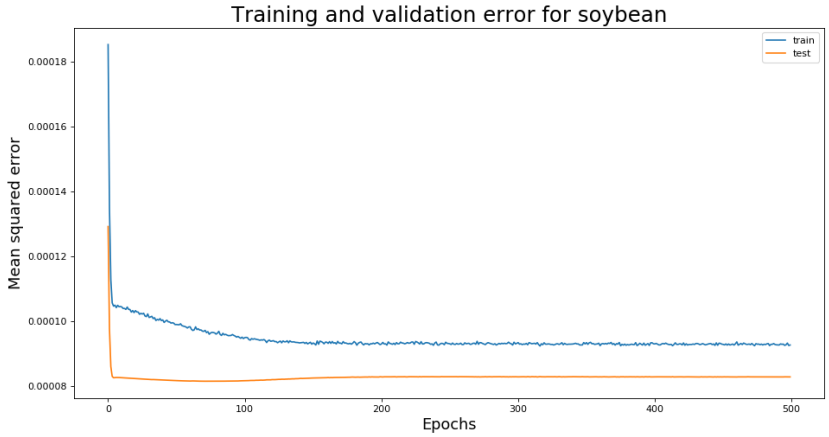


Figure 24: Training error and validation error during the optimization process for soybean.

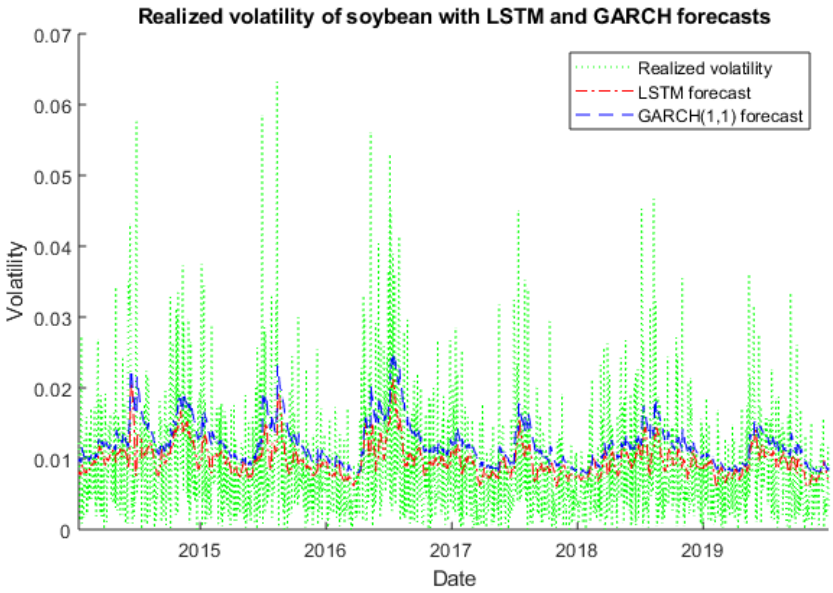


Figure 25: Realized volatility (green dotted line) of soybean alongside the LSTM (red dash dotted line) and GARCH(1,1) (blue dashed line) forecasts.

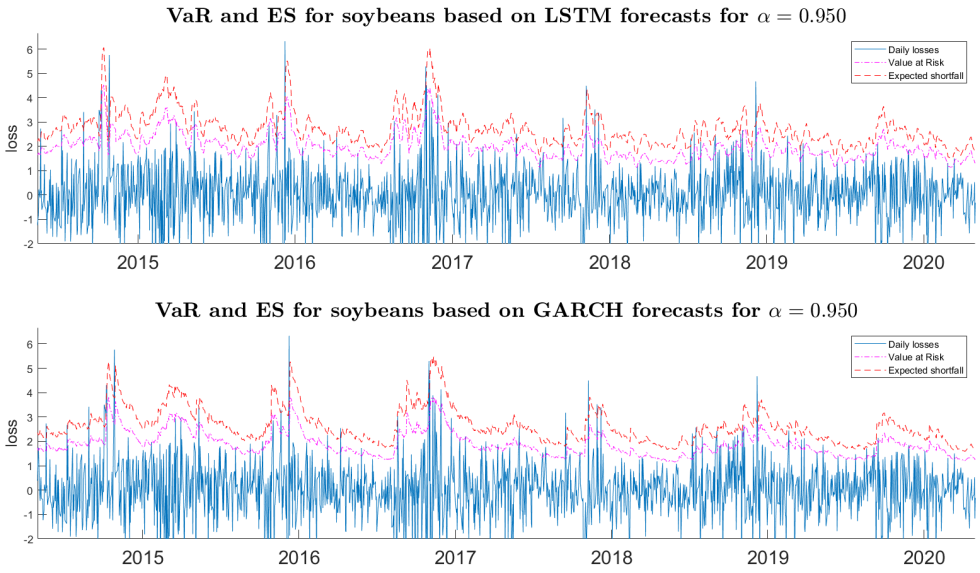


Figure 26: VaR and ES estimations for $\alpha = 0.950$ based on LSTM (top) and GARCH(1,1) (bottom) volatility forecasts for gold. The VaR and ES estimates are based on a VWHS with a rolling window of 1000 losses.

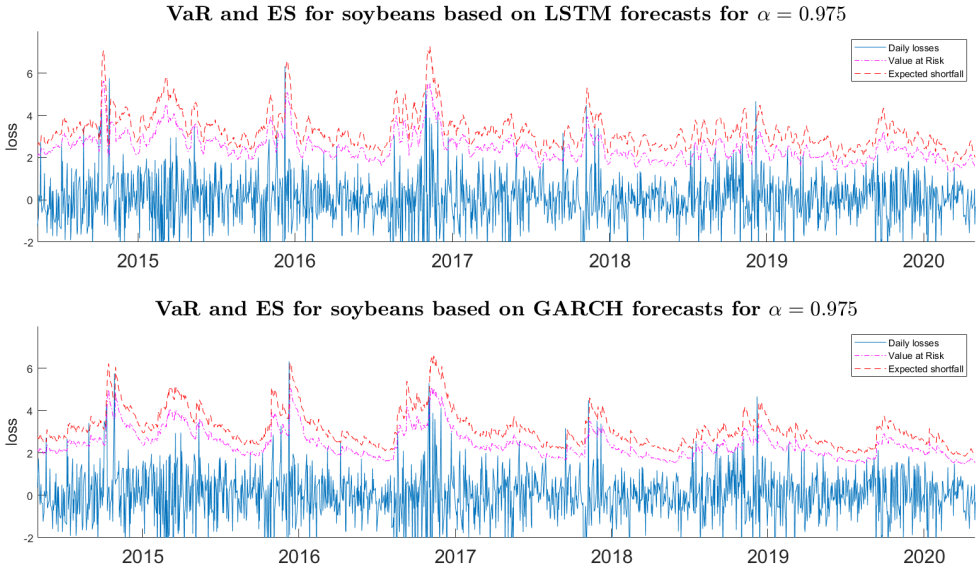


Figure 27: VaR and ES estimations for $\alpha = 0.975$ based on LSTM (top) and GARCH(1,1) (bottom) volatility forecasts for gold. The VaR and ES estimates are based on a VWHS with a rolling window of 1000 losses.

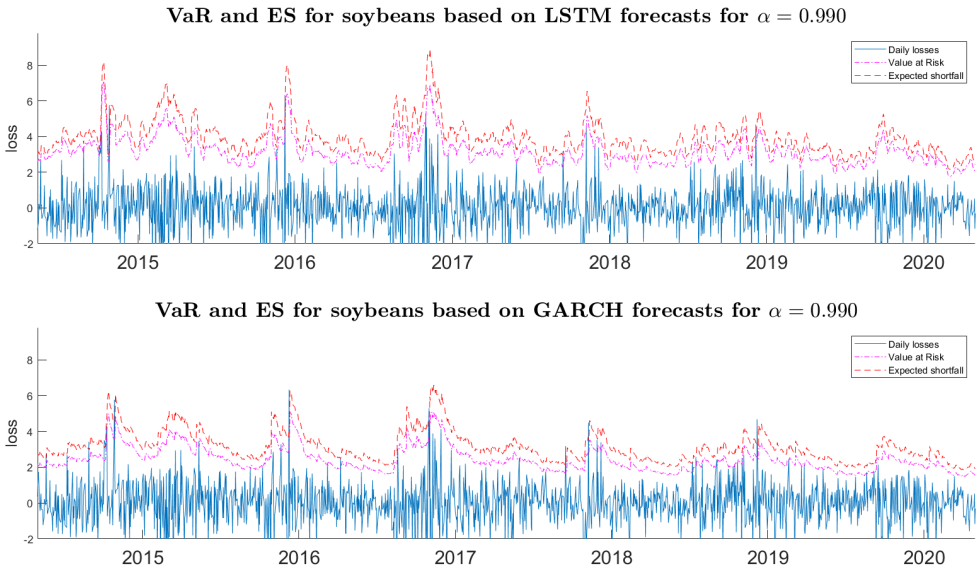


Figure 28: VaR and ES estimations for $\alpha = 0.990$ based on LSTM (top) and GARCH(1,1) (bottom) volatility forecasts for gold. The VaR and ES estimates are based on a VWHS with a rolling window of 1000 losses.

D Tables

Table 10: Mean squared error (MSE) and mean absolute errors (MAE) from the realized variance of each commodity for the two different approaches on the test data set.

	Oil		**Gold**		**Soybean**	
Model	Test MSE	Test MAE	Test MSE	Test MAE	Test MSE	Test MAE
LSTM	2.085E-04	0.0105	3.335E-05	0.0044	6.254E-05	0.0059
GARCH(1,1)	2.360E-04	0.0120	3.947E-05	0.0052	7.232E-05	0.0068

E Code and scripts

All code utilized can be provided upon request. Everything related to neural networks were conducted in Python, with graphing and the tests conducted in Matlab.