Erasmus University Rotterdam

Erasmus School of Economics

Master Thesis Quantitative Finance

Pairs Trading Using Machine Learning: An Empirical Study

Author:
R.W.J. van der Have
Student ID:
384898

Supervisor Erasmus University:
Prof. Dr. D. J. C. van Dijk
Second assessor Erasmus University:
Dr. H.J.W.G. Kole

Supervisors Deloitte:
Dr. Ir. H. E. Eisma
Dr. G. Diepen

December 18, 2017

Abstract

Pairs trading is a quantitative trading strategy that exploits financial markets that are out of equilibrium. By identifying a pair of stocks that historically move together, and assuming that their price difference is mean-reverting, an investor can profit from deviations from the mean by taking a long-short position in the chosen pair. Throughout the years, several trading frameworks and methods have been established in order to optimize this strategy. These methods, in particular the stochastic (residual) spread method, are mainly based on the more traditional estimation techniques, such as the Expectation Maximization algorithm. Since machine learning techniques are becoming more popular in finance, we propose to develop a framework for pairs trading using neural networks. This thesis analyzes the performance of neural networks in pairs trading applied to Exchange Traded Funds (ETFs) both statistically and economically, and compares the performance with the more traditional methods. The results show that recurrent neural network is superior compared to the other methods, since it generates the largest returns, around 11%, as well as the highest Sharpe and Sortino ratios.

Keywords : Pairs Trading, Distance Method, Cointegration Method, Time Series, Stochastic Spread, Machine Learning, Feedforward Neural Network, Recurrent Neural Network

This page would be intentionally left blank if we would not wish to inform about that.

Contents

1 Introduction

Pairs trading is a quantitative trading strategy that exploits financial markets that are out of equilibrium. The strategy is widely used by hedge funds and investment banks. The idea behind pairs trading is as follows. First, a pair of assets is selected, that are known to historically move together and have some sort of long-run relationship. Using the assumption that the spread, defined as the difference in price between the paired assets, is mean-reverting, deviations from the mean can be exploited. In case of a deviation from the mean of the spread, an investor should take a short position in the overvalued asset and a long position in the undervalued asset. As soon as the spread converges back to its mean, the investor should unwind both positions, resulting in a profit. Even though this may sound as an intuitive approach, sophisticated econometric techniques can be used in all the steps involved in pairs trading.

To exploit the opportunities that pairs trading offers, two main aspects need to be optimized. These aspects are identifying the best pairs as well as implementing and executing a trading strategy, which may involve capturing the characteristics of the spread or forecasting the spread. Throughout the years, several methods have been used to optimize these steps. Recently, more research has become available on applying machine learning techniques in finance, in particular time series prediction. However, there is little literature available on machine learning applied to pairs trading. Dunis et al. (2006, 2015) are the main authors in this area, but their applications are limited to specific spreads, such as the gasoline crack spread. Therefore, one of our two main contributions will be applying machine learning methods to an entire asset class and comparing the performance of these methods to the more traditional methods.

Our second contribution to the literature will be applying these techniques to Exchange Traded Funds (ETFs). ETFs are products that track indices and try to follow the underlying assets as close as possible. With the increasing popularity and easier accessibility of ETFs, it is interesting to investigate the performance of pairs trading using these instruments. The fact that there is only little published literature available on this topic, while literature on pairs trading with stocks is available to a larger extent, results in more incentive to apply the pairs trading techniques to ETFs rather than stocks.

These two contributions can be summarized in the following research question:
To what extent does pairs trading in the ETF market benefit from applying machine learning methods compared to the more traditional methods?

This research is relevant for both scientific and practical purposes. As discussed above, this research has a contribution to the literature that is twofold, which is developing machine learning techniques for pairs trading and comparing these with the more traditional

methods, as well as applying these techniques to ETFs. From a practical point of view, this research can be interesting for hedge funds, investment banks and other trading firms that use and have interest in pairs trading. This is due to the fact that better, more efficient and well developed pairs trading methods can lead to more profit for the particular firm.

Pairs trading is a strategy developed by a team of quants of the Morgan Stanley group somewhere in the mid-1980s. Until the early 2000s, it was mainly used in practice, resulting in little literature being available. One of the first main articles introducing pairs trading was conducted by Gatev et al. (1999)[1]. They introduce the distance method, in which they match stocks into pairs by means of the minimum distance between normalized historical prices. They show that a simple trading rule, which is executing a trade when the spread deviates two standard deviations from its mean, yield relatively large annualized excess returns of up to 11% that typically exceed conservative transaction-cost estimates. As stated by Krauss (2015), the distance method is one of the most investigated pairs trading frameworks, since it is relatively easy to implement the framework in practice due to its simplicity, transparency and non-parametric character. These advantages and its use in practice, make the distance method a good reference point in this research to see how the machine learning techniques perform.

One of the first parameterized methods, that is also used in practice, is the cointegration method as thoroughly explained by Vidyamurthy (2004). Vidyamurthy (2004) selects the paired assets based on the cointegration relationship between two financial instruments. The idea behind this strategy is that two cointegrated assets will follow the same long term trend and will return back to their mean in case of deviations. Using cointegration has the advantage that the choice of a certain pair can be explained statistically while also confirming the desired mean-reversion of the pair. The main methods used for cointegration testing are the Engle & Granger (1987) or the Johansen (1991) test. Even though Vidyamurthy (2004) does not provide empirical results of the cointegration method, it is a framework that can form as a base for subsequent research. Caldeira & Moura (2013) use cointegration to select pairs on a Brazilian stock index. Using the trading rule proposed by Gatev et al. (1999), they find excess returns of more than 16% per year for the identified pairs, which shows that the cointegration approach can result in large profits. The cointegration method will be used to identify the pairs to use in the neural networks and the traditional methods.

The more traditional methods involve modeling the mean-reverting characteristics of

[1]Gatev et al. conducted their research in two parts, which are Gatev et al. (1999) and Gatev et al. (2006). Gatev et al. (1999)introduced their trading strategy, while Gatev et al. (2006) extended their research with more recent data to show that the strategy remained profitable. Note that Gatev et al. (2006) was eventually published. For referencing purposes, we refer to Gatev et al. (1999).

the spread, which are used to generate the optimal trading signals. Elliott et al. (2005) assume that the spread is driven by a latent state variable following an Ornstein-Uhlenbeck process. This assumption makes it possible to explicitly model the mean-reversion behavior of the spread using a state space model. Therefore, a main advantage is that their framework revolves around the mean-reversion of the spread, which is fundamental for pairs trading. Also, the parameters of their linear and Gaussian state space model are estimated using the Expectation Maximization algorithm and the Kalman filter, which makes the model completely tractable. Elliott et al. (2005) capture the mean-revering properties of the spread and use these to determine the appropriate investment decisions. This approach is known as the stochastic spread method.

In literature, Avellaneda & Lee (2010), and D'Aspremont (2011) both apply a variant of the stochastic spread method to their spread time series. Avellaneda & Lee (2010) focus more on explaining the variance of the spread and do not apply the trading rule, while D'Aspremont (2011) trades out-of-sample with several pairs and finds an average Sharpe Ratio of 6% before transaction costs.

Do et al. (2006) criticize the methodology proposed by Elliott et al. (2005) by stating that "the model restricts the long run relationship, since the two assets chosen must provide the same return such that any departure from it will be expected to be corrected in the future" (p. 9). In practice, this is a difficult restriction since it is rather odd to find two assets with identical returns. Therefore, Do et al. (2006) suggest the so-called stochastic residual spread method, which uses the mispricing on return level, instead of price level. Unfortunately, the stochastic residual spread method has not been applied to empirical data. The stochastic spread method and stochastic residual spread method are used as the more traditional methods, since they assume linearity in the model, Kalman filter and EM algorithm, for the spread.

As explained earlier, one of our main contributions is applying machine learning techniques to pairs trading and comparing the results to the above mentioned methods. Machine learning techniques are fairly unexplored in the field of pairs trading, since most of the relevant articles have limited applications to only a few selected securities, but they provide a promising direction of further research. (Krauss, 2015). The methods relevant to this research are the feedforward neural network (FNN) and the recurrent neural network (RNN). Both methods are widely used and have shown to be reliable when it comes to forecasting stationary time series (Brezak et al., 2012), while also being the methods applied to pairs trading by the main authors in this area, Dunis et al. (2006, 2015). A neural network can be described as an incremental mining technique, since it allows incorporating new data submissions in the already trained neural network, without having to process the old information again. As Franses & van Dijk (2000) describe, neural

networks have shown to be a very good approximation to almost all nonlinear functions. Given the fact that time series data is nonlinear in general, using neural networks might be a better choice to predict and model the stock prices than a regular linear framework. Neural networks are able to detect nonlinearities in data, without being given a prior specification of any nonlinear relationship, which differentiates them from the traditional methods that assume a linear relationship.

On the described machine learning techniques in combination with pairs trading, not many articles are available. However, Brezak et al. (2012) provide arguments for the use of FNN and RNN and evaluate the performance of both methods for stock price prediction. Olden (2016) states that neural networks require a stationary time series in order to be able to predict the stock price. Since the time series of the spread is supposed to be stationary, these methods can be applied to pairs trading as well. The few articles on machine learning in combination with pairs trading that are available provide the following results. Dunis et al. (2006) apply artificial neural networks to model the gasoline crack spread. By forecasting the spread using neural networks and executing a fairly easy trading strategy, they show that neural networks can be profitable. In particular, using a recurrent neural network results in profits of 15% on average before transaction costs. However, their framework results in a very active trading strategy and therefore, the impact of transaction costs is large. Dunis et al. (2015) apply neural networks in an adjusted framework to the corn/ethanol crush spread. In this case, they find more satisfying results because of the adjustments made in the trading strategy, which results in less trades and therefore a profit after transaction costs of about 20%. These positive results when applying neural networks to specific spreads motivate applying the methods to ETFs even more.

The data that will be used in this research contains the time series of a universe of ETFs. After the data is obtained, we will apply the distance method, cointegration method, stochastic (residual) approach, feedforward neural network and recurrent neural network. First to select the ETF pairs, then to execute the trading strategy. The results will be economically compared by constructing a portfolio for each method, calculating the Sharpe (1994) and Sortino & Price (1994) ratio, and maximum drawdown (Pereira Camara Leal & Vaz de Melo Mendes, 2005), and accessing the statistical significance of these ratios against one another.

Using the pairs selected by the cointegration method, the main results show that the recurrent neural network is the best performing method, since it generates the largest returns (around 11%) with average monthly returns that are statistically significant on a 1% significance level when using a larger number of pairs. Also, it generates (one of the) highest Sharpe and Sortino ratios. In most cases, the Sharpe ratio of the recurrent

neural network are significantly larger than the ratios of the other methods. The main drawback of the recurrent neural network is that it has the largest maximum drawdown of all methods. However, it can be argued that for some investors, depending on their utility function and attitude towards risk, the drawdown is offset by the size of the return. Finally, subperiod analysis shows that the recurrent neural network is the only method that behaves as expected in pairs trading, since it generates most of its return in the most volatile period. This leads to the conclusion that the use of a recurrent neural network is beneficial to pairs trading.

The remainder of the thesis is organized as follows. Section 2 explains the methodology used in this research, while the data is described in section 3. Section 4 discusses the results, and section 5 both concludes and elaborates on possible further research.

2 Methodology

In this section, the methodology of the research is discussed. Section 2.1 introduces one of the first pairs trading frameworks in the literature, known as the distance method, which will serve as a benchmark model. Section 2.2 discusses the pair selection method for the more sophisticated methods, which is the cointegration method. Section 2.3 elaborates on the methods we use to either capture the mean-reversion in the spread or forecast the changes in the spread, as well as on the corresponding trading rules for each method. Section 2.4 discusses the economic comparison of the techniques.

2.1 Distance Method

The distance method has been introduced by Gatev et al. (1999) and is still one of the main methods used for pairs trading in practice. This method introduced pairs trading to the academic literature and established pairs trading as a true capital market anomaly (Krauss, 2015). Gatev et al. (1999) select pairs by minimizing the sum of squared deviations between two stock price series and trade when the spread between the chosen stocks diverges more than two historical standard deviations from its mean. The main advantage of the distance method is that it is model-free, which makes the strategy easy to implement. Being model-free also means that the strategy is free from misspecification and misestimation. Due to these advantages, the distance method is still most used pairs trading strategy in practice.

Pair selection: Gatev et al. (1999) found through interviews that this approach approximates the way how traders choose their pairs. First, they split up the sample and define a formation period and a trading period. In the formation period, the pairs are formed and the historical mean and standard deviation of the spreads are calculated. During the trading period, the trading strategy is executed for the different selected pairs to generate returns that are further analyzed. Gatev et al. (1999) normalize the prices of all the stocks to the first day of the formation period. A matching partner for each stock is found by minimizing the sum of squared distances between the two normalized price series, denoted by P_i and P_j for respectively stock i and j. This distance, which is the price difference between the series i and j, is known to be the spread, $S_{i,j}$, between the two assets, which is at time t defined as

$$S_{t,i,j} = P_{t,i} - P_{t,j}. \tag{1}$$

Gatev et al. (1999) use the Euclidean squared distance, leading to the average sum of

squared distances, $\overline{SSD_{t,i,j}}$, being given as

$$\overline{SSD_{t,i,j}} = \frac{1}{T_f}\sum_{t=1}^{T_f} S_{t,i,j}^2 = \mathrm{V}(S_{i,j}) + \left(\frac{1}{T_f}\sum_{t=1}^{T_f} S_{t,i,j}\right)^2, \tag{2}$$

where T_f is defined as the end of the formation period and we use that the sample variance of the spread, V(.), is equal to

$$\mathrm{V}(S_{i,j}) = \frac{1}{T_f}\sum_{t=1}^{T_f} S_{t,i,j}^2 - \left(\frac{1}{T_f}\sum_{t=1}^{T_f} S_{t,i,j}\right)^2. \tag{3}$$

This way, the method identifies a matching partner, a certain stock j, for every stock i.

Trading strategy: Gatev et al. (1999) also base their trading rule on practice. During the trading period, positions are opened and closed when a price diverges more than two historical standard deviations from the historical mean of the spread, which is mathematically:

$$S_{t,i,j} \geq \mu + 2 \cdot \sigma \quad \vee \quad S_{t,i,j} \leq \mu - 2 \cdot \sigma, \tag{4}$$

where $T_f + 1 \leq t \leq T$ with T being the end of the trading period, and the historical mean (μ) and historical standard deviation (σ) are calculated over the formation period, where the mean is defined as

$$\mu = \frac{1}{T_f}\sum_{t=1}^{T_f} S_{t,i,j} \tag{5}$$

and the historical standard deviation is equal to the square root of (3). As soon as the spread converges back to its mean, the positions unwind and profit is made. In case the spread has not converged at the end of the trading period, profits and losses are calculated on the last day of the interval.

The distance method has a few main drawbacks, as stated by Krauss (2015). First of all, Gatev et al. (1999) identify a paired stock for every stock. This could be a possible disadvantage of the method, since it could be that one of stocks does not have proper paired stock in the asset universe, but is still paired with one, since the distance method is set up for each stock to have a partner. Since the particular stock and its partner are not a 'true' pair, it is likely that using this pair to trade with will result in a loss. Secondly, minimizing (2) results in the variance of spread being minimized, which results in less deviations from the mean and therefore less potential profits. Finally, the selection of the pairs cannot be statistically motivated. It is not tested whether there exists a mean-reverting, long-run relationship between the chosen pairs. This leads to higher divergence risk, which will result in more unprofitable trades with pairs of which the spread does not

converge back to the mean. Krauss (2015) used the methodology of Gatev et al. (1999) on an extended dataset. They find that 32% of the pairs selected using the distance method diverge. These drawbacks indicate that selecting pairs using the distance method is a rather suboptimal selection metric, motivating the choice for pair selection method based on a statistical relationship. Therefore, we will propose such a pair selection method and subsequently more sophisticated trading strategies. The distance method will serve as benchmark for the proposed strategies, since it still is the most used trading strategy in practice.

2.2 Pair Selection: Cointegration Method

A pair selection method based on a statistical relationship between two stocks, is the cointegration method. Vidyamurthy (2004) thoroughly explained the cointegration method and created the foundation for a pair selection method that is also used in practice. Cointegrated instruments are expected to follow the same long-run trend and show mean reverting behaviour, which is an important statistical characteristic for pairs.

The concept of cointegration was introduced by Engle & Granger (1987). Define p_i to be the natural logarithm of the price series P_i of asset i. Then, cointegrated assets i and j have the following characteristics. Their price series p_i and p_j are are integrated of order 1, $I(1)$, which means they are non-stationary,. However, there exists a linear combination of both assets resulting in a stationary, i.e. $I(0)$, time series $S_{t,i,j}$ that can be written as:

$$S_{t,i,j} = p_{t,i} - \beta p_{t,j} - \alpha, \tag{6}$$

where α and β are parameters that are estimated. Note that the time series $S_{i,j}$ is the same as the residual of regressing p_i on p_j. For pairs trading, it is considered to be the spread between the two assets.

To test two assets for cointegration, Engle & Granger (1987) provide a procedure consisting of two steps. Given that p_i and p_j are non-stationary, the parameters of the cointegration relation as defined in (6) need to be estimated using Ordinary Least Squares (OLS). Thereafter, the computed spread $S_{i,j}$ needs to be tested for stationarity. This can be done using the well-known Augmented-Dickey-Fuller test (Dickey & Fuller, 1979). Once it has been shown that the spread is stationary, it can be concluded that assets i and j are cointegrated and can therefore be considered to be a pair.

Unfortunately, Vidyamurthy (2004) does not provide any empirical results of the cointegration method. However, subsequent research does, using the trading rule proposed by Gatev et al. (1999). For example, Dunis et al. (2010) apply the cointegration approach to high frequency data. They find that the cointegration method is able to give

a good indication of the profitability of the pair in a high frequency setting. Caldeira & Moura (2013) use cointegration to select pairs on a Brazilian stock index. They find excess returns of more than 16% per year for the identified pairs, which shows that the cointegration approach can result in large profits. Furthermore, Huck & Afawubo (2015) show that the cointegration approach can generate high, stable and robust returns of up to 5% per month, while the distance method is unable to provide significant returns over the same time period. This indicates that the cointegration method is a better pair selection method than the distance method. Additionally, this shows that employing the cointegration method for pairs selection can serve as a foundation in order to optimize the trading strategy. Therefore, we will use the cointegration method to select the pairs that will be further evaluated using extensive time series analysis or machine learning techniques.

2.3 Trading Strategies

After selecting the pairs using the cointegration method, we are able to use more extensive trading strategies. The trading framework proposed by Gatev et al. (1999) is considered to be a model-free and ad-hoc trading strategy, where a trade is executed when the spread deviates more than two historical standard deviations from its mean. It is clear to see that this does not optimize the profit, as the optimal point in time to execute a trade might be at a later time. Therefore, we propose to either capture the mean-reversion in the spread using extensive time series analysis following the approach of Elliott et al. (2005) or forecasting the spread using the machine learning framework from Dunis et al. (2006). For both frameworks, the corresponding trading rules will be used, which will hopefully result in a better timing of the trades and therefore more profit. Section 2.3.1 explains two more advanced time series approaches to model the mean-reversion of the spread, while section 2.3.2 proposes machine learning techniques for forecasting and trading.

2.3.1 Capturing the Mean-Reverting Properties

One possible way to optimize the trading strategy, is using extensive time series analysis on the spread and capturing the mean-reverting properties of the time series. Elliott et al. (2005) and Do et al. (2006) both propose to use a mean-reverting process to model the spread. They show that this process results in a state space model that can be estimated using the Expectation Maximization (EM) algorithm and the Kalman filter. This method has several advantages over framework discussed in section 2.1. For instance, the mean-reverting process ensures that the mean-reversion needed for pairs trading is captured. This way, we can try to more optimize the timing of the trade by using the mean-reverting

properties in the trading rule. Furthermore, estimating the parameters of the state space model using the EM algorithm and Kalman filter makes the model completely tractable. These advantages should result in a framework that yield more profit than the distance method as discussed in 2.1.

2.3.1.1 Stochastic Spread Method

Elliott et al. (2005) assume that the spread is driven by a latent state variable ξ following an Ornstein-Uhlenbeck process, which is defined by

$$d\xi_t = \kappa(\theta - \xi_t)dt + \sigma dB_t, \tag{7}$$

where θ is the mean of the state variable, κ the speed of mean-reversion and B_t is a standard Brownian Motion. Elliott et al. (2005) set the spread between stock i and j, $S_{t,i,j}$, equal to the state variable plus a Gaussian noise, as given by

$$S_{t,i,j} = \xi_t + H\omega_t, \tag{8}$$

where $S_{i,j}$ is observed and defined as in (6), and $\omega_t \sim \mathcal{N}(0,1)$. This way, the spread is mainly driven by the mean-reverting process and slightly by the effect of a Gaussian noise term. Due to this noise term, a state space model is needed. As Elliott et al. (2005) do not give an explicit explanation for the use of the state space model, we explain the noise term as follows. The ETF data used consists of daily closing prices. These ETFs consist of multiple assets that might be listed on different exchanges. It does not have to be the case that these exchanges are in the same timezone, which means the closing prices are not by definition the closing prices as the closing of the market is at a different time. Also, in case the exchanges are in the same time zone it could be the case that the closing price report on one exchange is the price at 16:55, while the other exchange report the 17:05 price as closing price. This could result in noise in the data, and therefore in the mean-reverting process, which motivates the use of the state space model.

Making use of the fact that the solution to (7) is Markovian, the equation can be rewritten to a discrete time transition equation, which is a simple AR(1) process, resulting in the following:

$$\xi_{t+1} = A + B\xi_t + C\epsilon_{t+1}, \tag{9}$$

where $A = \frac{\theta}{\kappa}\tau$, $B = 1 - \kappa\tau$ and $C = \sigma\sqrt{\tau}$, τ represents the time interval between two observations and ϵ is a random process with a standard normal distribution. It holds that $A \geq 0$ and $0 < B < 1$, and thus it should hold that $\theta \geq 0$ and $0 < \kappa < \frac{1}{\tau}$. Rewriting the

definitions for A, B and C gives the following definitions for κ, θ, σ:

$$\kappa = \frac{1-B}{\tau}, \tag{10}$$

$$\theta = \frac{A}{\tau}\kappa, \tag{11}$$

$$\sigma = \sqrt{\frac{C^2}{\tau}}. \tag{12}$$

The measurement equation can be written as

$$S_{t,i,j} = \xi_t + H\omega_t, \tag{13}$$

Following the approach of Elliott et al. (2005), the parameters (A, B, C, H) in (9) and (13) are iteratively estimated using the Expectation Maximization (EM) algorithm, the Kalman filter and Kalman smoother. A detailed description of the solution for the EM algorithm, the update equations in the Kalman filter and the smoothing equations can be found in appendix A. The EM algorithm will provide estimates for the parameters (A, B, C, H) based on the observations where $1 \leq t \leq T_f$, which is the same formation period as for the distance method. These estimates can be used to calculate parameters in the mean-reverting process in (7), κ, θ and σ, as defined in respectively (10), (11) and (12). Elliott et al. (2005) provide a trading rule that makes use of these parameters. They state to execute a pairs trade when

$$S_{t,i,j} \geq \theta + c_{i,j}(\frac{\sigma}{\sqrt{2\kappa}}) \quad \vee \quad S_{t,i,j} \leq \theta - c_{i,j}(\frac{\sigma}{\sqrt{2\kappa}}) \tag{14}$$

for $T_f + 1 \leq t \leq T$. The variable $c_{i,j}$ will be arbitrarily optimized over the in-sample period for every pair i and j individually, that is, the $c_{i,j}$ that results in the highest profit in-sample will be used out-of-sample. The positions are unwinded when

$$\theta - \epsilon \leq S_{t,i,j} \leq \theta + \epsilon, \tag{15}$$

where ϵ is equal to 0.01% of the value of the spread. This trading rule is comparable to the rule proposed by Gatev et al. (1999) in section 2.1. However, the main difference is the Gaussian noise term in (13). The differences in results with the distance method will mostly depend on the importance of this noise term (represented by parameters H), that is, its size compared to the variance of the mean-reverting process ξ_t.

The approach of Elliott et al. (2005) has rarely been applied to empirical data in the academic literature. Avellaneda & Lee (2010), and D'Aspremont (2011) both apply a variant of the stochastic spread method to their spread time series. Avellaneda &

Lee (2010) focus more on explaining the variance of the spread and do not apply the trading rule, while D'Aspremont (2011) trades out-of-sample with several pairs and finds an average Sharpe Ratio of 6% before transaction costs.

2.3.1.2 Stochastic Residual Spread Method

However, Do et al. (2006) criticize the methodology proposed by Elliott et al. (2005). Do et al. (2006) state that the model restricts the long run relationship, since the two assets chosen must provide the same return such that any departure from it will be expected to be corrected in the future. To overcome this issue, Do et al. (2006) suggest the so-called stochastic residual spread method, which uses the mispricing on return level instead of price level, which means that the spread will be defined as the difference in returns instead of difference in prices. Also, this gives the possibility to adjust the spread for differences in exposure to risk factors.

They propose to use the same mean-reverting Ohlenstein-Uhlenbeck process as Elliott et al. (2005), as defined in (8). This results in the same transition equation as given in (9). Next, Do et al. (2006) use the Asset Pricing Theory of Ross (1976) to create a completely tractable model of mean-reverting relative pricing for two asset based on their returns. They use that a relative Asset Pricing Theory on two assets i and j at time t can be written as

$$R_{t,i} = R_{t,j} + \mathbf{\Gamma}' \boldsymbol{r}_{\boldsymbol{t}}^{\boldsymbol{f}} + e_t, \tag{16}$$

where the return R_t^i for asset i at time t is defined as

$$R_{t,i} = \log\left(\frac{P_{t,i}}{P_{t-1,i}}\right), \tag{17}$$

and $\mathbf{\Gamma}$ is a vector of exposure differentials to risk factors $\boldsymbol{r}_{\boldsymbol{t}}^{\boldsymbol{f}}$. This way, the mean-reverting process is adjusted for the differences in exposure to risk factors. Do et al. (2006) give the example of using the Fama-French three-factor model (Fama & French, 1993) for the risk factors, which are the excess return on the market portfolio ($R_{t,m}$), and excess returns of the Small (market capitalization) Minus Big (SMB) and the High (book-to-market ratio) Minus Low (HML) portfolios. To extend this example, we propose to use the Carhart four-factor model (Carhart, 1997). The Carhart four-factor model is essentially the Fama-French three-factor model with a momentum factor added. This momentum factor captures the tendency that assets that are rising, will continue to rise and vice

versa. These four factors give the following definition for $\boldsymbol{r_t^f}$:

$$\boldsymbol{r_t^f} = \begin{bmatrix} R_{t,m} \\ SMB_t \\ HML_t \\ MOM_t \end{bmatrix}. \tag{18}$$

Using (16), Do et al. (2006) change the measurement equation in (13) to

$$S_{t,i,j} = x_t + \boldsymbol{\Gamma r_t^f} + H\omega_t, \tag{19}$$

where the spread $S_{t,i,j}$ is now defined as

$$S_{t,i,j} = R_{t,i} - \beta R_{t,j}, \tag{20}$$

where β still defined as in (6). Important to note is that when the price series are I(1), the returns series are I(0), i.e. stationary. This means that any linear combination of return series remains stationary. Therefore, we are able to define the spread as in (20), which makes the comparison with the other methods, since the characteristics of the cointegration relation between the prices in (6) are still used.

Once again, the parameters in (9) and (19) can be estimated using the Expectation Maximization algorithm and the Kalman filter. However, due to the extra, exogenous term in the measurement equation some of the dynamics change. Appendix B elaborates on the changed dynamics, the new solutions for the EM algorithm and the changed equations of the Kalman filter and smoother.

Furthermore, we use the same strategy as for the stochastic spread method, as described in section 2.3.1.1. Note that the main difference is that the spread is now defined as the difference in returns, see (17). Similar to the stochastic spread method, the approach of Do et al. (2006) has not been used on empirical data in the academic literature as well. Krauss (2015) describes the return restriction Do et al. (2006) propose as difficult, since it is rather odd to find two assets with identical returns. However, the idea is to have a spread that is mean-reverting, which is the case for two returns series and therefore, this method can be valuable for pairs trading.

As described by Cummins & Bucca (2012), one of the major limitations of the stochastic (residual) spread method is the fact that the Gaussian properties of the Ornstein-Uhlenbeck process is not in line with the stylized facts of financial data, such as the assumption of linearity or constant volatility the methods use. However, it can be argued that the analytic simplicty of the Ornstein-Uhlenbeck process offsets this limitation.

Still, it remains an open (empirical) question whether the analytic simplicity of the process is sufficient to offset its limitations and consequently being a valuable pairs trading framework that improves the non-parametric frameworks, such as the distance method.

2.3.2 Forecasting Using Neural Networks

The methods introduced in section 2.3.1 rely on the more traditional methods, such as the Kalman filter and EM algorithm, that assume a linear relationship in the spread. Throughout the years, machine learning techniques have been introduced in finance, and in particular in trading. The more used machine learning techniques are neural networks. As Franses & van Dijk (2000) describe, neural networks have shown to be a very well approximation to almost all nonlinear functions. Given the fact that time series data is nonlinear in general, using neural networks might be a better choice to predict and model the stock prices than a regular linear framework. Neural networks are able to detect nonlinearities in data, without given a prior specification of any nonlinear relationship. Also, a neural network is able to incorporate new data in the network without having to process the old information again (Lam, 2004). Since the more traditional methods do have to process the old information again, this is a major advantage.

In academic literature, only Dunis et al. (2006, 2015) apply neural networks directly to pairs trading. Their motivation revolves around the strong ability of neural networks to detect nonlinearities in time series data. They develop a trading framework using neural networks and apply it to specific spreads, such as a gasoline crack spread and a corn/ethanol crush spread. Dunis et al. (2006) define the spread as in equation (20). They state that the advantage of defining the spread in this manner, that it is possible to present returns with more conventional percentage return/risk profiles. Their framework consists of forecasting the change in the spread using several feedforward neural networks and recurrent neural networks. The forecasts are compared to a certain threshold and in case the forecast exceeds the threshold, a trade is executed. Dunis et al. (2006) show that their frameworks result in returns ranging from 19% at most for the feedforward neural networks to 41% for the recurrent neural networks, while Dunis et al. (2015) reports returns of around 42% for both neural networks. Due to the profitability of these methods in specific applications, it is interesting to see how the performance of the neural networks holds when applying these to the ETF market. Therefore, we will use the feedforward and recurrent neural networks in this research, based on the approach of Dunis et al. (2006, 2015). Respectively section 2.3.2.1 and 2.3.2.2 discuss these methods in more detail.

2.3.2.1 Feedforward Neural Network

Following the approach of Dunis et al. (2006), the first neural network to be discussed is the feedforward neural network (FNN). It is considered to be the most used neural network for time series forecasting (Brezak et al., 2012). Medeiros & Teräsvirta (2006) wrote one of the main articles on modeling neural networks for time series, and in particular using FNN. Therefore, we will follow their methodology of the neural network. Neural networks consist of an input node or layer, one or several hidden layers and an output layer and/or node. The input layer contains the values of the explanatory variables. The hidden layer(s) process these values through a series of nonlinear functions to the output layer. Essentially, a neural network can therefore be seen as a nonlinear regression model. All these layers consist of nodes, where the input nodes are connected with the hidden nodes (also known as hidden units), and the hidden nodes are connected with the output node(s). These connections are represented by weights, that determine the importance of the incoming information from a certain node to the receiving node. The weights are essentially the parameters of the model that need to be estimated. In the FNN, the information is processed in only one direction, which is forward from the input layer, through the hidden layer(s), to the output layer. An example of a FNN with a single hidden layer with two hidden units can be found in figure 1.

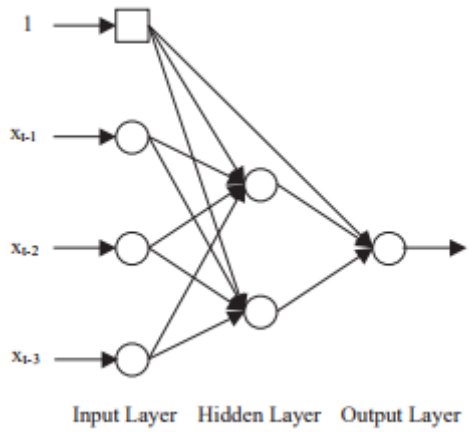


Figure 1: **Graphical illustration of a feedforward neural network.** The image shows three input nodes, one hidden layer with two hidden units, and an output layer with the resulting output node. The direction of the process is illustrated by the arrows. *Source: Balkin (1997).*

The input used in the input layer are p lags of the time series, where we allow $p > 1$. Besides the lags of the time series, it is also possible to use other input, such as investor sentiment and volume data. However, the effect of these inputs on the spread of two assets is unknown. Therefore, we decide to only use the lags of the time series, as proposed by Dunis et al. (2006). Medeiros & Teräsvirta (2006) define the model in the specification

from input to output of the FNN with M hidden units as

$$S_{t,i,j} = G(\boldsymbol{x}_t; \boldsymbol{\Psi}) + \epsilon_t = \boldsymbol{\phi}'\tilde{\boldsymbol{x}}_t + \sum_{m=1}^{M} \lambda_m F(\tilde{\boldsymbol{\omega}}_m'\boldsymbol{x}_t - b_m) + \epsilon_t, \tag{21}$$

where $G(\boldsymbol{x}_t; \Psi)$ is a nonlinear function of the input variables $\boldsymbol{x}_t$ and parameter vector $\boldsymbol{\Psi} = [\boldsymbol{\alpha}', \lambda_1, ..., \lambda_M, \tilde{\boldsymbol{\omega}}_1', ..., \tilde{\boldsymbol{\omega}}_M', \beta_1, ..., \beta_h]'$, and $\tilde{\boldsymbol{x}}_t = [1, \boldsymbol{x}_t']'$. The vectors $\tilde{\boldsymbol{\omega}}_1', ..., \tilde{\boldsymbol{\omega}}_M'$ represent the weight vectors between the input and hidden nodes, while the variables $\lambda_1, ..., \lambda_M$ represent the weights between the hidden and output nodes. The function $F(\tilde{\boldsymbol{\omega}}_m'\boldsymbol{x}_t - b_m)$ is the activation function in the hidden unit, for which we use the logistic sigmoid function:

$$F(\tilde{\boldsymbol{\omega}}_m'\boldsymbol{x}_t - b_m) = \frac{1}{1 + e^{-(\tilde{\omega}'\boldsymbol{x}_t - b_m)}}, \tag{22}$$

where $\tilde{\boldsymbol{\omega}_i} = [\tilde{\omega_{1i}}, ..., \tilde{\omega_{pi}}]'$ and b_i is the bias term. This bias term is added to increase the flexibility of the model to fit the data. Also, in (21), $\boldsymbol{\phi}'\tilde{\boldsymbol{x}}_t$ can be seen as a linear node in the model, which makes it possible to directly compare the performance of the network to the performance of a simple AR(p) model, since this model is nested in the network. After going through the hidden layers, the output computed is the forecast of the spread at time $t + 1$.

Brezak et al. (2012) propose to estimate the parameters iteratively using a training algorithm known as the error backpropagation (EBP) algorithm. This algorithm uses a simple gradient method to the minimize the error function, that is the sum of squared errors between the in-sample output of the network and the true value, for (21). The EBP uses the value of the error function of the previous iteration to improve the parameters further in the current iteration. The combination of weights minimizing the error function is therefore the solution of the network.

To make sure that the network is identified, that is the weights take on extreme values, weight decay and restriction is used (Krogh & Hertz, 1992). In each iteration, the weights are updated. The weight decay decreases the weights by a small amount, to keep the sizes of the weights small. The weight restriction ensures that the weights do not become too small or too large. If that is the case, the weights are decreased or increased by the amount needed to bring them back in range. Weight decay and restriction is implemented in most of the standard software packages for neural networks.

In order to reduce the possibility of overfitting the neural network, the in-sample dataset is divided into three parts, which are the training, validation and testing set. The training set is used to fit the model on the data, that is, estimating the parameters. The validation set is directly used to prevent overfitting. Every iteration, the parameters estimated based on the training set are used to fit the validation set. If the performance

on the training set increases, but the performance on the validation set remains the same or decreases, it means that we are overfitting the model and hence, we should stop training. Finally, the test set is used to assess the performance of the trained network on a completely new dataset. We choose to divide the data 60/20/20 for the three different sets, to have a sufficient sample for the training as well as the testing of the network.

Furthermore, decisions need to be made on the number of lags, the number of hidden layers and number of hidden units in each hidden layer. We try to find one general, optimal architecture of the network for all pairs, since this is computational optimal. The decisions on the hidden layers and hidden units are made based on the specific-to-general approach, as explained by Medeiros & Teräsvirta (2006), which means that we start with a network with one hidden layer (hidden unit) and work towards h hidden layers (hidden units) and choose the number of hidden layers (hidden units) that optimizes the performance of the network. Also, the number of lags is taken into account, which means that we will simultaneously determine the number of lags and apply the specific-to-general approach. This is done by applying the specific-to-general approach for each number of lags and evaluating the performance, which is the sum of squared errors over test set of the network. This way, each architecture is evaluated based on its performance on a completely new dataset, which gives a clear indication of the effectiveness of the network. We find that in general the best performance is reached for a network with five lags, two hidden layers and three hidden units. A similar, but more detailed description on the decision with regards to the architecture, and replicability can be found in Appendix D. Using this architecture, the weights are estimated for each pair individually. With two hidden layers, (21) changes to

$$S_{t,i,j} = G(\boldsymbol{x}_t; \boldsymbol{\Psi}) + \epsilon_t = \boldsymbol{\phi}'\tilde{\boldsymbol{x}}_t + \sum_{q=1}^{Q} \nu_q F(\boldsymbol{\lambda}_m' F(\tilde{\boldsymbol{\omega}}_m' \boldsymbol{x}_t - b_{1,m}) - b_{2,q}) + \epsilon_t, \tag{23}$$

where $\boldsymbol{\lambda}_m = [\lambda_{1i}, ..., \lambda_{pi}]'$ represents the weight vector corresponding to the weights between the nodes in the first and second hidden layer, ν_q the weights between the Q hidden nodes in the second hidden layer and the output node.

For the trading strategy, Dunis et al. (2006) propose to use several filters in order to decide on the moment to trade based on the forecasts. One of their better performing filters is their threshold filter, which works as follows. If the predicted spread is larger than the level of the filter, $X_{i,j}$, i.e. $\hat{S}_{t+1,i,j} > X_{i,j}$, then go or stay long in the spread, while for the predicted spread being smaller than minus the level of the filter, i.e. $\hat{S}_{t+1,i,j} < X_{i,j}$, a short position in the spread should be taken. In case of the spread being in between plus and minus the level of the filter, no position is taken. The level of the filter is arbitrarily

optimized in-sample in the same way as for the variable $c_{i,j}$ in (14). Dunis et al. (2006) state that with accurate predictions of the spread, it is possible to filter out traders that are smaller than the level of the filter which improves the risk/return rate of the model.

2.3.2.2 Recurrent Neural Network

The second neural network Dunis et al. (2006) use is the recurrent neural network (RNN). The main difference with FNN is that RNN uses its own output from the hidden layer(s) at time $t-1$ as new input for the hidden layer(s) in order to train itself at time t. The output from the previous hidden layer is saved and stored in the context layer, before it is processed as input for the current layer. An example of a RNN with a single hidden layer can be found in figure 2.

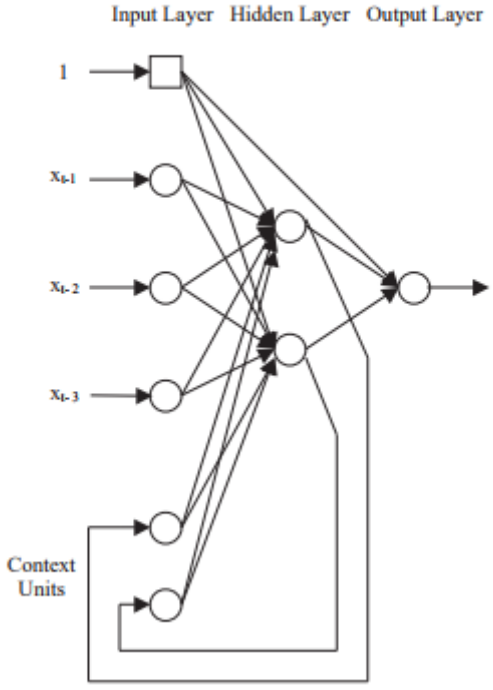


Figure 2: **Graphical illustration of a recurrent neural network.** The image shows three input nodes, one hidden layer with two hidden units and their two corresponding context units, and an output layer with the resulting output node. The direction of the process is illustrated by the arrows. In particular, the arrows going from the hidden units to the context units shows that the output from the hidden layer is in the model again. This illustrates the difference between the recurrent and feedforward neural networks. *Source: Balkin (1997).*

Where a FNN only uses the information at the current time (i.e. p lags) to predict the value at the next time, a RNN has the advantage that its recurrent properties allow the network to also look at and remember the information from the past (Sundermeyer et

al., 2014). In particular for predicting a time series with mean-reverting properties, being able to use long-term dependencies should be useful. To get into more detail, the lags input in the FNN are used to determine the weights, but the information is not stored in the network. In other words, for the value at the current time t, the lags $t-1$ until $t-p$ are used, but the values before $t-p$ are left out. For a RNN, the weights are also estimated based on the lags, but the information of the hidden layer is stored in the context unit and is used again at later times. Thus, compared to the FNN, for the value at the current time t, the lags $t-1$ until $t-p$ are used, but also the output of the hidden layer at time $t-1$. The output of the hidden layer at time $t-1$ is based on the lags $t-2$ until $t-p-1$ and the output of the hidden layer at time $t-2$, etc. This way, the past information is stored and used in the network, which makes it easier to model the long-term dependencies.

As discussed for FNN, the input used in the input layer are lags of the time series of the spread. The output produced is once again the forecast of the spread at time $t+1$. Due to the recurrent characteristics of the network, (21) changes to

$$S_{t,i,j} = G(\boldsymbol{x}_t; \boldsymbol{\Psi}) + \epsilon_t = \boldsymbol{\phi}' \tilde{\boldsymbol{x}}_t + \sum_{m=1}^{M} \lambda_m F(\tilde{\boldsymbol{\omega}}_m' \boldsymbol{x}_t + \tilde{\boldsymbol{w}}_l' \boldsymbol{h}_{t-1} - b_m) + \epsilon_t, \tag{24}$$

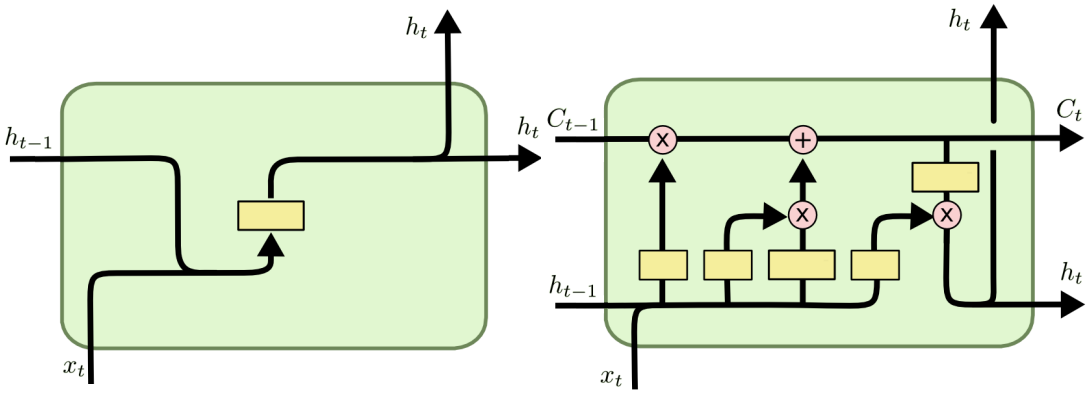
where $\tilde{\boldsymbol{w}}_l' \boldsymbol{h}_{t-1}$ represents the recurrent part, that is, $\tilde{\boldsymbol{w}}_l$ denotes the weight vector between the context layer and hidden layer, and $\boldsymbol{h}_{t-1}$ denotes the output of the hidden layer at time $t-1$, defined by

$$\boldsymbol{h}_{t-1} = F(\tilde{\boldsymbol{\omega}}_m' \boldsymbol{x}_{t-1} + \tilde{\boldsymbol{w}}_l' \boldsymbol{h}_{t-2} - b_i). \tag{25}$$

In practice, using the EBP algorithm for RNN results in the vanishing or exploding gradient problem, as described by Hochreiter & Schmidhuber (1997). As discussed, the EBP uses the value of the error function (sum of squared errors) of the previous iteration to improve the parameters further in the current iteration. With RNNs, it could be the case that, due to its recurrent characteristics, the value of the error function either vanishes (i.e. becomes very small) or explodes (i.e. becomes extremely large), which makes training the network hard or even impossible to do.

This results in the network losing its ability to take into account the correlation between the long-term behaviour of the time series and the current value. Since the long-term behaviour is important for pairs trading, essentially, that is the mean-reverting behaviour, we have to overcome this issue. In order to do so, Hochreiter & Schmidhuber (1997) propose to use a special kind of RNN, that is a Long-Short Term Memory (LSTM) network. The main difference between the LSTM and standard RNN as discussed above, is that the dynamics in the hidden layer of the LSTM change in order to overcome the

vanishing gradient problem. The core idea behind the LSTM is to have a cell state, $\boldsymbol{C}_t$, in the hidden layers, which is used to store and process (part of) the information from all previous times, instead of only using the output of the hidden layer at the previous time. The cell state interacts with the information from the previous hidden layer and the new input through so-called gates. Gates are used to determine what information is removed, kept or added to the cell state, based on the information from the previous hidden layer and new input. Where in (24) the information is processed through one single nonlinear function $F(.)$, for example the logistic sigmoid function as defined in (22), the gates in the LSTM makes use of several nonlinear functions. Figure 3a illustrates a hidden layer in a regular RNN, whereas figure 3b illustrates a hidden layer in the LSTM.



(a) Hidden layer of a regular RNN

(b) Hidden layer of a LSTM

Figure 3: **A closer look at the hidden layers of a regular RNN (a) and LSTM (b).** The yellow rectangle represent nonlinear functions where the new input $\boldsymbol{x}_t$ and output from the previous hidden layer $\boldsymbol{h}_{t-1}$ are processed through. The red circles containing either a multiplication or addition sign represent the corresponding linear operation. Furthermore, the variable $\boldsymbol{C}_t$ denotes the cell state at time t.
Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

The LSTM consists of three gates, which are the forget gate, input gate and output gate. The forget gate, as illustrated in figure 4, determines what information is going to be removed from the previous cell state $\boldsymbol{C}_{t-1}$. In order to so, the information from the previous state $\boldsymbol{h}_{t-1}$ and new input $\boldsymbol{x}_t$ are processed through a logistic sigmoid function as defined in (22), which gives

$$\boldsymbol{f}_t = \frac{1}{1 + e^{-(\tilde{\boldsymbol{\omega}}_f' \boldsymbol{x}_t + \tilde{\boldsymbol{w}}_f' \boldsymbol{h}_{t-1} - b_f)}}, \tag{26}$$

where $\tilde{\boldsymbol{\omega}}_f'$ and $\tilde{\boldsymbol{w}}_f'$ represent the weight vectors for respectively the input $\boldsymbol{x}_t$ and previous hidden layer $\boldsymbol{h}_{t-1}$ for the forget gate, while b_f is the bias term. The output of a sigmoid function, in this case represented by $\boldsymbol{f}_t$, is between 0 and 1 for each element in $\boldsymbol{C}_{t-1}$, where 0 represents leaving that specific element out of the next cell state $\boldsymbol{C}_t$ and 1 represents

keep the specific element completely in the next cell state. Therefore, the previous state $\boldsymbol{C}_{t-1}$ is multiplied by $\boldsymbol{f}_t$, such that the decisions made in the forget gate are processed in the cell state.

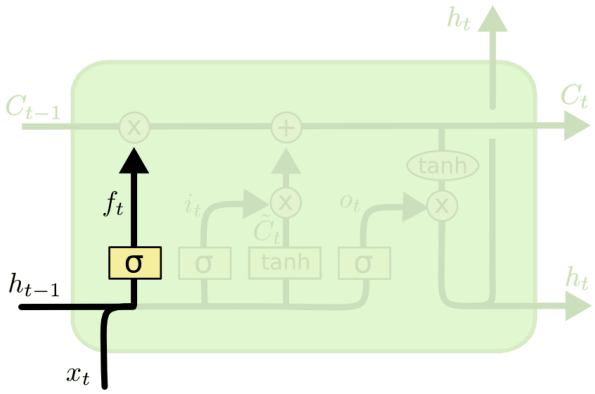


Figure 4: **Graphical illustration of the forget gate.** The yellow rectangle containing the σ represents the logistic sigmoid function as in (22) where the new input $\boldsymbol{x}_t$ and output from the previous hidden layer $\boldsymbol{h}_{t-1}$ are processed through, resulting in output $\boldsymbol{f}_t$.
Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

The input gate is used to decide what information in $\boldsymbol{C}_{t-1}$ to update as well as generate a set of possible update values $\tilde{\boldsymbol{C}}_t$ to add to the previous cell state. This process is illustrated in figure 5a. Once again, a sigmoid function as in (22) decides on the information to update, resulting in output $\boldsymbol{i}_t$, which again consists of values between 0 and 1, given by

$$\boldsymbol{i}_t = \frac{1}{1 + e^{-(\tilde{\boldsymbol{\omega}}_I' \boldsymbol{x}_t + \tilde{\boldsymbol{w}}_I' \boldsymbol{h}_{t-1} - b_I)}}, \tag{27}$$

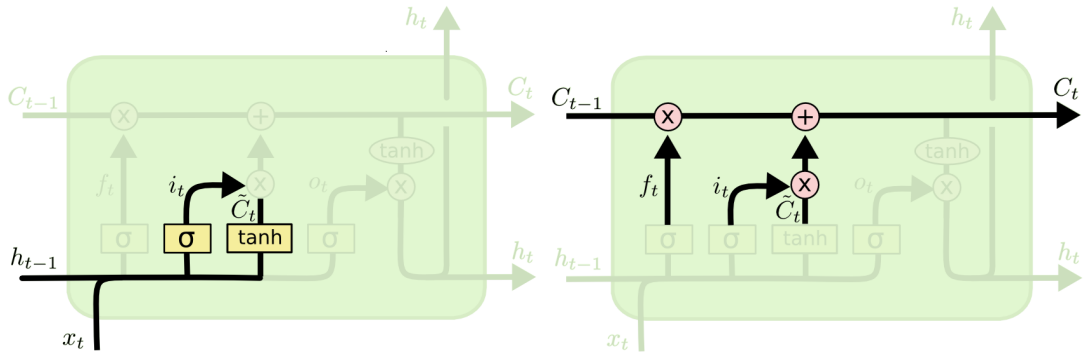
where $\tilde{\boldsymbol{\omega}}_I'$ and $\tilde{\boldsymbol{w}}_I'$ represent the weight vectors for respectively the input $\boldsymbol{x}_t$ and previous hidden layer $\boldsymbol{h}_{t-1}$ for the input gate, while b_I is the bias term. At the same time, possible update values for each element in $\boldsymbol{C}_{t-1}$, denoted by $\tilde{\boldsymbol{C}}_t$, are generated by means of a hyperbolic tangent function, given by

$$\tilde{\boldsymbol{C}}_t = \frac{e^{-(\tilde{\boldsymbol{\omega}}_C' \boldsymbol{x}_t + \tilde{\boldsymbol{w}}_C' \boldsymbol{h}_{t-1} - b_C)} - e^{-(\tilde{\boldsymbol{\omega}}_C' \boldsymbol{x}_t + \tilde{\boldsymbol{w}}_C' \boldsymbol{h}_{t-1} - b_C)}}{e^{-(\tilde{\boldsymbol{\omega}}_C' \boldsymbol{x}_t + \tilde{\boldsymbol{w}}_C' \boldsymbol{h}_{t-1} - b_C)} + e^{-(\tilde{\boldsymbol{\omega}}_C' \boldsymbol{x}_t + \tilde{\boldsymbol{w}}_C' \boldsymbol{h}_{t-1} - b_C)}}, \tag{28}$$

where $\tilde{\boldsymbol{\omega}}_C'$ and $\tilde{\boldsymbol{w}}_C'$ represent the weight vectors for respectively the input $\boldsymbol{x}_t$ and previous hidden layer $\boldsymbol{h}_{t-1}$ for the update values, while b_C is the bias term. The function in (28) returns possible update values $\tilde{\boldsymbol{C}}_t$ that range between -1 and 1. These values are multiplied by $\boldsymbol{i}_t$, which gives a set of candidate values scaled by how much each element in the previous state will be updated. Next, the new cell state $\boldsymbol{C}_t$ is computed as follows:

$$\boldsymbol{C}_t = \boldsymbol{f}_t \cdot \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \cdot \tilde{\boldsymbol{C}}_t, \tag{29}$$

as illustrated in 5b.



(a) **The nonlinear functions.** (b) **The linear interactions.**

Figure 5: **Graphical illustration of the input gate.** Figure (a) represents the nonlinear functions the input is processed through in the input gate, whereas figure (b) illustrates the linear interactions between the outputs from the forget gate and input gate, resulting in new cell state $\boldsymbol{C}_t$. The yellow rectangle containing the σ represents the logistic sigmoid function as in (22) where the new input $\boldsymbol{x}_t$ and output from the previous hidden layer $\boldsymbol{h}_{t-1}$ are processed through, resulting in output $\boldsymbol{i}_t$. The yellow rectangle containing the σ represents the hyperbolic tangent function as in (28) where the new input $\boldsymbol{x}_t$ and output from the previous hidden layer $\boldsymbol{h}_{t-1}$ are processed through, resulting in candidate values for the cell state denoted by $\tilde{\boldsymbol{C}}_t$. *Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/*

The final step in the hidden layer is to decide what to output, which is determined by the output gate as illustrated in figure 6. The output of the hidden layer $\boldsymbol{h}_t$ is a filtered version of the cell state and consists of two components. First, the output from the previous hidden layer and the new input are processed through a sigmoid function, as defined in (22), which is similar to what happens in the hidden layer of a normal RNN, resulting in output $\boldsymbol{o}_t$, as given by

$$\boldsymbol{o}_t = F(\tilde{\boldsymbol{\omega}}'_m \boldsymbol{x}_{t-1} + \tilde{\boldsymbol{w}}'_l \boldsymbol{h}_{t-1} - b_i). \tag{30}$$

Second, the new cell state $\boldsymbol{C}_t$ is processed through a hyperbolic tangent function as defined in (28) and multiplied by $\boldsymbol{o}_t$, which changes (25) to

$$\boldsymbol{h}_t = \boldsymbol{o}_t \cdot \frac{e^{\boldsymbol{C}_t} - e^{-\boldsymbol{C}_t}}{e^{\boldsymbol{C}_t} + e^{-\boldsymbol{C}_t}} \tag{31}$$

This way, the new output only consists of the parts of the cell state as based on the output of previous hidden layer and the new input. All together, (24) changes to

$$S_{t,i,j} = \boldsymbol{\phi}' \tilde{\boldsymbol{x}}_t + \sum_{m=1}^{M} \lambda_m \left(F(\tilde{\boldsymbol{\omega}}'_m \boldsymbol{x}_t + \tilde{\boldsymbol{w}}'_l \boldsymbol{h}_{t-1} - b_m) \cdot \frac{e^{\boldsymbol{C}_t} - e^{-\boldsymbol{C}_t}}{e^{\boldsymbol{C}_t} + e^{-\boldsymbol{C}_t}} \right) + \epsilon_t. \tag{32}$$

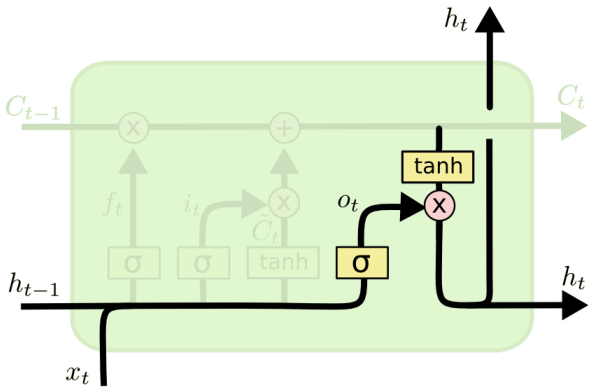


Figure 6: **Graphical illustration of the output gate.** The yellow rectangle containing the σ represents the logistic sigmoid function as in (22) where the new input $\boldsymbol{x}_t$ and output from the previous hidden layer $\boldsymbol{h}_{t-1}$ are processed through, resulting in output $\boldsymbol{o}_t$. The yellow rectangle containing the σ represents the hyperbolic tangent function as in (28) where the new cell state $\boldsymbol{C}_t$ is processed through. The two outputs are multiplied, resulting in the output of the hidden layer $\boldsymbol{h}_t$.
Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Once again, similarly for (33), the LSTM with two hidden layers is defined as

$$S_{t,i,j} = \boldsymbol{\phi}'\tilde{\boldsymbol{x}}_t + \sum_{q=1}^{Q} \nu_q \left(F\left(\boldsymbol{\lambda}'_m \left(F(\tilde{\boldsymbol{\omega}}'_m \boldsymbol{x}_t + \tilde{\boldsymbol{w}}'_l \boldsymbol{h}_{t-1} - b_m) \cdot \frac{e^{\boldsymbol{C}_{1,t}} - e^{-\boldsymbol{C}_{1,t}}}{e^{\boldsymbol{C}_{1,t}} + e^{-\boldsymbol{C}_{1,t}}} \right) - b_{2,q} \right) \cdot \frac{e^{\boldsymbol{C}_{2,t}} - e^{-\boldsymbol{C}_{2,t}}}{e^{\boldsymbol{C}_{2,t}} + e^{-\boldsymbol{C}_{2,t}}} \right) + \epsilon_t. \tag{33}$$

The trading strategy for the RNN is the same as for FNN as discussed in 2.3.2.1.

2.4 Economic Evaluation

To evaluate the performance of the discussed methods, several economical measures are used. The portfolios for all strategies are evaluated using the Sharpe (1994) and Sortino (Sortino & Price, 1994) ratios, as well as the Maximum Drawdown (Pereira Camara Leal & Vaz de Melo Mendes, 2005).

2.4.1 Portfolio Construction

After identifying the pairs, a portfolio is created and the strategies will be applied. This means that there is an individual portfolio for every strategy and therefore, the performance of the strategy can be measured by means of the cumulative profit made on the associated portfolio. The cumulative profit is determined following the approach of Gatev et al. (1999). For the first trade a short position of \$1 is taken in the relatively overpriced stock, while for the relatively underpriced stock a long position of \$1 is taken. This way,

a zero-cost portfolio is constructed, which is a portfolio that is self-financing by taking a long and short position in two assets for the same amount. After the initial trade, the trade is either unwinded (multiple times) before the end of the trading period or, in case of no convergence, only unwinded at $t = T$. This results in one or multiple cashflows for each pair. Since the gains and losses are calculated over the long and short positions of \$1, Gatev et al. (1999) state that the payoffs can be interpreted as excess returns. The open positions are valued daily, to be able to compute a return series over the entire period.

Furthermore, besides determining the pairs in the formation period using the cointegration method in 2.2, estimating the parameters for the stochastic spread methods in 2.3.1 and initizialing the neural networks in 2.3.2, the formation period is also used to determine what potentially could be the most profitable pairs to use in the trading period following the approach of Gatev et al. (1999). For each method individually, the corresponding trading strategy is used in the formation period for all the possible pairs. Subsequently, the Sharpe ratio is calculated for all pairs, which gives the possibility to rank the pairs based on their risk-adjusted performance. Using this ranking, a certain number of best performing pairs over the formation period can be selected. In this research, the top 5, 20 and 50 performing pairs over the formation period are selected for each method individually. We do not consider more than 50 pairs, since we saw that the results do not significantly improve when using a larger number of pairs. Note that for the different methods the best performing pairs can differ because each method has its own trading rule.

2.4.2 Sharpe Ratio

The Sharpe (1994) ratio ($\mathcal{SR}$) adjusts the accumulated return on the portfolio for the risk taken, and can be defined as

$$\mathcal{SR}_i = \frac{\bar{r}_{p,i} - r_f}{\sigma_{p,i}}, \tag{34}$$

where $\bar{r}_{p,i}$ is the average portfolio return using strategy i, r_f represents the riskfree rate and $\sigma_{p,i}$ is the standard deviation of the portfolio returns using strategy i. Note that the subscript p denotes that the variable refers to a portfolio.

2.4.3 Sortino Ratio

Sortino & Price (1994) consider upside risk to be no 'real' risk, since it results in a larger positive return. Therefore, the difference between the Sortino and Sharpe ratio is that the Sortino ratio ($\mathcal{STR}$) only takes into account the downside risk. This results in the

following expression for the Sortino ratio:

$$\mathcal{STR}_i = \frac{\bar{r}_{p,i} - r_f}{\sqrt{\frac{1}{T}\sum_{t=1}^{T}(r_{t,i} - MAR)^2 * \mathbb{1}_{[r_{t,i}-MAR<0]}}}, \tag{35}$$

where $r_{t,i}$ is the return at time t using strategy i, r_f is the riskfree rate and MAR is the minimum acceptable return, which is the minimum return the investor wants to generate. In our case, we use the return of an ETF representing the entire market over the trading period as MAR.

2.4.4 Maximum Drawdown

Where the Sharpe and Sorentino ratios adjust the return for the risk taken, the Maximum Drawdown ($\mathcal{MDD}$) indicates the downside risk over a certain period of time by calculating the maximum loss from a peak to trough in that period. Pereira Camara Leal & Vaz de Melo Mendes (2005) define W_t to be the value of the portfolio at time t. Then, let W_k be a local maximum and W_l be the next local minimum. Thus, $W_k > W_{k+1} \geq ... \geq W_l$ with $l - k \geq 1$ and $k \geq 1$. They define a drawdown as

$$X_q = \log\left(\frac{W_l}{W_k}\right), \tag{36}$$

where $q = 1, 2, ...$ Over the entire out-of-sample period, this will results in N drawdowns, $X_1, ..., X_N$, with $N < T$. The maximum drawdown over the entire period is then defined as

$$\mathcal{MDD}_\rangle = \min(X_{1,i}, ..., X_{N,i}), \tag{37}$$

where i denotes the strategy.

2.4.5 Statistical Interference of the Sharpe Ratio

To test whether the Sharpe ratio is significantly different from the ratio of another strategy, the bootstrap two-sided test as proposed by Ledoit & Wolf (2008) can be used. They provide a two-sided test for the null hypothesis that the difference between the ratios is zero at significance level α by constructing a bootstrap confidence interval with confidence level $1 - \alpha$. The test rejects the null hypothesis if zero is not in the generated interval. The advantage of the approach is that resampling can be done from the observed data. Unfortunately, there are no such tests available for the Sortino ratio and the Maximum Drawdown. However, the significance of the difference in Sharpe ratios will provide some good insights in the performance of the models relatively to each other and the conclusions will possibly also hold for the Sortino ratio.

2.4.6 Transaction Costs

It is not the aim of the research to find a proper measure or estimate for the transaction costs. However, transaction costs can have a large implication on the results. For example, strategy i is able to generate a return of 60% and needs 75 trades over the out-of-sample period, while strategy j generates a returns a 30% but only needs 15 trades in order to do so. The net profit, after transaction costs, will be higher for strategy j than for strategy i, because the impact of transaction costs will be much higher for the latter, while the initial results gave the impression that strategy i was the better and more profitable strategy. We will overcome this issue without making any assumptions on the amount or percentage transaction costs per trade. For each strategy, the return and other economic measures will be reported, together with the number of trades needed to get to this return. This way, we can calculate the break-even transaction costs (BETC), which are the transaction costs per trade in order to break-even on the initial investment. The BETC will give an indication for investors whether it would be profitable after transaction costs to use the strategy, given their personal transaction costs.

3 Data

This section describes the data, its sources and the selection procedure used in the research.

Since there are no lists or databases available on all existing ETFs, we choose to use data on all the ETFs listed on the NYSE Arca from the Datastream database. The US ETF market is far bigger than the European market, which results in the NYSE Arca containing the largest variety of ETFs worldwide. Therefore, this might give more possibilities when it comes to pairs trading. Currently, the NYSE Arca contains 1,237 active and 514 dead ETFs. Because of the large growth of the ETF market in the past years, nearly half of these ETFs have been established after 2010. To have a sample size that results in a sufficient amount of data to generate reliable forecasts and make a sufficient division between the formation and trading period, we choose to use a sample running from 01/01/2011 until 18/08/2017. This results in 1,731 observations and eliminates all the ETFs established after 31/12/2010, which results in a total number of 634 active and 247 dead ETFs. We choose to include the dead ETFs in our sample for two reasons. First, not including the dead ETFs might lead to selection bias in our data, because only including the active ETFs might to different and/or biased results which are not representable for the entire universe of ETFs. Second, at time $t-1$ it is not possible to know if and which ETFs will be dead at time t, which is also a valid reason to include the ETFs.

We conduct daily time series of the prices after dividend and splits. Following the approach of Caldeira & Moura (2013), we eliminate the less liquid ETFs. We choose to eliminate ETFs that traded less than 2.25% of the total trading days the ETF was active. Caldeira and Moura (2013) motivate their approach by stating that less liquid instruments may involve greater operational costs and difficulty in setting up a covered call. Using this criterion, 283 ETFs are eliminated because of illiquidity, resulting in 598 ETFs to trade with. In total, this gives $\binom{598}{2} = 178,598$ possible pairs. Executing the cointegration method for all 178,598 possible pairs is not considered to be very efficient. Therefore, we want to use sector or category specifiers of the ETFs to improve the efficiency. Also, it is more likely to find pairs within sectors, since assets within the same sector are more likely to follow the same trend. We conduct the corresponding category specifier for each ETF from Barchart[2]. Appendix E shows a breakdown of the number ETFs in a specific category. Furthermore, we conduct the factors in the Carhart four-factor model from the Kenneth French data library[3].

[2]Link to Barchart.com
[3]Link to Kenneth French data library

The performance of all methods is evaluated over the entire trading period and three subperiods in the trading period. As described above, the data consists of 1,731 observations. We choose to divide the entire period using the 60/40 rule of thumb for the formation and trading period. This means that the formation period consists of 1,230 observations, while the trading period contains a total of 501 observations, which is almost two trading years consisting of 252 trading days. Thus, the formation period runs from 01/01/2011 until 17/08/2015 and the trading period runs from 18/08/2015 until 18/08/2017. Note that the formation period is the sample which is divided in the training, validation and test set for the neural networks.

As discussed by Krauss (2015), pairs trading strategies tend to be more profitable in periods of large volatility. Large volatility results in more opportunities, since the spread tends to deviate more and by a larger amount from its mean. To investigate this claim for our methods, we divide the entire trading period in subperiods based on the state of a benchmark portfolio, representing the market. As discussed in section 3, the asset universe consists of ETFs in several sectors. These sectors range from large cap stocks to bonds, as well as from emerging markets to the European market. To be able to find a benchmark that represents the entire universe, we use a portfolio of the Vanguard Total Stock World Stock ETF (VT), which is an ETF that (attempts to) represent the entire stock universe, and the iShares Core U.S.Aggregate Bond ETF (AGG), representing the U.S. bond market. Table 8 in Appendix E shows that 76 of the total number of ETFs are in the Bonds category, which is 12.7% of the total. Therefore, the benchmark portfolio consists of 87.3% VT and 12.7% AGG. By dividing the trading period in three equally large subperiods of 167 trading days, three different states of the market are created. This means that subperiod one runs from 18/08/2015 until 17/04/2016, the second-period from 18/04/2016 until 17/12/2016, while the final subperiod lasts from 18/12/2016 until 18/08/2017, since these periods show a large change in volatility compared to each other. Figure 7 shows the returns of the market portfolio over the entire trading period, while also the division for the subperiods is illustrated.

As can be seen in the figure, the first subperiod is the most volatile, while the volatility decreases in the second subperiod. In the third subperiod, the return grow steadily, indicating low volatility. This is confirmed by table 1, which shows the volatility in each of the three periods.

Table 1: **The monthly volatility for the different subperiods.**

	Period 1	Period 2	Period 3
σ	4.39%	3.47%	1.66%

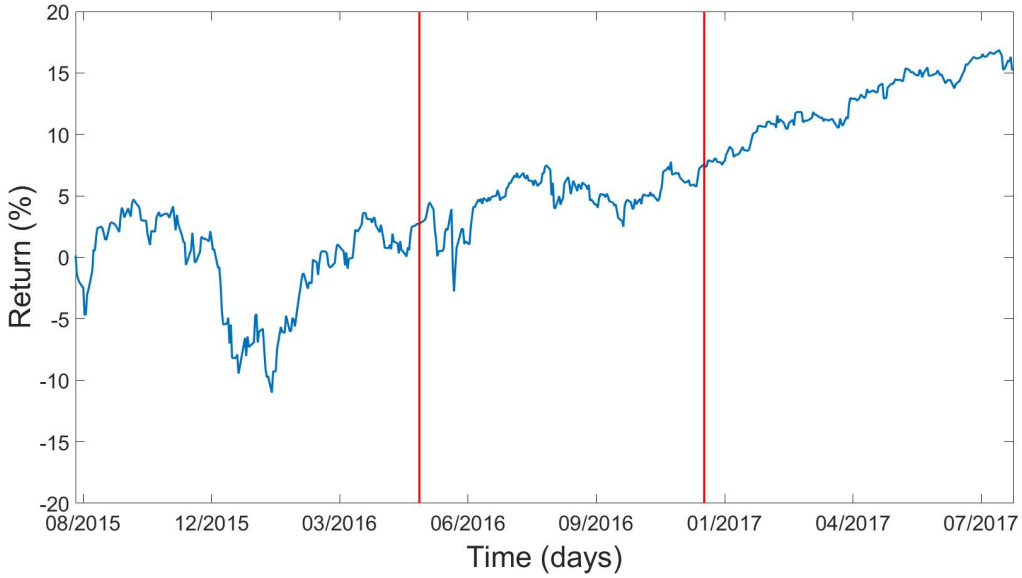


Figure 7: **The cumulative returns over time of the benchmark portfolio, which represents the market.** The vertical red lines indicate the seperation between the subperiods.

Based on figure 7 and table 1, we expect the performance of the methods to be the best in the first subperiod, while the methods will probably struggle to generate a large return in the third subperiod.

4 Results

This section discusses the results obtained by applying the methodology discussed in chapter 2.

4.1 Pair Selection

As discussed in section 2.2, the pairs are selected based on their cointegration relation. The number of selected pairs per sector can be found in table 2.

Table 2: **Breakdown of the number of selected ETF pairs per category.**

Categories	# of pairs	# of possible pairs	% of possible pairs
Asia	566	1,482	38.19%
Biotech	36	210	17.14%
Bonds	3,380	5,700	59.30%
Commodities	86	240	35.83%
Consumer	134	506	26.48%
Currency	34	182	18.68%
Emerging Markets	434	1,122	38.68%
Energy	114	506	22.53%
Europe	135	342	39.47%
Financial Services	137	600	22.83%
Income	183	506	36.17%
Industrials	6	56	10.71%
International	774	2,162	35.80%
Large Cap	913	4,032	22.64%
Materials	4	12	33.33%
Metals	61	272	22.43%
Mid Cap	577	1,640	35.18%
Multi Cap	14	42	33.33%
Natural Resources	6	72	8.33%
Power	20	132	15.15%
Real Estate	103	210	49.05%
Small Cap	693	1,332	52.03%
Technology	93	506	18.38%
Utilities	16	42	38.10%
Total	8,519	21,906	38.89%

First of all, we see that a total number of 8,519 pairs is identified. Taking table 10 into account shows that the sectors with the most ETFs, are also the sectors with the most pairs. The Bonds sector has the largest percentage of pairs, 59.30%, while the Natural Resources sector has the smallest percentage of pairs, 8.33%. Also, the Small Cap and

Real Estate sectors have large perentages of pairs selected, respectively being 52.03% and 49.05%. For the other sectors, the percentage of pairs selected ranges from 10.71% to 39.47%.

4.2 Strategy Performance

The most important result is to see which method has the best relative economic performance. In order to do so, the performance of the methods is first evaluated over the entire period. The distance method, stochastic spread method, stochastic residual spread method, feedforward neural network and recurrent neural network are denoted by respectively DM, SSM, SRSM, FNN and RNN. Figure 8 shows how the returns evolve over time for the different methods.

As can be seen in figure 8a, the returns exhibit high volatility, whereas this volatility decreases in both figure 8b and 8c. This means that the more pairs used, the less volatility exhibited by the portfolio. Furthermore, it stands out that the RNN is able to perform consistently and generate a large return as well. Only the SSM seems to be able to come close in terms of return when using the top 5 pairs. To get into greater detail, table 3 reports the return, monthly volatility (defined as the standard deviation of the returns), number of trades and BETC over the entire trading period, while table 11 shows the economic measures. Monthly implies that these measures are originally calculated over the entire period, but have been scaled by the number of trading days (21) in one month.

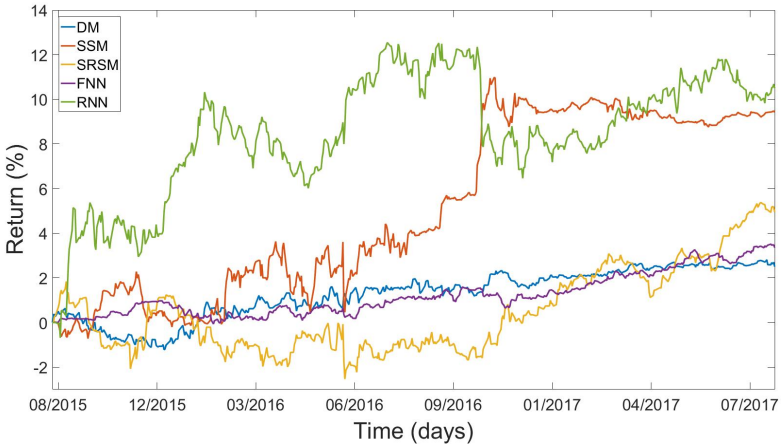
The results that stands out most in table 3 is the consistent performance of the RNN in terms of return independent of the number of pairs used. The RNN is able to generate a return ranging from 9.81% to 12.46% for all the three situations. which is considerably larger than the other methods. In particular in the case where the top 5 pairs is used, the RNN manages to generate a return of 11.07% in only 51 trades, which results in corresponding BETC of 0.217%. This means that the RNN is profitable after transaction costs if the costs are not more than 0.217% per trade. Also, the average monthly returns of the RNN are significant for the top 20 and 50 pairs on a 1% significance level, while the other methods show less or no statistical significant average monthly returns in these cases.

For the top 5 pairs, the only method that comes relatively close to the BETC of the RNN is the distance method, while for the top 20 and 50 pairs, the distance method has a higher BETC than the RNN. However, the difference in return is that large for all situations that at least for the top 5 and top 20 pairs, and possibly for the top 50 as well, an investor will always prefer the RNN over the distance method despite the difference in BETC, as long as his actual transaction costs make the trades profitable.

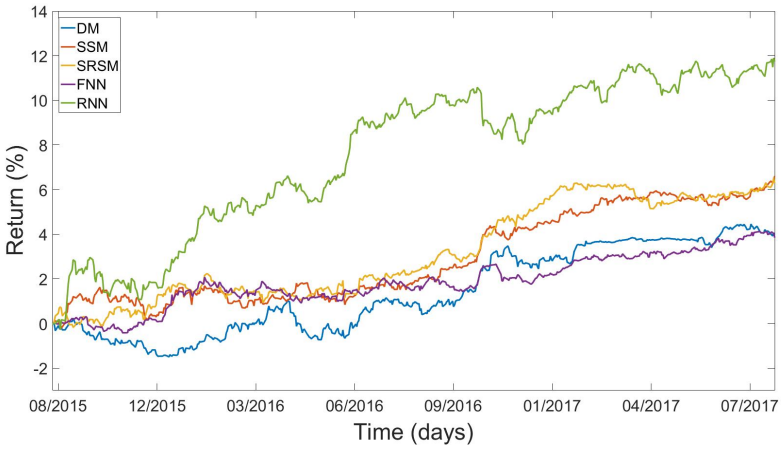
A drawback of the RNN could be the relatively high monthly volatility compared to

Table 3: **Total excess return, average monthly return, monthly volatility, number of trades and BETC over the entire trading period using the 5, 20 and 50 best performing pairs in formation period.** This table reports the total excess returns, average monthly return, monthly volatility, number of trades and BETC for the different methods over the entire trading period, which is 18/08/2015 until 18/08/2017. Top 5, 20 and 50 respectively denotes the number of best performing pairs in the formation period that are used in the trading period. The asterisks indicate that the average monthly return of the method for this particular case is statistically significant on a 1%(***), 5%(**) or 10%(*) significance level. The results of the conducted t-test can be found in Appendix F.

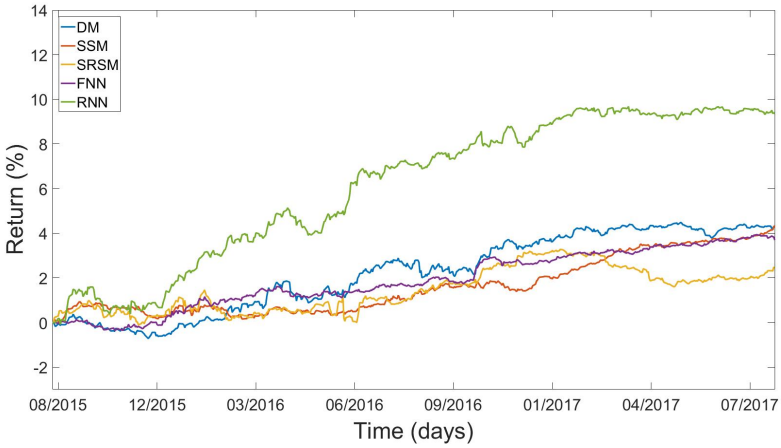
	Top 5					**Top 20**					**Top 50**				
	Return	μ	σ	Trades	BETC	Return	μ	σ	Trades	BETC	Return	μ	σ	Trades	BETC
DM	2.54%	0.11%	0.63%	22	0.115%	4.05%	0.17%*	0.57%	52	0.078%	4.42%	0.18%**	0.46%	118	0.037%
SSM	9.88%	0.40%	1.44%	384	0.026%	6.83%	0.28%**	0.58%	1,244	0.005%	4.43%	0.18%***	0.30%	3,110	0.001%
SRSM	5.08%	0.21%	1.15%	478	0.011%	6.49%	0.26%**	0.57%	2,396	0.003%	2.49%	0.10%	0.49%	6,018	0.000%
FNN	3.41%	0.14%**	0.38%	194	0.018%	3.93%	0.16%*	0.46%	392	0.010%	3.76%	0.16%***	0.29%	1,124	0.003%
RNN	11.07%	0.44%	1.80%	52	0.217%	12.46%	0.49%***	0.92%	332	0.038%	9.81%	0.39%***	0.58%	866	0.011%



(a) Returns over time using the 5 best performing pairs in the formation period



(b) Returns over time using the 20 best performing pairs in the formation period



(c) Returns over time using the 50 best performing pairs in the formation period

Figure 8: **Return over time using the top 5, 20 and 50 pairs.** The return using the different methods for the top 5 (a), 20 (b) and 50 (c) pairs. The returns of the distance method, stochastic spread method, stochastic residual spread method, feedforward neural network and recurrent network are illustrated by the blue, red, yellow, purple and green lines respectively.

Table 4: **Sharpe ratio, Sortino ratio and maximum drawdown over the entire trading period using the 5, 20 and 50 best performing pairs in formation period.** This table reports the monthly Sharpe ratio, monthly Sortino ratio and maximum drawdown for the different methods over the entire trading period, which is 18/08/2015 until 18/08/2017. Top 5, 20 and 50 respectively denotes the number of best performing pairs in the formation period that are used in the trading period.

	Top 5			**Top 20**			**Top 50**		
	$\mathcal{SR}$	$\mathcal{SRT}$	$\mathcal{MDD}$	$\mathcal{SR}$	$\mathcal{SRT}$	$\mathcal{MDD}$	$\mathcal{SR}$	$\mathcal{SRT}$	$\mathcal{MDD}$
DM	0.17	0.22	1.84%	0.29	0.39	1.73%	0.40	0.51	1.08%
SSM	0.27	0.38	3.12%	0.48	0.67	1.45%	0.61	0.76	0.77%
SRSM	0.18	0.25	4.27%	0.46	0.63	1.29%	0.21	0.26	1.67%
FNN	0.37	0.46	1.03%	0.35	0.46	1.15%	0.53	0.67	0.50%
RNN	0.25	0.35	5.90%	0.54	0.77	2.53%	0.67	0.97	1.23%

the other methods. In all three cases, the RNN has the highest volatility. It can be argued that this relatively high volatility is caused by its larger return over the trading period, since the other methods generate a smaller return and therefore exhibit less volatility. Also, we see that the volatility decreases when the number of pairs increases.

When taking the economic measures in table 11 into account, we see that the Sharpe and Sortino ratio are amongst the highest using the top 5 pairs, and the highest in case of the top 20 and 50. In particular when using the top 50 pairs, the RNN generates a relatively large Sharpe and Sortino ratio of 0.67 and 0.97 respectively, whereas the methods closest to this performance is the SSM with 0.61 and 0.76 respectively. Also, the large positive difference between the Sortino and Sharpe ratios using the top 20 and 50 pairs indicate that the RNN is able to limit its downside risk while generating a large return. However, for the top 5, the difference is smaller, which in combination with the large maximum drawdown shows that the downside risk is relatively larger than in the other cases. As already explained from figure 8a, the returns of RNN exhibit quite large volatility for a smaller number of pairs used, resulting in a large drawdown. Table 3 already showed that when the number of pairs increases, the volatility decreases, which also expresses itself in the decreasing maximum drawdown and increasing Sharpe and Sortino ratios. Since we know that the return remains stable, the ratios can only increase if the (downside) volatilty increases. Thus, in all aspects, the RNN is the best performing method.

Analyzing the results of the other methods, the distance method is outperformed in terms of return by all methods except for the FNN using the top 20 and 50 pairs. Based on these findings, it could be argued that the non-parametric distance method is inferior to the other, parametric, methods. To a certain extent, this is correct. The distance method lacks in terms of return, however, it needs far less trades than the other methods when

the number of pairs increases, to get to its return. Where the RNN compensates with a large return, the other methods are unable to do so, which results in the distance method being able to beat the SSM, SRSM and FNN in terms of the BETC. Therefore, after transaction costs, the distance method might still be able to outperform these methods.

However, in terms of the economic measures, the distance method is outperformed in terms of Sharpe and Sortino ratios by considerable amounts, except for the SRSM when using the top 50 pairs. When looking at the maximum drawdown, the distance methods shows average values compared to the other methods. This means that even though the distance method is able to generate its return in less trades, all the other measures show that the method is being outperformed by its parametric alternatives, which takes away the value of the small number of trades.

The 9.88% return of the SSM for the top 5 comes close to the return of the RNN, which is 11.07%. However, as already noted, the number of trades is considerably larger than for the RNN (384 vs. 51), which makes the BETC of the SSM considerably smaller. Also, the SSM is unable to keep up its performance in terms of return when more assets are involved, while the number of trades increases quite rapidly as well. Even though the SSM attempts to capture the mean-reverting characteristics of the spread, it gives an investor no reason to use it, when the RNN is able to do the same in a better and more consistent way. The same holds for the SRSM, for which the return remains stable as the number of pairs increases but the number of trades increases dramatically compared to the other methods. Despite the return, it is fair to state that taking into account the risk factors and subsequently using the trading rule of Elliott et al. (2005), will result in little to no profit after transaction costs have been taking into account.

The large number of trades is mainly caused by the threshold of the spread methods being relatively small compared to the two times sigma rule of Gatev et al. (1999). This is shown in table 12 and illustrated in figures 9-11 in Appendix G. As presented in table 12, the average relative threshold increases as the number of pairs increases. From the increasing threshold would be expected that the number of trades would not increase a lot (note that the number of trades will always increase because more pairs are used), since the threshold will be triggered less. However, in particular for the SRSM, as the number of pairs increases, the smaller thresholds trigger far more trades than for a smaller number of pairs, which explains the large increase in number of trades. For the both the FNN and RNN, the relative threshold is considerably smaller than for the SRSM and SSM. However, the threshold is used differently than for the spread methods, since a trade is triggered if the predicted change of the spread exceeds the threshold. Therefore, as shown by the figures, the results are different. The results show that the number of trades remains small independent of the value of the relative threshold.

Whereas the RNN is the best performing method, the FNN provides rather disappointing results. The returns of the FNN are lower than the ones of the RNN, SRSM, SSM for all the different situations, and also lower than the distance method in case of 20 and 50 pairs. Because of the relative low number of trades the FNN needs, especially compared to the SSM and SRSM, it has the third highest BETC in all cases. When looking at the Sharpe and Sortino ratios, and the maximum drawdown, the FNN is able to perform best in case of the top 5 pairs, where it generates the largest ratios of all methods. Using more pairs, the other methods are able to outperform the FNN in most cases. Furthermore, the maximum drawdown of the FNN is consistently the lowest. However, due to the lack in return as well as the rather disappointing ratios when using more pairs, the use of the FNN is not beneficial to pairs trading compared to the more traditional methods. This shows that the benefits of the RNN over the FNN, that is, the ability to have a long-short term memory, make it a more powerful and more profitable method for pairs trading.

Summarizing our findings over the entire trading period, the RNN proved to be the best performing method in terms of returns, BETC, Sharpe and Sortino ratio. Its maximum drawdown is somewhat larger than the other methods, but this does not outweigh the other positive results. This shows that the use of machine learning techniques, in particular the RNN, is beneficial for pairs trading.

4.3 Statistical Interference of Sharpe Ratio

As shown in section 4.2, the RNN is the best performing method in absolute terms of return and economic measures. To be able to make statistical interference on these results, the significance of the difference in Sharpe ratios is evalatuated using the bootstrap two-sided test proposed by Ledoit & Wolf (2008), as discussed in section 2.4.5. Table 5 shows the differences between the Sharpe ratios reported in table 11 as well as the significance of the differences.

One of the most notable results in table 5 is that the difference in Sharpe ratio between the RNN and the other methods is only significant in a few cases. When using the top 5 pairs, the RNN does not significantly outperform the other methods. For the top 10 pairs, the RNN significantly outperforms the distance method, the SSM and the SRSM, and is still not outperformed by any of the other methods. The RNN also significantly outperforms the distance method and FNN when using the top 20 pairs. Since none of the methods is able to outperform the RNN, and the RNN outperforms some of the other methods for a certain number of pairs, the results show that the RNN is once again the best performing method. Also, no investor would choose a different method with a higher, but not significantly higher, Sharpe ratio than the RNN, when the RNN is able to generate a much larger return for a comparable amount of risk.

Table 5: **The difference and corresponding significance in Sharpe ratios.** The difference and corresponding significance in Sharpe ratios between the methods (table 11). The Sharpe ratio of the method in the column was deducted from the Sharpe ratio in the row. If a cell is colored gray, the difference is statistically significant at a significance level of 5%.

		DM	SSM	SRSM	FNN	RNN
Top 5	DM	0.00	-0.11	-0.01	-0.20	-0.08
	SSM	0.11	0.00	0.09	-0.09	0.03
	SRSM	0.01	-0.09	0.00	-0.19	-0.07
	FNN	0.20	0.09	0.19	0.00	0.12
	RNN	0.08	-0.03	0.07	-0.12	0.00
Top 20	DM	0.00	-0.19	-0.17	-0.06	-0.24
	SSM	0.19	0.00	0.02	0.13	-0.06
	SRSM	0.17	-0.02	0.00	0.12	-0.07
	FNN	0.06	-0.13	-0.12	0.00	-0.19
	RNN	0.24	0.06	0.07	0.19	0.00
Top 50	DM	0.00	-0.21	0.19	-0.13	-0.27
	SSM	0.21	0.00	0.40	0.07	-0.07
	SRSM	-0.19	-0.40	0.00	-0.32	-0.46
	FNN	0.13	-0.07	0.32	0.00	-0.14
	RNN	0.27	0.07	0.46	0.14	0.00

The results in section 4.2 showed that the distance method was outperformed in absolute terms of returns and economic measures by the other methods, leading to the suspected conclusion that the distance method is indeed an outdated method. Table 5 confirms this suspicion, since the distance method is in most cases significantly outperformed in terms of the Sharpe ratio. The methods unable to significantly outperform the distance method are respectively the SRSM and RNN for the top 5, the FNN for the top 20 and 50. The only exception is for the top 50 pairs, where the distance method is able to outperform the SRSM. Besides these cases, the distance method is always significantly outperformed, from which can be concluded that the other, parametric, methods indeed surpass the distance method.

Furthermore, the SSM and SRSM are only able to significantly outperform the distance method in most of the cases, while the latter is significantly outperformed by the FNN when using the top 5 pairs and all the other methods using the top 50. Besides these significant differences, the SSM and SRSM do not outperform and are not outperformed by any of the methods. The FNN is also able to significantly outperform the distance method for the top 5 pairs, while it is, as discussed, able to significantly outperform the SRSM in case of the top 20 and 50.

Because of the performance of the RNN, the few cases the SSM, SRSM and FNN outperform other methods than the distance method, it can still be concluded that the

RNN is the best performing method.

4.4 Subperiod Analysis

Where section 4.2 evaluates the performance of the methods over the entire period and finds that RNN is the best and most consistent performer, this section elaborates on the performance over three different subperiods of the trading period. As discussed in 3, it could be the case, that the performance per method differs over time, for example caused by a more or a less volatile market. Table 6 and table 7 show the returns, monthly volatility, number of trades and BETC, and the economic measures respectively, based on the subperiods defined in section 3.

As can be seen in table 6, the RNN generates most of its return in the first subperiod independent of the number of pairs used. Thus the most volatile state in the market is where the most profit is made, as can be explained by the spread deviating more often from its mean, which raises more trading opportunities. The return in the first subperiod slightly decreases as the number of pairs increases, while it is the aim to generate more return when more assets are used. For the second subperiod, the return increases with a considerable amount, from 1.95% to 4.03%, as the number of pairs increases. The number of trades in the third subperiod is the smallest for all the number of pairs, which is once again as expected, since this period exhibits small volatility. When using the top 50 pairs, the return shows a large decrease compared to the a smaller number of pairs, which shows that more unprofitable trades have been made in that period Overall, excluding some exceptions, the RNN is (one of) the best performing methods in terms of return in every subperiod independent of the number of pairs, which confirms and partially explains the findings in section 4.2.

Table 7 shows that the Sharpe and Sortino ratios in the first subperiod are considerably larger than for the second and third subperiod for the top 5 and 20 pairs. The ratios of the second subperiod are similar to the ones of the first subperiod when using the top 50 pairs, which can also be explained from the returns being of similar values. Also, the ratios increase as the number of pairs increases. Since it was shown earlier that the return remain equal or slightly decrease, the increase in ratios can only be caused by a decrease in the (downside) volatility. The same holds for the second and third period, where the increase in ratios is larger than the increase in returns in table 6. As found in section 4.2, the volatility decreases over the entire period as the number of pairs decreases in all subperiods as well.

Despite the excellent returns, Sharpe and Sortino ratios, the RNN exhibits the largest maximum drawdown of all methods in most cases. The maximum drawdown decreases for all periods as more pairs are used, which is due to the decrease in volatility and therefore

Table 6: **Excess return, monthly volatility, number of trades and BETC over the entire trading period using the 5, 20 and 50 best performing pairs in formation period.**

		Top 5				**Top 20**				**Top 50**			
		Return	σ	Trades	BETC	Return	σ	Trades	BETC	Return	σ	Trades	BETC
DM	Period 1	0.51%	0.78%	11	0.047%	-0.13%	0.60%	21	-0.006%	0.97%	0.47%	49	0.020%
	Period 2	1.26%	0.71%	14	0.090%	3.20%	0.66%	29	0.110%	2.59%	0.57%	64	0.040%
	Period 3	0.76%	0.31%	2	0.382%	0.88%	0.43%	15	0.058%	0.72%	0.30%	47	0.015%
SSM	Period 1	2.00%	1.37%	115	0.017%	1.82%	0.71%	389	0.005%	0.46%	0.37%	924	0.000%
	Period 2	7.97%	2.07%	130	0.061%	2.33%	0.58%	392	0.006%	0.96%	0.30%	997	0.001%
	Period 3	-0.22%	0.53%	145	-0.002%	2.53%	0.42%	478	0.005%	2.97%	0.21%	1,203	0.002%
SRSM	Period 1	-0.55%	1.35%	141	-0.004%	1.25%	0.66%	636	0.002%	0.53%	0.56%	1,667	0.000%
	Period 2	0.61%	1.27%	125	0.005%	3.13%	0.57%	752	0.004%	2.05%	0.57%	1,850	0.001%
	Period 3	5.02%	0.78%	215	0.023%	1.97%	0.47%	1,008	0.002%	-0.09%	0.32%	2,454	0.000%
FNN	Period 1	0.25%	0.36%	36	0.007%	1.25%	0.54%	98	0.013%	1.33%	0.36%	240	0.006%
	Period 2	0.73%	0.43%	49	0.015%	0.77%	0.48%	106	0.007%	1.42%	0.29%	312	0.005%
	Period 3	2.40%	0.36%	60	0.040%	1.86%	0.36%	102	0.018%	0.97%	0.23%	334	0.003%
RNN	Period 1	6.87%	2.01%	25	0.275%	6.08%	1.04%	143	0.043%	4.45%	0.70%	315	0.014%
	Period 2	1.40%	1.95%	19	0.074%	2.69%	0.90%	100	0.027%	4.03%	0.62%	274	0.015%
	Period 3	2.49%	1.40%	7	0.356%	3.23%	0.82%	67	0.049%	1.07%	0.40%	191	0.006%

Table 7: **Sharpe ratio, Sortino ratio and maximum drawdown over the entire trading period using the 5, 20 and 50 best performing pairs in formation period.**

		Top 5			Top 20			Top 50		
		$\mathcal{SR}$	$\mathcal{SRT}$	$\mathcal{MDD}$	$\mathcal{SR}$	$\mathcal{SRT}$	$\mathcal{MDD}$	$\mathcal{SR}$	$\mathcal{SRT}$	$\mathcal{MDD}$
DM	Period 1	0.08	0.11	1.84%	-0.03	-0.03	1.72%	0.26	0.33	1.08%
	Period 2	0.20	0.26	0.94%	0.61	0.89	0.75%	0.55	0.73	0.88%
	Period 3	0.29	0.34	0.32%	0.30	0.41	0.58%	0.37	0.45	0.69%
SSM	Period 1	0.18	0.25	2.45%	0.32	0.45	1.45%	0.15	0.20	0.77%
	Period 2	0.48	0.66	3.10%	0.56	0.80	0.98%	0.39	0.48	0.45%
	Period 3	-0.07	-0.08	1.33%	0.70	0.94	0.66%	1.68	2.26	0.16%
SRSM	Period 1	-0.05	-0.07	3.84%	0.24	0.32	1.29%	0.12	0.15	1.35%
	Period 2	0.08	0.10	2.47%	0.76	1.07	0.78%	0.46	0.56	1.17%
	Period 3	0.75	1.15	1.94%	0.42	0.57	1.15%	-0.14	-0.15	1.67%
FNN	Period 1	0.09	0.10	1.03%	0.29	0.39	1.07%	0.47	0.60	0.48%
	Period 2	0.20	0.25	0.94%	0.25	0.33	0.76%	0.68	0.91	0.38%
	Period 3	0.76	1.00	0.62%	0.58	0.73	0.42%	0.48	0.53	0.24%
RNN	Period 1	0.42	0.64	3.71%	0.72	1.10	1.87%	0.79	1.18	1.21%
	Period 2	0.08	0.11	5.40%	0.38	0.50	2.32%	0.81	1.19	0.70%
	Period 3	0.20	0.31	1.96%	0.46	0.67	1.41%	0.34	0.43	0.58%

a smaller probability that a (large) drawdown will occur. However, the large return and Sharpe and Sortino ratios compensate for the drawdown, as the drawdown is smaller than the difference in return with the other methods.

Furthermore, the large returns of the RNN results in a relatively large amount of BETC when using the top 5 pairs compared to the other methods. As the return remain the same or decrease slightly and the number of trades increases when more pairs are used, the BETC decrease. However, for the top 20 pairs, the BETC are still of a relatively high value. Similarly to the findings for the entire period, the only method able to keep up with the BETC of the RNN is the distance method, which even has better BETC for the top 20 and top 50 pairs. Still, the returns of the RNN are considerably larger than the distance method. Also, the returns of the distance method for each period are inconsistent and even negative at times as the number of pairs used changes. This once again shows that the distance method, an intuitive non-parametric framework, is not as profitable as a parametric, more econometric based method such as the RNN independent of the state of the market.

For the top 5 and top 20 pairs, the FNN generates most of its return in the third subperiod, the least volatile period of the market. Similar to the return in the first subperiod of RNN, as the number of pairs increases, the return of FNN in the third subperiod decreases from 2.40% to 0.97%. The returns in the first and second subperiod

both increase with a considerable amount, showing that the performance in the more volatile period increases as the number of pairs increases. Still, the FNN is outperformed in terms of returns in every subperiod by the RNN as well as for most periods by the other methods, as found for the entire trading period.

The returns exhibit quite low volatility, which results in relatively high Sharpe and Sortino ratios in table 7. Also, the ratios are considerably larger for the second and third subperiod when using the top 5 pairs (Sortino: 0.25 and 1.00), than the ratios of the RNN, where for example the Sortino ratios are equal to 0.11 and 0.31 respectively for the top 5 pairs. However, as discussed, the returns of FNN are relatively small and are likely not able to exceed the transaction costs, which makes the high ratios less valuable than they could have been.

Furthermore, the SSM and SRSM show quite inconsistent returns over the subperiods as the number of pairs changes. Also, the number of trades is relatively large and increases dramatically for both methods, decreasing the BETC to around 0.001%, that both methods will likely not be profitable after transaction costs. The SSM even generates a negative return of 0.22% in the third period when using the top 5 pairs, while the SRSM generates a return of -0.55% in the first subperiod. In the other cases, the returns in these subperiods when using a different number of pairs, take on values of around 0.5% to 3%, which even more shows the inconsistency of the methods.

As a result of the negative returns, the SSM and SRSM return negative ratios for the top 5 pairs in respectively the third and first subperiod, while the distance method has a negative ratio for the top 20 pairs in the first subperiod. Notable of these negative ratios is that the Sortino ratio is equal or less than the Sharpe ratio, which means there is more downside risk involved, which makes sense in case of negative returns.

One of the most outstanding Sharpe and Sortino ratio is generated by the SSM. When using the top 50 pairs, it is able to generate a Sharpe ratio of 1.68 and a Sortino ratio of 2.26. Compared to the other methods, these values are quite extreme. Also, a ratio approach or exceeding 2.0 are considered to be excellent. The high ratios are once again caused by a return (2.97%) with a very low volatility (0.21%) and as it appears, even less downside risk.

As concluded in section 4.2, the RNN performs better and more consistently than the other methods. Also, it shows the behaviour that we would expect from a pairs trading method, as it generates more return and better economic measures in a volatile period than in a less volatile period.

4.5 Performance per sector

In table 8, the return per sector and weights of the sector in the portfolio are shown for the RNN. Analyzing the returns for the top 5 pairs, it is clear to see that the large return is

Table 8: **The returns and weights in the portfolio per sector for the RNN.** Note that the sectors not shown in the table have no pairs among at most the top 50 pairs.

	Top 5		**Top 20**		**Top 50**	
	Return	Weights	Return	Weights	Return	Weights
Asia	15.55%	60.00%	20.18%	25.00%	20.18%	10.00%
Biotech					10.20%	4.00%
Bonds			2.00%	15.00%	1.16%	28.00%
Commodities					52.46%	2.00%
Consumer			5.63%	10.00%	6.02%	8.00%
Energy					37.50%	2.00%
Europe					-1.87%	2.00%
Large Cap			5.29%	25.00%	5.25%	16.00%
Metals			89.76%	5.00%	81.52%	4.00%
Mid Cap	5.22%	20.00%	5.22%	5.00%	5.22%	2.00%
Small Cap	3.49%	20.00%	3.21%	15.00%	2.79%	22.00%

mainly caused by the pairs within the Asia sector, in which a return of 15.55% is generated. Also for the top 20 and 50, the Asia sector is responsible for the largest part of the return, taking the weight in the portfolio into account. The return in the Asia sector keeps increases when using 20 pairs instead of 5, which means that more assets with a positive return in the trading period are among the top 20 pairs for the RNN. However, because of the portfolio construction, go short (long) one dollar in the overvalued (undervalued) stock, the weight of the Asia sector in the portfolio decreases. Furthermore, for the top 20 and 50 assets, RNN generates a return of respectively 89.76% and 81.52% in the metals sector, but due to the weight of 5% and 4% in the entire portfolio, the effect of this return is small. This means that the fund allocation in the portfolio is rather sub-optimal. This issue could possibly be solved by using a mean-variance portfolio (Markowitz & Todd, 2000), which allocates the funds of the portfolio based on the historical mean and variance of the assets, in this case the pairs. This would in the case of pairs trading mean that for the better pairs identified in the mean-variance portfolio, an investor goes long and short in the corresponding assets for a larger amount. Despite this drawback, the RNN is still able to generate a sufficiently large return over the trading period and forms a good starting point in optimizing the portfolio further.

5 Conclusion

This thesis examines the use of feedforward and recurrent neural networks for pairs trading using ETFs following the approach of Dunis et al. (2006, 2015). The neural networks are compared based on their economic performance to the more traditional methods that capture the characteristics of the mean reversion in the spread, which are stochastic spread method (Elliott et al., 2005) and stochastic residual spread method (Do et al., 2006). The main purpose of this thesis was to find an answer the following research question:
To what extent does pairs trading in the ETF market benefit from applying machine learning methods compared to the more traditional methods?

As presented in section 4, the RNN is able to consistently generate a return of around 11%, which is at least 5 to 6% more than the traditional methods. Furthermore, the RNN is able to significantly outperform the traditional methods in some cases in terms of the Sharpe ratio, whereas the other methods are unable to outperform the RNN, which shows that the RNN is able to generate its returns with a relatively low volatility. The values of the Sortino ratio are (among) the highest of the all methods for the RNN as well. Since these values are also higher than the Sharpe ratios, it indicates that the returns of the RNN exhibit relatively small downside volatility. The main drawback of the RNN is that its maximum drawdown is the largest in all cases, however, this is offset by the high value of the returns. Subperiod analysis shows that the RNN generates most of its return in the most volatile period. When the ETF market becomes less volatile, the performance of the RNN decreases, as expected from a pairs trading point of view. Whereas the RNN is the best performing method, the FNN lacks in terms of return, since it only generates a return of around 3%. Still, the FNN shows a proper performance when looking at the other methods, however, this does not make up for the low returns.

All together, the good performance of the RNN shows that its ability to remember and use long-term dependencies is beneficial for pairs trading and results in more return and higher (significant) economic measures than the other, traditional, methods. The FNN is unable to keep up with the RNN in terms of returns, but shows a similar, but worse, performance for the economic measures. Therefore, it is safe to say that the RNN is able to perform better than the other methods and is consequently more beneficial to pairs trading.

The research can be extended in the following ways. Dunis et al. (2006, 2015) use several thresholds in their research. We decided to choose the threshold filter, since Dunis et al. (2006) showed that this was the best performing filter. Since their research is limited to a specific spread, it could be the case that one of the other filters is more

profitable in the ETF market and could possibly improve the performance of the FNN as well. Furthermore, as we saw in 4.5, going long and short for one dollar in each pair seems to be a rather suboptimal way of optimizing the profit. Therefore, a possible extension is to allocate the total amount using a mean-variance approach (Markowitz & Todd, 2000), which takes into account the historical risk and return of a pair for the portfolio construction, thus go long and short for larger amount in what seems to be a more profitable pair.

References

Avellaneda, M., & Lee, J.-H. (2010). Statistical arbitrage in the US equities market. *Quantitative Finance*, *10*(7), 761–782.

Balkin, S. D. (1997). *Using recurrent neural networks for time series forecasting.*

Brezak, D., Bacek, T., Majetic, D., Kasac, J., & Novakovic, B. (2012). A comparison of feed-forward and recurrent neural networks in time series forecasting. *2012 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr).*

Caldeira, J., & Moura, G. V. (2013). Selection of a Portfolio of Pairs Based on Cointegration: A Statistical Arbitrage Strategy.

Carhart, M. (1997). On Persistence in Mutual Fund Performance. *Journal of Finance*, *52(1)*, 57–82.

Cummins, M., & Bucca, A. (2012). Quantitative Spread Trading on Crude Oil and Refined Products Markets. *Quantitative Finance*, *12*(12), 1857–1875.

D'Aspremont, A. (2011). Identifying small mean-reverting portfolios. *Quantitative Finance*, *11*(3), 351–364.

Dickey, D. A., & Fuller, W. A. (1979). Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *Journal of the American Statistical Association*, *74*(366), 427–431.

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

Do, B., & Faff, R. (2010). Does simple pairs trading still work? *Financial Analysts Journal*, *66*(4), 83–95.

Do, B., Faff, R., & Hamza, K. (2006). A new approach to modeling and estimation for pairs trading. *Proceedings of 2006 Financial Management Association European Conference*, *Monash Uni*, 31.

Doering, P. (2017). Relative Pricing Efficiency in the ETF Markets – Evidence from Pairs Trading.

Dunis, C. L., Giorgioni, G., Laws, J., & Rudy, J. (2010). Statistical Arbitrage and High-Frequency Data with an Application to Eurostoxx 50 Equities. *Review Literature And Arts Of The Americas*, 1–31.

Dunis, C. L., Laws, J., & Evans, B. (2006). Modelling and trading the gasoline crack spread: A non-linear story. *Derivatives Use, Trading & Regulation*, *12*(1), 126–145.

Dunis, C. L., Laws, J., Middleton, P. W., & Karathanasopoulos, A. (2015). Trading and hedging the corn/ethanol crush spread using time-varying leverage and nonlinear models. *The European Journal of Finance*, *21*(4), 352–375.

Elliott, R. J., van der Hoek, John, & Malcolm, W. P. (2005). Pairs trading. *Quantitative Finance*, *5*(3), 271–276.

Engle, R. F., & Granger, C. W. J. (1987). Co-Integration and Error Correction: Representation, Estimation, and Testing. *Econometrica*, *55*(2), 251.

Fama, E. F., & French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, *33*(1), 3–56.

Franses, P. H., & van Dijk, D. (2000). *Nonlinear Time Series Models in Empirical Finance*. Cambridge University Press.

Gatev, E., Goetzmann, W. N., & Rouwenhorst, K. G. (1999). Pairs trading: Performance of a relative-value arbitrage rule. *Working paper, Yale School of Management's International Center for Finance*.

Gatev, E., Goetzmann, W. N., & Rouwenhorst, K. G. (2006). Pairs trading: Performance of a relative-value arbitrage rule. *Review of Financial Studies*, *19*(3), 797–827.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780.

Huck, N., & Afawubo, K. (2015). Pairs trading and selection methods: is cointegration superior? *Applied Economics*, *47*(6), 599–613.

Johansen, S. (1991). Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models. *Econometrica*, *59*(6), 1551–1580.

Krauss, C. (2015). Statistical Arbitrage Pairs Trading Strategies: Review and Outlook. *IWQW Discussion Paper Series*.

Krogh, A., & Hertz, J. A. (1992). A simple weight decay can improve generalization. In *Advances in neural information processing systems* (pp. 950–957).

Lam, M. (2004). Neural Network Techniques for Financial Performance Prediction: Integrating Fundamental and Technical Analysis. *Decision Support Systems*, *37*, 567–581.

Ledoit, O., & Wolf, M. (2008). Robust Performance Hypothesis Testing with the Sharpe Ratio. *Journal of Empirical Finance*, *15*, 850–859.

Markowitz, H. M., & Todd, G. P. (2000). *Mean-variance analysis in portfolio choice and capital markets* (Vol. 66). John Wiley & Sons.

Medeiros, M. C., & Teräsvirta, T. (2006). Building Neural Network Models for Time Series : A Statistical Approach. *Journal of Forecasting*, *25*, 49–75.

Nath, P. (2003). High Frequency Pairs Trading with U.S. Treasury Securities: Risks and Rewards for Hedge Funds. *SSRN Electronic Journal*(November 2003).

Olden, M. (2016). Predicting Stocks with Machine Learning.

Pereira Camara Leal, R., & Vaz de Melo Mendes, B. (2005). Maximum Drawdown. *The Journal of Alternative Investments*, *7*(4), 83–91.

Ross, S. A. (1976). The Arbitrage Theory of Capital Asset Pricing. *Journal of Economic Theory*, *13*, 341–360.

Sharpe, W. F. (1994). The Sharpe Ratio. *The Journal of Portfolio Management*, *21*(1), 49–58.

Sipilä, M. (2013). Algorithmic Pairs Trading: Empirical Investigation of Exchange Traded Funds. , 77.

Sortino, F. A., & Price, L. N. (1994). Performance Measurement In a Downside Risk Framework. *Journal of Investing*, *3*(3), 59–64.

Sundermeyer, M., Alkhouli, T., Wuebker, J., & Ney, H. (2014). Translation Modeling with Bidirectional Recurrent Neural Networks Human Language Technology and Pattern Recognition Group. *EMNLP*, 14–25.

Vidyamurthy, G. (2004). *Pairs Trading: Quantitative Methods and Analysis*. John Wiley & Sons.

Appendix

In this section the appendix is given.

A EM Algorithm for the Stochastic Spread Method

Following the approach of Elliot et al. (2005), we know that ξ_t based on $\mathcal{I}_t$ happens to be normal:

$$\xi_t|\mathcal{I}_t \sim \mathcal{N}(\hat{\xi}_{t|t}, P_{t|t}), \tag{A1}$$

where

$$\hat{\xi}_{t|t} = E[\xi_t|\mathcal{I}_t], \tag{A2}$$
$$P_{t|t} = E[(\xi_t - \hat{\xi}_{t|t})(\xi_t - \hat{\xi}_{t|t})'|\mathcal{I}_t]. \tag{A3}$$

In other words, $\hat{\xi}_{t|t}$ is our best estimate, while $P_{t|t}$ denotes its uncertainty. With this knowledge and the given the transition equation in (9), we can formalize the prediction step as follows:

$$\hat{\xi}_{t+1|t} = A + B\hat{\xi}_{t|t}, \tag{A4}$$
$$P_{t+1|t} = BP_{t|t}B' + C^2. \tag{A5}$$

Using the joint normal distribution Lemma, the update equations become:

$$\hat{\xi}_{t+1|t+1} = \hat{\xi}_{t+1|t} + P_{t+1|t}(P_{t+1|t} + H^2)^{-1}(S_{t+1,i,j} - \hat{\xi}_{t+1|t}), \tag{A6}$$
$$P_{t+1|t+1} = P_{t+1|t} - P_{t+1|t} - P_{t+1|t}(P_{t+1|t} + H^2)^{-1}P_{t+1|t}. \tag{A7}$$

For $t \leq N$ and using backward recursion, the smoothing equations are:

$$\hat{\xi}_{t|T} = \hat{\xi}_{t|t} + P_{t|t}B'P^{-1}_{t+1|t}(\hat{\xi}_{t+1|T} - \hat{\xi}_{t+1|t}), \tag{A8}$$
$$P_{t|T} = P_{t|t} - P_{t|t}B'P^{-1}_{t+1|t}(P_{t+1|t} - P_{t+1|T})P^{-1}_{t+1|t}BP_{t|t}, \tag{A9}$$
$$P_{t+1,t|T} = P_{t+1|T}P^{-1}_{t+1|t}BP_{t|t}. \tag{A10}$$

Furthermore, for the state space model, the joint density of observing the states $\xi_{0:T}$ and data $S_{1:T,i,j}$ is given by

$$\begin{aligned}\mathcal{L}(S_{1:T,i,j}, \xi_{0:T}|A, B, C, D) = \frac{T}{2}\log|H^{-2}| - \frac{1}{2}\sum_{t=1}^{T}(S_{t,i,j} - \xi_t)'H^{-2}(S_{t,i,j} - \xi_t) \\ + \frac{T}{2}\log|C^{-2}| - \frac{1}{2}\sum_{t=1}^{T}(\xi_t - A - B\xi_{t-1})'C^{-2}(\xi_t - A - B\xi_{t-1}) \\ + \text{constants}. \qquad \text{(A11)}\end{aligned}$$

Maximizing (A11) with respect to our parameters A, B, C, H assuming states ξ_t are known, subsequently assuming the states are unknown and making use of the Kalman filter and smoother, results in a single complete EM loop to be given by:

$$A = \frac{1}{T}\sum_{t=1}^{T}(\hat{\xi}_{t|T} - F\hat{\xi}_{t-1|T}), \qquad \text{(A12)}$$

$$B = (\sum_{t=1}^{T}\hat{\xi}_{t|T}\hat{\xi}'_{t-1|T} + P_{t,t-1|T})(\sum_{t=0}^{T-1}\hat{\xi}_{t|T}\hat{\xi}'_{t|T} + P_{t|T})^{-1}, \qquad \text{(A13)}$$

$$\begin{aligned}C^2 = \frac{1}{T}\sum_{t=1}^{T}(\hat{\xi}_{t|T}\hat{\xi}'_{t|T} + P_{t|T} - B[\hat{\xi}_{t-1|T}\hat{\xi}'_{t|T} + P_{t-1,t|T}] - [\hat{\xi}_{t|T}\hat{\xi}'_{t-1|T} + P_{t,t-1|T}]B' \\ + B[\hat{\xi}_{t-1|T}\hat{\xi}'_{t-1|T} + P_{t-1|T}]B' - \hat{\xi}_{t|T}A' - A\hat{\xi}'_{t|T} + AA' \\ + A\hat{\xi}'_{t-1|T}B' + B\hat{\xi}_{t-1|T}A), \qquad \text{(A14)}\end{aligned}$$

$$H^2 = \frac{1}{T}\sum_{t=1}^{T}(S_{t,i,j}S'_{t,i,j} - \hat{\xi}_{t|T}S'_{t,i,j} - S_{t,i,j}\hat{\xi}'_{t|T} + [\hat{\xi}_{t|T}\hat{\xi}'_{t|T} + P_{t|T}]). \qquad \text{(A15)}$$

The number of iterations needed before the values converge will mainly depend on the the starting values and stopping criterion of the EM algorithm.

B EM Algorithm for the Stochastic Residual Spread Method

As discussed in section 2.3.1.1, the exogeneous term in (13) changes the dynamics of the estimation of the parameters. Compared to the prediction and updates equations in Appendix A, (A6) changes as follows:

$$\hat{\xi}_{t+1|t+1} = \hat{\xi}_{t+1|t} + P_{t+1|t}(P_{t+1|t} + H^2)^{-1}(S_{t+1,i,j} - \hat{\xi}_{t+1|t} - \Gamma \boldsymbol{r_t^f}). \tag{A16}$$

Except for (A6), (A4) - (A10) remain the same. Note that the conditional variance in the measurement equation is not affected by the extra term, since this term is known at the current time.

Furthermore, the joint density in (A11) changes to:

$$\begin{aligned}\mathcal{L}(S_{1:T,i,j}, \xi_{0:T}|A, B, C, H) = \frac{T}{2}\log|H^{-1}| - \frac{1}{2}\sum_{t=1}^{T}(S_{t,i,j} - \xi_t - \Gamma \boldsymbol{r_t^f})' H^{-1}(S_{t,i,j} - \xi_t - \Gamma \boldsymbol{r_t^f}) \\ + \frac{T}{2}\log|C^{-1}| - \frac{1}{2}\sum_{t=1}^{T}(\xi_t - A - B\xi_{t-1})' C^{-1}(\xi_t - A - B\xi_{t-1}) \\ + \text{constants}. \end{aligned} \tag{A17}$$

Maximizing (A17) with respect to our parameters A, B, Γ, C, H assuming states ξ_t are known, subsequently assuming the states are unknown and making use of the Kalman filter and smoother, results in a single complete EM loop to be given by:

$$A = \frac{1}{T}\sum_{t=1}^{T}(\hat{\xi}_{t|T} - F\hat{\xi}_{t-1|T}), \tag{A18}$$

$$B = (\sum_{t=1}^{T}\hat{\xi}_{t|T}\hat{\xi}'_{t-1|T} + P_{t,t-1|T})(\sum_{t=0}^{T-1}\hat{\xi}_{t|T}\hat{\xi}'_{t|T} + P_{t|T})^{-1}, \tag{A19}$$

$$\Gamma = (\sum_{t=1}^{T}(S_{t,i,j}(\boldsymbol{r_t^f})' - \hat{\xi}_{t|T}(\boldsymbol{r_t^f})'))(\sum_{t=1}^{T}\boldsymbol{r_t^f}(\boldsymbol{r_t^f})')^{-1} \tag{A20}$$

$$\begin{aligned} C^2 = \frac{1}{T}\sum_{t=1}^{T}(\hat{\xi}_{t|T}\hat{\xi}'_{t|T} + P_{t|T} - B[\hat{\xi}_{t-1|T}\hat{\xi}'_{t|T} + P_{t-1,t|T}] - [\hat{\xi}_{t|T}\hat{\xi}'_{t-1|T} + P_{t,t-1|T}]B' \\ + B[\hat{\xi}_{t-1|T}\hat{\xi}'_{t-1|T} + P_{t-1|T}]B' - \hat{\xi}_{t|T}A' - A\hat{\xi}'_{t|T} + AA' \\ + A\hat{\xi}'_{t-1|T}B' + B\hat{\xi}_{t-1|T}A), \end{aligned} \tag{A21}$$

$$\begin{aligned} H^2 = \frac{1}{T}\sum_{t=1}^{T}(S_{t,i,j}S'_{t,i,j} - \Gamma'\boldsymbol{r_t^f}S_{t,i,j} - S_{t,i,j}(\boldsymbol{r_t^f})'\Gamma - \hat{\xi}_{t|T}S'_{t,i,j} - S_{t,i,j}\hat{\xi}'_{t|T} \\ + \Gamma'\boldsymbol{r_t^f}(\boldsymbol{r_t^f})'\Gamma + [\hat{\xi}_{t|T}\hat{\xi}'_{t|T} + P_{t|T}] + \Gamma'\boldsymbol{r_t^f}\hat{\xi}'_{t|T} + \hat{\xi}_{t|T}(\boldsymbol{r_t^f})'\Gamma). \end{aligned} \tag{A22}$$

Compared to the solutions (A12) - (A15), the solution for Γ, (A20), has been added, while the solution for H^2 changed to (A22). Since it is not quite common to have exogenous variables in the measurement equation (Γ is set to zero in general), the derivation of the solution for Γ can be found in Appendix C.

C Derivation of the solution for Gamma

The solution is obtained by maximizing of the joint density in (A17) with respect to $\mathbf{\Gamma}'$ using matrix derivation rules as follows:

$$\begin{aligned}
\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{\Gamma}'} &= -\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}\mathbf{\Gamma}'}\,\mathrm{Tr}\left[H^{-1}\sum_{t=1}^{T}(y_t - \mathbf{\Gamma}'\boldsymbol{r}_t^f - \xi_t)(y_t - \mathbf{\Gamma}'\boldsymbol{r}_t^f - \xi_t)'\right] \\
&= -\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}\mathbf{\Gamma}'}\mathrm{Tr}(H^{-1}\sum_{t=1}^{T}(y_t y_t' - \mathbf{\Gamma}'\boldsymbol{r}_t^f y_t' - y_t(\boldsymbol{r}_t^f)'\mathbf{\Gamma} - \xi_t y_t' - y_t\xi_t' + \mathbf{\Gamma}'\boldsymbol{r}_t^f(\boldsymbol{r}_t^f)' \\
&\quad \mathbf{\Gamma} + \xi_t\xi_t' + \mathbf{\Gamma}'\boldsymbol{r}_t^f\xi_t' + \xi_t(\boldsymbol{r}_t^f)'\mathbf{\Gamma})) \\
&= \frac{1}{2}\sum_{t=1}^{T}[-(\boldsymbol{r}_t^f y_t H^{-1})' - (H^{-1}y_t(\boldsymbol{r}_t^f)') + (H^{-1}\mathbf{\Gamma}'\boldsymbol{r}_t^f(\boldsymbol{r}_t^f)') + (\boldsymbol{r}_t^f(\boldsymbol{r}_t^f)'\mathbf{\Gamma}H^{-1})' \\
&\quad + (\boldsymbol{r}_t^f\xi_t'H^{-1})' + (H^{-1}\xi_t(\boldsymbol{r}_t^f)')]
\end{aligned}$$

Setting the derivative equal to zero gives

$$\begin{aligned}
0 &= \frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\mathbf{\Gamma}'} \\
0 &= H^{-1}\sum_{t=1}^{T}\left[y_t(\boldsymbol{r}_t^f)' - \mathbf{\Gamma}'\boldsymbol{r}_t^f(\boldsymbol{r}_t^f)' - \xi_t(\boldsymbol{r}_t^f)'\right] \\
0 &= \sum_{t=1}^{T}\left[y_t(\boldsymbol{r}_t^f)' - \mathbf{\Gamma}'\boldsymbol{r}_t^f(\boldsymbol{r}_t^f)' - \xi_t(\boldsymbol{r}_t^f)'\right] \\
\mathbf{\Gamma}'\sum_{t=1}^{T}\boldsymbol{r}_t^f(\boldsymbol{r}_t^f)' &= \sum_{t=1}^{T}\left[y_t(\boldsymbol{r}_t^f)' - \xi_t(\boldsymbol{r}_t^f)'\right] \\
\mathbf{\Gamma}' &= \left(\sum_{t=1}^{T}\left[y_t(\boldsymbol{r}_t^f)' - \xi_t(\boldsymbol{r}_t^f)'\right]\right)\left(\sum_{t=1}^{T}\boldsymbol{r}_t^f(\boldsymbol{r}_t^f)'\right)^{-1}
\end{aligned}$$

The solution for a full EM loop, that is including the Kalman filter and smoother equations, can be obtained using that

$$E(\xi_t) = \hat{\xi}_{t|T}$$

which gives the following solution for $\mathbf{\Gamma}'$:

$$\mathbf{\Gamma}' = \left(\sum_{t=1}^{T}\left[y_t(\boldsymbol{r}_t^f)' - \hat{\xi}_{t|T}(\boldsymbol{r}_t^f)'\right]\right)\left(\sum_{t=1}^{T}\boldsymbol{r}_t^f(\boldsymbol{r}_t^f)'\right)^{-1}$$

D On the replication of the neural networks

To program the neural networks, we used the Neural Networks Toolbox in MATLAB. In particular, we used the `feedforwardnet` function for the FNN and the `lstmLayer` for the RNN.

As discussed in section 2.3.2.1, the hidden units, hidden layers and number of lags are determined simultaneously. This means that we created four forloops in our script, where one loops over the pairs, one loops over a number of hidden units, one over a number of hidden layers and one over the number of lags. The loops iterate over a certain set of values, with minimum 1 and the following maximums:

- The maximum number of hidden layers used is three. It is known that in general one or two hidden layers is more than sufficient for time series prediction (Medeiros & Teräsvirta, 2006). To verify that this is the case for our data, we test for three layers as well.
- The maximum number of hidden units used is five, which is a relatively large number resulting in a large network.
- The maximum number of lags used is ten.

This way, all computational achievable and realistic combinations are tested, resulting in an optimal, general architecture in terms of hidden layers, hidden units and lags. The criterion for the optimal, general architecture, that is a specific number of hidden layers, hidden units and number of lags, is to have the smallest average sum of squared errors based on the test set over all pairs. The main motivation behind finding one optimal architecture is the computational complexity when switching between different architectures. It was memory-wise not possible for the computers we used to switch between the architectures for different pairs.

As discussed in section 2.3.2.1 and as can be seen from the results in table 9, we find that the optimal (average) performance is reached for two hidden layers, three hidden units, and five lags. This is the architecture we use for all pairs. Within the architecture, the parameters (weights) are estimated for each pair individually.

Table 9: **The average sum of squared errors for the different scenarios calculated over the test set using the FNN.** As can be seen from the table, the value of the sum of squared is the smallest for two hidden layers, three hidden units and five lags. 'Hidden unit' is denoted by hu. The values in the table need to be multiplied by 10^{-3}.

	1 hidden layer					**2 hidden layers**					**3 hidden layers**				
	1 hu	2 hu	3 hu	4 hu	5 hu	1 hu	2 hu	3 hu	4 hu	5 hu	1 hu	2 hu	3 hu	4 hu	5 hu
1 lag	0.1774	0.1539	0.1720	0.1585	0.1703	0.1738	0.1513	0.1643	0.1643	0.1543	0.1621	0.1534	0.1753	0.1608	0.1565
2 lags	0.1505	0.1496	0.2015	0.1752	0.1806	0.1475	0.1998	0.1893	0.2057	0.1935	0.1532	0.1672	0.1747	0.1629	0.1794
3 lags	0.1479	0.1544	0.1648	0.1961	0.1647	0.1760	0.1717	0.1551	0.1721	0.1517	0.2212	0.1322	0.1766	0.1463	0.1738
4 lags	0.1497	0.1560	0.1674	0.1471	0.1561	0.1513	0.1960	0.1432	0.1899	0.1520	0.1610	0.1401	0.2009	0.1541	0.1581
5 lags	0.1634	0.1473	0.1357	0.1412	0.1569	0.1667	0.1435	0.1269	0.1492	0.1821	0.1820	0.1506	0.1696	0.1598	0.1815
6 lags	0.1699	0.1658	0.1618	0.2010	0.1859	0.1678	0.2034	0.1404	0.1866	0.1611	0.1601	0.1638	0.1857	0.1315	0.1405
7 lags	0.1949	0.1650	0.1423	0.1949	0.1575	0.1589	0.1603	0.1887	0.2108	0.1722	0.1650	0.1675	0.1524	0.1693	0.1754
8 lags	0.1587	0.1650	0.1462	0.1658	0.1579	0.1627	0.1706	0.1730	0.1771	0.1950	0.1756	0.1501	0.1731	0.1940	0.1631
9 lags	0.1857	0.1516	0.1865	0.1616	0.1888	0.1919	0.1497	0.1420	0.1421	0.1650	0.1779	0.2034	0.1927	0.1674	0.1748
10 lags	0.1549	0.1568	0.1530	0.1615	0.1541	0.1614	0.1755	0.1691	0.1596	0.1555	0.1460	0.1864	0.1507	0.1548	0.1958

E Breakdown of the number of ETFs per category

Table 10: **Breakdown of the number of ETFs per category.**

Categories	# of ETFs in category
Asia	39
Biotech	15
Bonds	76
Commodities	16
Consumer	23
Currency	14
Emerging Markets	34
Energy	23
Europe	19
Financial Services	25
Income	23
Industrials	8
International	47
Large Cap	64
Materials	4
Metals	17
Mid Cap	41
Multi Cap	7
Natural Resources	9
Power	12
Real Estate	15
Small Cap	37
Technology	23
Utilities	7

F Significance of monthly returns

Table 11: **Monthly average return, t-statistic and p-value over the entire trading period using the 5, 20 and 50 best performing pairs in formation period.** This table reports the monthly average return (μ_r), t-statistic and p-value for the different methods over the entire trading period, which is 18/08/2015 until 18/08/2017. Top 5, 20 and 50 respectively denotes the number of best performing pairs in the formation period that are used in the trading period.

	Top 5			**Top 20**			**Top 50**		
	μ_r	t-statistic	p-value	μ_r	t-statistic	p-value	μ_r	t-statistic	p-value
DM	0.11%	0.80	0.215	0.17%	1.41	0.087	0.18%	1.91	0.034
SSM	0.40%	1.31	0.101	0.28%	2.30	0.016	0.18%	2.90	0.004
SRSM	0.21%	0.86	0.198	0.26%	2.22	0.018	0.10%	1.01	0.162
FNN	0.14%	1.76	0.046	0.16%	1.67	0.055	0.16%	2.54	0.009
RNN	0.44%	1.18	0.126	0.49%	2.57	0.009	0.39%	3.22	0.002

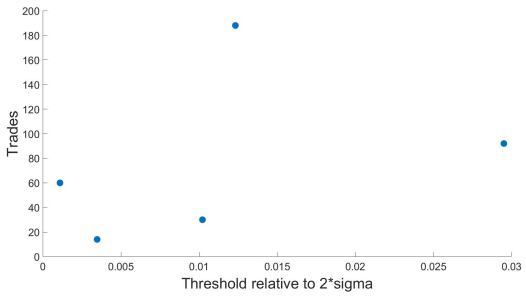
G Evaluation of thresholds

The thresholds $c_{i,j}$ and $X_{i,j}$ are evaluated for the top 5, 20 and 50 pairs. They are compared relatively to the rule of Gatev et al. (1999), which means that the value of the threshold is divided by two times the standard deviation of the spread between i and j. Table 12 shows the average, minimum and maximum threshold for the corresponding number of pairs.

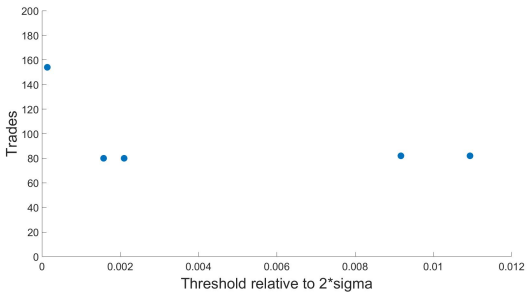
Table 12: **Mean, minimum and maximum threshold relative to the rule of Gatev et al. (1999) for all methods.**

	Top 5			**Top 20**			**Top 50**		
	μ	min	max	μ	min	max	μ	min	max
SSM	0.011	0.001	0.030	0.300	$6.93{\cdot}10^{-4}$	1.650	0.452	$4.90{\cdot}10^{-4}$	2.740
SRSM	0.005	$1.38{\cdot}10^{-4}$	0.011	0.004	$7.50{\cdot}10^{-5}$	0.019	0.008	$7.16{\cdot}10^{-5}$	0.055
FNN	$3.21{\cdot}10^{-5}$	$9.93{\cdot}10^{-5}$	$1.74{\cdot}10^{-4}$	$1.02{\cdot}10^{-4}$	$9.43{\cdot}10^{-6}$	$2.40{\cdot}10^{-4}$	$1.26{\cdot}10^{-4}$	$3.84{\cdot}10^{-6}$	$8.21{\cdot}10^{-4}$
RNN	$2.82{\cdot}10^{-4}$	$3.30{\cdot}10^{-5}$	$6.86{\cdot}10^{-4}$	$6.68{\cdot}10^{-4}$	$2.15{\cdot}10^{-5}$	0.004	$9.78{\cdot}10^{-4}$	$2.15{\cdot}10^{-5}$	0.009

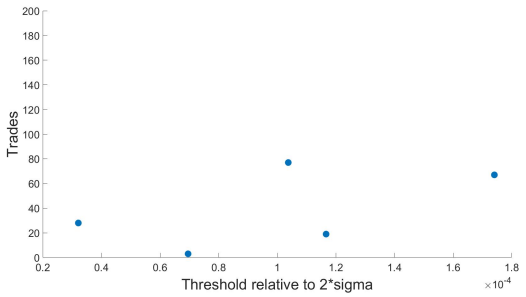
Figure 9, 10 and 11 illustrate the relative threshold against the number of trades for the top 5, 20 and 50 pairs respectively.



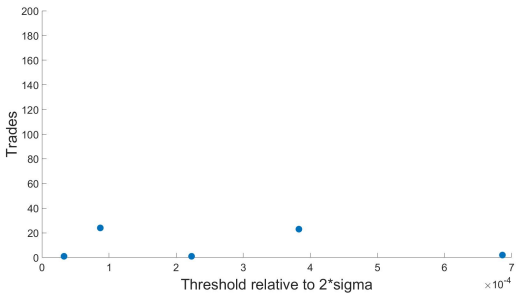
(a) Scatter plot for the SSM



(b) Scatter plot for the SRSM

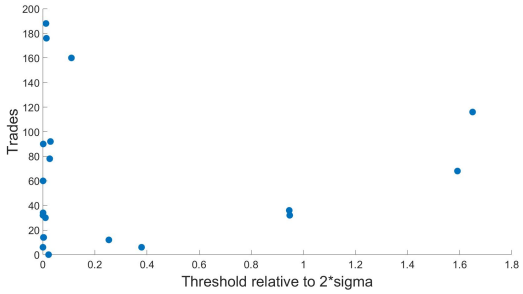


(c) Scatter plot for the FNN

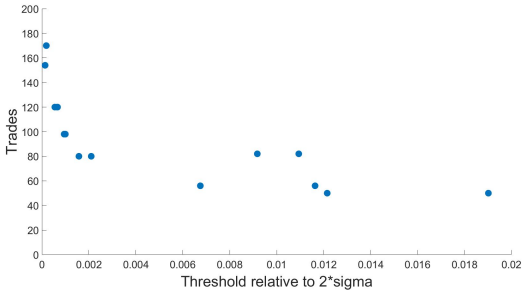


(d) Scatter plot for the RNN

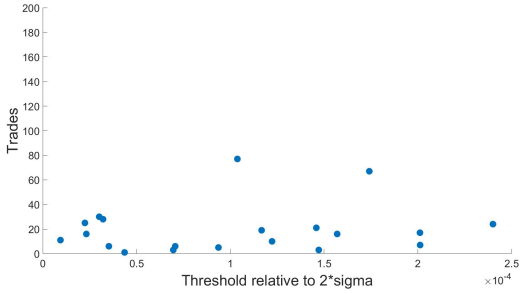
Figure 9: **Scatterplot of the threshold relative to two times the corresponding sigma against the number of trades for the top 5 pairs.**



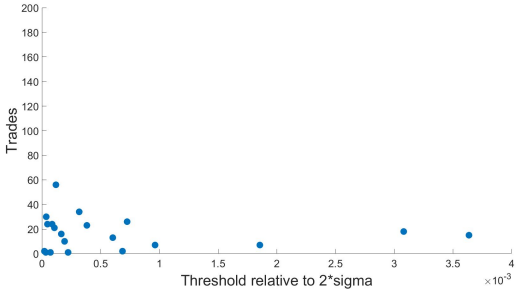
(a) Scatter plot for the SSM



(b) Scatter plot for the SRSM

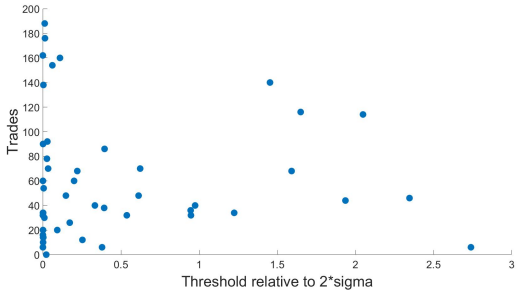


(c) Scatter plot for the FNN

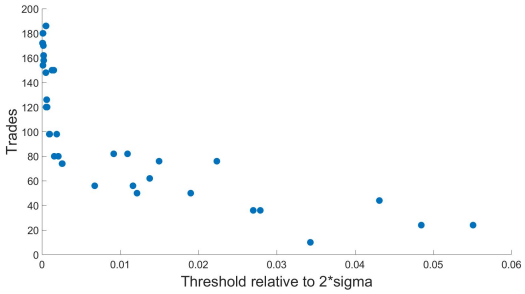


(d) Scatter plot for the RNN

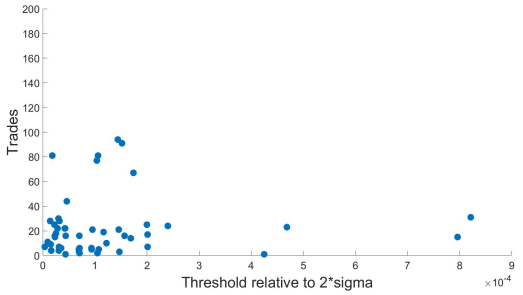
Figure 10: **Scatterplot of the threshold relative to two times the corresponding sigma against the number of trades for the top 20 pairs.**



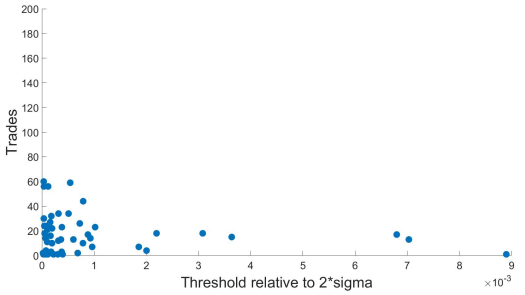
(a) Scatter plot for the SSM



(b) Scatter plot for the SRSM



(c) Scatter plot for the FNN



(d) Scatter plot for the RNN

Figure 11: **Scatterplot of the threshold relative to two times the corresponding sigma against the number of trades for the top 50 pairs.**