Combining empirical mode decomposition with neural networks for the prediction of exchange rates

Jacques Mouton

21635749

Dissertation submitted in partial fulfillment of the requirements for the degree Master of Engineering in Computer and Electronic Engineering at the Potchefstroom Campus of the North-West University

Supervisor: A J Hoffman

November 2014

NORTH-WEST UNIVERSITY
YUNIBESITI YA BOKONE-BOPHIRIMA
NOORDWES-UNIVERSITEIT
POTCHEFSTROOM CAMPUS

Acknowledgements

I would like to express my gratitude to everyone that assisted, supported and guided me throughout this study:

- To Prof. A. J. Hoffman in his capacity as supervisor. Thank you for all the patience, support, advice and wisdom.
- To my parents and family. Thank you for the endless encouragement and love. All your support means the world to me and I love you very much.
- To my fellow students and friends in office 212. Thank you for the true friendship, fun times and interesting conversations.
- To the National Research Foundation. Thank you for the financial support.

Abstract

The foreign exchange market is one of the largest and most active financial markets with enormous daily trading volumes. Exchange rates are influenced by the interactions of a large number of agents, each operating with different intentions and on different time scales. This gives rise to nonlinear and non-stationary behaviour which complicates modelling. This research proposes a neural network based model trained on data filtered with a novel Empirical Mode Decomposition (EMD) filtering method for the forecasting of exchange rates.

One minor and two major exchange rates are evaluated in this study. Firstly the ideal prediction horizons for trading are calculated for each of the exchange rates. The data is filtered according to this ideal prediction horizon using the EMD-filter. This EMD-filter dynamically filters the data based on the apparent number of intrinsic modes in the signal that can contribute towards prediction over the selected horizon. The filter is employed to filter out high frequency noise and components that would not contribute to the prediction of the exchange rate at the chosen timescale. This results in a clearer signal that still includes nonlinear behaviour. An artificial neural network predictor is trained on the filtered data using different sampling rates that are compatible with the cut-off frequency. The neural network is able to capture the nonlinear relationships between historic and future filtered data with greater certainty compared to a neural network trained on unfiltered data.

Results show that the neural network trained on EMD-filtered data is significantly more accurate at prediction of exchange rates compared to the benchmark models of a neural network trained on unfiltered data and a random walk model for all the exchange rates. The EMD-filtered neural network's predicted returns for the higher sample rates show higher correlations with the actual returns, and significant profits can be made when applying a trading strategy based on the predictions. Lower sample rates that just marginally satisfy the Nyquist criterion perform comparably with the neural network trained on unfiltered data; this may indicate that some aliasing occurs for these sampling rates as the EMD low-pass filter has a gradual cut-off, leaving some high frequency noise within the signal.

The proposed model of the neural network trained on EMD-filtered data was able to uncover systematic relationships between the filtered inputs and actual outputs. The model is able to deliver profitable average monthly returns for most of the tested sampling rates and forecast horizons of the different exchange rates. This provides evidence that systematic predictable behaviour is present within exchange rates, and that this systematic behaviour can be modelled if it is properly separated from high frequency noise.

Keywords: Empirical Mode Decomposition (EMD), neural networks, exchange rates, forecasting.

Table of Contents

List of Tables

List of Figures

List of abbreviations

EMD	Empirical Mode Decomposition
IMF	Intrinsic Mode function
ANN	Artificial Neural Network
EMH	Efficient Market Hypothesis
GARCH	General Autoregressive Conditional Heteroskedasticity
ARIMA	Autoregressive Integrated Moving Average
RMSE	Root-mean-square-error
SVR	Support Vector Regression
MAE	Mean-average-error
MAPE	Mean-average-percentage-error
DS	Directional symmetry
BPNN	Backpropagation Neural Network
GRNN	General Regression Neural Network
MIMO	Multiple-input Multiple-output
MASE	Mean-absolute-scaled-error
MI	Mutual Information

1. Introduction

"A great discovery solves a great problem, but there is a grain of discovery in the solution of any problem. Your problem may be modest, but if it challenges your curiosity and brings into play your inventive faculties, and if you solve it by your own means, you may experience the tension and enjoy the triumph of discovery"

- George Polya

1.1 Introduction

Financial modelling aims to improve insight into financial markets. The goal of this dissertation is to make a contribution to the financial modelling domain by creating a model that is able to simulate and predict foreign exchange rates.

Trading on the foreign exchange market averaged $5.3 trillion a day in 2013, making it one of the largest markets in the world [1]. Several agents operate simultaneously on a market of this size, each with its own motivations and time horizons. Every agent, from governments, financial institutions, businesses and intraday traders to long term investors influence the market by their actions, either directly or indirectly. These interactions with the market and each other give rise to extremely complex market interactions that are commonly characterised by nonlinear and non-stationary behaviour [2]. This presents several challenges for an investor wishing to exploit the movements in the exchange rate data series in order to generate returns. The first challenge is the identification of the time horizon on which the most significant returns can be generated if trading costs and data predictability are taken into account. Secondly the noise and data irrelevant to the selected forecast horizon must be removed in order to maximize the signal-to-noise ratio at this time scale. Finally techniques must be selected that are able to consistently model the systematic nonlinear relationships between historic and future behaviour.

Time series analysis traditionally seeks for a suitable model to fit the data; this is complicated by the fact that the data is typically non-stationary, with non-linear relationships between past and future values and behaviour occurring simultaneously at different time scales. Empirical Mode decomposition (EMD) is a technique designed to decompose a signal into its intrinsic modes [3], with each mode constrained to a limited frequency band and has seen wide usage in the area of financial analysis. What makes EMD attractive in financial analysis is that it is an empirically based technique that is *a posteriori* and adaptive, allowing the data to speak for itself. No *a priori* assumptions are required, as is the case with traditional time-frequency techniques such as Fourier or wavelet analyses. The time-frequency components obtained from EMD can simplify this task by allowing one to investigate the series for one intrinsic mode function (IMF) at a time and over time horizons that are optimal for the respective IMFs. While EMD is traditionally used to analyse the individual modes of a time series, usage of the technique as a filter has also been identified [4], [5]. An advantage of EMD-filtering is that the data still retains its nonlinearity and non-stationarity, which is not the case when using conventional filtering techniques.

While EMD filtering can be used to separate the systematic behaviour of the time series from noise at the selected time scale, a technique is still required to model future behaviour based on the historic values of the filtered signal. Artificial Neural Networks (ANN's) are a widely used machine learning technique that simulates the structure of a biological neural network in order to model arbitrary relationships between a set of inputs and a set of outputs. The structure of the neural network consists of nodes distributed across input, hidden and output layers, connected by weighted connections and activation functions [6]. This structure gives neural networks the built-in property to identify nonlinear relationships between input and output variables, making it ideal for application to nonlinear domains such as financial prediction.

This research proposes an ANN model applied to data filtered with a novel EMD-filtering technique for multi-step prediction of foreign exchange rates. The purpose of the prediction will be to maximize the returns of an investor by identifying the most exploitable forecast horizon and the optimal sampling period using empirical methods. The input data forming part of the training set will be filtered to improve the signal to noise ratio for the selected forecast horizon at the appropriate time scale. The EMD-filtered ANN model will be tested on multiple exchange rates and will be compared with an ANN applied to unfiltered data as well as to a random walk model in terms of accuracy of predictions and simulated returns on an investment.

1.2 Problem Statement

The problem investigated in this dissertation involves the analysis and prediction of exchange rate time series with the goal of exploiting the predictability in order to consistently generate positive returns. A prediction model must be designed and implemented in order to accomplish this task. This model must possess the following characteristics to accurately predict the exchange rates:

- Operate on timescales that are optimal for the generation of significant returns on investment.
- Be able to increase data clarity for a chosen timescale by dynamically filtering out noise and data not contributing to the predictions.
- Be able to capture the non-linear relationships between past and future exchange rate data.

The problem can be stated into a hypothesis:

- **H_0**: The returns generated by the proposed model that incorporates dynamic filtering (EMD-based filters) and nonlinear modelling (ANN's) are not significantly higher than the returns generated by benchmark models in which these techniques are not incorporated.

1.3 Research Objectives

The objective of this dissertation is to document the research concerning the design, implementation and testing of a foreign exchange rate model that can accurately predict exchange rates in a manner that is exploitable to an investor. This is done in several stages.

The first stage is a study of the background regarding the foreign exchange market, neural networks and empirical mode decomposition. The purpose of the study is to understand the problem domain of exchange rate prediction as well as the mathematical properties and characteristics of the techniques that will be used. The study is conducted while taking into consideration the potential application to the forecasting of financial time series.

The second stage is a detailed literature survey that investigates the literature of the fields crucial to this research. These fields are the prediction of financial time series using EMD based techniques and prediction of exchange rates in both the financial and computational intelligence domains. Based on this thorough survey it is possible to make informed decisions regarding the structure of the proposed model and the choice of performance evaluation criteria and benchmark models.

The next stage entails the design of the proposed EMD-filtered ANN model. The methodology is completed in several steps. Step one is the determination of optimal forecast horizons and trading times for the different exchange rates. Step 2 is the design and implementation of the EMD-filter, as well as the determination of ideal filter cut-off periods. Step 3 is the design of the neural network. This includes determination of the architecture, including the number of input and hidden neurons as well as the training procedure. The benchmark models and performance evaluation criteria are also designed as part of this stage.

Stage four is the implementation, verification and validation of the model components, the completed model as well as the results generated by the models. Verification of the EMD-filter is completed in order to gauge the impact that the filtered training data will have on the prediction results. Verification of the neural network ensures that the neural network is sufficiently regularised after training. A comparative performance analysis is conducted between the forecasting results of the proposed and benchmark models using a predefined set of performance evaluation criteria. Finally the generated returns are validated in order to test the null-hypothesis mentioned in section 1.2.

The dissertation is concluded by the final chapter in which the detailed observations are summarised and recommendations are made.

1.4 Research limitations

The research contains some limitations:

- Only a small number of exchange rates are used in order to restrict the required amount of analyses and simulations to the time and resources that were available for this study. The exchange rates chosen may however not be representative of all exchange rates as each one could contain unique underlying dynamics.
- Only historical exchange rate data is used in this research. Other macro- and micro-economic indicators may hold additional explanatory power, but determining which additional indicators to incorporate into the models is beyond the scope of this research.
- Performance evaluation is completed with historical back testing, with the design of the simulations restricting access during training of the models to data that will not be available at the theoretical time of forecast. The limitation is that the behaviour of exchange rates may not be stationary. Underlying characteristics may change over time, diminishing the performance of the model.
- An empirical approach is followed in the design of the model. The contribution to the theoretical knowledge in the field of financial prediction is limited, but it still contributes knowledge to the understanding of underlying exchange rate behaviour and the design of empirical models.

1.5 Dissertation structure

This rest of the dissertation is organised as follows:

Chapter 1, which has already been covered, presents an introduction to the problem of exchange rate modelling and the challenges faced by such models. A precise statement of the problem is included, as well as the objectives of this research.

Chapter 2 provides some background information that is essential to the understanding of this dissertation. Background information is provided on three topics: the foreign exchange market, neural networks and empirical mode decomposition (EMD). The foreign exchange market is discussed in order to present an introduction to the domain in which the problem has been identified. Neural networks and EMD are presented as techniques with which modelling can be accomplished.

Chapter 3 provides a review of literature relevant to foreign exchange rate prediction. This chapter is divided into three subsections: prediction of exchange rates in the financial domain, prediction of exchange rates in the in the computational intelligence domain and financial prediction and analyses using EMD.

Chapter 4 discusses the methodology and analyses which were used in order to design the EMD-filtered ANN model. This chapter includes the collection of exchange rate data, data pre-processing, selection of time scales, neural network design and the choice of comparative models and performance evaluation criteria.

Chapter 5 provides the results of the research. Verifications of the EMD-filter, neural network and composite model are done separately. The results of the EMD-filtered model are also validated using 11 fold cross validation and two-sample t-tests.

Chapter 6 concludes the dissertation with a summary of the observations made in this research and gives recommendations on possible related future research that can be conducted.

2. Background

"An investment in knowledge still yields the best returns"

- Benjamin Franklin

2.1 Introduction

The analysis and prediction of financial time series have seen much attention since the founding of structured financial markets, and for good reason. With a projected knowledge of future behaviour it is possible to accomplish a wide variety of tasks. Maximization of returns on investments, minimization of risk and optimization of asset allocations are a few of the reasons why accurate modelling of financial time series is important. The foreign exchange market, which is one of the largest financial markets in the world, offers plenty of opportunities for investors to exploit the price movements in order to generate a profit. This is partly due to a huge amount of independent agents executing trades at different time scales.

This chapter aims to provide an overview of several topics that are important in this research. The problem domain of the foreign exchange market is discussed, providing information on the advantages, hindrances and approaches to foreign exchange trading. Background information is then given on neural networks, which are commonly used for modelling of nonlinear systems such as exchange rates. Finally the technique of empirical mode decomposition is discussed. The discussion includes the method by which a signal is decomposed, comparisons with other time-frequency techniques and filtering using EMD.

2.2 Foreign exchange market

The foreign exchange market is one of the largest markets in the world, with trading averaging $5.3 trillion per day in 2013 [1]. Trading on this market involves selling a specific currency and buying another at a specified rate, which is called the foreign exchange rate. Exchange rates are free floating and thus determined by a plethora of influences including the economic state of both countries of the currency pair, political events and even the psychological behaviour of investors. All the agents interact with the market on different time scales creating complex nonlinear behaviour, the prediction of which is a nontrivial matter. It is thus important to be able to model this behaviour at different time scales. This section gives an overview of foreign exchange rate prediction, the prediction approaches used and the difficulties that the nonlinear nature of the data gives rise to.

2.2.1 Reasons to predict foreign exchange

Almost all international business practices are influenced by exchange rates. Some of the reasons to predict exchange rates are listed below [7]:

- Buying and selling at the appropriate times in order to profit from fluctuations at different time scales.
- In order to protect foreign assets and proceeds from international operations.
- Optimize international cash management.
- Evaluate foreign investments.
- Reduce costs of protective measures by accurately analysing risks.

Prediction of foreign exchange rates is therefore not just an endeavour for investors, but for any business with international exposure.

2.2.2 Advantages of foreign exchange trading

Trading on the foreign exchange market holds many advantages over trading on other markets such as the stock or bond markets.

24 hour market: This means that trades can be executed at any time, and does not have to fall in specified trading hours.

Low bid-ask spreads: Pips are the smallest unit by which an exchange rate can change, and is generally the 4th decimal place of the exchange rate. Take an example of the EUR/USD exchange rate. If the original rate is 1.3204 and has moved to 1.3205, then the price has increased by one pip. Exchange rate brokers generally offer different selling and buying prices, with the difference between those prices called the bid-ask spread. For commonly traded currency pairs, called major currency pairs, a spread of 3 to 7 pips is common. This is quite low and encourages frequent trading [8].

Highly liquid: With trading averaging $5.3 trillion daily in 2013, the exchange rate market is the most liquid market in the world [1]. This makes quick execution of trades at low trading costs possible.

Sell before you buy: When trading in exchange rates, a trader selling one currency is simultaneously buying another. This means that the trader does not have to possess additional liquidity in order to complete a trade.

2.2.3 Technical analysis

There is an ongoing debate in the financial community on the efficiency of markets. The Efficient Market Hypothesis (EMH) states that prices already reflect all available information, and that the only way to obtain excess returns is to invest in riskier assets [9]. Several levels of efficiency have been defined [10]:

- Weak form EMH: Security prices reflect all available historical trading information such as price or trading volume.
- Semi-strong form EMH: Security prices reflect all publicly available information such as historical prices, fundamental economic data, balance sheets and quality of management.
- Strong form EMH: Security prices reflect all relevant information, even privileged inside information.

Technical analysis is the study of price and volume movements with the purpose of forecasting price trends. The philosophy behind technical analysis is that it is possible to identify and exploit trends in security prices irrespective of the fundamental reason for change in security price, provided that the trend adjustment process is slow enough [11]. If the EMH is true in any of its forms, then technical analysis would be pointless. All historical price information is publically available, and should already be reflected in the security price if EMH holds true. Technical analysis assumes inefficiency in the market to process and reflect all historical information.

There is however evidence of foreign exchange market inefficiency. Simulations and statistical analyses show operational inefficiencies which can be exploited, and a multitude of empirical studies shows that it is possible to make significant returns using only historic price and volume data [12–14].

2.2.4 Nonlinearity

Even if it is accepted that historic prices and returns hold value in forecasting exchange rates, the forecasting of these rates is still a difficult venture. This is partly due to the inherent nonlinearity of the time-series. Major [2] and minor [15] exchange rates have been proven to contain nonlinear behaviour, and nonlinear models outperform linear models in forecasting these rates [16]. If historical data is used for building a forecasting model, the need to incorporate nonlinear techniques becomes apparent. The following sections will discuss artificial neural networks as a form of nonlinear modelling, as well as Empirical Mode Decomposition, which is used for nonlinear signal analysis. The motivations for using these techniques are covered in chapter 3.

2.3 Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational models inspired by the central nervous system of animals. A neural network is a set of interconnected neurons, where each neuron is a simple processing node. This section will give an overview of the operation of biological and artificial neurons, neural network structures, neuron transfer functions and backpropagation learning methods.

2.3.1 Biological neuron structure

In general, the biological neuron consists of the central body cell, input poles and output poles. The body is called the soma, while the input poles are called dendrites, and the output poles are called axons, as seen in Figure 2-1 [17]. Electrical impulses are transmitted from the axon terminals to dendrites via synapses which vary in conductivity, thus adjusting the intensity of the signal. The receiving neuron sums the signals received through the dendrites in order to determine its excitation level. If the excitation level exceeds the excitation threshold it transmits its own impulse, propagating the signal [18]. A human brain consists of about 85 billion neurons, with each neuron connected to around 5000 other neurons [19]. This gives a human brain immense processing capacity with advanced predictive, cognitive and classification abilities.

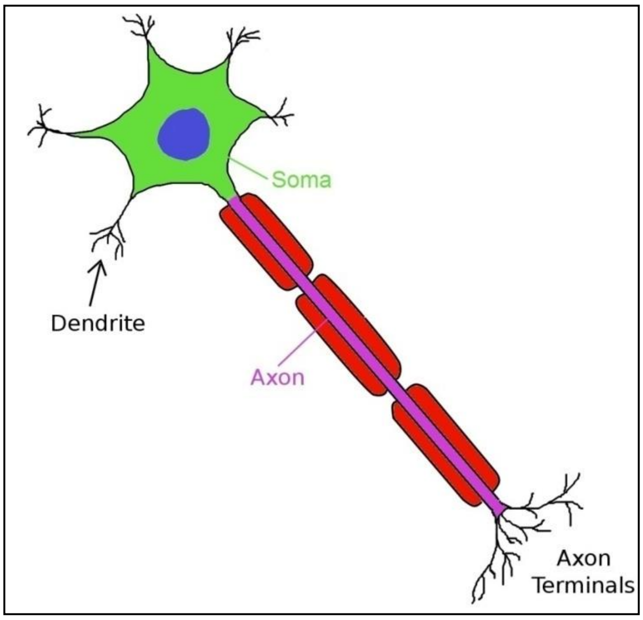

Figure 2-1: Biological neuron

2.3.2 Artificial neuron structure

Just as the biological neural network consists of biological neurons, an artificial neural network consists of artificial neurons. An artificial neuron is a mathematical simplification of the biological neuron, and consists of input, summation, activation and output nodes. Figure 2-2 shows the structure of the artificial neuron. x_1 to x_n are the neuron inputs, similar to the dendrites of the biological neuron. Each input is assigned a respective weight, w_1 to w_n, which acts as a multiplier and can have a positive or negative value. The weighted sum of the inputs, net is calculated by summation of the weighted inputs as seen in equation 2-**1**:

$$net = \sum_{i=1}^{n} w_i x_i \qquad (\text{2-1})$$

This weighted sum of inputs is given as the input of the transfer function φ in order to determine the output of the neuron, o, shown in equation 2-**2** :

$$o = \varphi(net)$$

$$o = \varphi(\textstyle\sum_{i=1}^{n} w_i x_i) \qquad (\text{2-2})$$

The transfer function will be discussed in greater detail in section 2.3.4.

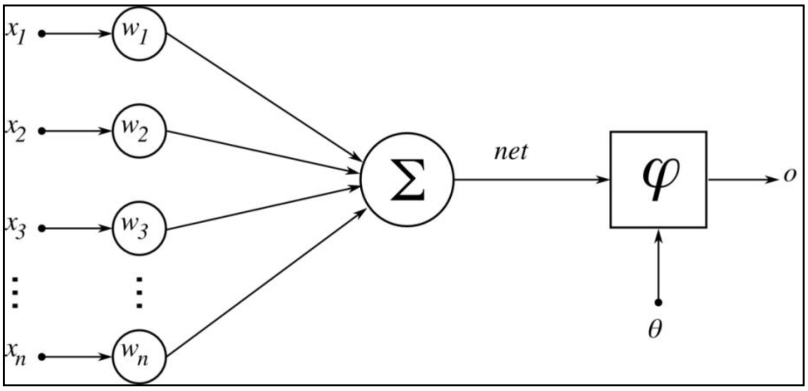

Figure 2-2: Artificial neuron

2.3.3 Artificial neural networks

The artificial neuron is in itself only a simple processing node, and shows its true computational ability when used in an interconnected structure, called a neural network. In a biological neural network, the following types of neurons can be distinguished: input, working (hidden) and output nodes. The input nodes act as the receptors and provide the working nodes with signals. The working nodes act as processors of the signals by providing weighted paths to the outputs, which function as the actuators of the biological system. Artificial neural networks use the same classification of neurons with collections of input, working and output neurons called the input, hidden and output layers respectively [18].

The way the different layers are connected to each other is called the architecture of the neural network, and can be divided into two basic types: cyclic (recurrent) and acyclic (feed forward). A cyclic neural network is generally an unordered neural network where the paths lead in multiple directions. A fixed separation of layers does not exist as a neuron can be an input, hidden and/or output neuron, being able to receive or transmit a signal to any other neuron. This makes the training of cyclic neural networks a complex task. In contrast, the signals in an acyclic neural network all lead in a single direction. Input, hidden and output layers are well defined and separated [18]. Figure 2-3 shows a multilayer feed forward neural network that consists of 3 input neurons, 4 hidden neurons in a single layer and 2 output neurons. Multilayer feed forward neural networks can contain more than one hidden layer or even zero hidden layers. Zero hidden layers should only be chosen when the data is linearly separable. One hidden layer is generally considered to be sufficient for any mapping of one finite space to another. Two hidden layers can theoretically approximate smooth mapping to any accuracy, but shows no significant performance increase when applied to practical data [20], [21].

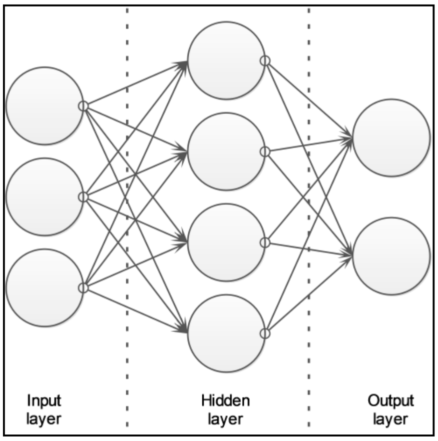

Figure 2-3: Multilayer feed forward neural network

2.3.4 Transfer functions

The transfer function is an integral part of a neuron, and the choice thereof is a nontrivial matter. Table 2-1 lists commonly used transfer functions along with the equations of the relation between the input value net and output value o [22].

Table 2-1: List of common neuron transfer functions and equations

Transfer function	Equation	
Linear transfer functions		
Linear	$o = net$	
Positive linear	$o = net$ $o = 0$	$net \geq 0$ $net < 0$
Saturating linear	$o = 0$ $o = net$ $o = 1$	$net < 0$ $0 \leq net \leq 1$ $net > 1$
Symmetric saturating linear	$o = -1$ $o = net$ $o = 1$	$net < -1$ $-1 \leq net \leq 1$ $net > 1$
Sigmoid transfer functions		
Log sigmoid	$o = \frac{1}{1 + e^{-net}}$	
Hyperbolic tangent sigmoid	$o = \frac{2}{1 + e^{-2net}} - 1$	
Basis transfer functions		
Triangular basis	$o = 1 - \lvert net \rvert$ $o = 0$	$-1 \leq net \leq 1$ $net < -1$ **or** $net > 1$
Radial basis	$o = e^{-net^2}$	
Hard limit transfer functions		
Hard limit	$o = 0$ $o = 1$	$net < 0$ $net \geq 0$
Symmetrical hard limit	$o = -1$ $o = 1$	$net < 0$ $net \geq 0$

Sigmoid type transfer functions are the most commonly used in the hidden layers of multilayer perceptrons due to the nonlinear relationship of the function. The sigmoid functions also have other desirable properties such as smoothness and non-local behaviour. The property of smoothness is desirable due to the ease of calculating derivatives, which makes backpropagation training more effective. The non-local behaviour property means that the function gives a non-zero output in the infinite domain, unlike the basis transfer functions [23]. The tangent-sigmoid function shows the best performance in nonlinear applications, but is more computationally intensive than the log sigmoid and linear functions [24]. The use of linear functions is common in the output layer of multilayer perceptrons. Results show negligible difference between sigmoidal and linear functions in the output layer, as the nonlinearities are already captured by the sigmoidal hidden layer [24].

2.3.5 Backpropagation training

The structure of a neural network allows for accurate modelling of almost all input/output relationships, but is only as good as the neural network training allows. A neural network is considered to be trained when the ideal weight of each neuron input has been found. This is generally accomplished using a technique called backpropagation training.

In order to implement backpropagation training, matching input and output observations are needed. The majority of these observations will be used as training data, called the training set. A smaller percentage of the observations are used as a testing set, and an optional validation set can be employed in order to avoid over fitting the data. The goal of backpropagation training is to minimize the error between the training value and the network output for a specific input by adjusting the neuron weights. This process is repeated for all the observations in the training set, and stops when the generalisation stops improving, indicated by an increase in the error of the validation and testing observations.

Several algorithms have been created in order to find the point of minimum error in the most efficient manner. Some of the more common algorithms include: Levenberg-Marquardt, Broyden-Fletcher-Goldfarb-Shanno (BFGS), resilient backpropagation, scaled conjugate gradient, Fletcher-Powell conjugate gradient, Polak-Ribiére conjugate gradient and one step secant. Of these algorithms, the Levenberg-Marquardt algorithm is accepted to have the fastest training time (convergence rate) as well as the best accuracy for relatively small neural networks with a few hundred weights [25], [26]. This makes the Levenberg-Marquardt backpropagation algorithm ideal for the proposed application.

See Appendix A for the complete Levenberg-Marquardt algorithm derivation and implementation. The algorithm can be summarised as follows:

1) Generate the total error with randomly generated initial weights.
2) Adjust the weights using equation A-22.

3) Evaluate the total error with the new weights.
4) If the total error is increased retract this steps and repeat step 2 and increase the combination coefficient μ by a predefined factor.
5) If the total error is decreased then keep the new weights and decrease the combination coefficient by the same factor used in step 4.
6) Repeat from step 2 with the new weights until the total error is smaller than the required threshold value.

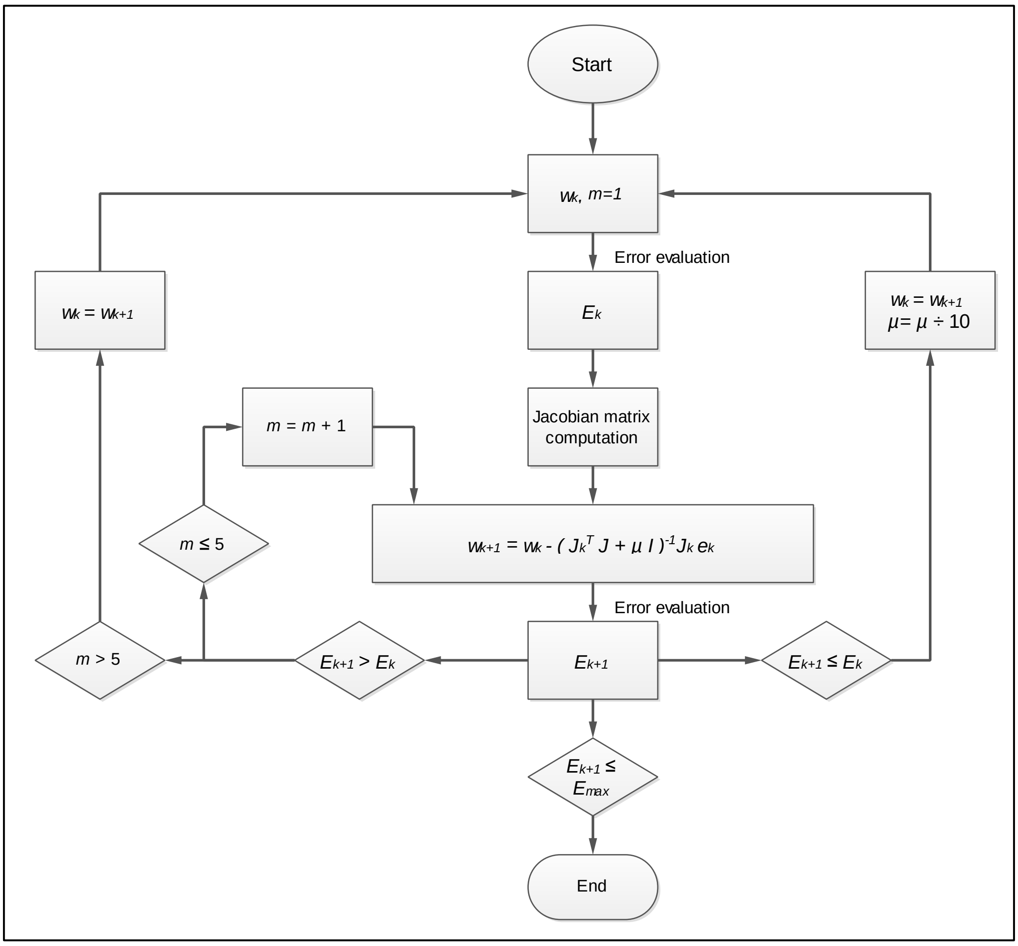

Figure 2-4: Block diagram for Levenberg-Marquardt training algorithm

An example of the Levenberg-Marquardt training process is shown in Figure 2-4, where w_k is the current weight, w_{k+1} is the next weight, E_{k+1} is the current total error, E_k is the previous total error, E_{max} is the required error value, μ is the combination coefficient and m is the index of outputs [27]. In this system the number of outputs is 5 and the combination coefficient is modified by a factor of 10.

2.4 Empirical Mode Decomposition

Empirical Mode Decomposition (EMD) is a technique introduced by Huang et al. in 1996 for nonlinear and non-stationary time series analysis. Although originally designed for use with the Hilbert transform in order to examine energy-frequency distributions, EMD has seen widespread use as a standalone method in various fields of study [3].

2.4.1 Intrinsic Mode Functions

Empirical Mode Decomposition is built on the presupposition that a signal consists of superimposed modes, with each mode representing a band limited frequency response. These modes are called Intrinsic Mode Functions (IMFs). A function is considered to be an IMF if the following conditions are met:

a) For the whole data set the difference in the number of extrema and zero crossings must be less than or equal to one.
b) At any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima must be zero or below the stopping threshold.

Figure 2-5 provides a visual example of the envelopes and mean signals.

2.4.2 The sifting process

The only requirement for a one dimensional signal to be decomposed with EMD is that at least three extrema are required (one maximum and two minima or two maxima and one minimum). A systematic way must be followed in order to decompose a signal into its intrinsic modes. This process is called the sifting process, and is completed as follows for a time signal $x(t)$:

1) Identify local extrema for $x(t)$.
2) Connect all local maxima using a cubic spline interpolate to obtain the upper envelope $\mathrm{x_{up}(t)}$ and connect the local minima using a cubic spline interpolate to obtain the lower envelope $\mathrm{x_{low}(t)}$
3) Obtain the mean envelope:

$$m(t) = \left(x_{up}(t) + x_{low}(t)\right)/2 \qquad (2\text{-}3)$$

4) Extract the difference variable:

$$d(t) = x(t) - m(t) \tag{2-4}$$

5) Check whether the stopping criterion, SC, is met where

$$\sum_{t=1}^{T} \frac{(d_j(t) - d_{j+1}(t))^2}{d_j^2(t)} < SC \tag{2-5}$$

where $\mathrm{d_j(t)}$ is the *j*th iteration's difference variable.

If the criterion is met then denote $d(t)$ as the *i*th IMF and replace $x(t)$ with the residual

$$r(t) = x(t) - d(t) \tag{2-6}$$

(Note that $r(t) = m(t)$. The reason for using different denotations is for clarity in the final phase of iteration where $r(t)$ will become the final residue.)

6) If the stopping criterion is not met replace $x(t)$ with $d(t)$.
7) Repeat steps 1 through 6 until the residue $r_n(t)$ has at most only one local extremum. r_n then becomes the final residual value and concludes the sifting process.

An example of the sifting process can be seen in Figure 2-5. This example illustrates an iteration of the sifting process applied to an extract of the USD/EUR exchange rate.

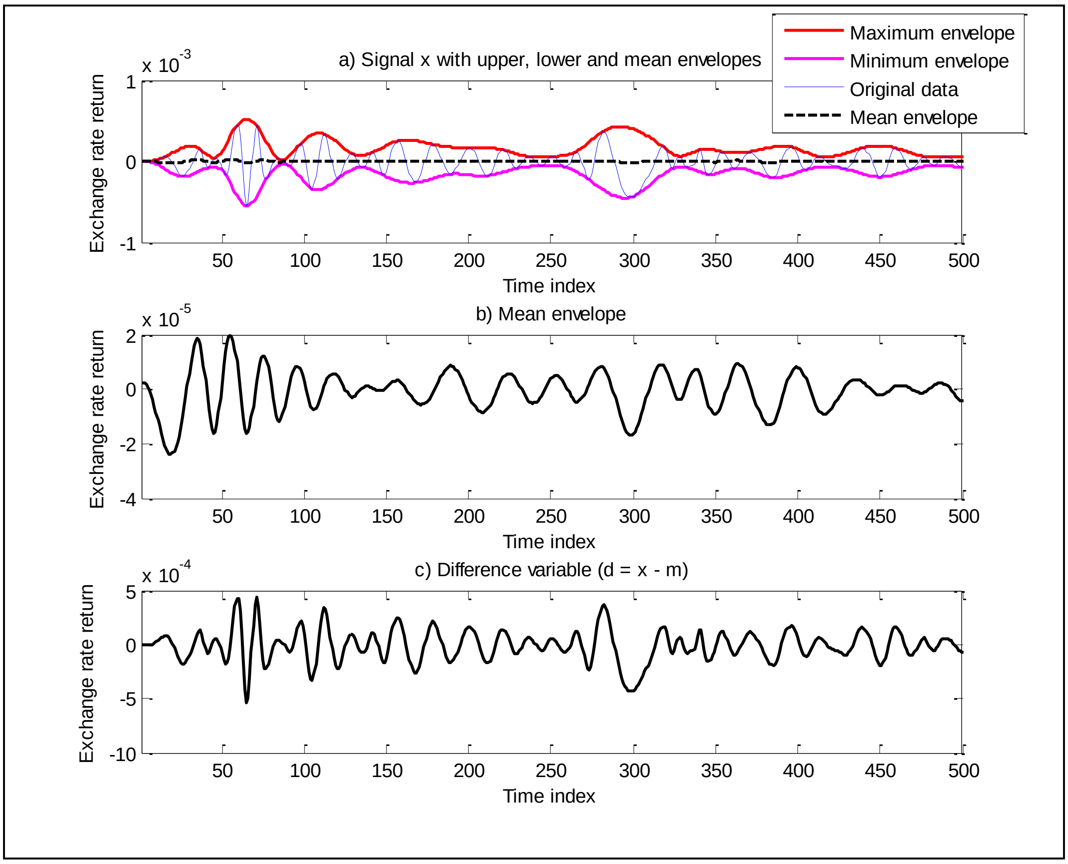

Figure 2-5: **Sifting process example: a) Signal x with upper, lower and mean envelopes; b) Mean envelope; c) Difference between data and mean envelope**

The upper and lower envelopes are generated by fitting a cubic spline to the upper and lower extrema of the data. The mean envelope is calculated using Equation 2-**3**. Equation 2-**4** is then used to calculate the difference variable. If the sifting process is continued until the stopping condition is satisfied, the data will be decomposed into its IMFs and a residue. Figure 2-6 shows the completed result of the sifting process. IMFs 1 to 6 are extracted in ascending order until the stopping criterion is satisfied, leaving only the residue. The results of EMD are given in the time domain, which makes understanding intuitive. Each IMF represents the time response within a certain frequency band, as determined by the local oscillations in the data. These IMFs and the residue can also be summed in order to recompose to original signal.

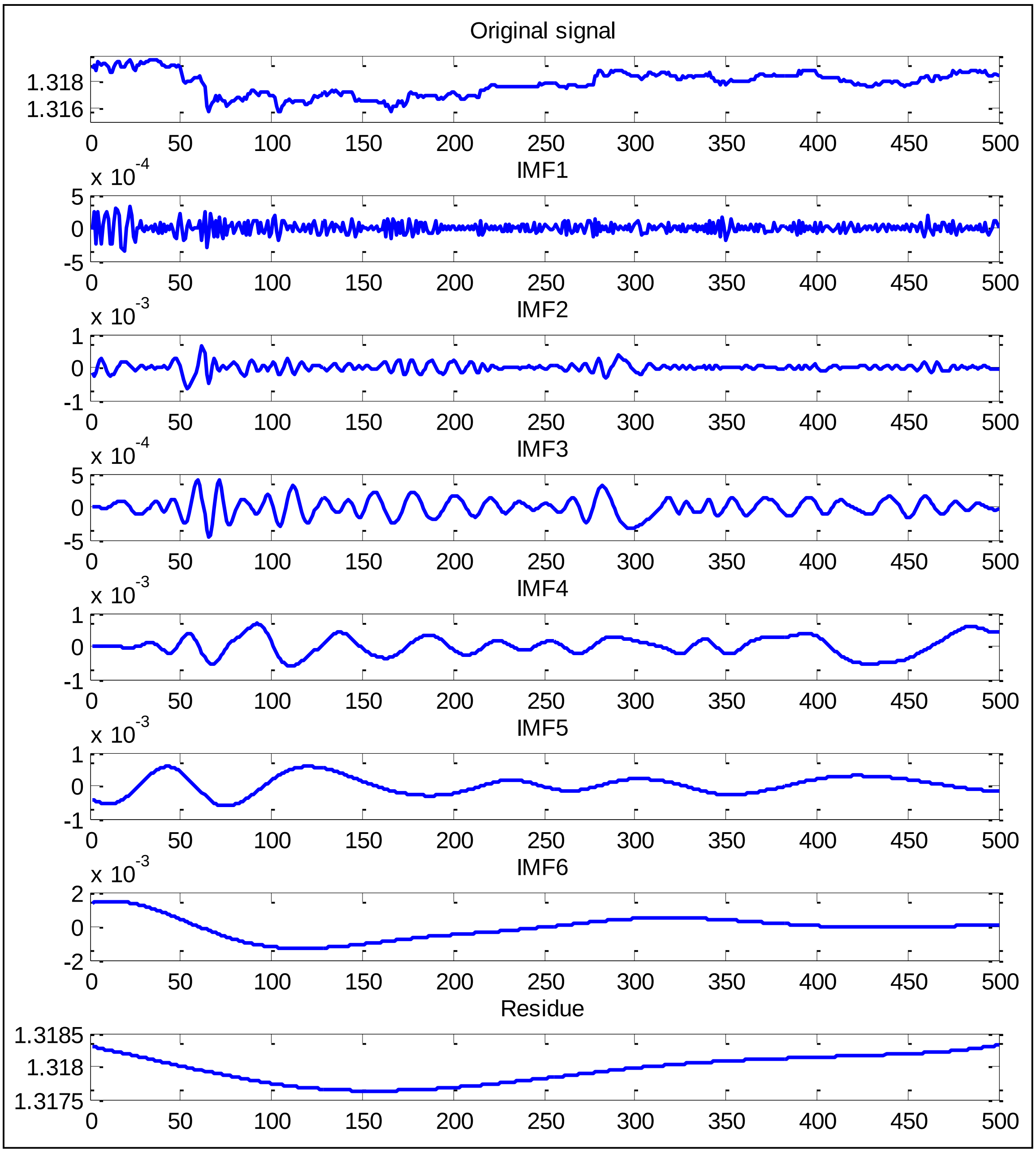

Figure 2-6: Original signal, IMFs and residue that result from the sifting process

2.4.3 Filtering using EMD

It has been suggested that EMD can be used to filter the data. This is due to the IMFs, which can be viewed as band-pass filtered components of the composite signal[4], [5]. The sum of IMFs can be used to create the desired filtered signal, whether it is a band-pass, low-pass or high-pass that is required. Table 2-2 shows the equations used to obtain the respective filters.

Table 2-2: EMD filter composition

Filter type	Equation
High-pass	$High\ pass = \sum_{n=1}^{m} IMF_n$ where IMF_m is the IMF that is closest to but still above the cut-off frequency
Low-pass	$Low\ pass = Residue + \sum_{n=m}^{p} IMF_n$ where IMF_m is the IMF closest to but below the cut-off frequency and IMF_p is the last IMF (not including the residue)
Band-pass	$Band\ pass = \sum_{n=p}^{q} IMF_n$ where IMF_p and IMF_q are the IMFs closest to the respective cut-off frequencies

EMD based filters have the ability to retain nonlinear and non-stationary characteristics of the original signal, which makes it useful when applied to nonlinear and non-stationary data such as exchange rates [4]. The different low-pass filtered signals that can be obtained from the example data set of the EUR/USD log returns can be seen in Figure 2-7. The dashed line represents the unfiltered signal and the solid line represents the EMD low-pass filtered signal.

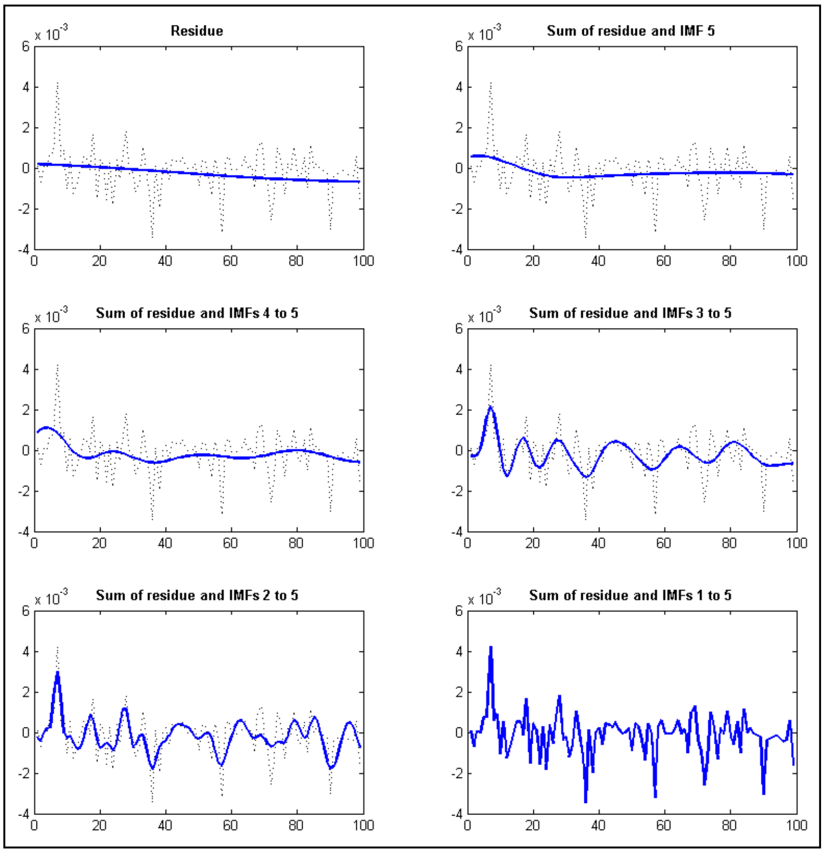

Figure 2-7: Different low-pass filtered versions of the EUR/USD log returns using EMD filtering

2.4.4 Comparison with other time-frequency techniques

In order to accurately represent a nonlinear and non-stationary time series, a method has to fulfil the following conditions [3]:

a) Complete: No information can be lost in the process of representing the data. This condition ensures the precision of expansion.
b) Orthogonal: The same data must not be represented by multiple components ensuring positivity of energy and preventing leakage between the components.
c) Local: Ensures ability to manage non-stationary data by identifying events by the time of occurrence.
d) Adaptive: Ensures ability to manage non-stationary and nonlinear data by adapting to local variations of the data.

The spectrogram is a basic time-frequency decomposition technique. It involves applying a Fourier spectral analysis on a fixed width window sliding along the time axis. This delivers a time-frequency distribution. The spectrogram is easy to implement and fast to compute when using the fast Fourier transform, but has several limitations. The window is assumed to be piecewise stationary, which is not always the case. A further problem is the window size: In order for a localized event, the window width must be narrow, but increased frequency resolution requires a longer time series. Furthermore, the window may be piecewise stationary, but the window size may not coincide with the stationary time scales. This makes the spectrogram a less than ideal technique for time-frequency decomposition [3].

Wavelet analysis is similar to the spectrogram with respect to being windowed Fourier spectral analyses, but differs in the fact that the window is adjustable as determined by the wavelet function. This makes the wavelet analysis extremely versatile, especially for analysing data with gradual frequency changes, but some limitations still exist. The wavelet function has to be predefined based upon assumptions about the data. This limits the applicability of the technique, especially in cases where the underlying behaviour of data is entirely unknown. Some leakage can also occur when using certain wavelet functions, violating the orthogonality condition [3]. Time variance also occurs, as a different set of wavelet coefficients are obtained for a shifted signal.

EMD is entirely empirically based with no formal mathematical assumptions. The technique attempts to decompose the series into the number of frequencies apparent in the data while giving the results in the time domain, making interpretation more intuitive. Table 2-3 highlights the differences between EMD and the other time-frequency domain techniques [28].

Table 2-3: Comparison of frequency domain techniques

	Spectrogram	**Wavelets**	**EMD**
Basis	*A priori*	*A priori*	*A posteriori*
Domain	Frequency	Time-frequency	Time
Suitable for non-stationary data	No	Yes	Yes
Suitable for nonlinear data	No	No	Yes
Suitable for asymmetric cycles	No	Yes	Yes
Orthogonal	No	Yes (depending on wavelet function)	Yes

2.4.5 End effect issues

EMD fits a cubic spline to the maxima and minima during the sifting process. This approach creates a well behaved envelope for the interior extrema, but suffers at the end points due to a lack of extrema that can be identified beyond the end of the available time span. The traditional sifting method instructs to extrapolate the cubic spline until the end of the window, but this approach suffers from large deviations at the end points, especially in the higher frequency IMFs, which then propagate through the rest of the IMFs due to the sequential nature of the sifting process. The easiest way to address this issue is by viewing the end points as both maxima and minima. This eliminates the extreme behaviour at the end points, but at the cost of the preservation of unique data movements, which is undesirable for applications of time-series prediction. Another approach is the extension of the data beyond the end points, allowing the spline to be fitted over extrema over the entire data range. Several methods exist that serve to extend the data such as: wave extension, local straight line, self similarity, overlapping sliding windows and mirror extension [3], [29–32]. These methods improve behaviour at end points significantly, allowing well behaved cubic splines while still retaining unique oscillations.

2.5 Chapter summary

This chapter provides background information on important concepts that feature in this dissertation.

Section 2.2 focused on the foreign exchange market and the importance of forecasting exchange rates. Some of the advantages offered by and difficulties encountered when implementing exchange rate trading are also discussed.

Section 2.3 gives an overview of Artificial Neural Networks. The components, architecture and training of neural networks are also described.

Section 2.4 provides background information on EMD and IMFs. IMF conditions are discussed as well as the sifting process to extract IMFs. Filtering using EMD is described, a comparison of EMD with other time-frequency methods is provided and solutions to the end effect issues are given.

3. Literature review

“More gold has been mined from the thoughts of men than has ever been taken from the earth”

- Napoleon Hill

3.1 Introduction

Foreign exchange rate prediction is a topic that has interested researchers for many years, and has gained much attention in recent years due to the advent of computational intelligence methods and affordable computational power. This has caused a schism between the approaches used by the financial and computational intelligence communities. The financial community value interpretability very highly, and have therefore mainly used linear models with which informed decisions can be made. These models tend to focus more on the fundamental reasons for change in exchange rates than the prediction performance. However, the computational intelligence community values prediction accuracy above interpretability, often making use of black-box techniques to identify patterns in the time-series data [14]. These models tend to be empirically driven and offer little insight into the fundamental reasons for price movements. This chapter will provide a review of the literature surrounding exchange rate prediction from both the financial and computational intelligence perspectives. Special attention will be given to computational intelligence techniques used in combination with Empirical Mode Decomposition.

3.2 Foreign exchange rate prediction in the financial domain

This section will give an overview of the trends in exchange rate prediction research in the financial domain. The early macroeconomic models of the 1970s, random walk models popular in the 1980s to early 1990s and modern nonlinear models will be discussed.

3.2.1 Early research

In the 1970s, a monetary view of the economy was the accepted norm. It was accepted that exchange rates are relative prices of the currency pair nation's currencies, and that any movements were simply the act of reaching equilibrium for demand and supply of international assets [33]. The monetary models of the time differed to a great extent in both the macro-economic factors that were included and in the view of assets prices. Two notable asset pricing systems were the flexible price view suggested by Frenkel [34], which assumed asset prices were flexible over time, and the sticky price view suggested by Bilson [35], which assumed that asset prices are fixed over a short period of time. Some of the models even combined the price views and factored in inflation and interest rates [36].

While the monetary approaches were being followed, the Efficient Market Hypothesis (EMH) also saw a rise in popularity [9]. The view at the time was that markets showed weak form efficiency, and that prediction using only historical data is worthless because it was already factored into the current exchange rate [37], [38].

The early research laid the groundwork for using macroeconomic factors in order to develop models that are interpretable from the financial domain [14].

3.2.2 Random walk models

The monetary models of the 1970s were questioned when Meese and Rogoff [39] compared the flexible and sticky price view models with a random walk model. They found that the out-of-sample prediction accuracy of these models was no better than a random walk model, and gave rise to the acceptance of the random walk hypothesis. This hypothesis states that exchange rate time series follow a random walk, and that any attempt at prediction is futile. Several other researchers have confirmed that foreign exchange rates appear to follow a random walk. Backus [40] describes the empirical support behind the monetary models as weak, and supports the random walk hypothesis. Wolff [41] investigated several published univariate time series models and found that none outperform a simple random walk model.

While the random walk hypothesis gathered support, several researchers opposed this view. Somanath [42] provided evidence that models that take lagged historical values into account outperform random walk models for both in and out-of-sample tests. Hakkio [43] investigated the conditions other researchers used to identify random walks and found that they have low explanatory power, as supported by Monte Carlo tests. Shinasi and Swamy [44] confirmed Meese and Rogoff's observations that a fixed coefficient model does not outperform a random walk, but found that a variable coefficient model does. Using variance ratio tests of five exchange rate pairs, Liu [45] found evidence rejecting the random walk hypothesis.

The foreign exchange rate prediction research in the 1980s to 1990s primarily focused on the random walk hypothesis, and whether it is possible to beat a random walk model. The debate around random walks has still not been settled. In 2008, Azad [46] tested modern Asian exchange markets, and found that daily timescales appear to follow a random walk, while longer weekly ones reject the random walk hypothesis.

3.2.3 Nonlinear models

Linear models dominated most of the early exchange rate research, but some researchers began exploring nonlinear modelling due to the poor fits of out-of-sample data. Hsieh [2] found that five major exchange rate pairs showed nonlinear behaviour, and that a Generalised AutoRegressive Conditional Heteroskedasticity (GARCH) model fitted this behaviour well. Recently, Nusair [15] showed that minor exchange rate pairs also show nonlinear behaviour. Taylor and Peel [47] investigated monetary models and found that the economic fundamentals used were significant but showed nonlinear behaviour, which may explain the poor out-of-sample performance of the earlier models. When researchers tested their new nonlinear models, the standard comparison was still the random walk model. Bleaney and Mizen [48] rejected the random walk hypothesis by using a cubic model to outperform both random walk and linear models in terms of prediction accuracy. Brooks [49] noted minor improvements over the random walk model when compared to the tested linear (autoregressive moving average) and nonlinear (GARCH and ANN) models. Creedy, Lye and Martin [50] showed that their nonlinear model, which derived the stationary density as the solution for a continuous time error correction model, was able to outperform the random walk model. Pavlidis, Paya and Peel [16] compared a nonlinear model with a random walk and linear autoregressive model for Frank/Sterling and Dollar/Sterling exchange rates. The nonlinear model outperformed all the models for the Dollar/Sterling rate, but failed to beat the linear model for the Frank/Sterling, which could suggest that not all exchange rates show nonlinear behaviour.

The shift from linear to nonlinear models in the financial domain proved to lead to higher prediction accuracy. Modern computational intelligence techniques can be optimised for nonlinear applications, and are thus ideal for exchange rate prediction.

3.3 Foreign exchange rate prediction in the computational intelligence domain

The theory behind computational intelligence methods have existed for a long time, but have not seen widespread use due to the limited access to computational power. With the advent of affordable computational power in the early 1990s, computational intelligence methods have increased in popularity.

Artificial Neural Networks (ANNs) have been the dominant computational technique in exchange rate prediction. This is due to the ability of ANNs to accurately describe the non-linear relationships between inputs and outputs, and the large amount of data available to properly train the network. In one of the earliest applications of ANNs on exchange rates, Refenes et al. [51] showed that it is possible to make up to 20% returns in 60 days using only historical exchange rate data. There were some limitations to their research, such as zero transaction costs and a risk free lending rate, but it proved to be a good proof of concept, with single-step-ahead trend prediction accuracy of 66%.

Staley and Kim [52] investigated the feasibility of forecasting the CAD/USD exchange rate. Using historical returns and the interest rates as inputs and the expected returns as outputs, they were able to correctly predict 59% of directional trends between 1991 and 1993.

Hann and Steurer [53] compared the performance of neural networks with linear monetary approaches and concluded that the nonlinear neural networks clearly outperformed the linear models. Monthly and weekly exchange rate data was used in this study, and led to an interesting observation: the neural network outperformed the linear model for the weekly data, but produced similar results for the monthly data. They conclude that nonlinearities only appear on certain timescales, and that neural networks are not needed for the linear timescales.

Yao and Tan [54] investigated the efficiency of several exchange rates and found that they were not highly efficient and that it should be possible to forecasts these rates. Using a neural network model, they showed that significant returns can be made on out-of-sample predictions with just historical data and no extensive market knowledge.

Since the early 2000s, studies have focused on the improvement of neural network predictions in several ways. Some work aims to optimise the architecture and training of the neural network while others create combinations of multiple ANNs or hybrid models which incorporate other techniques with ANNs. Some researchers use genetic algorithms to optimize the neuron weights or number of inputs [55], [56], while others employ fuzzy logic [57] or ensemble neural networks [58].

Yu et al. [59] use an adaptive smoothing neural network in order to improve generalisation of a multilayer feed forward network. This smoothing retains the nonlinearities of the data while removing unwanted noise. Their results show improved prediction accuracy for the three tested exchange rates, as well as faster training convergence and better generalisation.

Ardiansyah et al. [60] conducted a survey of 55 articles published on exchange rate forecasting using neural networks between 1996 and 2012. They classified the articles based on the input data types, forecast horizons and performance evaluation techniques.

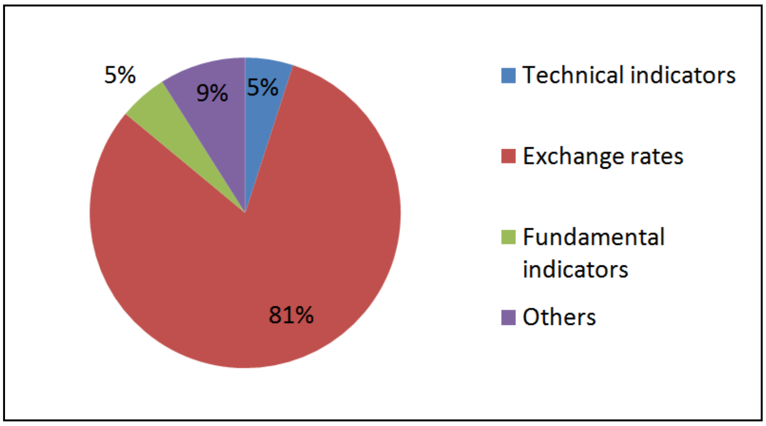

Figure 3-1: Articles classified by input data types

Figure 3-1 shows the distribution of the input types used. 81% of the articles reviewed used the exchange rates or processed versions thereof as the inputs to the neural network. The distribution of forecast horizons can be seen in Figure 3-2. The majority of research used daily data and forecast a single step ahead. This may be attributed to the ease by which daily data can be obtained and the fact that the research focussed more on the accuracy of predictions than the exploitability of the exchange rates at those horizons.

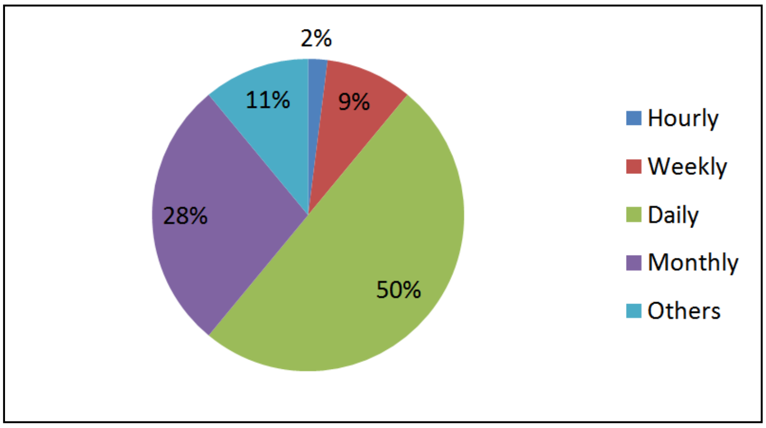

Figure 3-2: Articles classified by forecast horizons

Figure 3-3 shows the distribution of performance evaluation criteria. 69% of researchers used statistical techniques such as mean-square-errors. Trends are also used as a more practical measure of performance as it can be used as buy/sell signals in order to test trading strategies based on the data.

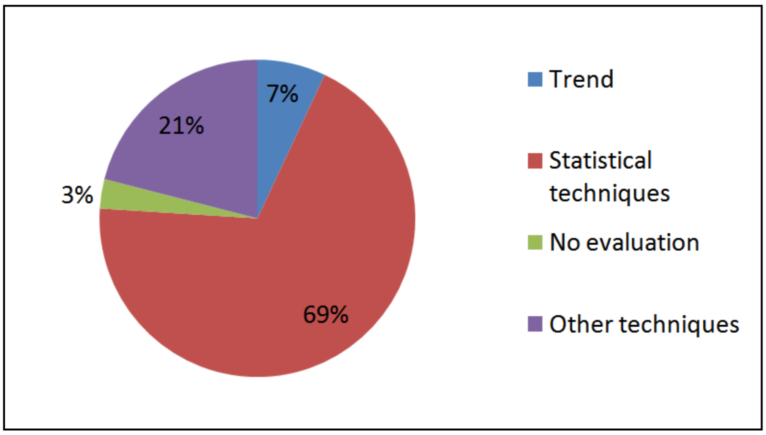

Figure 3-3: Articles classified by performance evaluation criteria

In conclusion it can be stated that ANNs have seen widespread use in the forecasting of exchange rates. The results of ANN exchange rate forecasting is mostly positive, with the ANN models outperforming the benchmark models in most cases and hybrid models offering additional utility or accuracy.

3.4 Financial prediction using Empirical Mode Decomposition

Empirical Mode Decomposition (EMD) was introduced in 1996 by Huang et al. [3], but has only recently been used in the area of financial prediction. The earliest published use of the method on financial time series can be attributed to Huang et al. in 2003 [4], who used it to analyse the frequency characteristics of mortgage rates. He also noted the filtering ability that EMD provides, and used an EMD filter to examine market volatility at different time scales.

Zhang, Lai and Wang [61] applied EMD to crude oil prices for analysis purposes. Using EMD, they were able to identify fluctuations caused by supply and demand, long term trends and shocks caused by significant events. Lai and Wang then collaborated with Yu [62] in order to combine EMD with machine learning techniques for prediction of daily crude oil prices. Their method consists of three steps:

1) The crude oil price time series is decomposed into a number of IMFs and a residue using EMD.
2) A three layer feed forward neural network is used to forecast the IMFs and residue individually.
3) The sum of the predicted values is calculated and given as inputs to an adaptive linear neural network (ALNN), which is a single layer neural network with linear transfer functions. The output of the ALNN is the final prediction.

The proposed model was tested against single feed-forward neural networks and Autoregressive Integrated Moving Average (ARIMA) techniques for single-step-ahead prediction, where it outperformed the other models in the criteria of root-mean-square-error (RMSE).

Wang et al. [63] continued the trend of financial prediction using EMD and computational intelligence techniques with a similar structure as the one used by Yu et al., but used another computational intelligence method called support vector regression (SVR). SVR works on the principle of mapping inputs to a higher dimensional feature space, where linear relationships can be found between variables. SVR is also trained using supervised learning approaches, but always converges on the global minimum where ANNs can sometimes converge on local minima. Their method uses EMD to decompose the Shanghai Securities Index into its IMF and residual components. Each of the components was then predicted using support vector regression and the final prediction can then be obtained by summation of the predicted components. A diagram of the method can be seen in Figure 3-4. The proposed model was compared with a single SVR model for single and multi-step-ahead predictions, where the hybrid model outperformed the single SVR in terms of RMSE and mean-average-error (MAE).

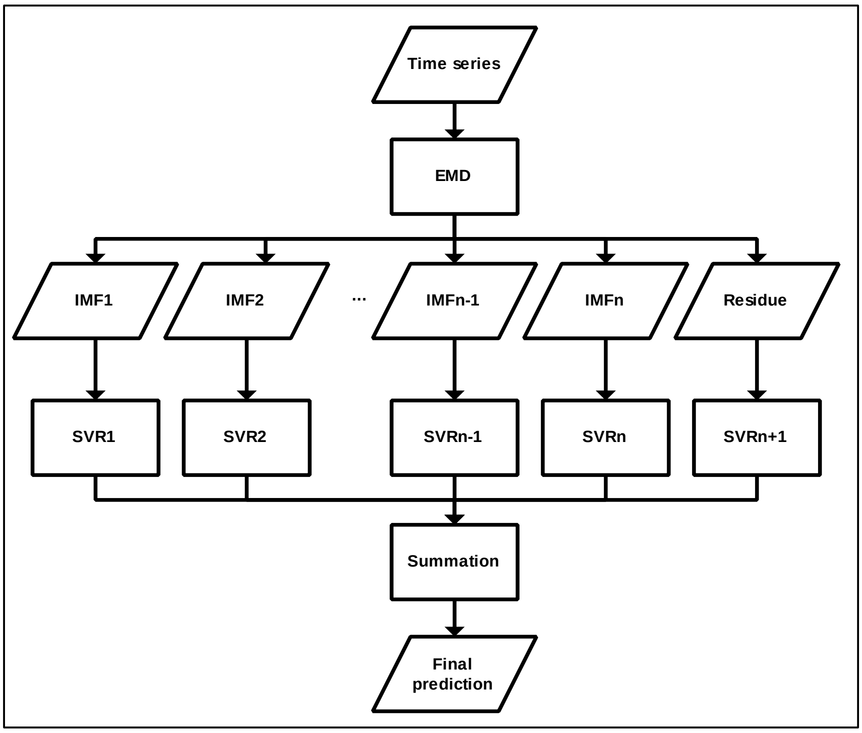

Figure 3-4: The hybrid EMD-SVR method used by Wang, Fu and Lin

Several examples exist where EMD-SVR hybrid models are used to forecast exchange rates. Fu [64] used a method identical to the one used by Wang (Figure 3-4) to forecast the decomposed signals and sum the predicted signals for the final prediction. The proposed model was applied to the weekly average EUR/RMB time series, and a single SVR was used as a benchmark model. The EMD-SVR model outperformed the SVR model by a large margin in terms of RMSE, MAE and mean-average-percentage-error (MAPE). Lin et al. [65] used the same hybrid method to forecast the three daily exchange rates of USD/NTD, JPY/NTD and RMB/NTD. The EMD-SVR model outperformed single SVR and ARIMA models in all performance evaluation criteria (RMSE, MAE, MAPE and directional symmetry).

Yang and Lin [66] created a hybrid EMD-back-propagation neural network (BPNN) model in order to forecast the daily USD/NTD exchange rate. This model decomposed the time series into IMFs and a residue and gave them all to a single BPNN as inputs. This BPNN was then responsible for calculating the final predicted value. The proposed model outperformed a random walk model in all the performance evaluation criteria (RMSE, MAE, MAPE, directional symmetry).

Yu et al [67] created an EMD-BPNN classifier to forecast financial crises using historical exchange rate time series. This classifier selected only the most significant IMFs of the decomposed signal using connection weights of a pre-trained network as the selection criteria. The proposed model is compared to a large number of statistical and computation models such as a logit model, probit model, linear discriminant analysis model, quadratic discriminant analysis model, BPNN and general regression neural network (GRNN). The proposed model outperformed the other models in the criteria of predicting the occurrence of financial crises with an out-of-sample accuracy of up to 98.39%. They also conclude that the neural network models without EMD outperformed the statistical models, indicating that ANNs are well suited for currency crisis forecasting.

Xiong [68] used a framework similar to the one in Figure 3-4, but using feed forward neural networks instead of SVR in order to investigate multiple-step-ahead predictions of crude oil prices. Using daily data, they investigate the accuracy of forecasts for forecast horizons of 4, 8,12,16,20 and 24 days ahead using several methods and models. Direct and multiple input multiple output (MIMO) multi-step-ahead forecasting strategies with the proposed EMD-FNN model perform the best, especially with the longer horizon forecasts. The iterated multi-step-ahead forecasting strategy, where single-step-ahead forecasts are iterated to obtain a multi-step-ahead forecast, has the worst performance, never outperforming any of the other strategies. All the strategies and models outperform the random walk model in the criteria of mean-absolute-scaled-error (MASE) and directional symmetry.

It can be concluded that in all the research conducted on EMD-based financial prediction, the inclusion of EMD adds value to the models by increasing prediction accuracy. The application of EMD to financial forecasting has only seen a rise in popularity in recent years, and the research around it has some shortcomings:

- A lack of practical performance measures such as possible returns that can be made over the time period if trading rules are applied to the predictions. Statistical measures such as error sizes do not necessarily translate into real world trading or investment performance.
- The arbitrary choice of data sampling rate and forecast horizons. The data sample times seem to be chosen simply for convenience of availability, and the forecast horizons are normally only one-step-ahead. Even in cases where multi-step-ahead forecasts are conducted, the horizons seem to be simply chosen at random.
- The EMD technique is used to allow the separate modelling of the signal at different time scales, without optimally using the filtering capabilities of EMD to discard information that will not contribute to forecasting over the selected horizon.

3.5 Chapter summary and conclusions

This chapter provided a review of literature in the area of exchange rate prediction in general and financial prediction using EMD in particular.

Section 3.2 provided a review of literature of exchange rate prediction in the financial domain. The early models and debate around the random walk hypothesis are discussed and the literature stressing the importance for nonlinear models was examined.

Section 3.3 provided a review of literature concerning exchange rate prediction using neural networks. The advantages of the nonlinear neural network models are explored along with novel innovations and hybrid methods in the computational intelligence domain.

Section 3.4 provided a review of literature where EMD was applied to financial analysis and prediction. The methods and frameworks of the different studies were discussed and the shortcomings of current studies were identified.

The literature studied provided many insights into exchange rate modelling and helped to identify areas where the current body of knowledge can be improved upon. The findings are summarised below:

- It is possible to forecast exchange rates using only historical data.
- The random walk model has been a benchmark for prediction models since the early 1980's.
- Computational Intelligence methods like ANN's and SVR's have proven to be successful in modelling exchange rates.
- Hybrid models that combine EMD with computational intelligence methods show improved prediction accuracy compared to the computational intelligence models that do not incorporate EMD.
- The computation intelligence community do not investigate practical performance measures such as simulated returns, but focus more on minimizing model errors.
- The data sample times and forecast horizons are almost always chosen arbitrarily.
- While the analysis capabilities of EMD is used, it is not fully exploited to perform filtering prior to forecasting.

The objectives of this study can be clearly defined after consideration of these findings. The aim of this study is to design an exchange rate prediction model that:

- Uses historical exchange rate data to predict future behaviour.

- Methodically chooses data sample times and prediction horizons to optimize returns to be generated.
- Uses the computational intelligence method of artificial neural networks.
- Uses a novel EMD-filtering method to improve ANN performance.
- Outperforms a random walk model in practical performance measures like simulated returns as well as modelling performance measures like RMSE and directional symmetry.

The research will contribute to the current body of knowledge in several ways. The viability of using EMD-filtering to improve forecasting performance will be evaluated, a methodical means of choosing prediction horizons will be designed and the practical performance of hybrid EMD ANN models will be assessed.

4. Methodology

"I think you can have a ridiculously enormous and complex data set, but if you have the right tools and methodology then it's not a problem."

- Aaron Koblin

4.1 Introduction

Neural networks are well known for their predictive capabilities and would be ideal for the prediction of nonlinear and non-stationary exchange rate data. However, the successful use of neural networks requires specific conditions to be satisfied to ensure that a robust classifier will be obtained producing reliable results that represent systematic underlying behaviour rather than to be dominated by the noise that is inherently part of the data.

Lawrence [69] summarised the design of neural networks in 5 steps:

1) Problem definition: Choose the information that will be used and what the desired output should be.
2) Pre-processing: Collect the data and process it to the desired representation.
3) Define the neural network architecture: Select number of inputs, hidden nodes and outputs.
4) Train the network: Compile training sets and train the network in order to find the optimal node weights.
5) Test the trained neural network: Test the neural network in a realistic situation and adjust parameters as necessary

In this case we try to develop a classifier that can distinguish between signals that are expected to either display positive or negative returns over a pre-determined future time horizon. We are hoping that the expected future time behaviour will be apparent from historical time behaviour, as represented by a potentially large number of historical time samples. If however too many time samples are used as input variables for the neural network, resulting in a large number of model parameters, the accuracy of the predictions can be expected to deteriorate due to over fitting of the model and hence lack of generalisation ability [70]. Thus a relatively small number of inputs are required that are still able to capture those aspects of behaviour that will display a systematic relationship with expected future behaviour. The main problem can thus be defined as identifying which of the many historic time samples to use.

The high frequency components contain the most detailed information, but are represented by too many samples and can be expected to contain much noise that will not contribute to forecasts over the selected time horizon. A technique is required that assists in the choice of the optimal combination of input data and forecast horizons. EMD-filtering effectively transforms the original data to a reduced data set, where each sample represents a combination of the original samples; if the cut-off frequency is correctly chosen these filtered samples should be more suitable for prediction over the desired horizon.

EMD offers several benefits over convention filtering:

- Behaviour belonging to the same mode can be isolated into a single sub-signal (or IMF).
- It is possible to determine which of these IMFs contain the most prominent signal behaviour by measuring the variance of each IMF.
- It is possible to only use that portion of the original signal that can realistically be expected to contribute towards prediction over the selected forecast horizon.
- The resampled samples that are obtained from the filtering would allow selection of a small enough set of features to be used as neural network inputs while still covering a long enough historical time window to capture the nature of signal behaviour at the respective time scale.

The EMD-filters would be able to provide the neural network model with adaptable and meaningful input features, and would conceivably result in greater prediction accuracy. This chapter documents the design process of such an EMD-filtered neural network model. The rest of the chapter is organized as follows:

Section 4.2 describes the data collection process, 4.3 the data pre-processing and section 4.4 the filtering procedure. In section 4.5 we consider the neural network design while in section 4.6 we describe the benchmark models and in section 4.7 the performance evaluation criteria. Section 4.8 summarises and concludes this chapter.

4.2 Data collection

Exchange rate pairs are generally divided into two categories: major and minor. There are currently seven major exchange rate pairs: EUR/USD, USD/JPY, GBP/USD, USD/CAD, USD/CHF, AUD/USD and NZD/USD. These pairs all feature the US Dollar, and are the most frequently traded pairs on the foreign exchange market [1]. The major pairs have the lowest bid/ask spread due to their high liquidity. The minor or exotic currency pairs generally feature at least one less frequently traded currency, while the other currency can be another minor currency or a major currency. The bid/ask spread of these currency pairs are usually larger than the bid/ask spread of major currency pairs due to the lower liquidity of at least one of the currencies. Three exchange rates will be chosen for this study. The EUR/USD and USD/JPY exchange rates represent major exchange rate pairs, while the USD/ZAR exchange rate is chosen to represent the minor exchange rates. This selection of exchange rates will allow us to investigate the forecasting accuracy and exploitability of both major and minor exchange rates.

The choice of sampling period of the data is of critical importance in this study, as a major component of the research is to explore the exploitability of foreign exchange time series at different timescales. It is therefore necessary to start off with data that can be analysed at many different time scales. The shortest sampling times for exchange rate data available for this research are one minute samples. Using one minute samples allows the evaluation of high frequency trading, as well as multi-step-ahead forecasts and the use of longer sampling periods by resampling the data. The use of high frequency data will also allow for sufficient low pass filtering even on the shortest practical prediction horizons (10 minute horizons and longer).

A calendar period of one year is chosen as population for model development for the purposes of this research. The motivation behind this choice is threefold:

- A year’s worth of exchange rate data should provide sufficient independent time periods in order to verify the consistency of model behaviour.
- Sufficient observations are present in order to provide data for training of the neural network, even when using longer sampling periods.
- It could be detrimental if the training period is too large, as the underlying nature of behaviour may change over time. If the year’s data is divided into months, with each previous month used as a training period for a model that is subsequently tested on the next month, then the risk of market behaviour significantly changing from training to testing period will be minimized.

In summary, the following data is chosen:

- Exchange rate pairs: EUR/USD, USD/JPY, USD/ZAR
- Time period: 1 January 2013 to 31 December 2013
- Sample time: 1 minute closing rate

The data is obtained from the Straighthold Investment Group's LiteForex server accessed through Metatrader 4's user interface. An average bid ask spread was calculated from the LiteForex server's data for each of the exchange rates in 2013. The values in Table 4-1 will used as the bid/ask spreads in all simulations.

Table 4-1: Bid/ask spreads for the exchange rates

Exchange rate	Bid/ask spread (pips)
EUR/USD	0.0004
USD/JPY	0.0004
USD/ZAR	0.0020

4.3 Data pre-processing

4.3.1 Transformation

In order to make meaningful predictions, the choice of variable to predict is of extreme importance. All the data obtained is in the form of the closing rate. While predictions of the closing rate are possible, it is not ideal as exchange rates are in general non-stationary time series. As a result the descriptive statistics such as means, variances and correlations that are used as part of performance evaluation criteria may not be meaningful as it may be dominated by long term upward or downward trends. The performance criteria would accordingly lose significance if the descriptive statistics are not reliable.

The return of a time-series is a profit that can be made on an investment, and is commonly used in the area of financial prediction to transform the data into a stationary time series. The simplest form is the single period return, given by equation 4-1:

$$R(t_f) = \frac{V(t_f) - V(t_i)}{V(t_i)} \qquad (4\text{-}1)$$

where $R(t_f)$ is the return on the investment at time t_f, $V(t_f)$ is the final value of the investment and $V(t_i)$ is the initial value of the investment.

Logarithmic returns (log returns) also indicate the return of an investment, but offer additional benefits. While the log return and simple single period return are relatively similar for extremely small returns such as those generated by high frequency exchange rate trading, the log return offers improved dependability. This comes into effect when calculating compound returns: the simple singe period compounded return is calculated by multiplying the small returns, which could lead to arithmetic underflow where significant data is too small to store within the computer memory. The compounded log return is calculated by addition of the small returns, making the answer more reliable. Log returns have the additional benefit of compressing the range of return values, preventing a few large outliers to dominate the statistics extracted from the time series. The logarithmic returns are given by equation 4-2:

$$r(t_f) = ln\left(\frac{V(t_f)}{V(t_i)}\right) \qquad (4\text{-}2)$$

where $r(t_f)$ is the log return on the investment at time t_f, $V(t_f)$ is the final value of the investment and $V(t_i)$ is the initial value of the investment.

Due to the small return that can be expected from high frequency trading, the data will be transformed to log returns using equation 4-2 after the necessary resampling is done.

4.3.2 Resampling and forecast horizons

The original data is based on a one minute sampling period. In order to investigate the exploitability of the data in a meaningful way, the data has to be resampled to other timescales. A structured method to determine the choice of forecast horizon and sample period is required and needs to take the following factors into consideration:

- The bid/ask spread of the time series (extremely short forecast horizons will result in profits that are insignificant compared to the bid/ask spread, resulting in losses even if the direction was correctly predicted).
- The maximum possible returns that can be made with an idealised trading strategy if all the predictions are 100% accurate.
- The historical time period to be covered by the input set in order to sufficiently represent the nature of signal behaviour.

A simulation using the idealised trading strategy was conducted in order to find the prediction horizon at which the maximum possible returns can be generated for a time series after providing for bid/ask spread losses. The returns are calculated using the performance evaluation measure and trading strategy to be discussed in section 4.7. The simulation was run for all three time series and consists of the following steps:

1) Divide the data of the entire year into 12 equally sized windows in order to represent monthly data.
2) Resample each monthly window to obtain time series with sampling periods of 1 minute to 1000 minutes.
3) Assume 100% prediction accuracy and performing a trade after each forecast horizon.
4) Calculate the average returns per month at each forecast horizon, taking the bid/ask spread into account.

Figure 4-1, Figure 4-2 and Figure 4-3 show the monthly returns that can be achieved for the EUR/USD, USD/JPY and USD/ZAR exchange rates respectively if trading occurs over different forecast horizons at 100% prediction accuracy. Note that the forecast horizon used in this analysis is equal to one sampling period.

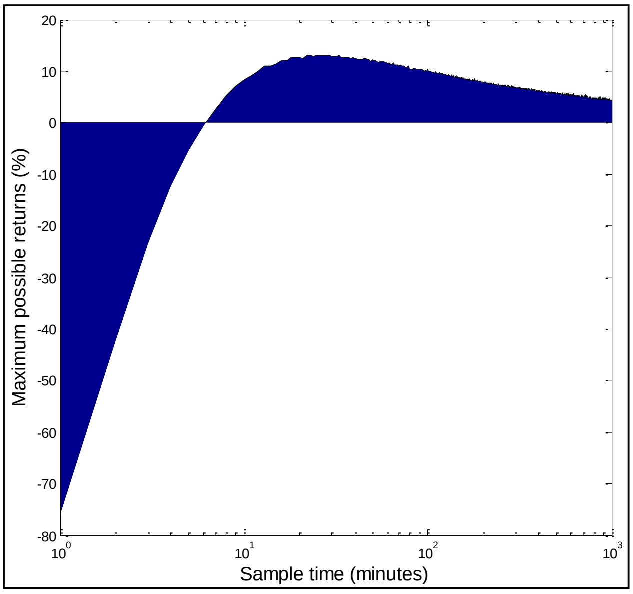

Figure 4-1: EUR/USD monthly average idealised returns at different sample times

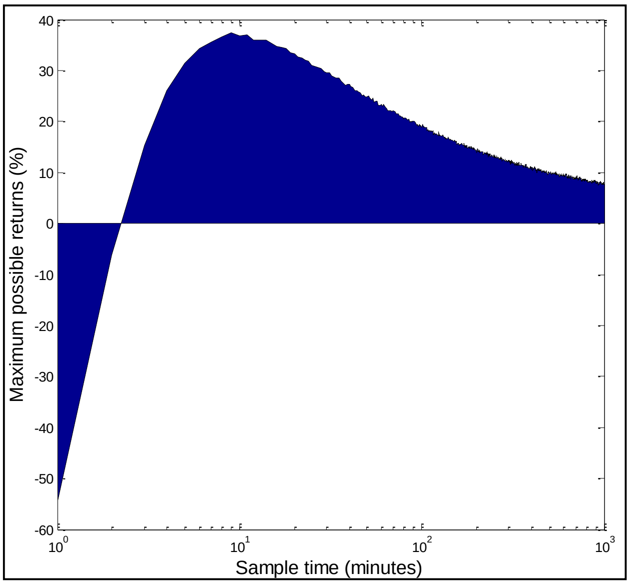

Figure 4-2: USD/JPY monthly average idealised returns at different sample times

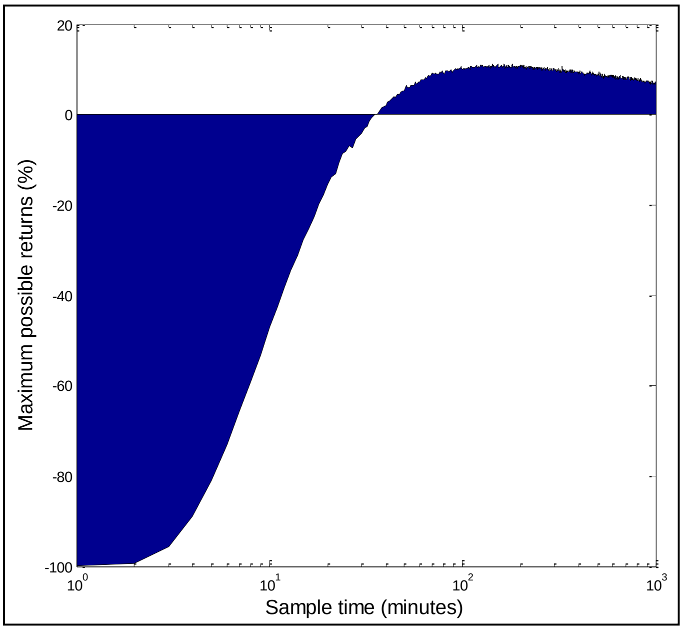

Figure 4-3: USD/ZAR monthly average idealised returns at different sample times

Table 4-2 shows the minimum forecast horizon for positive returns and the forecast horizon at which maximum returns can be found. The EUR/USD and USD/JPY exchange rates have relatively similar responses with positive returns that occur at low forecast horizons due to the low bid/ask spreads. The USD/ZAR exchange rate, with its much higher bid/ask spread requires a long forecast horizon to offset the trading costs.

Table 4-2: Minimum sample time to obtain positive returns and sample time to obtain maximum returns for the chosen exchange rates

Exchange rate	Minimum sample time for positive returns (minutes)	Sample time at which maximum returns are found (minutes)
EUR/USD	7	33
USD/JPY	3	9
USD/ZAR	37	181

The timescales have now been identified at which idealised trading must occur in order to maximize returns. Next it is required to choose data sample periods which coincide with the optimal forecast horizons. Table 4-3 shows the chosen sample times and forecast horizons for the different exchange rates.

Table 4-3: Choice of sample times and forecast horizons for the exchange rates

Exchange rate	Sample period (minutes)	Forecast horizon (number of samples)	Forecast horizon (minutes)
EUR/USD	1	35	35
	5	7	35
	7	5	35
	35	1	35
USD/JPY	1	10	10
	2	5	10
	5	2	10
	10	1	10
USD/ZAR	15	12	180
	30	6	180
	60	3	180
	90	2	180
	180	1	180

The choice of sample periods and prediction horizons considers the following aspects:

- The forecast horizon in minutes was chosen to coincide with the rates at which trading must occur in order to deliver maximum idealized returns while still allowing several options for sample periods and numbers of samples at those horizons.
- For the EUR/USD and USD/JPY exchange rates the sample periods vary from the minimum available sample period of 1 minute up to sample periods equal to the full length of the forecast horizon. With these sample periods it should be possible to compare single and multi-step-ahead prediction accuracy for the different exchange rates. It can be reasoned that the exchange rates sampled at higher frequencies contain components that are of importance to the forecasting accuracy. In contrast to this view it is possible that the high frequency components increase the noise levels in the data which can be detrimental to forecasting accuracy.
- The choice of a minimum sample period of 15 minutes for the USD/ZAR exchange rate was made under the assumption that it would be unreasonable to expect acceptable forecasting accuracy at 1 minute sample times at a forecast horizon of 180 minutes.

4.4 Feature extraction enabled by an EMD filter

Feature extraction is the process by which the raw input data is transformed into input sets that, in combination with the target values, form the observations on which the models are trained and tested. The transformation of the exchange rate data to log returns and the resampling of the data represent the initial steps of the input data transformation. However, this data contains samples that are deemed irrelevant for the chosen forecast horizons. The higher frequency data is not essential to the longer forecast horizons, and can be expected to deteriorate prediction accuracy if included in the training of the neural network as it will significantly add to the noise in the signal.

A correctly defined low-pass filter would effectively reduce the noise in the data set while retaining the useful information in the signal, thereby improving the signal-to-noise ratio of the data. The problems with conventional linear filtering techniques are that the cut-off frequencies have to be chosen a priori and that nonlinearities in the relationships between historic and future data are lost upon filtering. In contrast the EMD-filter is empirically based and thus not only naturally isolates behaviour at different time scales, but also retains the nonlinear data relationships.

Conventional tapped delay line filters furthermore result in significant time delay between the filter inputs and filter output – this may cause the filtered version of the signal to only become available once a significant portion of the future movement to be exploited has already occurred. EMD filters are expected to have less of an impact in terms of time delays as they do not use a tapped delay line architecture.

Figure 4-4 illustrates the proposed EMD-filter. The transformed exchange rate data is decomposed using EMD into N IMFs and a residue, *R*. The signal is then low-pass filtered by the summation of IMFs j to N and R, where j is the index of the IMF with a frequency closest to but below a selected cut-off frequency. The low-pass filtered signal x is expressed in equation 4-3:

$$x = R + \sum_{n=j}^{N} IMF_n \qquad (4\text{-}3)$$

The selected cut-off frequency must ensure that all the IMFs retained can reasonably be expected to contribute to the predicted signal over the selected time horizon.

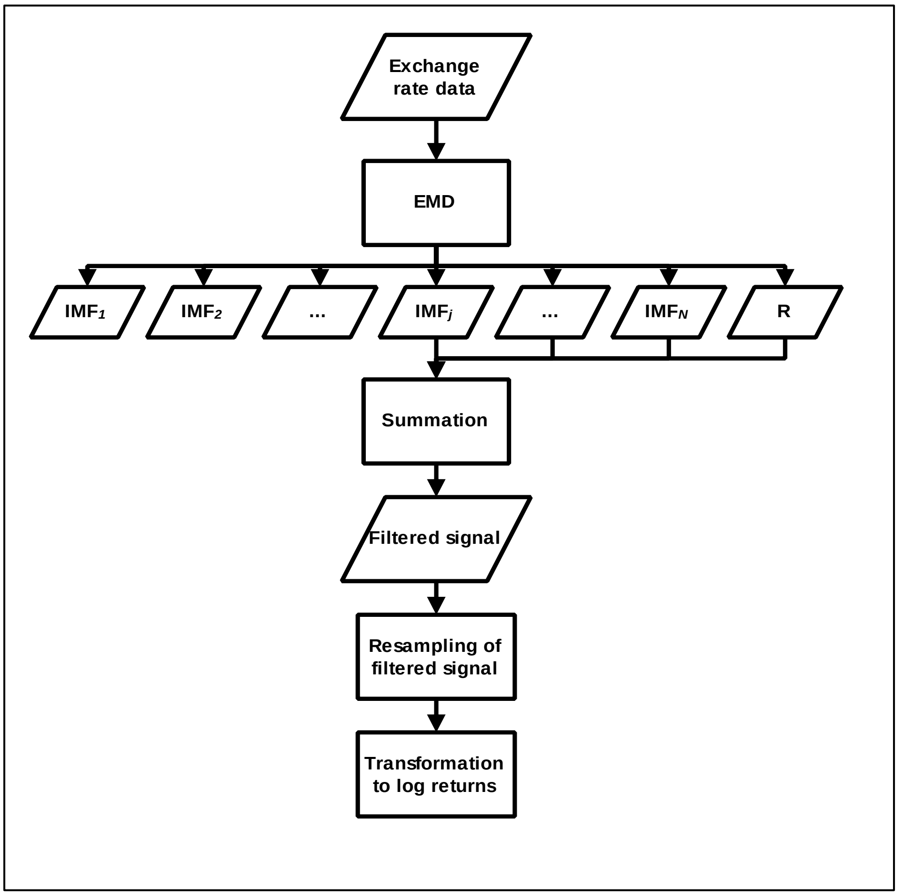

Figure 4-4: The proposed EMD-filter

4.4.1 Cut-off frequency selection

The cut-off frequency will be determined by using the principle of Nyquist's sampling theorem and empirical observations. Nyquist's sampling theorem states that the sampling frequency should be at least twice the highest frequency contained in the signal. If this condition is not met then aliasing occurs, which is the misrepresentation of a signal due to an insufficient sampling rate.

In our case we intend to predict the signal over a predetermined time horizon as explained in the previous section. We furthermore intend to compare the use of a different number of time steps, in each case using a different sampling period, in order to generate the prediction. As explained in Table 4-3 we need to decimate the signal from the initial 1 minute sampling period down to the various selected sampling periods, the longest being equal to the prediction horizon. Nyquist filtering is required to prevent aliasing from occurring before this decimation process can be implemented. In addition we intend to further restrict the bandwidth of the signal to maximize the signal-to-noise ratio in the historic time samples used for prediction over the chosen forecast horizon. The EMD-filter will then serve as an anti-aliasing filter with cut-off based on optimal prediction over the forecast horizon.

As stated in Section 4.4, only the IMFs that can be expected to contribute towards the prediction of the signal over the forecast horizon should be retained. In order to measure the contribution towards the predicted signal the mutual information (MI) between the historic and future time samples, extracted after EMD based filtering was applied with different effective cut-off frequencies, will be calculated. The selected cut-off frequency (corresponding to the retention of IMFs as from a specific number) will be based on maximizing the mutual information between historic and future time samples with lags equal to or exceeding the prediction horizon.

Mutual information is a criterion that is a measure of the dependence between any two variables [71], [72]. The mutual information, MI, of two variables of length n, X and Y, is given in equation 4-4

$$MI = \frac{1}{n}\sum_{i=1}^{n} \ln\left[\frac{f_{X,Y}(x_i, y_i)}{f_X(x_i)f_Y(y_i)}\right] \qquad (4\text{-}4)$$

where $f_{X,Y}(x_i, y_i)$ is the joint probability density of the i_{th} bitvariate sample pairs of X and Y, $f_X(x_i)$ is the univariate probability density of X in the point x_i and $f_Y(y_i)$ is the univariate probability density of Y in the point y_i.

If the two variables are entirely independent, the joint probability of the occurrence of the two variables would be equal to the product of the individual probabilities. Thus $f_{X,Y}(x, y)$ would be equal to $f_X(x)f_Y(y)$, resulting in a MI of 0. The accurate estimation of the mutual information is dependent on the accurate estimation of the joint and univariate probabilities. Following Fraser and Swinney [71], this is calculated using fixed bin histograms.

Using mutual information as evaluation criterion, a number of different filtering options will be compared for each exchange rate, intended to optimize prediction over the ideal forecast horizon and for the different sampling periods as previously determined.

The significant period, T_{fund}, of an IMF is calculated using the discrete Fourier transform given by equation 4-5:

$$T_{fund} = \frac{1}{maxpowerfreq(dftpower[k])} \qquad (4\text{-}5)$$

where $maxpowerfreq(dftpower)$ is the frequency at which the power spectral density $dftpower$ is at a maximum. $dftpower$ is calculated as follows:

$$dftpower[k] = \left| \sum_{n=0}^{N-1} IMF_n[n] e^{-i2\pi k \frac{n}{N}} \right|^2 \qquad (4\text{-}6)$$

Where N is the length of the IMF, n is the index of the current IMF and $k = 0,1,\ldots,N-1$.

The filtering options are implemented by retaining those IMFs for which the lengths of their significant periods exceed the following criteria, defined in terms of the sampling period and forecast horizon (the values in brackets are valid for a sampling period of 7 minutes):

- $\frac{1}{2}$ the sampling period (3.5 minutes)
- 1 time the sampling period (7 minutes)
- 2 times the sampling period (14 minutes)
- $\frac{1}{2}$ the forecast horizon (17.5 minutes)
- 1 time the forecast horizon (35 minutes)
- 2 times the forecast horizon (70 minutes)

The first two options do not satisfy the Nyquist criteria completely, but will still be included in the comparison in order to accurately judge the different filtering periods.

Using these different filtering options we calculated the MI for each exchange rate and for each sampling period over a number of different time lags; due to lack of space we show only the MI graphs for EUR/USD and for a 7 minute sampling period. Figure 4-5 shows an example of the mutual information of the current and lagged values of the log returns for the unfiltered signal. Note that the signal is first lagged by a number of steps equal to the forecast horizon before the analysis is done. A lag of 1 thus indicates 1 sample after the forecast horizon.

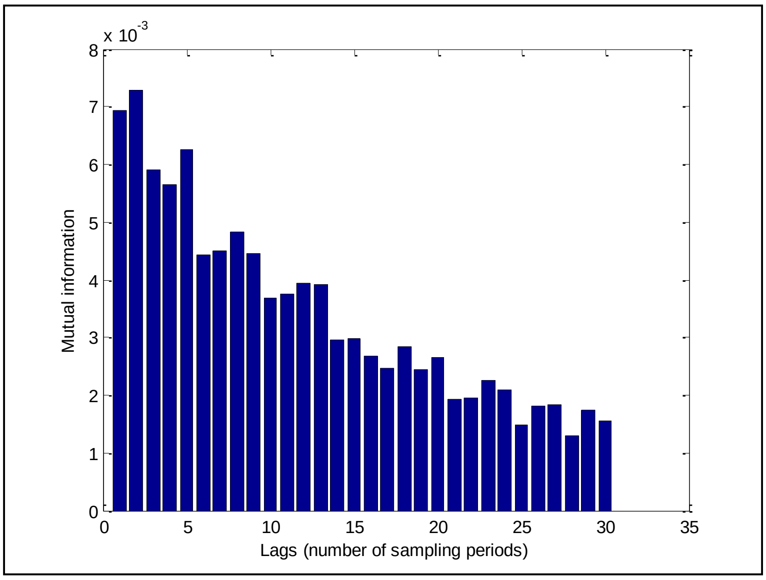

Figure 4-5: Mutual information for the unfiltered EUR/USD log returns sampled at 7 minutes and with a forecast horizon of 35 minutes.

If the high frequency data contains information that is irrelevant for predictions over the forecast horizon it can be expected that the filtered signals will display MI that is higher than the MI of the unfiltered signal for time lags equal to or exceeding the forecast horizon. Figures 4-6 to 4-11 display the MI for the different filtering options as explained above

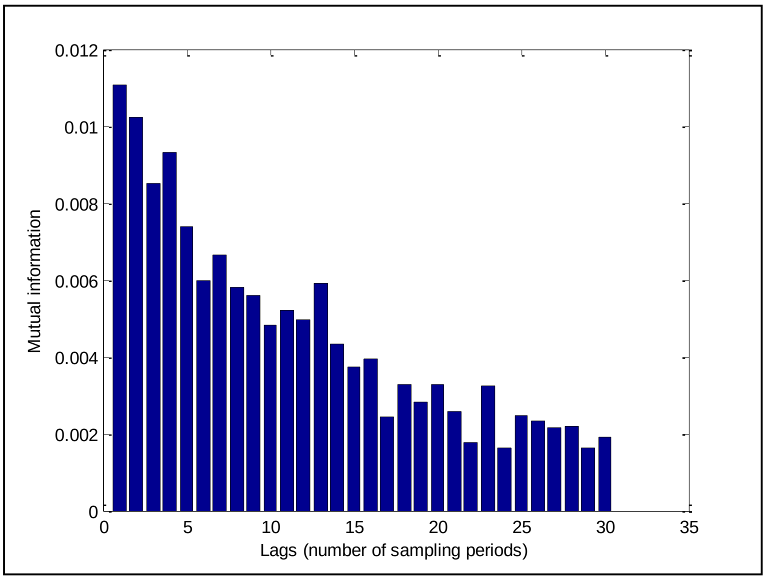

Figure 4-6: Mutual information for the EUR/USD log returns sampled at 7 minutes and with a forecast horizon of 35 minutes. Only IMFs with significant periods larger than half the sample rate (3.5 minutes) are retained.

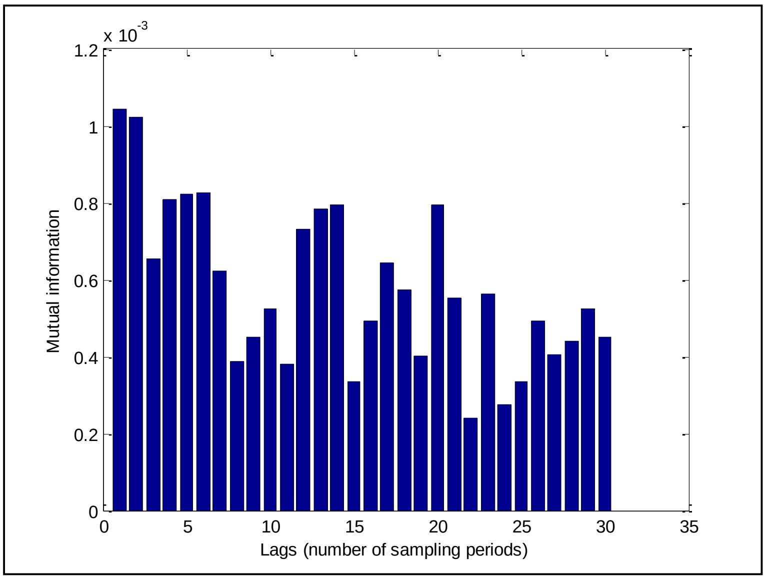

Figure 4-7: Mutual information for the EUR/USD log returns sampled at 7 minutes and with a forecast horizon of 35 minutes. Only IMFs with significant periods larger than the sample rate (7 minutes) are retained.

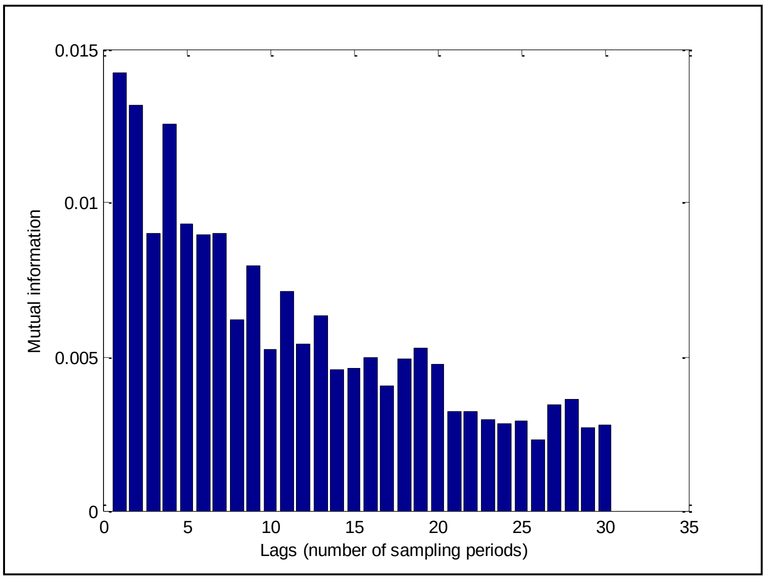

Figure 4-8: Mutual information for the EUR/USD log returns sampled at 7 minutes and with a forecast horizon of 35 minutes. Only IMFs with significant periods larger than twice the sample rate (14 minutes) are retained.

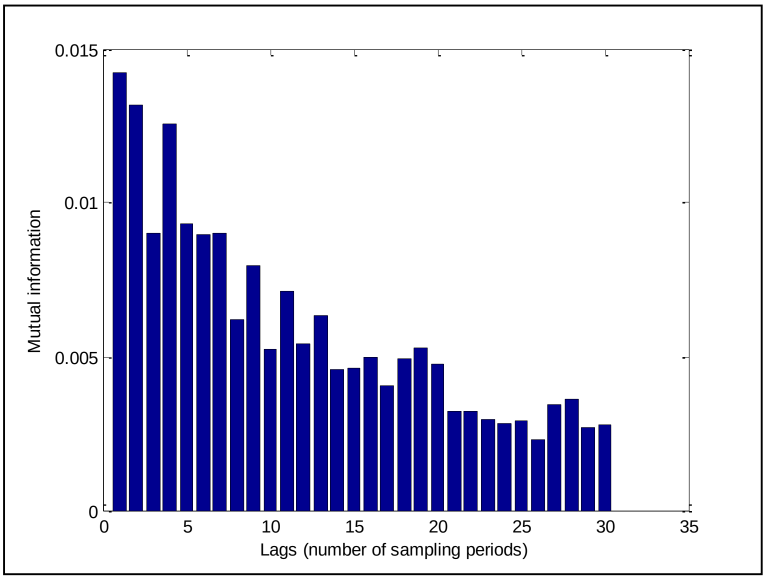

Figure 4-9: **Mutual information for the EUR/USD log returns sampled at 7 minutes and with a forecast horizon of 35 minutes. Only IMFs with significant periods larger than half the forecast horizon (17.5 minutes) are retained.**

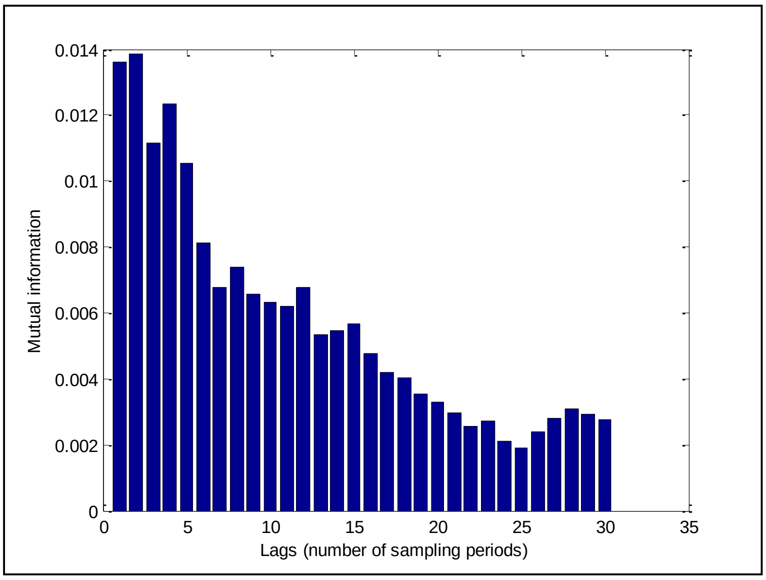

Figure 4-10: **Mutual information for the EUR/USD log returns sampled at 7 minutes and with a forecast horizon of 35 minutes. Only IMFs with significant periods larger than the forecast horizon (35 minutes) are retained.**

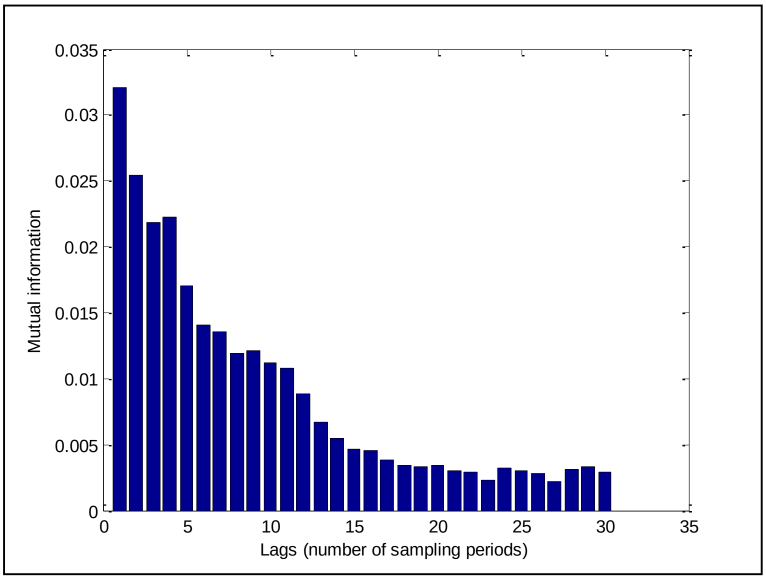

Figure 4-11: Mutual information for the EUR/USD log returns sampled at 7 minutes and with a forecast horizon of 35 minutes. Only IMFs with significant periods larger than twice the forecast horizon (70 minutes) are retained.

From the figures above it can be observed that the signal filtered to only retain IMFs with significant period of at least 2 times the forecast horizon has significantly higher levels of mutual information than any of the other signals. Tests on other exchange rates and at other sample times and forecast horizon combinations support these results. This cut-off frequency also coincides with the Nyquist frequency for the longest sampling period that equals the forecast horizon. Consequently a minimum significant period for IMFs of 2 times the forecast horizon will be used as the cut-off condition; this will not only maximize MI but will ensure that the Nyquist criterium will always be satisfied.

The mathematical conditions for the filter are given in equation 4-7

$$T_c \geq 2\, T_h \qquad (4\text{-}7)$$

where T_c is the cut-off period of the EMD-filter and T_h is the forecast horizon.

The filtering process is described below:

1) Decompose the time series into IMFs using the sifting process described in section 2.4.2.
2) Calculate the most significant period, T_{fund}, of each IMF using Fourier spectral analysis. This is done using equations 4-5 and 4-6:
3) Determine the index of the first IMF where $T_{fund} \geq 2T_h$

4) Use equation 4-3 to calculate the filtered signal

4.5 Neural network design

The design of the neural network encompasses the choice of the neural network architecture, the selection of inputs, the number of hidden nodes and the transfer functions used in each layer. These factors are of critical importance to the prediction accuracy of the proposed model and can therefore not be chosen arbitrarily. A combination of directives from literature as well as practical analyses will be used to determine the optimal choice of parameters.

4.5.1 Neural network architecture

The architecture and number of hidden layers have to be determined for the EMD-filtered neural network model. Section 2.3.3 states the background surrounding the choices in Table 4-4.

Table 4-4: Choice of architecture and number of hidden layers

Parameter	Choice	Rationale
Architecture	Feed forward	The feed forward neural network offers methodical design and a well controlled training process.
Hidden layers	1	The data is not linearly separable, making 0 hidden layers insufficient. Literature shows that more than one hidden layer shows comparable performance to a single hidden layer, but at the cost of increased complexity.
Transfer functions	Hyperbolic tangent sigmoid and linear	The hidden neurons will use a hyperbolic tangent sigmoid transfer function in order to capture the nonlinear relationships in the data. The output neuron will use a linear transfer function as it has been proven that nonlinear transfer functions in the output layer do not make significant contributions to prediction accuracy.

4.5.2 Input selection

The determination of meaningful neural network inputs is of critical importance to the forecasting performance of the proposed model. Literature showed that the exchange rate markets are not entirely effective, allowing forecasting using only historical data. The first problem is to determine the number of historical data points which should be provided to the neural network as inputs. It is possible to use the entire previous year's data as network inputs, but this will result in a very large number of network weights while much of the data is irrelevant to the forecasting horizon. The large number of inputs will lead to an even larger number of degrees of freedom in the network (weights to be trained) – this excessive modelling capacity tends to result in overtraining of the network (i.e. overfitting on the data in the training set) which reduces generalization capability and decreases prediction performance. Another method must therefore be found to help select a sufficiently small number of relevant network inputs.

First consider the maximum allowable inputs for the different exchange rates. There are no fixed formulas to determine the maximum number of inputs for a neural network, but several rules of thumb exists that have been empirically proven to be fairly accurate:

- The number of hidden neurons should be $\frac{2}{3}$ the size of the input layer plus the size of the output layer [21].
- The training set should exceed the number of weighted connections in the neural network by at least 10 times for effective training [73].

Equation 4-8 is used to calculate the number of weighted connections in a fully connected network that includes bias nodes:

$$W = (A+1)B + (B+1)C \qquad (4\text{-}8)$$

Where W is the number of weighted connections, A is the number of input neurons, B is the number of hidden neurons and C is the number of output neurons [74] .

Using the guideline that the number of hidden neurons should be $\frac{2}{3}$ the number of input neurons plus the number of outputs, the statement of $B = \frac{2}{3}A + 1$ can be made. As there will only be one output, the equation reduces to

$$W = (A+1)(\frac{2}{3}A + 1) + \left(\frac{2}{3}A + 1 + 1\right)1$$

Rearrange to solve for A:

$$A = 3\frac{(\frac{8W}{3} - \frac{23}{9})^{1/2}}{4} - \frac{7}{4} \qquad (4\text{-}9)$$

Under the assumption that the size of the training set (spanning one calendar month) should be at least 10 times the number of weighted connections, the maximum number of inputs can be calculated for each exchange rate and sample rate pair. These maximum inputs can be seen in Table 4-5.

Table 4-5: The maximum number of inputs for each neural network

Exchange rate	Sampling period	Size of training set	Maximum number of inputs
EUR/USD	1	30707	66
	5	6141	29
	7	4386	24
	35	877	10
USD/JPY	1	30712	66
	2	15356	46
	5	6142	29
	10	3071	19
USD/ZAR	15	1744	14
	30	872	10
	60	436	6
	90	290	5
	180	145	3

Frenay et al. [75] identified mutual information as an optimal method for feature selection in machine learning applications, specifically for regression of series with uniform, Laplacian or Gaussian noise. Cao and Soofi [76] also recommend mutual information as a feature selection criterion for nonlinear and noisy exchange rate data . See section 4.4.1 on how the mutual information is calculated. To evaluate the candidate inputs we first calculated the mutual information between the target data (EMD-filtered log returns at a specific time) and the possible lagged inputs (EMD-filtered log returns with sequential lags at least equal to the forecast horizon) for each month and then calculated the average mutual information over all the months. This average mutual information is calculated for historic samples lagged by up to 100 sampling periods with respect to the target value, and for every exchange rate, sampling period and forecast horizon. The analysis is limited to 100 historic sampling periods in order to limit the complexity of the neural network due to limits on the amount of historic data available for training [73].

Appendix C shows the average mutual information for the 100 lagged values. We use the method proposed by Cao and Soofi [76] and an adapted form of the method proposed by Fraser and Swinney for selecting the number of delayed inputs [71]. Subjective evaluation of the significant mutual information is the norm, but we propose the following technique in order to assist with the evaluation: the mean and standard deviation of the mutual information is calculated. Any lagged values with mutual information larger than the mean plus one standard deviation will be considered for inputs (as long as the addition of that input does not exceed the condition for the maximum number of inputs). The reasoning behind this is that the average mutual information for a sufficiently long noisy signal tends to be insignificant for signal prediction. Any lagged values within one standard deviation also do not contribute significantly to signal prediction. Thus only lagged values with mutual information larger than one standard deviation are considered. Table 4-6 shows the number of lagged values to be used as inputs for each case that is obtained using this method. The USD/ZAR exchange rate does not show any significant mutual information for the larger sample periods, and the inputs are chosen based on the size of the mutual information within the limits of the maximum allowed number of inputs.

Table 4-6: The number of lagged values that will be used as inputs for the different exchange rates and sample times

Exchange rate	Sample rate	Number of lagged values to use as inputs
EUR/USD	1	18
	5	16
	7	14
	35	4
USD/JPY	1	15
	2	15
	5	11
	10	11
USD/ZAR	15	4
	30	3
	60	3
	90	2
	180	2

4.5.3 Hidden neurons

The recommended number of hidden neurons is dependent on the number of inputs and outputs of the neural network and is described in equation 4-10:

$$\mathrm{B} = \frac{2}{3}A + C \qquad (4\text{-}10)$$

Where B is the number of hidden neurons, A is the number of input neurons and C is the number of output neurons. The number of hidden neurons for each case is calculated in Table 4-7.

Table 4-7: The number of hidden neurons for the different neural networks

Exchange rate	Sample rate	Number of hidden neurons
EUR/USD	1	13
	5	12
	7	10
	35	4
USD/JPY	1	11
	2	11
	5	8
	10	8
USD/ZAR	15	4
	30	3
	60	3
	90	2
	180	2

4.5.4 Neural network training

The training of the neural networks can commence now that the architecture has been determined. The training data has been verified to be sufficient for training the number of weighted connections as calculated in section 4.5.2.

In order to validate the results, it was decided to use one year of exchange rate data that is divided into 12 windows of equal size, with each window representing a month. With this division of the data it would be possible to train the neural network on one month's worth of data, and generate forecasts using the returns for the next month. With this approach a total of 11 months of data will be forecasted using 11 months' worth of training data. This will indicate if the accuracy of the forecasts depends on the selection of the specific training and test sets and hence allow better validation of the robustness of the results. The training of the neural network will proceed according to the following steps:

1) Divide the data into 12 equally sized windows
2) Use the initial window as training data. This is done by dividing the window into 3 sets: training, testing and validation sets. The observations in the testing data are divided randomly in the following ratios:
    - 70% of the observations will compose the training set
    - 15% of the observations will compose the in sample testing set
    - 15% of the observations will compose the validation set
3) Train the neural network using the Levenberg-Marquardt training algorithm described in section 2.3.5 and appendix A.
4) Forecast the returns over the duration of the following month by feeding the relevant lagged inputs to the trained neural network.
5) Repeat steps 2 to 4 for every subsequent month until returns for 11 months have been forecast.

Two approaches exist for multiple-step-ahead prediction and, dependent on the approach chosen, affects the target values used in the training procedure. The approaches are iterative or direct multi-step-ahead prediction. The iterative procedure forecast one-step-ahead and, using the obtained prediction, iterates until the required number of steps have been reached. The direct approach trains the neural network on target values that are already the required number of steps in the future. Literature shows the direct approach to be more accurate as errors are prone to compound when using the iterative approach [68].

4.5.5 The EMD-filtered neural network model

The EMD-filtered ANN tries to exploit the adaptive properties of EMD-based filters in order to improve the signal-to-noise ratio of the training data fed to a predictive feed forward neural network. The basic structure of the proposed EMD-filter was shown in Figure 4-4. The exchange rate data is decomposed using EMD into *n* IMFs and a residue. The cut-off for retention of IMFs is determined based on the MI maximization criterion developed in the previous section (which must also satisfy the Nyquist criterion). The IMFs from *j* to *n* and the residue, *R*, are summed in order to produce the non-linear adaptive low-pass filtered signal, where IMF_j is the IMF with a fundamental period larger than or equal to the selected cut-off period of two times the forecast horizon. This filtered signal is then resampled at the sampling periods chosen in Table 4-3 and transformed to log returns. The artificial neural network uses this signal for training, testing and forecasting purposes.

A more detailed architecture of the neural network is given in Figure 4-12. The configuration is that of an acyclic feed forward neural network with *n* inputs (i_1 to i_n), m hidden nodes (h_1 to h_m) and one output (o_1). The optimal number of inputs and hidden nodes were calculated in previous sections. The number of input neurons was calculated in section 4.5.2, and the number hidden neurons was calculated in section 4.5.3.

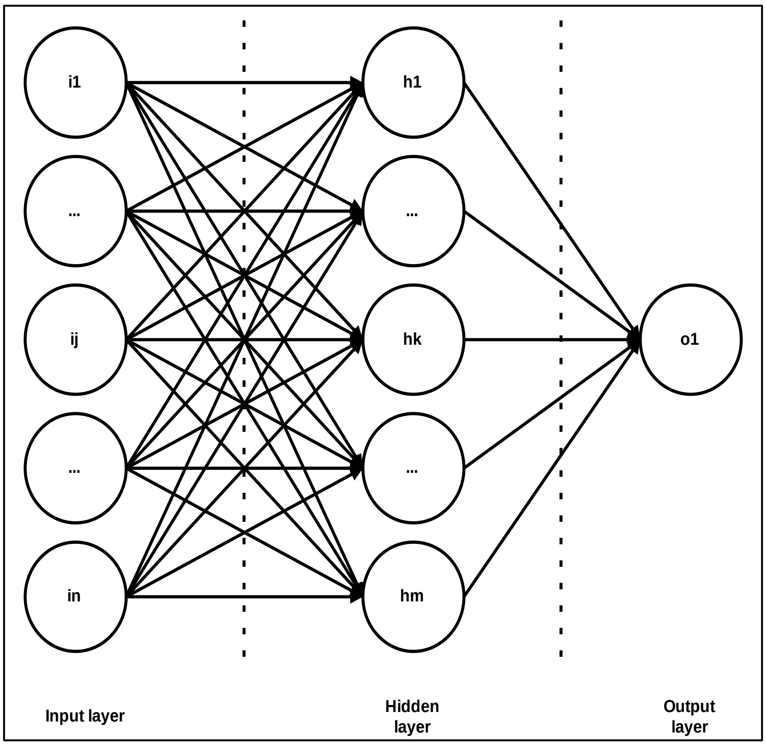

Figure 4-12: The feed forward neural network structure

4.6 Comparative models

Two other models will be evaluated in order to provide a comparison with the EMD-filtered neural network model. These models are a random walk model and an unfiltered neural network.

4.6.1 Random walk model

The random walk model has been a benchmark since the early 1980's, when Meese and Rogoff published their findings on monetary models [39]. The most basic form of a random walk is the naive random walk. This model assumes that the next value is equal to the most recent historic value. The naive random walk also serves as a baseline for market efficiency. According to the efficient market hypothesis, the most recent value will be the most relevant value as all data is already captured therein. If a model is able to consistently outperform the naive random walk model using publically available information, then this could indicate market inefficiency. A naive random walk model will thus be used as a benchmark model.

4.6.2 Unfiltered neural network

In order to accurately judge the effects that EMD-filtering has on forecasting performance it is necessary to compare the model with an unfiltered neural network. This feed forward neural network will have the exact same architecture as the EMD-filtered neural network with the same sample period and forecast horizons.

4.7 Performance evaluation

4.7.1 Performance evaluation criteria

Several criteria will be used to evaluate the prediction performance of the models. The naming conventions used for variables and indices are first introduced:

- t is the index of time values, from 1 to T, where T is the number of samples
- p_t is the predicted value at time t
- a_t is the actual value at time t

Following the research of Yu et al., Wang et al., Fu, Yang et al. and Xiong et al. [62–64], [66], [68], the statistical performance evaluation criteria of root-mean-square-error ($RMSE$) and directional symmetry (DS) are chosen.

$RMSE$ is a common statistical measure of the difference between the actual and predicted values and is given in equation 4-11

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(a_t - p_t)^2} \qquad (4\text{-}11)$$

For a random walk the RMSE should be approximately equal to the standard deviation of the signal.

DS is a statistical measure of a model's performance in the prediction of change from one period to the next, and is given in equation 4-12.

$$DS = \frac{100}{T}\sum_{t=1}^{T} d_t \qquad (4\text{-}12)$$

where

$$d_t = 1 \quad (a_t - a_{t-1})(p_t - p_{t-1}) \geq 0$$

$$d_t = 0 \qquad otherwise$$

For a random walk the DS should not be expected to exceed 0.5.

As additional performance evaluation measurement the Pearson correlation coefficient will also be calculated. The Pearson correlation coefficient is a measurement of the linear dependence between two variables, where a value of 1 is complete positive correlation, a value of 0 indicates no correlation and a value of -1 is complete negative correlation. The Pearson correlation coefficient ($corr$) is given in equation 4-13.

$$corr = \frac{cov(a,p)}{\sigma_a \sigma_p} \qquad (4\text{-}13)$$

where

$cov(a,p) = E[(a - \mu_a)(p - \mu_p)]$

E is the mathematical expectation

σ_a is the standard deviation of the actual time series

σ_p is the standard deviation of the predicted time series

μ_a is the mean value of the actual time series

μ_p is the mean value of the predicted time series

For a random walk the correlation would be expected to be approximately 0.

The final performance evaluation measure is the simulated compound returns generated by a basic trading strategy. Figure 4-13 describes the trading strategy used to calculate the simulated returns. The strategy evaluates whether the predicted returns for the next period will be positive or negative. If the expected returns are positive a long position should be taken, otherwise a short position should be taken. The transaction costs of the bid/ask spread are deducted whenever the position changes. Whenever a long position is held, the returns generated are calculated using the actual return rates. Because the actual and predicted rates of return are logarithmic, the calculation of the compound return is a simple task. The compound return over the time period (ret) is calculated using equation 4-14

$$ret = e^{log\,ret} - 1 \qquad (4\text{-}14)$$

where

$$logret = \sum_{t=1}^{T} a_t - transcost \qquad (4\text{-}15)$$

for every t where the predicted return is positive. When the strategy changes position, the transaction costs imposed by the bid-ask-spread are deducted. When the position is unchanged to the previous position, the transaction costs are zero.

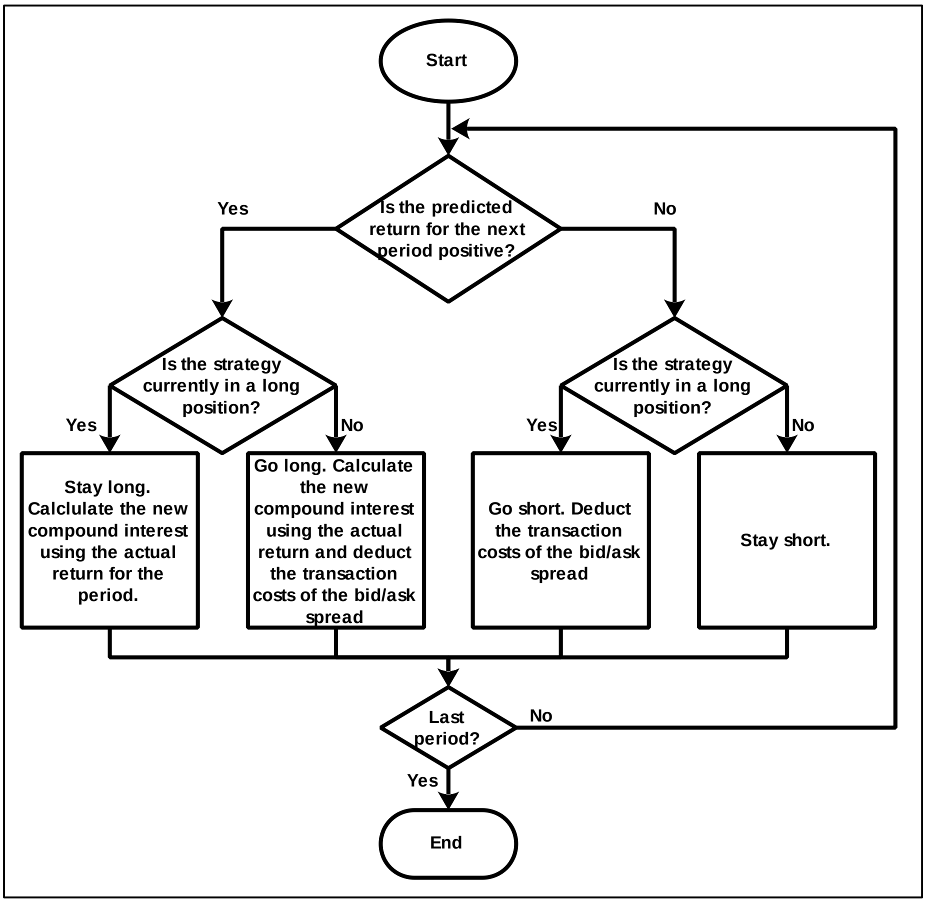

Figure 4-13: Flow diagram of the trading strategy used to calculate the simulated returns

4.7.2 Combinations of model parameters to evaluate

In order to accurately judge the performance of the proposed model, it is required to evaluate it for the complete set of cases to be considered. Combinations of the following fields will be used in these tests

- 3 models: EMD-filtered ANN, unfiltered ANN and random walk.
- 13 sampling periods: 4 sampling periods each for the EUR/USD and USD/JPY rates, and 5 sampling periods for the USD/ZAR rate, each with the same forecast horizon in terms of minutes but different in terms of number of samples.

- 11 training periods: February to December of 2013 (January will only be used as training data for February's forecasts, with each subsequent month using the previous month as training data).

This leads to a total of 429 separate cases for which a model training and evaluation exercise will be conducted. All the criteria described in section 4.7.1 will be evaluated in order to judge the performance of each of the combinations.

4.8 Chapter summary

This chapter provides the methodology used for the design of the EMD-filtered neural network. The sections in this chapter can be summarised as follows:

Section 4.2 selects the EUR/USD, USD/JPY and USD/ZAR exchange rates as the test data. The original exchange rates are sampled at one minute and the population that is considered includes the whole duration of 2013.

Section 4.3 covers the pre-processing of the data. The selection of forecast horizons is based on the maximum ideal returns of the respective exchange rates. Sample rates and forecast horizon combinations are then chosen that coincide with the maximum ideal return rate. This section also describes the rationale and method behind the log return transformation of the data.

Section 4.4 describes the EMD-filtering procedure. The logarithmic returns are filtered using EMD with a cut-off period that is equal to twice the forecast horizon. The cut-off period was determined empirically using the mutual information criterion.

Section 4.5 describes the choice of neural network architecture and the determination of the parameters and inputs. A 3 layer feed-forward neural network will be used with a single hidden layer. The number of inputs and hidden neurons are calculated for each exchange rate and sample time combination. The Levenberg-Marquardt algorithm will be used to train the neural net with the previous month's data.

Section 4.6 describes the benchmark prediction models whose results will be used for comparison with the EMD-filtered neural network. These two models are a naive random walk model and an unfiltered neural network with the same architecture and parameters as the EMD-filtered model.

Section 4.7 describes the criteria that will be used for performance evaluation as well as all the combinations of parameters that will be evaluated. Root-mean-square-error, directional symmetry, the Pearson correlation coefficient and simulated returns will be evaluated for the 429 possible model parameter combinations.

5. Results

"Results! Why, man, I have gotten a lot of results. I know several thousand things that won't work."

\- Thomas A. Edison

5.1 Introduction

This dissertation intends to test several hypotheses relating to the techniques that are being investigated:

- Can financial time series be divided into distinct behavioural modes, each predominantly acting on a specific time scale?
- How effective is the EMD technique to implement this division of a signal into intrinsic modes that are purely derived from information contained within the signal itself?
- Can a single IMF or a subset of IMFs be predicted into the future using a neural network technique to produce more accurate results compared to predictions obtained by applying the same technique to the original time series?

In order to investigate the above hypotheses a number of analyses were conducted to produce results based on which conclusions can be made. As explained in chapter 4 the EMD technique is used to decompose a financial time series into IMFs; those IMFs that are deemed not to contribute to systematic behaviour of the signal over the chose forecasting horizon are filtered out. A neural network is then used to predict future returns by using EMD-filtered historical returns.

The analysis of the forecasting performance is completed in the following parts. Firstly the EMD-filter is verified to improve the forecasting performance of the neural network model. Secondly the neural network is verified to give accurate and believable results without overfitting to the data. Thirdly the forecasting performance of the EMD-filtered ANN model is compared to the benchmark models for a specified set of performance evaluation criteria. Lastly the returns generated by the models are validated using 11 fold cross validation and two sample t-tests.

5.2 Verification of the EMD-filter

The EMD-filter is introduced in order to increase the predictability of the exchange rate for a specified forecast horizon. It is therefore of importance to verify that the filter contributes to the predictability before it is applied to the hybrid neural network model. The first verification entails the evaluation of the end-effect of the EMD-filter, as mentioned in section 2.4.5. The second verification is to evaluate the correlation between predicted returns and the filtered and unfiltered actual returns. Before the above verifications are performed we described in section 5.2.1 below how the filtering and subsampling is performed and illustrate through examples how the unfiltered, filtered and subsampled signals differ from each other.

5.2.1 Comparison of EMD-filtered signals and original signals

The EMD-filter is designed to filter out the higher frequency components that are not expected to contribute to the prediction of a signal at a chosen forecast horizon. The sequence in which this is done is of importance, as it can have a significant impact on the results. The sequence that was implemented in this research is:

- Obtain the one minute sampled exchange rate.
- Filter this exchange rate using the EMD-filter, filtering out IMFs with period smaller than twice the forecast horizon (70 minutes for the EUR/USD predictions).
- Subsample the filtered and unfiltered exchange rates to the required level (5, 7 and 35 minutes for the EUR/USD predictions).
- Convert the filtered and unfiltered exchange rates to logarithmic returns.

This sequence ensures that the data is adequately filtered. If the sequence is changed and the data is sampled before it is filtered it would be meaningless to filter the data sampled at times equal to the forecast horizon as there would be no IMFs to omit; more seriously this would have breached the Nyquist criterion for all but the highest sampling rates. If the returns are calculated before the data is sampled, then the sampled returns for all sampling rates would be calculated on a one sample basis. As we intended to demonstrate the impact of calculating returns based on different sampling rates, the decision was to first subsample before log returns are calculated.

Figure 5-1 gives an example of the filtered and unfiltered exchange rates for a 500 minute window for different sampling times. Figure 5-2 shows the log returns of the filtered and unfiltered signals for the corresponding exchange rates in Figure 5-1. It can be seen that while the signals with lower sampling rates do not accurately represent the 1 minute sampled unfiltered signal, their EMD-filtered versions more closely resemble the filtered 1 minute sampled signal.

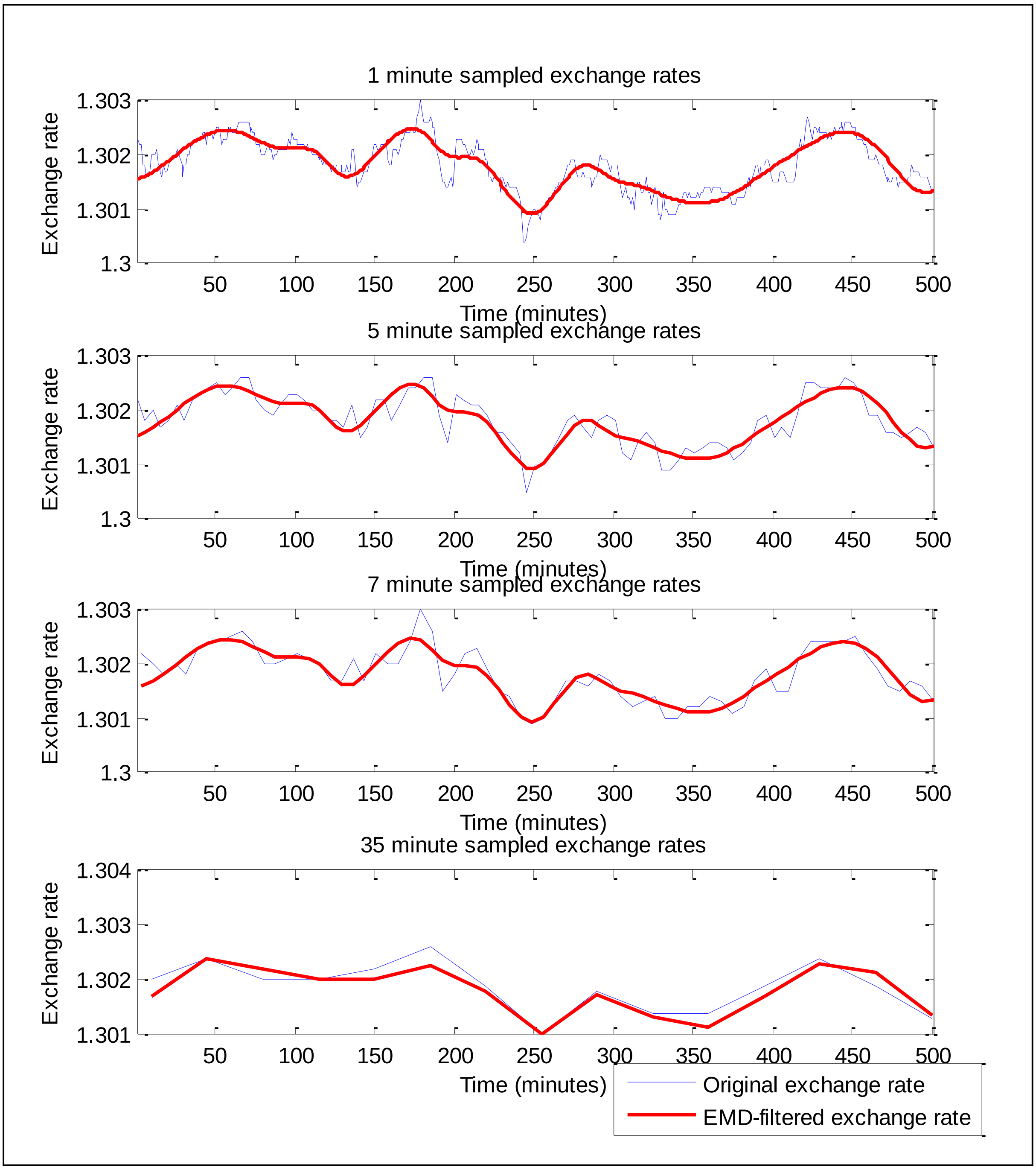

Figure 5-1: Example of the filtered and unfiltered EUR/USD exchange rates for a 500 minute window

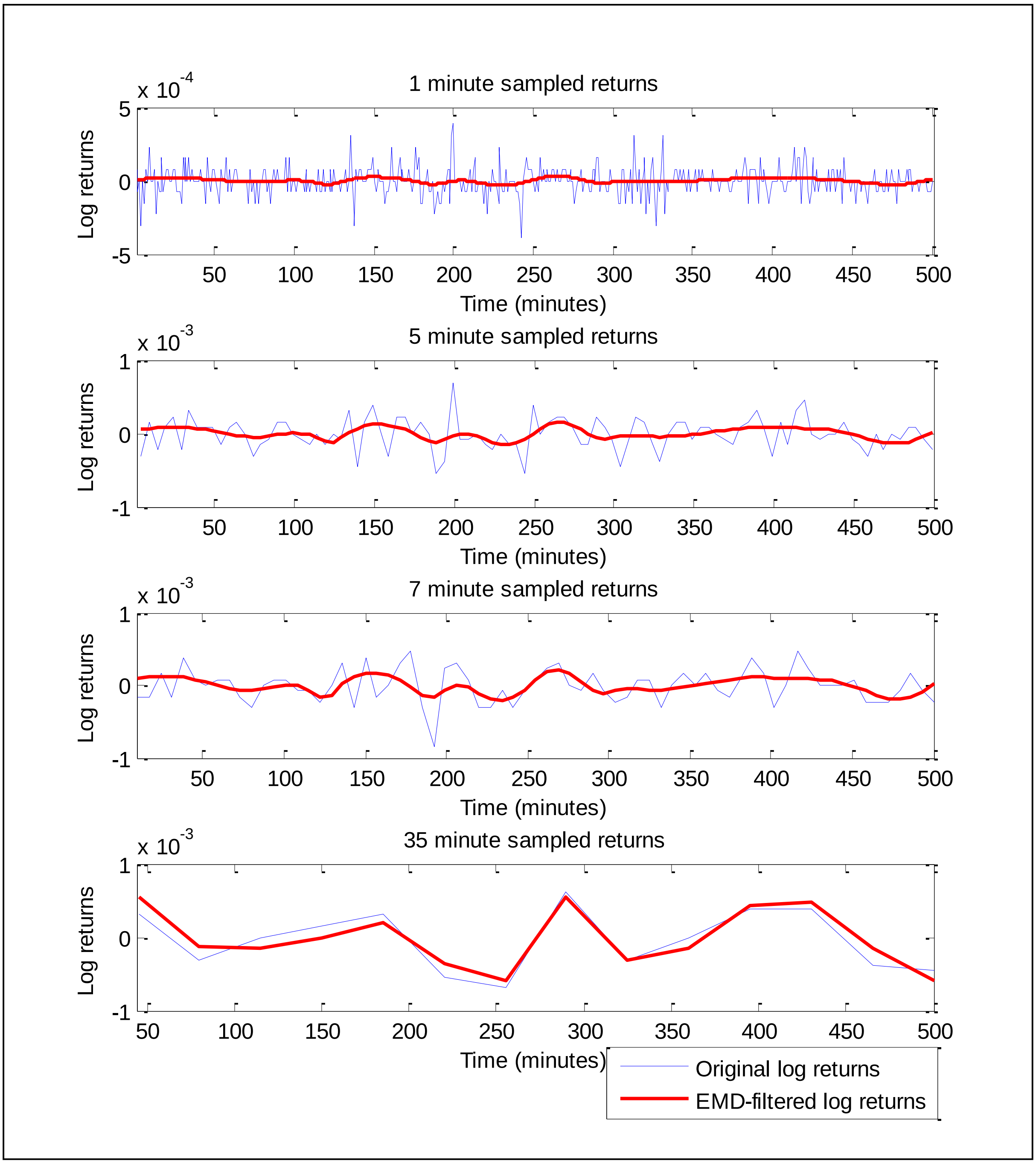

Figure 5-2: Example of the filtered and unfiltered EUR/USD log returns for a 500 minute window

5.2.2 Impact of EMD-filter end effect

In section 2.4.5 the end-effect problem was identified when using EMD. This problem occurs due to the required extrapolation of the cubic spline fitted envelopes at the extreme ends of the series. Mirror mode extrapolation was identified as a possible solution to this problem, and is implemented in the EMD code.

This section will evaluate the impact that the mirror mode extrapolation has on the end of the EMD-filtered time series.

For the technique that we have chosen a forecast must be made using a neural network fed with inputs taken from the EMD-filtered signal extracted just before the forecast period starts; the neural network output is supposed to accurately predict the EMD-filtered signal extracted after the forecast period has been completed. In order to evaluate the impact of the EMD end effect a comparison must therefore be made of the EMD filtered signal extracted just before the forecast period starts and the EMD-filtered signal extracted just after the forecast period has been completed. If these signals show similar behaviour and a high correlation over the period just prior to the forecast then the extrapolation should not have a significant negative impact on the accuracy of predictions.

Figures 5.3 to 5.15 show the comparison of the EMD-filtered log returns using available data from just before a prediction is made and the EMD-filtered data for an extended window. The size of the extension is equal to the forecast horizon as displayed in Table 4-3, while the filter cut-off period is 2 times the forecast horizon, as discussed in section 4.4.1. A forecast is made using a neural network trained on the limited window EMD-filtered data in order to provide an estimation of the forecasting accuracy of the EMD-filtered ANN.

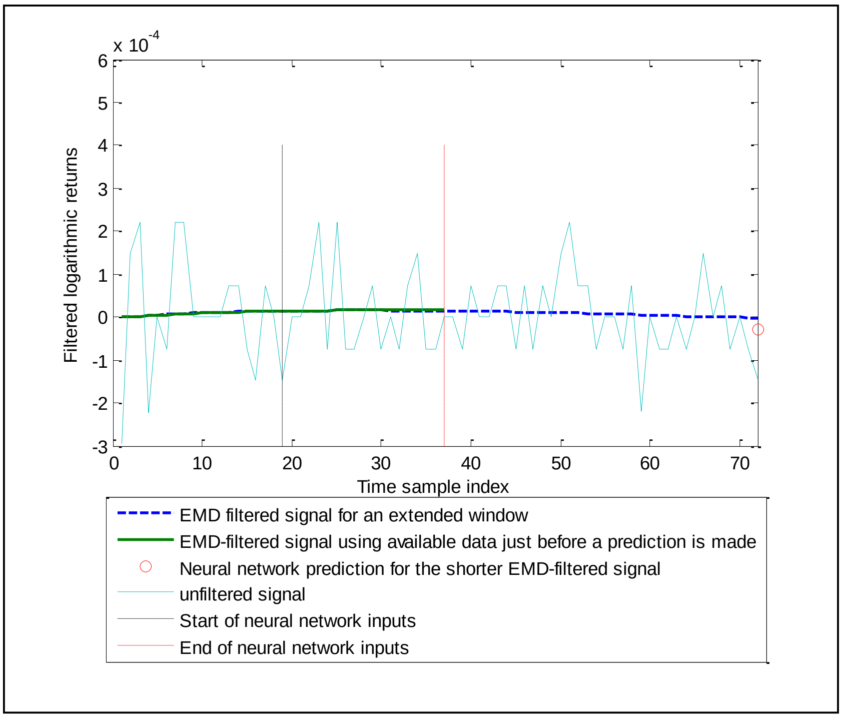

Figure 5-3: Analysis of the EMD-filter end effect for the 1 minute sampled EUR/USD log returns with a forecast horizon of 35 minutes

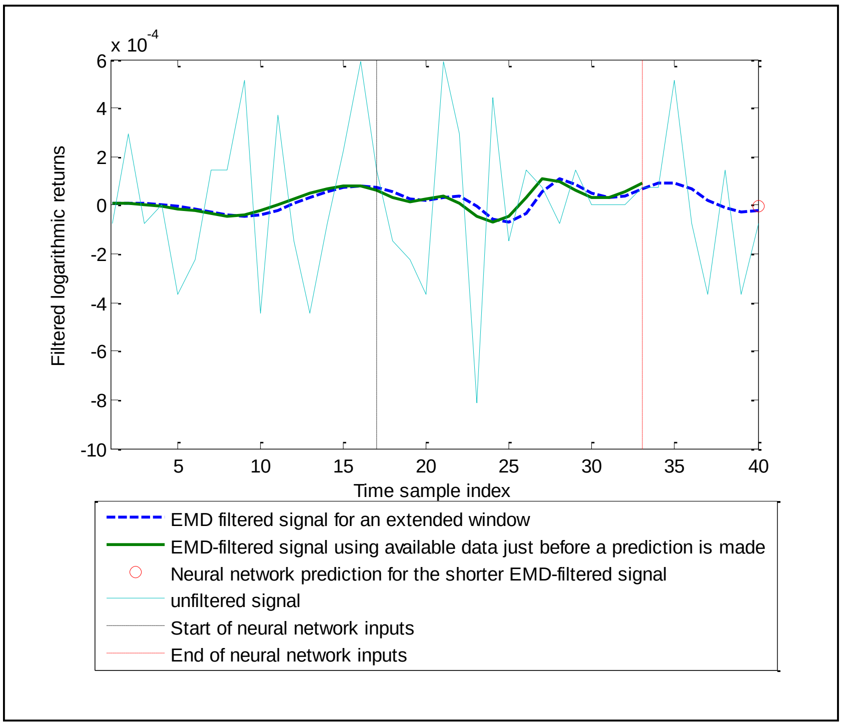

Figure 5-4: Analysis of the EMD-filter end effect for the 5 minute sampled EUR/USD log returns with a forecast horizon of 35 minutes

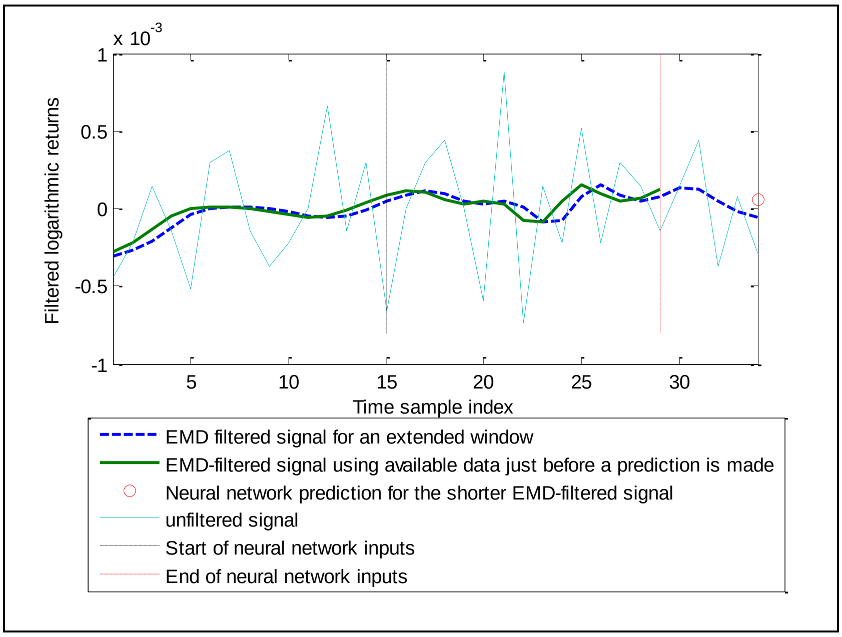

Figure 5-5: Analysis of the EMD-filter end effect for the 7 minute sampled EUR/USD log returns with a forecast horizon of 35 minutes

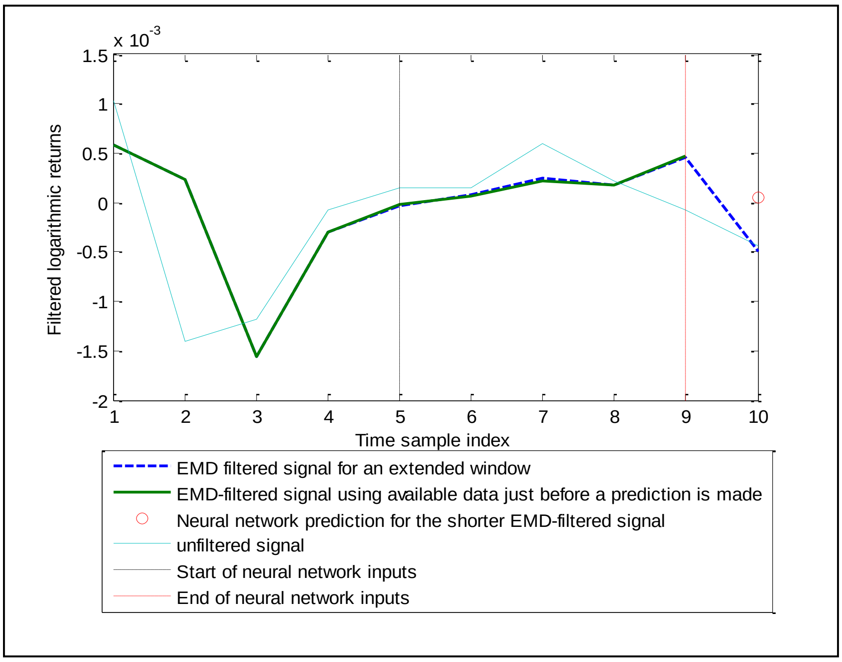

Figure 5-6: Analysis of the EMD-filter end effect for the 35 minute sampled EUR/USD log returns with a forecast horizon of 35 minutes

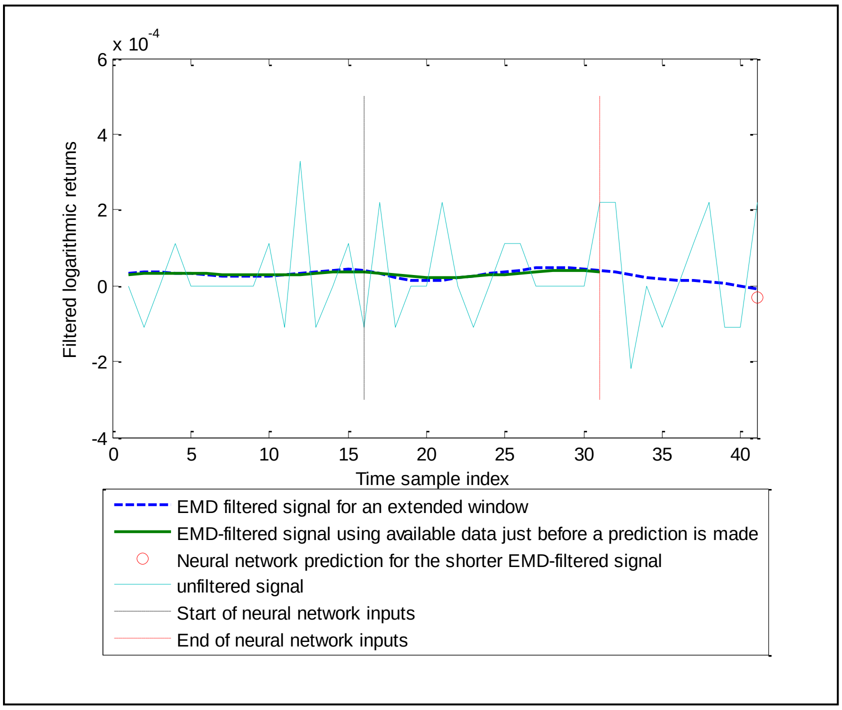

Figure 5-7: **Analysis of the EMD-filter end effect for the 1 minute sampled USD/JPY log returns with a forecast horizon of 10 minutes**

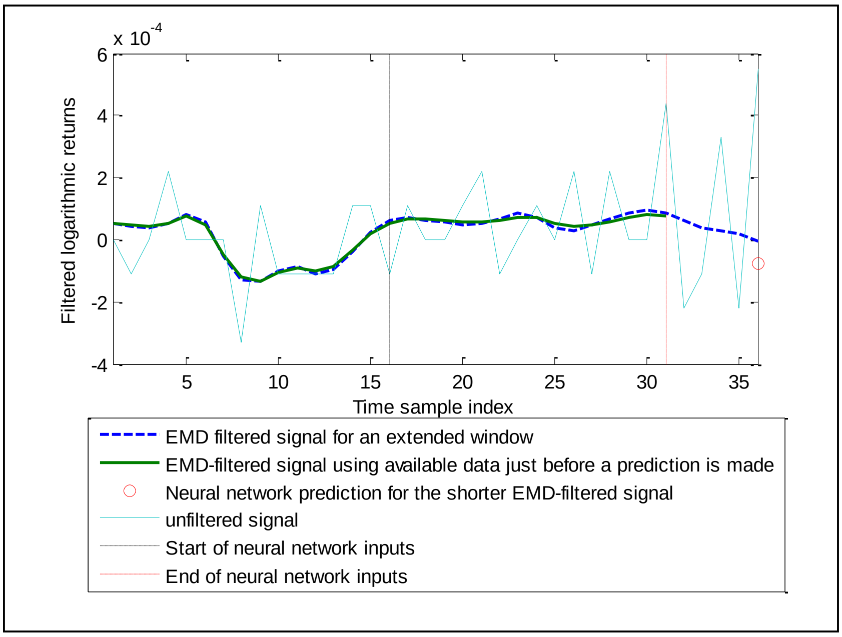

Figure 5-8: **Analysis of the EMD-filter end effect for the 2 minute sampled USD/JPY log returns with a forecast horizon of 10 minutes**

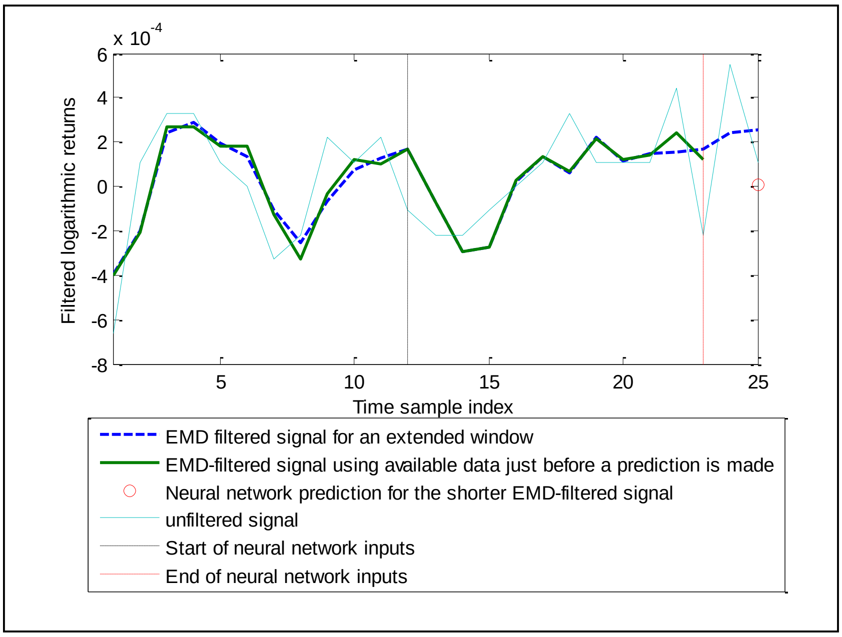

Figure 5-9: Analysis of the EMD-filter end effect for the 5 minute sampled USD/JPY log returns with a forecast horizon of 10 minutes

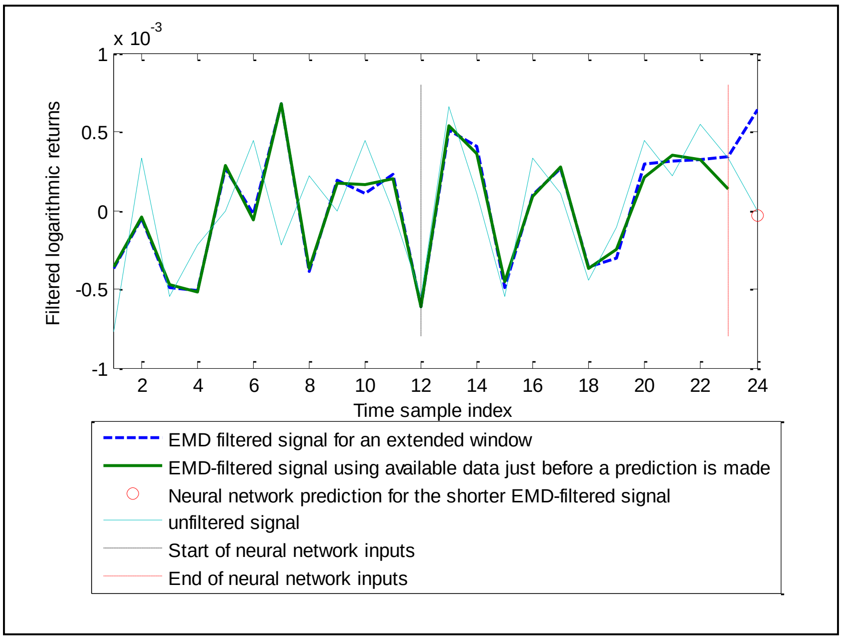

Figure 5-10: Analysis of the EMD-filter end effect for the 10 minute sampled USD/JPY log returns with a forecast horizon of 10 minutes

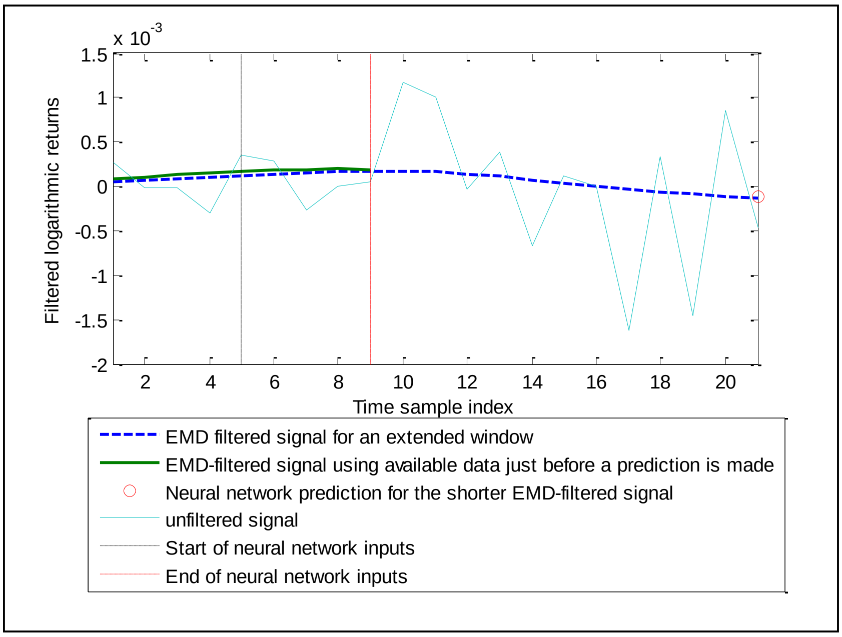

Figure 5-11: Analysis of the EMD-filter end effect for the 15 minute sampled USD/ZAR log returns with a forecast horizon of 180 minutes

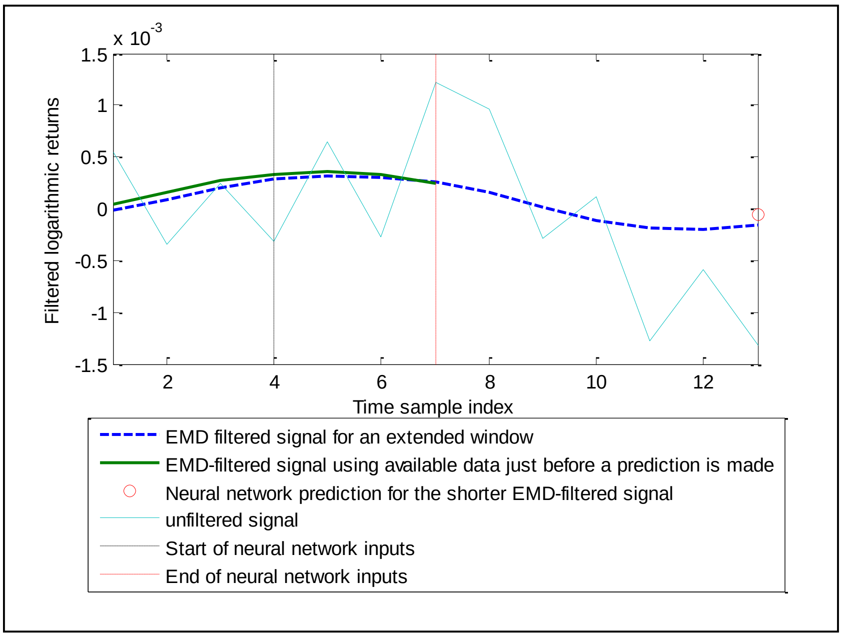

Figure 5-12: Analysis of the EMD-filter end effect for the 30 minute sampled USD/ZAR log returns with a forecast horizon of 180 minutes

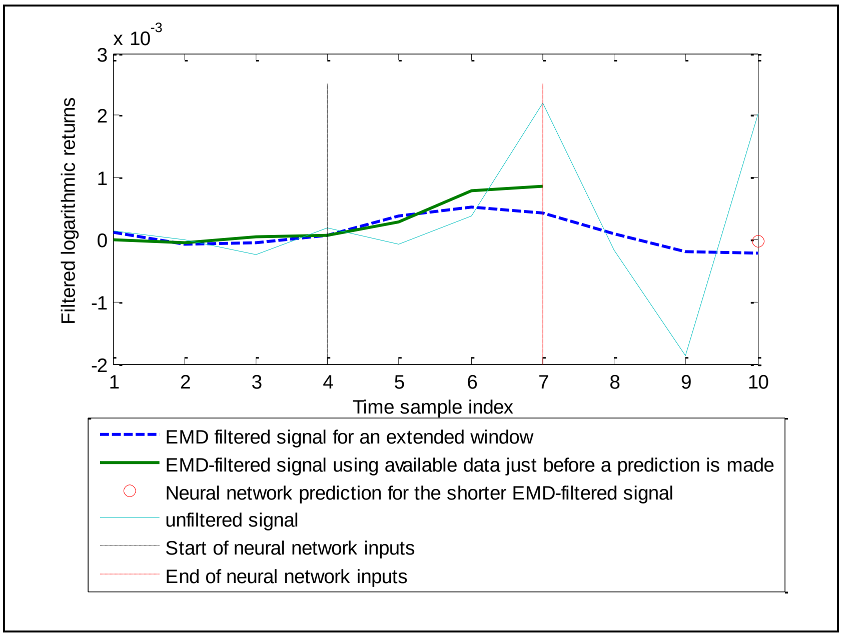

Figure 5-13: Analysis of the EMD-filter end effect for the 60 minute sampled USD/ZAR log returns with a forecast horizon of 180 minutes

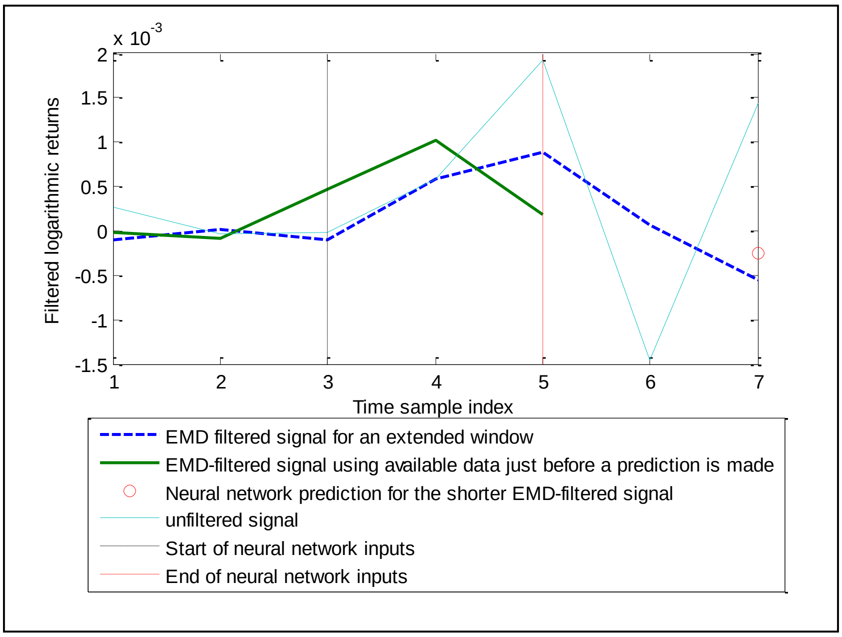

Figure 5-14: Analysis of the EMD-filter end effect for the 90 minute sampled USD/ZAR log returns with a forecast horizon of 180 minutes

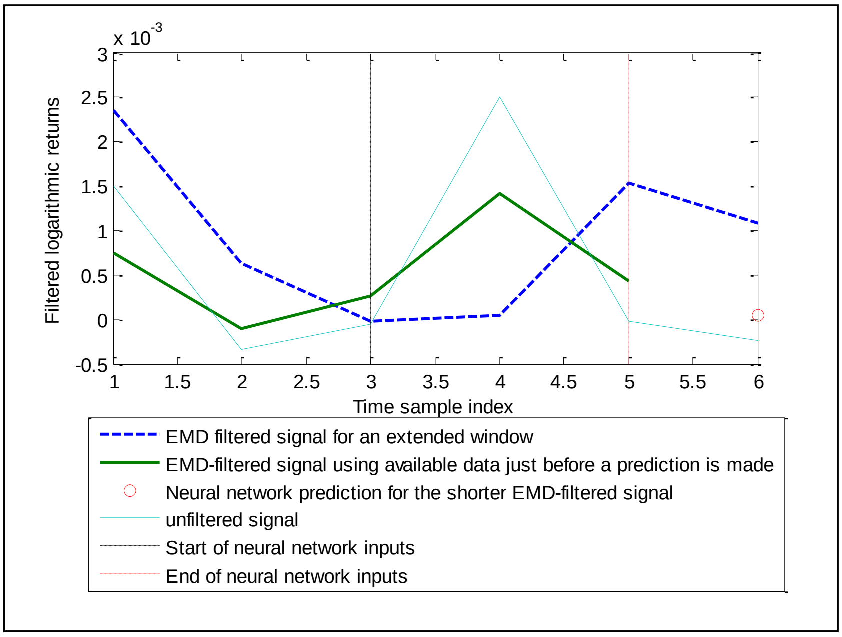

Figure 5-15: Analysis of the EMD-filter end effect for the 180 minute sampled USD/ZAR log returns with a forecast horizon of 180 minutes

From the above figures it can be seen that there is a close correspondence between the two EMD-filtered signals prior to the start of the forecast period. This correspondence however deteriorates as the sampling rate at which the log returns were calculated is decreased. The correlation between the limited and extended EMD signals will now be evaluated for the windows that will serve as inputs to the neural networks.

For the EUR/USD exchange rate the correlation of the input windows ranges from 0.8301 to 0.9977. The predictions with the limited window filtered signal and the neural net are relatively close to the value of the extended window at that time. This shows that inaccuracies arising from the mirror mode extrapolation technique should not have a significant negative impact on the neural network predictions.

The USD/JPY exchange rate shows behaviour similar to the EUR/USD rate. The two EMD windows are very closely correlated with correlations ranging from 0.9169 to 0.9866.

The USD/ZAR exchange rate graphs differ a bit from the previous observations. While there is still a strong correlation ranging from 0.8523 to 0.9355 for the shorter sample times (15 to 60 minutes), larger differences appear for the longer sample times. However this should not be a significant problem as the predicted returns are still relatively close to the value of the extended signal at the forecast horizon.

From these selected examples it is possible to observe that the use of mirror mode extrapolation as part of the EMD process ensures that the end effects of the filtered signals do not severely impact the forecasting performance of the neural network, and that the filtering technique is viable for exchange rate prediction. The ability of the ANN trained with the EMD-filtered data extracted prior to the forecast period to extract relationships between input and output data will be explored in more detail in section 5.3.

5.2.3 Correlation of predictions and actual filtered returns

The second part of the EMD-filter verification process is to evaluate the correlation between neural network predictions using filtered and unfiltered training data and filtered and unfiltered actual data. The EMD-filter technique will be verified to provide increased predictability if a stronger positive correlation exists between the EMD-filtered predictions and the EMD-filtered actual data compared to the correlation between the unfiltered predictions and the original data. It is also expected that a positive correlation should exist between the EMD-filtered predictions and the unfiltered actual data. Both the above mentioned correlations should ideally be larger than the correlation between the unfiltered predictions and the unfiltered or EMD-filtered actual data in order to prove that EMD-filtering improves exchange rate predictability. In this analysis the exchange rates are predicted for the period of 1 February to 31 December 2013. Table 5-1 shows these correlations. The following notation is used in the table:

- Corr(x,y) is the correlation between vectors x and y.
- A-Orig is the actual returns of the original unfiltered time series.
- A-EMD is the actual returns of the EMD-filtered time series.
- P-Orig is the predicted returns of a neural network trained using the original unfiltered data.
- P-EMD is the predicted returns of a neural network trained using EMD-filtered data that is available at the time of forecasting.

The correlation between the following pairs of vectors will be investigated:

- Corr(A-Orig,A-EMD): This is the correlation between the actual unfiltered returns and the actual EMD-filtered returns. This correlation is expected to be the highest as both are based on the original data. This is evaluated in order to determine if the filtered signal is still a good representation of the unfiltered data.
- Corr(A-EMD,P-EMD): The correlation between the actual EMD-filtered data and the predicted EMD-filtered data. The correlation should be high if the EMD-filtered signal is predictable.
- Corr(A-Orig,P-EMD): The correlation between the actual original returns and the EMD-filtered prediction is expected to be slightly lower than the above correlation due to the predictions being based off filtered data. The correlation should still be significant if the use of EMD-filters improves the predictability of the underlying signal.

- Corr(A-EMD,P-Orig): The correlation between the actual EMD-filtered returns and the predicted data should be lower than Corr(A-EMD,P-EMD) if the EMD-filter improves the predictability of the signal.
- Corr(A-Orig,P-Orig): The correlation between the original returns and the predicted returns using unfiltered data should be similar to Corr(A-EMD,P-Orig). If the use of EMD-filtering improves the predictability of the signal then this correlation should be lower than the correlation between the original data and the predictions based on EMD-filtered data (Corr(A-Orig,P-EMD)).

Table 5-1: Correlation between actual and predicted returns for the verification of the EMD-filter

Exchange rate	Forecast horizon (minutes)	Sample rate (minutes)	Corr (A-Orig, A-EMD)	Corr (A-EMD, P-EMD)	Corr (A-Orig, P-EMD)	Corr (A-EMD, P-Orig)	Corr (A-Orig, P-Orig)
EUR/USD	35	1	0.8808	0.6829	0.6005	0.0401	-0.0024
		5	0.9039	0.5663	0.5112	0.1873	0.1791
		7	0.9017	0.5097	0.4573	0.1802	0.1672
		35	0.9049	0.0675	0.0601	0.0305	0.0078
USD/JPY	10	1	0.9093	0.6345	0.5836	0.2170	0.2161
		2	0.9112	0.5585	0.5088	0.1345	0.1432
		5	0.9074	0.3025	0.2568	0.0769	0.0864
		10	0.9113	0.0227	0.0282	-0.0108	0.0152
USD/ZAR	180	15	0.9252	0.5015	0.4706	0.0739	0.0824
		30	0.9252	0.3626	0.3375	0.0391	0.0512
		60	0.9201	0.2393	0.2068	0.0146	0.0170
		90	0.9201	0.0907	0.0836	0.0372	0.0537
		180	0.9200	-0.0157	-0.0175	-0.0341	0.0152

The results in Table 5-1 confirm that the EMD-filter improves predictability of the exchange rates, with exception of the three cases where the sample period is equal to the forecast horizon. In all the other cases the correlation between the predicted and actual EMD-filtered signals is the highest, followed by the correlation of the EMD-filtered predictions and the unfiltered actual signal. The correlation for the predictions made with a neural network trained with unfiltered data is considerably lower than the predictions with the filtered data. This analysis provides additional confidence in the EMD-filtering technique when applied to data with smaller sample intervals.

5.3 Verification of the reliability of the neural network models

Neural network based models sometimes exhibit unpredictable behaviour, making it necessary to verify that the neural network is successfully trained. A sufficiently regularised neural network should ideally display gradual changes in output for a range of inputs. Insufficient regularisation would result in abrupt changes in the output that are not supported by systematic changes in the inputs. For a system where there is no relationship between the inputs and outputs, one would expect little or no changes in the output for a range of inputs.

The evaluation of the input-output relationships extracted from the data by the neural network is done by varying one input at a time between its extreme values, while keeping the other inputs at their mean values. The neural network used to predict the EUR/USD exchange rate at a sample time of 7 minutes and a forecast horizon of 35 minutes will be investigated. Some of the modelled relationships are displayed in Figures 5-16 to 5-19 below:

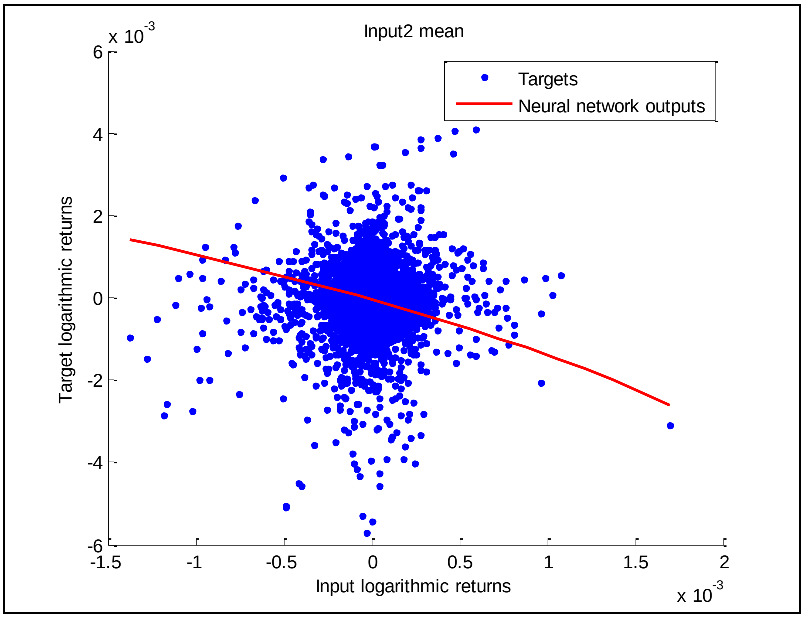

Figure 5-16: Scatter plot of input 2 and the target logarithmic returns for the EUR/USD 7 minute sampled EMD-filtered neural network. The prediction of the neural network is shown for the entire range input values while the other inputs maintain their mean values.

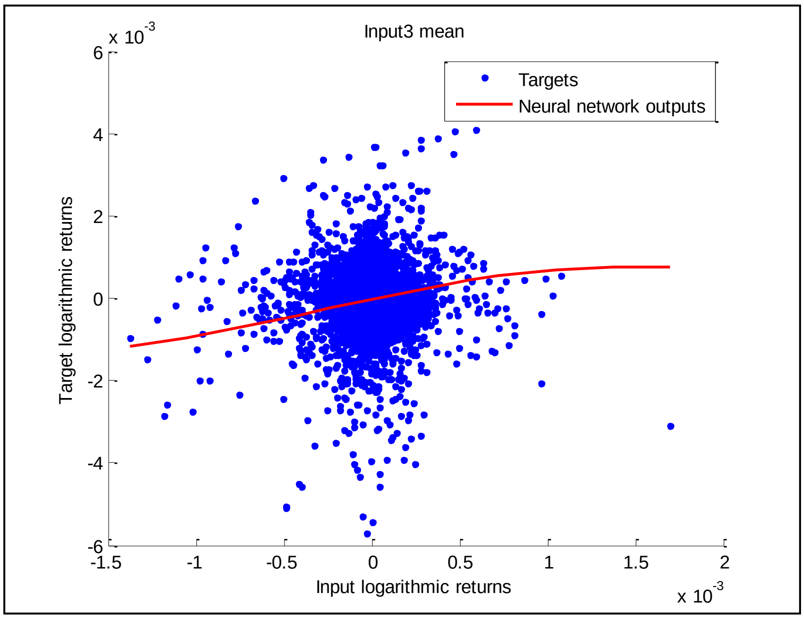

Figure 5-17: **Scatter plot of input 3 and the target logarithmic returns for the EUR/USD 7 minute sampled EMD-filtered neural network. The prediction of the neural network is shown for the entire range input values while the other inputs maintain their mean values.**

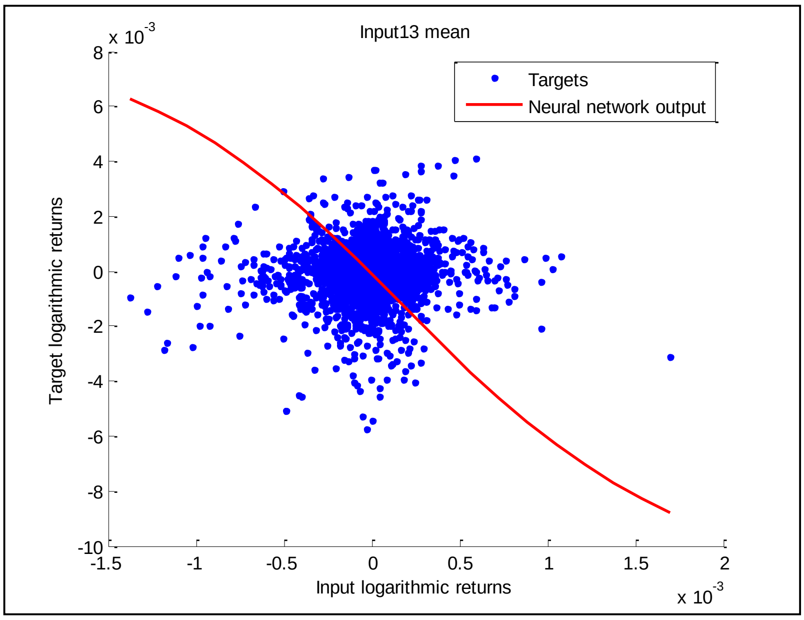

Figure 5-18: **Scatter plot of input 13 and the target logarithmic returns for the EUR/USD 7 minute sampled EMD-filtered neural network. The prediction of the neural network is shown for the entire range input values while the other inputs maintain their mean values.**

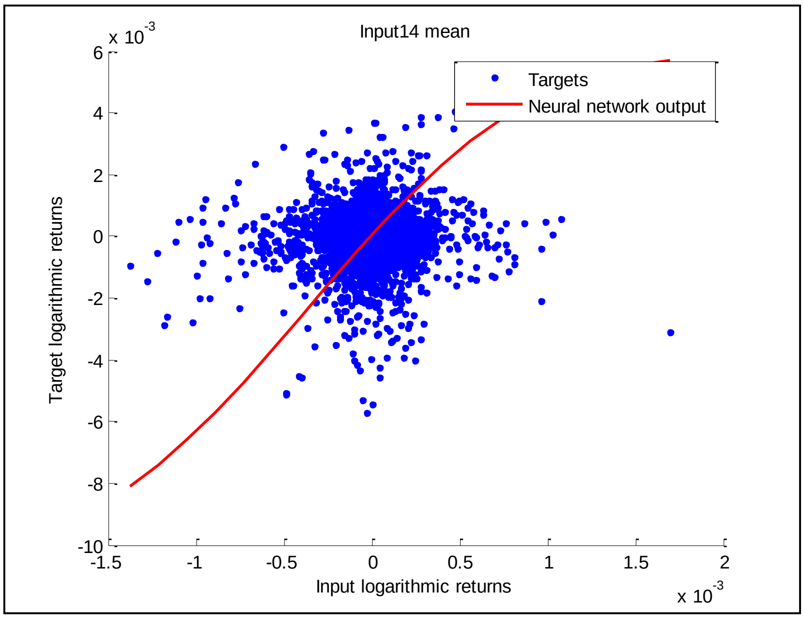

Figure 5-19: Scatter plot of input 14 and the target logarithmic returns for the EUR/USD 7 minute sampled EMD-filtered neural network. The prediction of the neural network is shown for the entire range input values while the other inputs maintain their mean values.

The figures display behaviour that is nonlinear and changes gradually over the range of inputs. This is to be expected for a properly regularised neural network, and verifies that the model has been trained in such a way that over-fitting to the data has been avoided, while still capturing gradual non-linear aspects of the input-output relationships. The other EMD-filtered neural network models for the different exchange rates at the different sample times have also been investigated, and show similar behaviour to the examples provided above. An exhaustive analysis of the inputs is not included as the number of inputs for all the neural networks number in the hundreds.

Figure 5-20 shows an example of the typical mean squared error (MSE) of the training, validation and test sets over the number of training epochs. The EMD-filtered neural network for the EUR/USD log returns sampled at 7 minutes stops training after 13 epochs as the validation checks fail after 7 epochs. This further ensures the regularisation of the trained model.

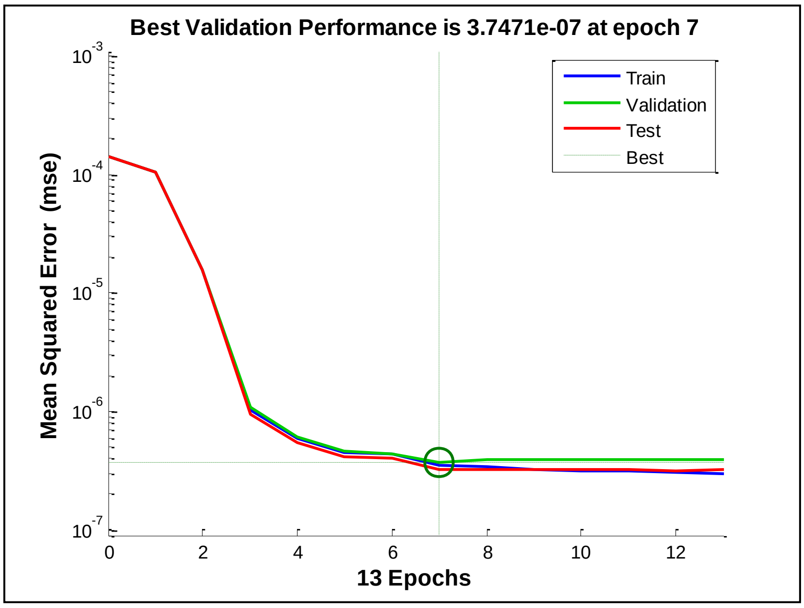

Figure 5-20: Mean squared error over the number of training epochs for the EUR/USD 7 minute sampled EMD-filtered neural network.

Figure 5-21 shows a histogram of the typical distribution of mean squared error after training has been completed. 83% of the errors for the EUR/USD 7 minute sampled EMD-filtered ANN lie within the zero error bin or the bins immediately adjacent to it, which shows regular and consistent behaviour of the trained model without the presence of a large deviations in model outputs resulting from noisy input data.

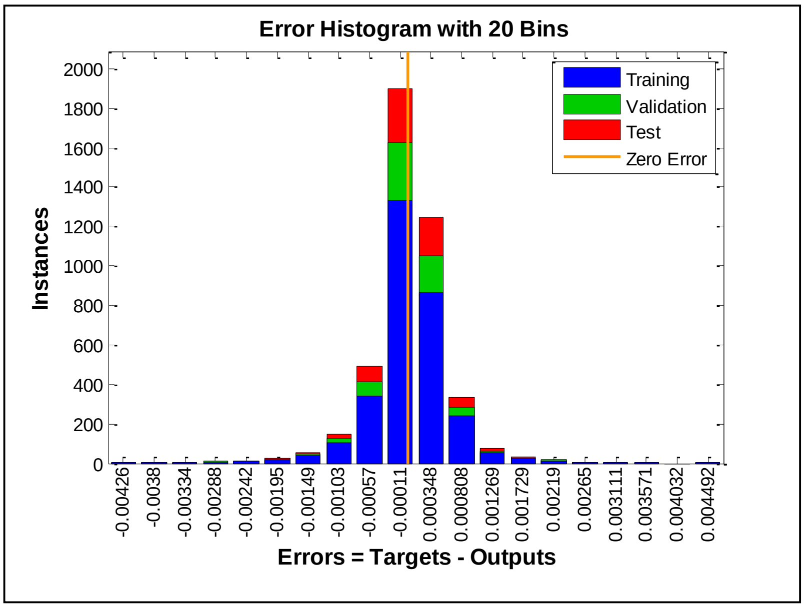

Figure 5-21: Histogram of the mean squared error of the trained EUR/USD 7 minute sampled EMD-filtered neural network.

Figure 5-22 shows plots of the linear regression of the target values relative to the output values for the training, testing and validation data for the EUR/USD 7 minute sampled EMD-filtered ANN. With an overall regression coefficient of 0.66124 target and output values show relatively high similarity without being identical. This provides additional confidence in the model as a relationship between the targets and outputs is shown to exist, but without the neural network overfitting to the training data.

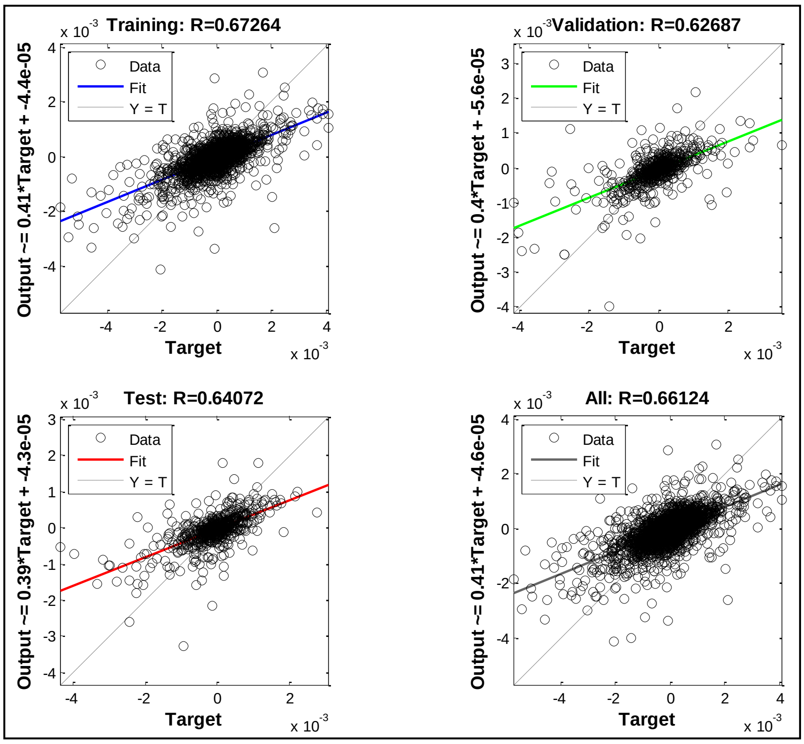

Figure 5-22: Linear regression of the target values relative to the output values for the EUR/USD 7 minute sampled EMD-filtered neural network.

5.4 Comparative performance analysis

The forecasting performance is analysed by comparing the root-mean-square-error, directional symmetry, correlation between actual and predicted returns and the simulated returns of the EMD-filtered ANN model with similar results produced by an unfiltered neural network and by a random walk model. For further insight into the performance the mean number of trades per month, mean return per trade, fraction of profitable trades, mean return per profitable trade and mean return per losing trade are also provided. When a prediction is made the next prediction will only start at the time that the previous prediction has ended. This allows the calculation of a realistic figure of the possible returns generated by the model for one month.

The forecasting performance of the models are analysed separately for each month from February to December 2013, after which the average monthly performance is calculated. See Appendix C for the separate monthly performance of the models.

Sample graphs of the first 50 predictions of the models are given in the figures below each table. This is included to illustrate the nature of the filtered signals for which the predictions were made and to gain further insight into the behaviour of the models. These figures only show the signal values after each forecasting period and not for all sample values, as the predictions are only generated once per forecasting period.

See section 4.5 for the parameters of the neural network models used to generate the predicted signals.

5.4.1 EUR/USD 35 minute forecast horizon performance

Table 5-2: EUR/USD average monthly forecasting performance using 1 minute samples with a 35 minute forecast horizon

Model	RMSE	DS (%)	Corr	Ret (%)	Average Trades per month	Mean return per trade (%)	Fraction of profitable trades	Mean return per profitable trade (%)	Mean return per losing trade (%)
EMD-filtered ANN	0.0006	68.39	0.6041	19.2524	475.00	0.0405	0.6067	0.0944	-0.0347
Unfiltered ANN	0.0007	51.38	-0.0005	-5.9576	274.81	-0.0252	0.3832	0.0995	-0.0973
Random walk	0.0010	33.56	-0.0048	-10.1056	458.63	-0.0220	0.2443	0.0958	-0.0541

The ideal monthly average compound returns for a model with 100% prediction accuracy is 38.47%.

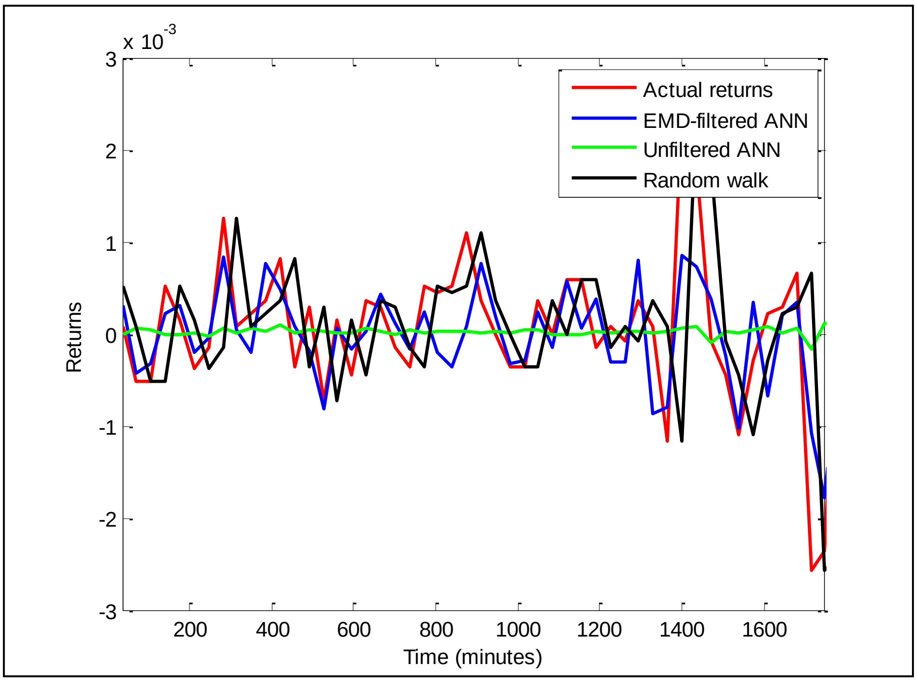

Figure 5-23: Actual and predicted returns for a sample of the EUR/USD exchange rate using 1 minute samples with a 35 minute forecast horizon.

Table 5-3: EUR/USD average monthly forecasting performance using 5 minute samples with a 35 minute forecast horizon

Model	RMSE	DS (%)	Corr	Ret (%)	Average Trades per month	Mean return per trade (%)	Fraction of profitable trades	Mean return per profitable trade (%)	Mean return per losing trade (%)
EMD-filtered ANN	0.0006	66.23	0.5146	17.1951	463.7273	0.0373	0.5914	0.0936	-0.0367
Unfiltered ANN	0.0007	54.13	-0.0057	-6.3521	300.9091	-0.0247	0.3763	0.0842	-0.0849
Random walk	0.0010	33.62	-0.0262	-10.3893	460.0000	-0.0225	0.2486	0.0931	-0.0549

The ideal monthly compound returns for a model with 100% prediction accuracy is 38.55%.

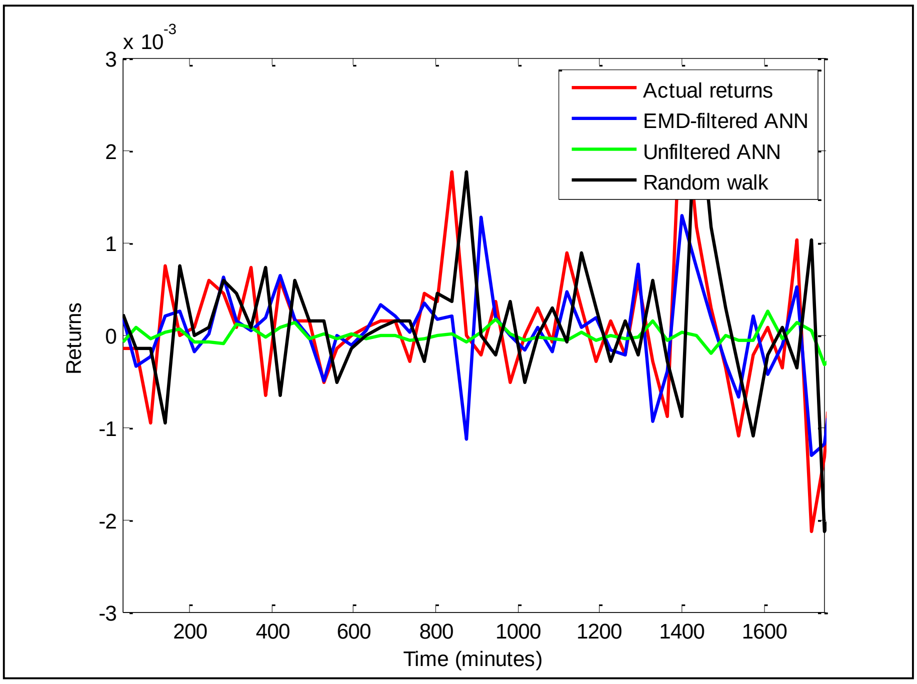

Figure 5-24: Actual and predicted returns for a sample of the EUR/USD exchange rate using 5 minute samples with a 35 minute forecast horizon.

Table 5-4: EUR/USD average monthly forecasting performance using 7 minute samples with a 35 minute forecast horizon

Model	RMSE	DS (%)	Corr	Ret (%)	Average Trades per month	Mean return per trade (%)	Fraction of profitable trades	Mean return per profitable trade (%)	Mean return per losing trade (%)
EMD-filtered ANN	0.0006	64.82	0.4574	14.1996	465.9091	0.0302	0.5679	0.0897	-0.0404
Unfiltered ANN	0.0007	53.14	0.0113	-7.7343	353.9091	-0.0253	0.3833	0.0750	-0.0825
Random walk	0.0010	33.72	-0.0126	-10.1657	455.8182	-0.0222	0.2448	0.0960	-0.0548

The ideal monthly compound returns for a model with 100% prediction accuracy is 38.61%.

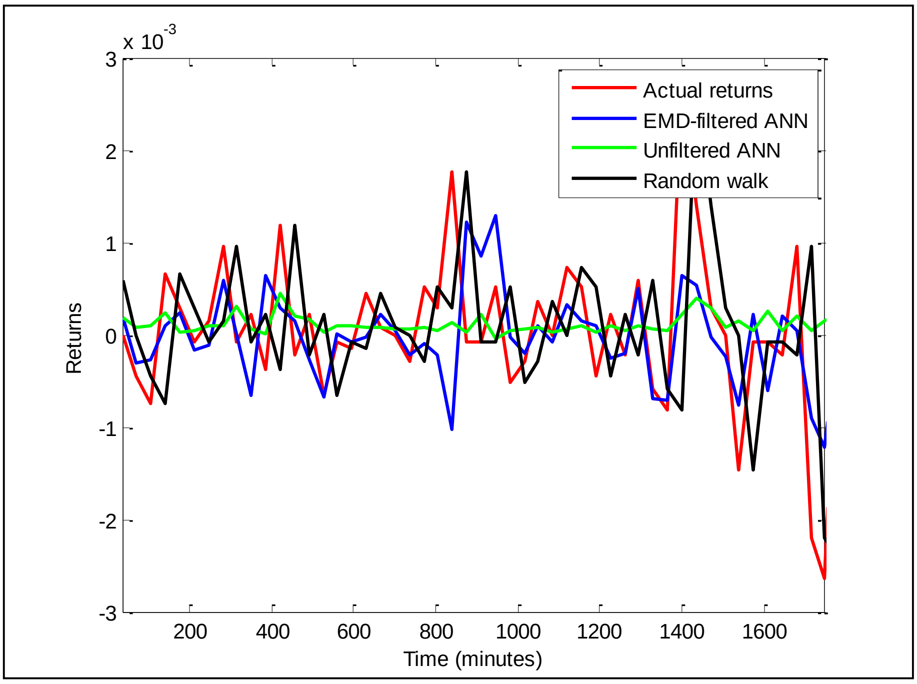

Figure 5-25: Actual and predicted returns for a sample of the EUR/USD exchange rate using 7 minute samples with a 35 minute forecast horizon.

Table 5-5: EUR/USD average monthly forecasting performance using 35 minute samples with a 35 minute forecast horizon

Model	RMSE	DS (%)	Corr	Ret (%)	Average Trades per month	Mean return per trade (%)	Fraction of profitable trades	Mean return per profitable trade (%)	Mean return per losing trade (%)
EMD-filtered ANN	0.0007	48.35	0.0578	-4.3859	327.6364	-0.0237	0.3821	0.0927	-0.0984
Unfiltered ANN	0.0008	45.84	-0.0023	-4.8192	184.6364	-0.0308	0.3940	0.1189	-0.1125
Random walk	0.0010	33.13	-0.0091	-10.6523	464.0909	-0.0229	0.2485	0.0903	-0.0543

The ideal monthly compound returns for a model with 100% prediction accuracy is 37.83%.

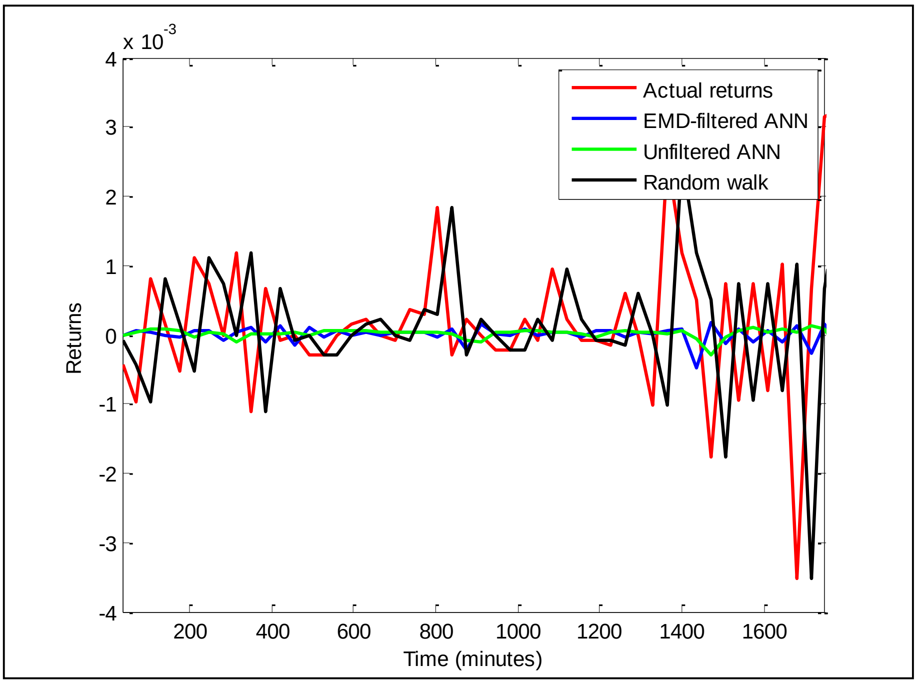

Figure 5-26: Actual and predicted returns for a sample of the EUR/USD exchange rate using 35 minute samples with a 35 minute forecast horizon.

5.4.2 USD/JPY 10 minute forecast horizon performance

Table 5-6: USD/JPY average monthly forecasting performance using 1 minute samples with a 10 minute forecast horizon

Model	RMSE	DS (%)	Corr	Ret (%)	Average Trades per month	Mean return per trade (%)	Fraction of profitable trades	Mean return per profitable trade (%)	Mean return per losing trade (%)
EMD-filtered ANN	0.0005	70.68	0.5753	65.7804	1694.2727	0.0387	0.6177	0.0976	-0.0301
Unfiltered ANN	0.0006	59.39	0.0262	-17.8490	1123.7273	-0.0161	0.4212	0.0625	-0.0569
Random walk	0.0009	34.27	-0.0311	-30.2572	1593.7273	-0.0190	0.2459	0.0803	-0.0391

The ideal monthly compound returns for a model with 100% prediction accuracy is 160.86%.

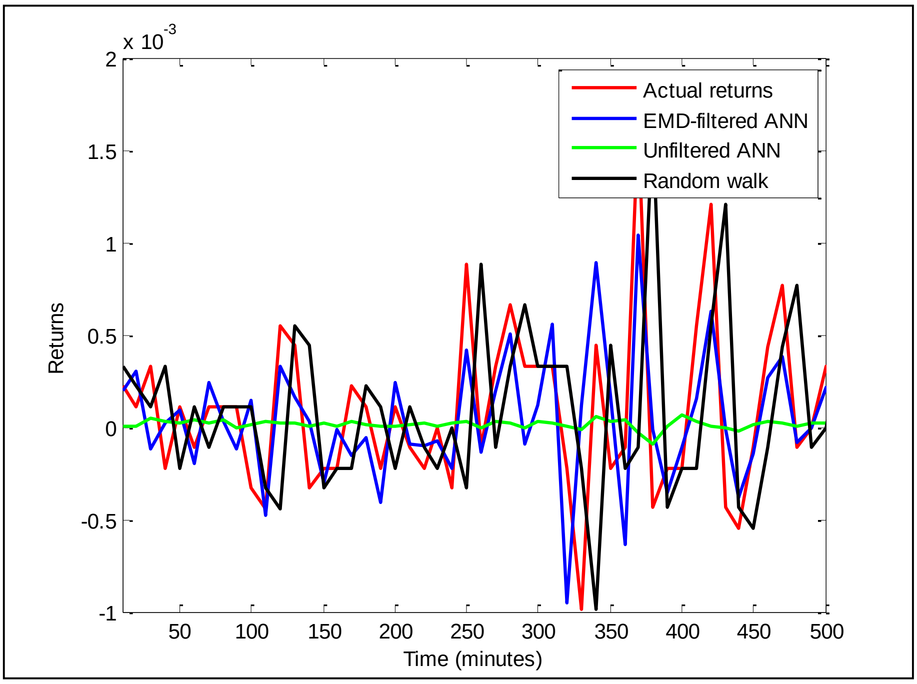

Figure 5-27: Actual and predicted returns for a sample of the USD/JPY exchange rate using 1 minute samples with a 10 minute forecast horizon.

Table 5-7: USD/JPY average monthly forecasting performance using 2 minute samples with a 10 minute forecast horizon

Model	RMSE	DS (%)	Corr	Ret (%)	Average Trades per month	Mean return per trade (%)	Fraction of profitable trades	Mean return per profitable trade (%)	Mean return per losing trade (%)
EMD-filtered ANN	0.0005	68.13	0.5017	53.7688	1694.1818	0.0316	0.5888	0.0932	-0.0312
Unfiltered ANN	0.0006	59.38	0.0041	-21.5194	1297.1818	-0.0170	0.4115	0.0571	-0.0536
Random walk	0.0009	34.09	-0.0284	-29.6208	1605.7273	-0.0184	0.2488	0.0796	-0.0387

The ideal monthly compound returns for a model with 100% prediction accuracy is 157.60%.

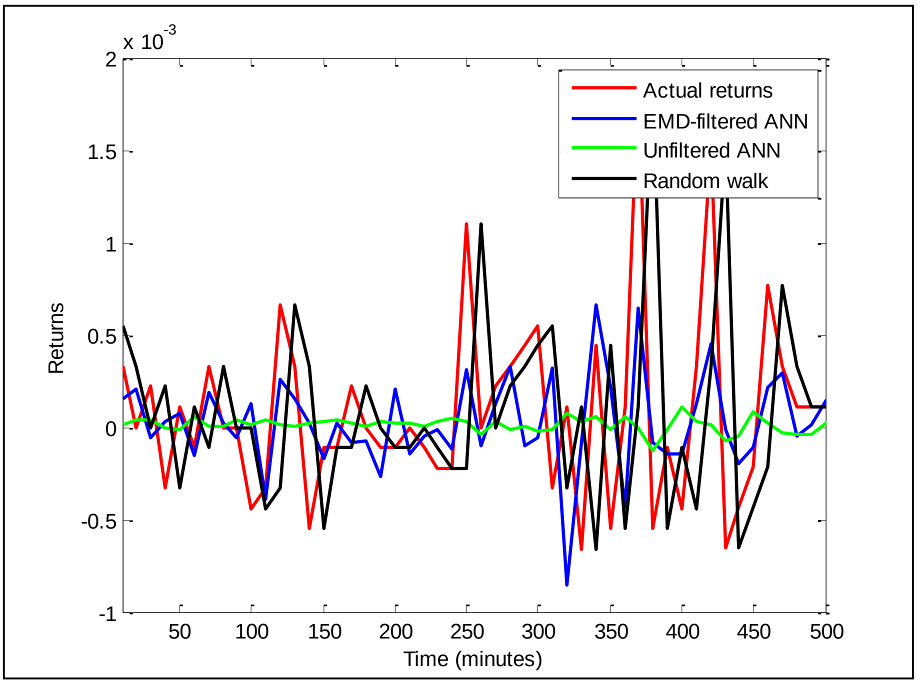

Figure 5-28: Actual and predicted returns for a sample of the USD/JPY exchange rate using 2 minute samples with a 10 minute forecast horizon.

Table 5-8: USD/JPY average monthly forecasting performance using 5 minute samples with a 10 minute forecast horizon

Model	RMSE	DS (%)	Corr	Ret (%)	Average Trades per month	Mean return per trade (%)	Fraction of profitable trades	Mean return per profitable trade (%)	Mean return per losing trade (%)
EMD-filtered ANN	0.0006	58.91	0.2481	11.3399	1718.0000	0.0064	0.4800	0.0782	-0.0372
Unfiltered ANN	0.0006	57.59	0.0030	-20.8346	1244.0000	-0.0174	0.3943	0.0686	-0.0586
Random walk	0.0009	34.74	-0.0121	-28.9341	1592.4545	-0.0182	0.2498	0.0817	-0.0388

The ideal monthly compound returns for a model with 100% prediction accuracy is 161.44%.

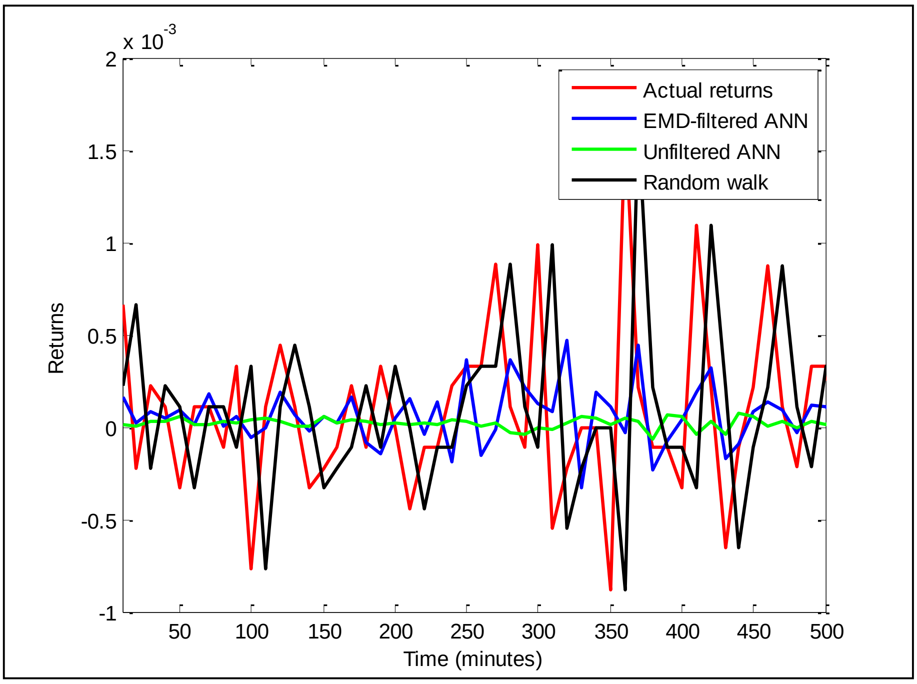

Figure 5-29: Actual and predicted returns for a sample of the USD/JPY exchange rate using 5 minute samples with a 10 minute forecast horizon.

Table 5-9: USD/JPY average monthly forecasting performance using 10 minute samples with a 10 minute forecast horizon

Model	RMSE	DS (%)	Corr	Ret (%)	Average Trades per month	Mean return per trade (%)	Fraction of profitable trades	Mean return per profitable trade (%)	Mean return per losing trade (%)
EMD-filtered ANN	0.0006	45.96	0.0279	-12.5176	952.0909	-0.0155	0.3660	0.1060	-0.0706
Unfiltered ANN	0.0006	56.62	0.0018	-17.4495	970.8182	-0.0175	0.4032	0.0735	-0.0644
Random walk	0.0009	34.09	-0.0285	-29.6137	1605.3636	-0.0184	0.2488	0.0796	-0.0387

The ideal monthly compound returns for a model with 100% prediction accuracy is 157.51%.

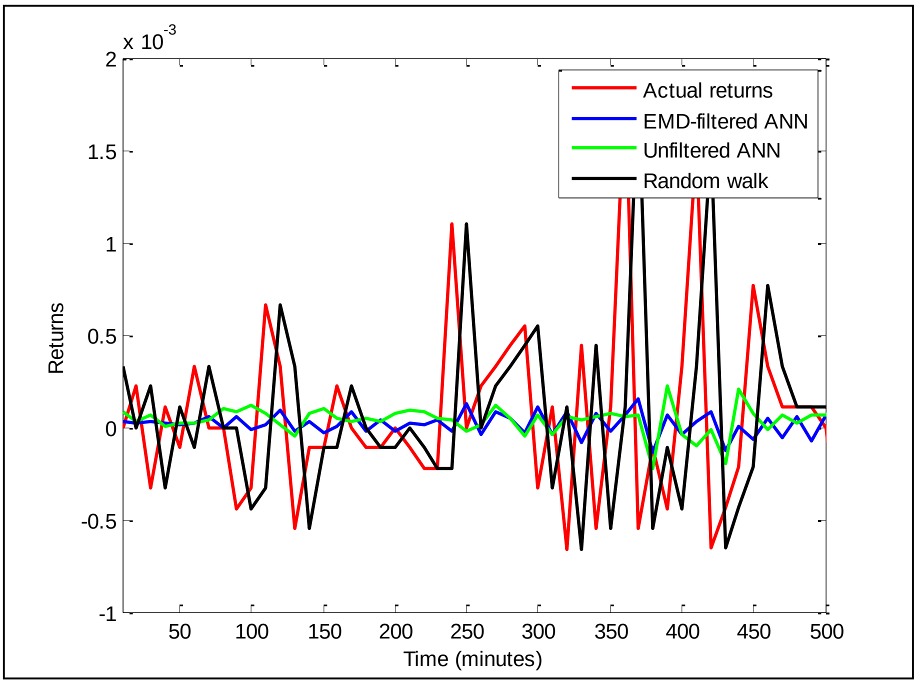

Figure 5-30: Actual and predicted returns for a sample of the USD/JPY exchange rate using 10 minute samples with a 10 minute forecast horizon.

5.4.3 USD/ZAR 180 minute forecast horizon performance

Table 5-10: USD/ZAR average monthly forecasting performance using 15 minute samples with a 180 minute forecast horizon

Model	RMSE	DS (%)	Corr	Ret (%)	Average Trades per month	Mean return per trade (%)	Fraction of profitable trades	Mean return per profitable trade (%)	Mean return per losing trade (%)
EMD-filtered ANN	0.0030	68.37	0.4639	3.8244	74.7273	0.0500	0.4603	0.4219	-0.2388
Unfiltered ANN	0.0033	50.82	0.0350	-4.2665	28.0000	0.3093	0.2613	0.5695	-0.3414
Random walk	0.0048	32.83	-0.0532	-15.3396	77.0000	-0.1979	0.1858	0.5021	-0.3363

The ideal monthly compound returns for a model with 100% prediction accuracy is 21.09%.

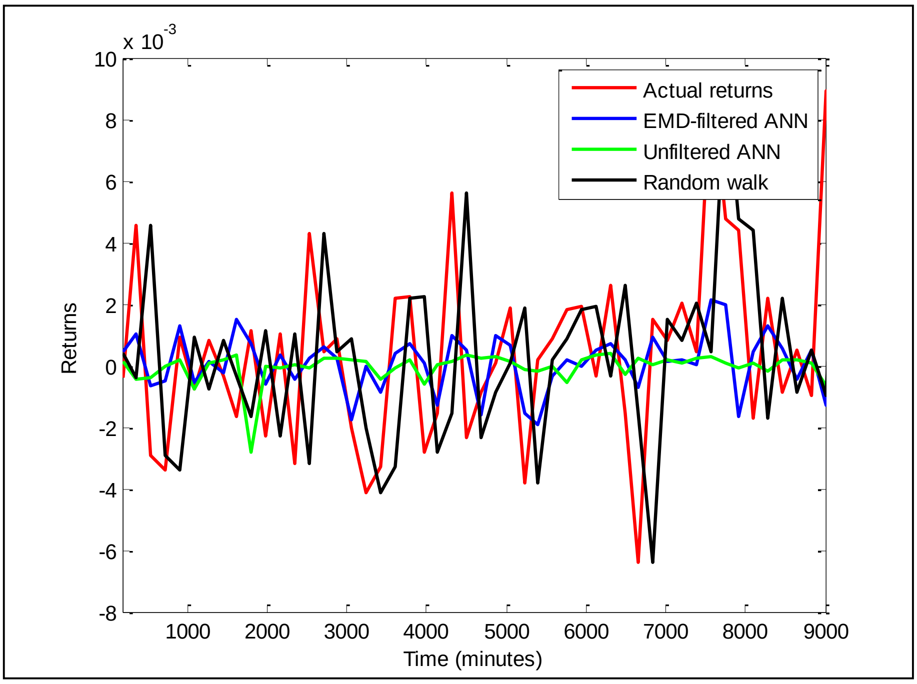

Figure 5-31: Actual and predicted returns for a sample of the USD/ZAR exchange rate using 15 minute samples with a 180 minute forecast horizon.

Table 5-11: USD/ZAR average monthly forecasting performance using 30 minute samples with a 180 minute forecast horizon

Model	RMSE	DS (%)	Corr	Ret (%)	Average Trades per month	Mean return per trade (%)	Fraction of profitable trades	Mean return per profitable trade (%)	Mean return per losing trade (%)
EMD-filtered ANN	0.0031	63.51	0.3561	1.4932	68.2727	0.0620	0.4442	0.4759	-0.2666
Unfiltered ANN	0.0033	53.34	0.0233	-6.1783	37.4545	-0.1325	0.3357	0.5146	-0.4825
Random walk	0.0048	33.46	-0.0531	-15.1006	74.6364	-0.2010	0.2165	0.4220	-0.3519

The ideal monthly compound returns for a model with 100% prediction accuracy is 21.24%.

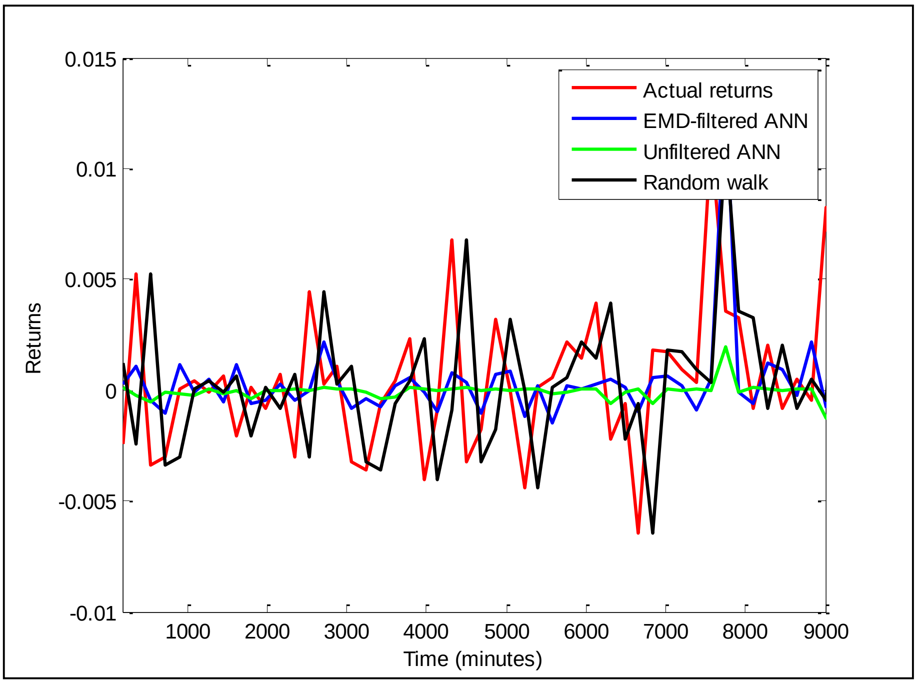

Figure 5-32: Actual and predicted returns for a sample of the USD/ZAR exchange rate using 30 minute samples with a 180 minute forecast horizon.

Table 5-12: USD/ZAR average monthly forecasting performance using 60 minute samples with a 180 minute forecast horizon

Model	RMSE	DS (%)	Corr	Ret (%)	Average Trades per month	Mean return per trade (%)	Fraction of profitable trades	Mean return per profitable trade (%)	Mean return per losing trade (%)
EMD-filtered ANN	0.0032	56.88	0.1995	-5.7342	73.4545	-0.0792	0.3734	0.3571	-0.3069
Unfiltered ANN	0.0033	50.82	0.0034	-7.6855	43.4545	-0.1640	0.3167	0.4526	-0.4891
Random walk	0.0047	33.14	-0.0391	-16.0368	76.5455	-0.2091	0.1861	0.4616	-0.3378

The ideal monthly compound returns for a model with 100% prediction accuracy is 20.82%.

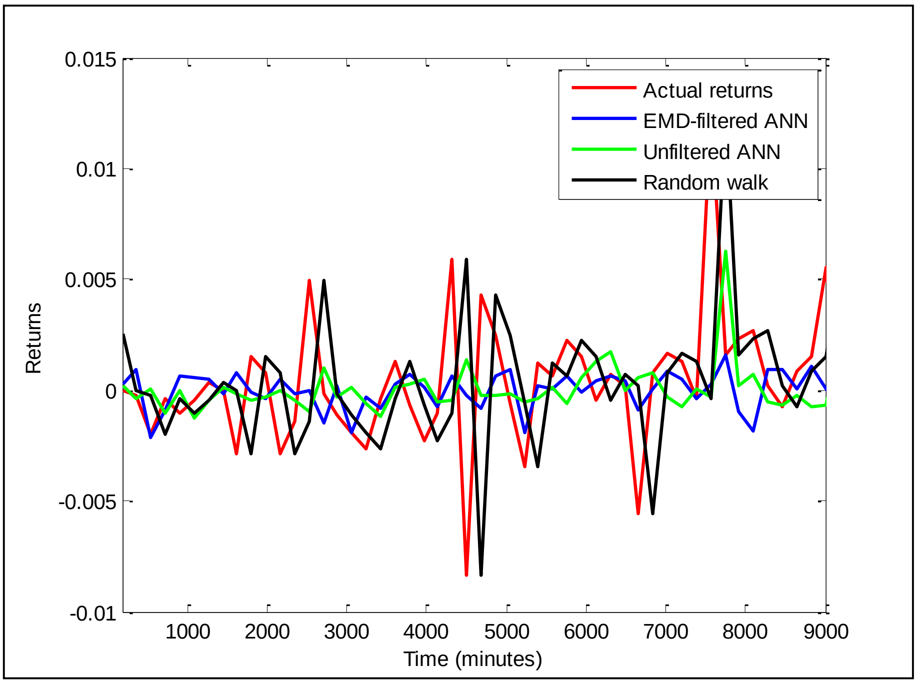

Figure 5-33: Actual and predicted returns for a sample of the USD/ZAR exchange rate using 60 minute samples with a 180 minute forecast horizon.

Table 5-13: USD/ZAR average monthly forecasting performance using 90 minute samples with a 180 minute forecast horizon

Model	RMSE	DS (%)	Corr	Ret (%)	Average Trades per month	Mean return per trade (%)	Fraction of profitable trades	Mean return per profitable trade (%)	Mean return per losing trade (%)
EMD-filtered ANN	0.0033	56.82	0.0930	-4.4896	42.9091	-0.1354	0.3053	0.4931	-0.4154
Unfiltered ANN	0.0034	54.29	0.0329	-7.4430	39.5455	-0.2944	0.2950	0.3783	-0.4332
Random walk	0.0047	33.14	-0.0391	-16.0368	76.5455	-0.2091	0.1861	0.4616	-0.3378

The ideal monthly compound returns for a model with 100% prediction accuracy is 20.83%.

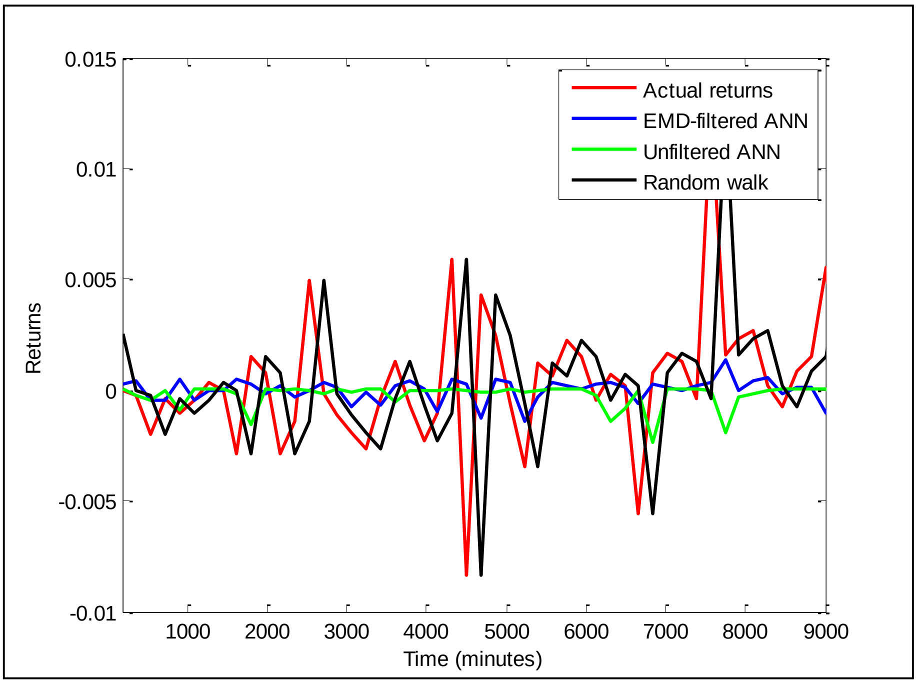

Figure 5-34: Actual and predicted returns for a sample of the USD/ZAR exchange rate using 90 minute samples with a 180 minute forecast horizon.

Table 5-14: USD/ZAR average monthly forecasting performance using 180 minute samples with a 180 minute forecast horizon

Model	RMSE	DS (%)	Corr	Ret (%)	Average Trades per month	Mean return per trade (%)	Fraction of profitable trades	Mean return per profitable trade (%)	Mean return per losing trade (%)
EMD-filtered ANN	0.0039	47.17	-0.0275	-6.3105	29.3636	-0.3465	0.2655	0.4014	-0.4435
Unfiltered ANN	0.0034	57.97	-0.0011	-8.8645	51.4545	-0.1229	0.3550	0.4114	-0.4855
Random walk	0.0047	33.058	-0.0399	-16.0487	76.1818	-0.2102	0.1886	0.4515	-0.3390

The ideal monthly compound returns for a model with 100% prediction accuracy is 20.74%.

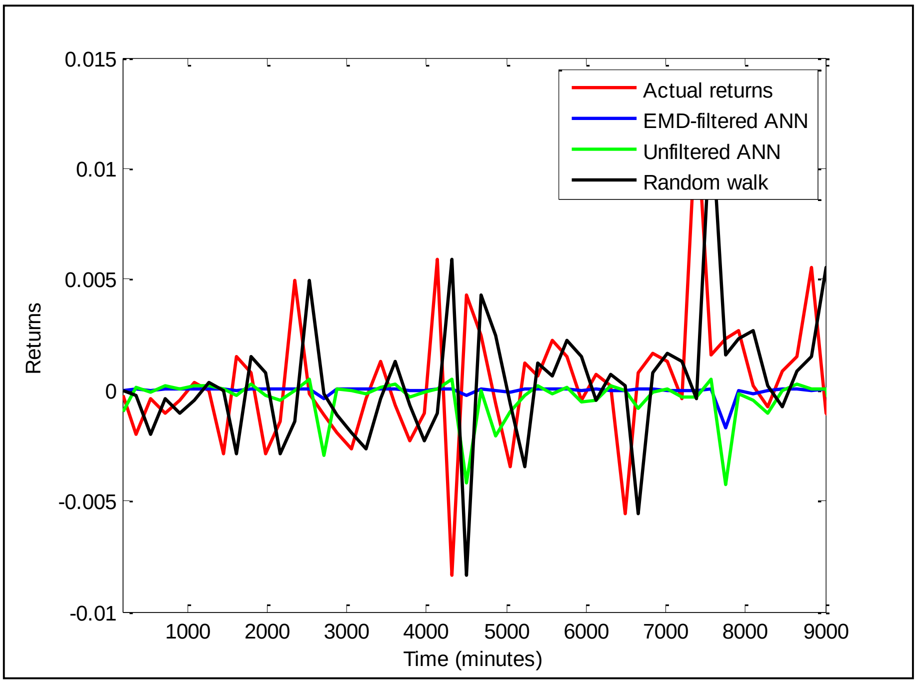

Figure 5-35: Actual and predicted returns for a sample of the USD/ZAR exchange rate using 180 minute samples with a 180 minute forecast horizon.

5.4.4 Observations

It can be seen that the correlations between the actual and predicted signals, the directional symmetry and the fraction of profitable trades provide complementary views of the capabilities of the forecasting techniques used. In general DS is somewhat higher than fraction of profitable trades due to the impact of trading costs, causing some trades for which the direction was correctly predicted to still generate a loss. These different measures for the accuracy and value of the forecasting techniques are however in support of each other for all cases investigated.

It is clear that the random walk model performs as would be expected: the return that is predicted is in general close to the last actual return before the prediction was made, hence the graph of predicted returns appears to be a time delayed version of actual returns with the delay equal to the forecast horizon (making the prediction useless, as confirmed by the correlation coefficient that is close to zero).

The ANN using unfiltered (and hence very noisy) input data finds it difficult to uncover any predictable patterns and therefore (also because the NN model is well regularized) produces predictions that are close to zero, once again with correlations that do not significantly deviate from zero.

On the other hand there are several cases where the EMD filtered ANNs uncover systematic relationships between filtered inputs and actual output, as confirmed by a percentage of profitable trades that is significantly above 50% and correlations that are significantly different from zero and positive in almost all cases.

It can be seen that while DS is significantly higher than 50% for all cases using the higher sampling rates on the filtered signals, this only leads to profitable trading for those exchange rates with relatively low bid-ask-spreads (EUR/USD and USD/JPY). In the case of the ZAR/USD the trading is only marginally profitable in spite of high DS values – the good prediction results are therefore largely nullified by high trading costs.

5.5 Validation

The results of section 5.4 allow us to reject the null hypothesis that the EMD-filtered ANN model does not deliver superior forecasting performance compared to the unfiltered and random walk models. In order to verify the rejection of this hypothesis a combination of cross-validation and t-testing will be implemented based on the performance evaluation criterion of simulated returns.

Cross-validation allows us to estimate how well the results of the statistical analysis will generalise to an independent data set and protects against misleading hypotheses. As the forecasting simulations were already carried out on a month-by-month basis, the data subsets for the cross-validation will be the monthly simulated returns. This will result in an 11-fold cross-validation.

Two sample t-tests will be performed on the 11 subsets of data in order to test the null hypothesis of superior forecasting performance of the EMD-filtered ANN model. The method for two sample t-testing follows:

Assume the two populations, X and Y are normally distributed. μ_x is the mean value of population X and σ_x is the standard deviation of population X. Similarly the mean of population Y is μ_y and the standard deviation is σ_y. The t-test statistic is then calculated using equation 5-1:

$$t = \frac{\mu_x - \mu_y}{\sqrt{\frac{{\sigma_x}^2}{n_x} + \frac{{\sigma_y}^2}{n_y}}} \qquad (5\text{-}1)$$

where n_x is the number of samples for population X and n_y is the number of samples for population Y.

The p-values corresponding to the t-values and degrees of freedom will then be calculated in order to effectively evaluate the level of significance of the t-statistic. Table 5-15, Table 5-16 and Table 5-17 show the results of the t-tests for the simulated returns of the EUR/USD, USD/JPY and USD/ZAR exchange rates respectively. The 11 months of predictions are the total populations of the respective models, with the individual months being separate samples of the total population. The EMD-filtered ANN is model X for all cases, with the unfiltered ANN and the random walk model alternately acting as model Y.

Using an alpha level of 0.1, all the results that are shown to be statistically significant are shaded.

Table 5-15: EUR/USD simulated return t-statistics for the two sample t-tests

Model X	Sample time (minutes)	Forecast horizon (minutes)	t-value	p-value	Model Y
EMD-filtered ANN	1	35	10.6670	<0.0001	Unfiltered ANN
EMD-filtered ANN	1	35	13.4147	<0.0001	Random walk
EMD-filtered ANN	5	35	10.0131	<0.0001	Unfiltered ANN
EMD-filtered ANN	5	35	12.1379	<0.0001	Random walk
EMD-filtered ANN	7	35	9.6948	<0.0001	Unfiltered ANN
EMD-filtered ANN	7	35	10.6122	<0.0001	Random walk
EMD-filtered ANN	35	35	0.3488	0.7345	Unfiltered ANN
EMD-filtered ANN	35	35	5.1955	0.0004	Random walk

Table 5-16: USD/JPY simulated return t-statistics for the two sample t-tests

Model X	Sample time (minutes)	Forecast horizon (minutes)	t-value	p-value	Model Y
EMD-filtered ANN	1	10	6.4604	0.0001	Unfiltered ANN
EMD-filtered ANN	1	10	7.4501	<0.0001	Random walk
EMD-filtered ANN	2	10	6.8867	<0.0001	Unfiltered ANN
EMD-filtered ANN	2	10	7.7251	<0.0001	Random walk
EMD-filtered ANN	5	10	5.8807	0.0002	Unfiltered ANN
EMD-filtered ANN	5	10	8.2053	<0.0001	Random walk
EMD-filtered ANN	10	10	1.1810	0.2649	Unfiltered ANN
EMD-filtered ANN	10	10	5.7660	0.0002	Random walk

Table 5-17: USD/ZAR simulated returns t-statistics for the two sample t-tests

Model X	Sample time (minutes)	Forecast horizon (minutes)	t-value	p-value	Model Y
EMD-filtered ANN	15	180	3.1719	0.0100	Unfiltered ANN
EMD-filtered ANN	15	180	8.7763	<0.0001	Random walk
EMD-filtered ANN	30	180	3.7639	0.0037	Unfiltered ANN
EMD-filtered ANN	30	180	9.5595	<0.0001	Random walk
EMD-filtered ANN	60	180	0.8908	0.3940	Unfiltered ANN
EMD-filtered ANN	60	180	8.2587	<0.0001	Random walk
EMD-filtered ANN	90	180	1.3960	0.1929	Unfiltered ANN
EMD-filtered ANN	90	180	6.8242	<0.0001	Random walk
EMD-filtered ANN	180	180	0.9530	0.3630	Unfiltered ANN
EMD-filtered ANN	180	180	6.7850	<0.0001	Random walk

The results in the above set of tables show that the EMD-filtered ANN predictions are significantly superior to both the unfiltered ANN and random walk predictions for all cases other than the lowest sampling rates.

5.6 Conclusions

This chapter presented and evaluated the forecasting performance of the proposed EMD-filtered ANN model. This was done in several parts: verification of the EMD-filter, verification of the reliability of the ANN, evaluation of the forecasting performance using select criteria and the validation of the statistical significance of the simulated returns of the model.

The verification of the advantage of using an EMD-filter was first performed. This was done in two parts: The first part entailed the verification that the end-effect issues that arise from using extrapolation techniques do not negatively impact the predictions. A shorter EMD window was compared to an extended EMD window in order to evaluate the effects. Strong positive correlations were observed between the overlapping portions of the signals in the different sized windows, meaning the end-effects would have minimal impact on the predictions. The second part evaluated the correlation between EMD-filtered and unfiltered actual returns and predictions made by neural networks that were trained on EMD-filtered and unfiltered data. The use of EMD-filtered training data proved to drastically improve the correlation between the predictions and the actual returns, proving that EMD-filtering does indeed provide benefits to forecasting performance.

The verification of the reliability of the ANN forming part of the hybrid EMD-filtered ANN model entailed the varying of a single input variable while the other inputs were kept at their mean values. It was observed that each input has a nonlinear relationship with the output, demonstrating that the ANN captured aspects of behaviour that cannot be modelled with a linear regression model. The relationship however displayed gradual changes in the output for small changes in inputs, without any rapid changes in output for small input changes that would be indicative of overtraining on the noise in the training data.

The following observations were made in the evaluation of the forecasting performance of the models:

- The random walk model delivers the worst performance of all the models for almost all the exchange rates, forecast horizons and sample times. It rarely returns positive simulated returns, and the correlation coefficient is close to zero, as would have been expected.
- The neural network trained on unfiltered data struggles to find relationships in the noisy data and predicts returns that are close to zero as a result. As with the random walk model, the correlation between actual and predicted returns is close to zero with DS close to 50%. This demonstrates that the ANN on its own did not manage to detect useful relationships

in the unfiltered data, as a result of the presence of high frequency noise in the training data.

- The neural network trained on EMD-filtered data was able to uncover systematic relationships between the filtered inputs and the actual output. The directional symmetry is significantly larger than 50% in most instances and profitable trades occur more frequently than unprofitable ones. A high correlation between the predicted and actual returns is common, and the model manages to deliver profitable average monthly returns for many sample rates and forecast horizons after taking into account realistic trading costs.

The EMD filter cuts off at a period equal to twice the forecast horizon. An observation was made on the bad performance of the EMD-filtered ANN when the sample rate was equal to the forecast horizon. This occurred in section 5.2.3 and section 5.4. In these cases the sampling frequency was approximately twice the cut-off frequency; the EMD-filter therefore barely met the Nyquist criterion. These marginal sampling rates lead to noisy log returns as the samples are spaced too far apart to follow the smooth underlying behaviour of the EMD-filtered signal. From the results for these cases it is clear that such marginal sampling rates fail to sufficiently isolate the signal from the noise and consequently the trained ANN fails to predict the returns as accurately as in cases where higher sampling rates were used.

The validation process proved that the EMD-filtered ANN produced returns that exceeded that of the unfiltered ANN and the random walk model for sample times that are smaller than the forecast horizons at a significance level of 90% for the EUR/USD and USD/JPY exchange rates. Improvement of the performance of the USD/ZAR predictions were however not as significant. This is attributed to the high bid-ask spread associated with the USD/ZAR exchange rate and the fact that the mean loss for a losing trade is more than twice the mean profit for a profitable trade. These conditions make significantly profitable returns extremely hard to obtain.

In summary, this chapter provided evidence that EMD-filtering significantly improved the forecasting performance of an ANN. The EMD-filtered ANN outperformed the unfiltered ANN and the random walk model in the majority of evaluations, and was statistically proven to provide significantly higher returns. The exceptions can be attributed either to filtering that was too marginal in terms of the Nyquist criterion, or to high bid-ask spreads.

6. Conclusions

"Prediction is very difficult, especially about the future"

- Niels Bohr

6.1 Research overview

The purpose of this research was to develop a model for the prediction of exchange rates with the goal of exploiting the predictability in order to consistently generate positive returns. A neural network based model trained on data filtered with a novel EMD-filtering method was proposed in order to accomplish this goal.

One minor (USD/ZAR) and two major (EUR/USD and USD/JPY) exchange rates were evaluated in this study. The initial step was to determine the ideal prediction horizon for each exchange rate while taking bid-ask spreads into account. The data was filtered using the EMD-filtering technique which is designed to filter out high frequency noise and components that are not expected to contribute towards prediction accuracy at the chosen forecast horizon. A neural network was then trained on this filtered data to predict future returns.

Prediction results showed that the neural network trained on the EMD-filtered data was significantly more accurate than a neural network trained on unfiltered data and a random walk model for all the exchange rates. The EMD-filtered neural network was able to make significant profits for the higher sample rates when a trading strategy was applied. The lower sample rates that just marginally satisfy the Nyquist criterion perform comparably with the neural network trained on unfiltered data, indicating that some high frequency noise may still be still be present due to the gradual cut-off of the EMD-filter.

The neural network trained on EMD-filtered data was shown to be able to uncover systematic relationships between filtered inputs and actual outputs, and provides evidence that such relationships can be modelled if separated from high frequency noise.

6.2 Detailed results observations

Several of the performance evaluation criteria offer complementary views of the capabilities of the forecasting accuracy of the models used. Directional symmetry and the fraction of profitable trades support each other for all cases investigated. The DS is slightly higher than the fraction of profitable trades due to the impact of trading costs. The correlation between predicted and actual returns also supports these two criteria, with higher correlation generally appearing with high DS and fractions of profitable trades. The RMSE criteria is however not a good indicator of practical performance of the models. The RMSE mostly remains constant across the models for all exchange rates at different sampling rates, and often shows similar values for the neural networks trained on filtered and unfiltered data, even though the filtered neural network generates significantly larger returns.

The random walk model, which uses the latest available data as the prediction, is the model with the worst forecasting performance. It rarely generates positive returns, has a very low fraction of profitable trades and shows correlations close to zero.

The neural network trained on the unfiltered data struggles to find any systematic relationships between the inputs and target values, and therefore gives correlations that are close to zero. The DS is close to 50%, which is no better than a random guess. The unfiltered neural network tends to undergo fewer transactions per month compared to the other models, thus minimizing transaction costs. The lower transaction costs are however offset by the larger loss made on losing transactions as an unprofitable position is often held much longer than with the other models.

The neural network trained on EMD-filtered data offers the best performance of the tested models. High correlations, directional symmetries and fractions of profitable returns are prevalent. The model generates significant simulated returns for most of the exchange rates at the higher sample rates. The high prediction accuracy minimizes the losses per losing transaction. There are two cases where the neural network trained on EMD-filtered data fails to outperform the unfiltered neural network. These are when the forecast horizon is equal to the sample period or when the transaction costs are too high. The explanation for the first case is that the filtered signal barely satisfies the Nyquist criterion, resulting in a noisy representation of the log returns. The explanation for the performance with higher transaction costs is that the penalisation for a wrong prediction is much higher, making it difficult to generate positive returns.

In general it can be stated that the above observations are in conflict with the Efficient Market Hypothesis, which states that all historic data is already represented in the current price, and the best prediction is the latest value. The EMD-filtered ANN technique that relies solely on publicly available historic prices clearly outperforms the Random Walk model, providing evidence that there are systematic relationships between historic and future prices, and that such relationships can be separated from the noise in price information to allow successful prediction of future behaviour.

6.3 Recommended future research

Several options are available for further analyses that will allow greater insight into exchange rate prediction or possibly improve the model performance:

- A comparison between EMD-filtering and other filtering techniques. This will provide insight into the advantages of EMD-filtering over conventional financial and signal processing filtering techniques including moving average, Butterworth or Chebyshev filters.
- A comparison between the performances of the neural network trained on EMD-filtered data with other architectures or computational intelligence techniques trained on the same EMD-filtered data. General regression neural networks and support vector machines have shown promising results when forecasting based historic data.
- Optimization of model parameters like number of inputs, number of hidden neurons, sample rate and forecast horizon. This can be completed in many ways. One option would be the use of genetic algorithms in order to find the optimal parameters. A population of agents can be assembled that forecast with different parameters, with only the most successful ones ‘reproducing’ and providing slightly mutated parameters to their offspring until maximum returns are generated.
- Testing on more exchange rates and for different periods. Only one minor and two major exchange rates were used and only tested for the data of 2013. It is possible that other exchange rates have fundamentally different behaviour or that underlying market behaviour has changed since 2013.
- Using other financial instruments with the neural network trained on the EMD-filtered data. It would be interesting to investigate if the model will work equally well on these instruments.
- Changing the order of subsampling and calculating log returns (i.e. first generating the log returns before subsampling) in order to generate smoother representations of log returns for lower sampling rates. With lower sampling rates it is possible to cover a larger historical time window of the signal using the same number of neural network inputs; presenting the neural network model with a longer time history of more accurately extracted log return samples may improve prediction accuracy.
- Integrating the model with an existing trading platform and evaluating the practical performance to determine if any practical factors will prevent the technique from achieving the same performance as in theory.

6.4 Concluding remarks

This research contributes to the current body of knowledge in the domain of computational intelligence applied to financial problems. More specifically our work provides evidence that the use of EMD as a filtering technique is practical and improves prediction accuracy of nonlinear forecasting techniques like neural networks applied to exchange rates. From the results obtained we can conclude that the reason for this is twofold. Firstly the EMD filter improves signal clarity, making it much easier for a computational intelligence technique to discover relationships between past and future values. Secondly the EMD filtering technique is dynamic and adaptable. It is entirely data driven and retains the nonlinear relationships in the data that are of importance to forecasting.

Based on our results we can state that the proposed model of a neural network trained on EMD-filtered exchange rate data is viable for generating significant returns on investment. It outperforms the benchmark models in most cases for the three exchange rates, providing evidence in favour of rejecting the Efficient Market Hypothesis.

7. Bibliography

[1] Bank for International Settlements, “Triennial Central Bank Survey: Foreign exchange turnover in April 2013 : preliminary global results,” 2013.

[2] D. Hsieh, “Testing for nonlinear dependence in daily foreign exchange rates,” *Journal of Business*, vol. 62, no. 3, pp. 339-368, 1989.

[3] N. E. Huang et al., “The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis,” in *Proceedings of the Royal Society London*, 1996, vol. 454, no. 1971, pp. 904-993.

[4] N. E. Huang, M.-li Wu, W. Qu, S. R. Long, S. S. P. Shen, and J. E. Zhang, “Applications of Hilbert-Huang transform to non-stationary financial time series analysis,” *Applied Stochastic Models in Business and Industry*, vol. 19, no. 4, pp. 361-361, Oct. 2003.

[5] P. Flandrin, “Empirical mode decomposition as a filter bank,” *IEEE signal processing letters*, vol. 11, no. 2, pp. 112-114, 2004.

[6] Laurene Fausett, *Fundamentals of Neural Networks*, 1st ed. Upper Saddle River: Prentice Hall, 1994.

[7] Business International, *Forecasting foreign exchange rates*. New York: Business International Corporation, 1972.

[8] MetaQuotes Software Corp., “MetaTrader 4.” 2014.

[9] E. Fama, “Efficient Capital Markets: A Review Of Theory And Empirical Work,” *The journal of Finance*, 1970.

[10] B. Zvi, K. Alex, and M. Alan, *Essentials of Investments*, 5th ed. The McGraw-Hill Companies, 2003.

[11] J. J. Murphy, *Technical Analysis of the Financial Markets*. New Jersey: New York Institute of Finance, 1999.

[12] M. D. Flood, “Market structure and inefficiency in the foreign exchange market,” *Journal of International Money and Finance*, vol. 13, no. 2, pp. 131-158, Apr. 1994.

[13] E. Abounoori, M. Shahrazi, and S. Rasekhi, “An investigation of Forex market efficiency based on detrended fluctuation analysis: A case study for Iran,” *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 11, pp. 3170-3179, Jun. 2012.

[14] T. Imam, “Intelligent Computing and Foreign Exchange Rate Prediction : What We Know and We Don ’ t,” *Progress in Intelligent Computing and Applications*, vol. 1, no. 1, pp. 1-15, 2012.

[15] S. A. Nusair, “Real exchange rate dynamics in transition economies: a nonlinear analysis,” *International Journal of Finance and Economics*, vol. 18, no. 2012, pp. 188-204, 2013.

[16] E. G. Pavlidis, I. Paya, and D. A. Peel, “Forecast Evaluation of Nonlinear Models : The Case of Long-Span Real Exchange Rates,” *Journal of Forecasting*, vol. 595, no. 2011, pp. 580-595, 2012.

[17] R. Llinas, “Neurons,” *Scholarpedia*, vol. 3, no. 8, 2008.

[18] J. Sima, “Introduction to neural networks,” Prague, 1998.

[19] F. A. C. Azevedo et al., “Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain.,” *The Journal of comparative neurology*, vol. 513, no. 5, pp. 532-41, Apr. 2009.

[20] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design*. Thomson Learning, 1996.

[21] J. Heaton, “Feedforward Backpropogation Neural Networks,” in *Introduction to Neural Networks for Java*, 2nd ed., 2008, p. Chapter 5.

[22] Mathworks, “Matlab documentation.” Cambridge, MA, 2012.

[23] W. Duch and N. Jankowski, “Survey of Neural Transfer Functions,” *Neural Computing surveys*, vol. 2, pp. 163-212, 1999.

[24] H. Yonaba, F. Anctil, and V. Fortin, “Comparing Sigmoid Transfer Functions for Neural Network,” *Journal of Hydrologic Engineering*, no. April, pp. 275-283, 2010.

[25] Ö. Kişi and E. Uncuoğlu, "Comparison of three back-propagation training algorithms for two case studies," *Indian Journal of Engineering & Materials Sciences*, vol. 12, no. October, pp. 434-442, 2005.
[26] M. T. Hagan and M. B. Menhaj, "Training Feedforward Networks with the Marquardt Algorithm," *IEEE TRANSACTIONS ON NEURAL NETWORKS*, vol. 5, no. 6, pp. 2-6, 1994.
[27] H. Yu and B. M. Wilamowski, "Levenberg–Marquardt Training," in *Industrial Electronics Handbook*, 2nd ed., CRC Press, 2010, pp. 1-16.
[28] P. M. Crowley, "How Do You Make A Time Series Sing Like a Choir ? Extracting Embedded Frequencies from Economic and Financial Time Series using Empirical Mode Decomposition," *Corpus*, no. July, pp. 1-33, 2011.
[29] J.-P. Zhao and D.-ji Huang, "mirror extending and circular spline function for empirical mode decomposition method.pdf," *Journal of Zhejiang University*, vol. 2, no. 3, pp. 247-252, 2001.
[30] R. Faltermeier and A. Zeiler, "Sliding empirical mode decomposition," *IEEE submissions*, no. 1, 2010.
[31] J. Wang, Y. Peng, and X. Peng, "Similarity searching based boundary effect processing method for empirical mode decomposition," *Electronics Letters*, vol. 43, no. 1, pp. 1-2, 2007.
[32] Z. Wu and N. Huang, "Ensemble empirical mode decomposition: a noise-assisted data analysis method," *Advances in Adaptive Data Analysis*, no. August, 2009.
[33] M. Beine, "Chapter 6 : exchange rate determination : theory," in *International Finance*, 2009.
[34] J. Frenkel, "A monetary approach to the exchange rate: doctrinal aspects and empirical evidence," *the scandinavian Journal of economics*, vol. 78, no. 2, pp. 200-224, 1976.
[35] J. F. O. Bilson, "The Monetary Approach to the Exchange Rate : Some Empirical Evidence," *Staff Papers - International Monetary Fund*, vol. 25, no. 1, pp. 48-75, 1978.
[36] J. Frankel, "On the mark: A theory of floating exchange rates based on real interest differentials," *The American Economic Review*, vol. 69, no. 4, pp. 610-622, 1979.
[37] M. Dooley and J. Shafer, *Analysis of short-run exchange rate behavior: March, 1973 to September, 1975*. 1976.
[38] D. Caves and E. Feige, "Efficient foreign exchange markets and the monetary approach to exchange-rate determination," *The American economic review*, vol. 70, no. 1, pp. 120-134, 1980.
[39] R. Meese and K. Rogoff, "Empirical exchange rate models of the seventies: Do they fit out of sample?," *Journal of international economics*, vol. 14, pp. 3-24, 1983.
[40] D. Backus, "Empirical models of the exchange rate: Separating the wheat from the chaff," *Canadian Journal of Economics*, vol. 17, no. 4, pp. 824-846, 1984.
[41] C. C. P. Wolff, "A Comparison of Forecasting Results," *International Journal of Forecasting*, vol. 4, pp. 605-607, 1988.
[42] V. S. Somanath, "Efficient exchange rate forecasts: Lagged models better than the random walk," *Journal of International Money and Finance*, vol. 5, no. 2, pp. 195-220, 1986.
[43] C. Hakkio, "Does the exchange rate follow a random walk? A Monte Carlo study of four tests for a random walk," *Journal of International Money and Finance*, vol. 5, no. 2, pp. 221-229, 1986.
[44] G. J. Schinasi and P. A. V. B. Swamy, "The out-of-sample forecasting performance of exchange rate models when coefficients are allowed to change," *Journal of International Money and Finance*, vol. 8, no. 3, pp. 375-390, 1989.
[45] C. Liu and J. He, "A Variance Ratio Test of Random Walks in Foreign Exchange Rates," *The Journal of Finance*, vol. 46, no. 2, pp. 773-785, 1991.
[46] A. S. M. S. Azad, "Random walk and efficiency tests in the Asia-Pacific foreign exchange markets: Evidence from the post-Asian currency crisis data," *Research in International Business and Finance*, vol. 23, no. 3, pp. 322-338, Sep. 2009.
[47] M. Taylor and D. Peel, "Nonlinear adjustment, long-run equilibrium and exchange rate fundamentals," *Journal of international money and finance*, vol. 19, pp. 33-53, 2000.

[48] M. Bleaney and P. Mizen, “Nonlinearities in Exchange Rate Dynamics: Evidence from Five Currencies, 1973–94,” *Economic Record*, vol. 12, no. 216, pp. 36-45, 1996.

[49] C. Brooks, “Linear and non-linear (non-) forecastability of high-frequency exchange rates,” *Journal of forecasting*, vol. 16, no. 1996, pp. 125-145, 1997.

[50] J. Creedy, J. Lye, and V. Martin, “A non-linear model of the real US/UK exchange rate,” *Journal of Applied Econometrics*, vol. 11, no. 6, pp. 669-686, 1996.

[51] A. N. Refenes, M. Azema-Barac, and S. A. Karoussos, “Currency exchange rate forecasting by error backpropagation,” in *Proceedings of the 25th Hawaii International Conference on System Sciences*, 1992, pp. 504-515.

[52] M. Staley and P. Kim, “Predicting the canadian spot exchange rate with neural networks,” in *Computational intelligence for financial engineering: Proceedings of the IEEE/IAFE 1995*, 1995, pp. 108-112.

[53] T. Hann and E. Steurer, “Much ado about nothing? Exchange rate forecasting: Neural networks vs. linear models using monthly and weekly data,” *Neurocomputing*, vol. 10, pp. 323-339, 1996.

[54] J. Yao and C. L. Tan, “A case study on using neural networks to perform technical forecasting of forex,” *Neurocomputing*, vol. 34, pp. 79-98, 2000.

[55] A. K. Nag and A. Mitra, “Forecasting daily foreign exchange rates using genetically optimized neural networks,” *Journal of Forecasting*, vol. 21, no. 7, pp. 501-511, Nov. 2002.

[56] J. Chang, C. Kuan, and Y. Lin, “Forecasting Exchange Rates by Genetic Algorithms Based Back Propagation Network Model,” *Intelligent Information Hiding and …*, pp. 791-794, Sep. 2009.

[57] V. Kodogiannis and A. Lolis, “Forecasting financial time series using neural network and fuzzy system-based techniques,” *Neural computing & applications*, vol. 11, pp. 90-102, 2002.

[58] G. Zhang and V. Berardi, “Time series forecasting with neural network ensembles: an application for exchange rate prediction,” *Journal of the Operational Research*, vol. 52, no. 6, pp. 652-664, 2001.

[59] L. Yu, S. Wang, and K. Lai, “Adaptive smoothing neural networks in foreign exchange rate forecasting,” *Computational Science*, vol. 3516, pp. 523-530, 2005.

[60] S. Ardiansyah, M. Majid, and J. Zain, “Foreign exchange forecasting by using artificial neural networks: A survey of literature,” *umpir.ump.edu.my*, no. September, 2012.

[61] X. Zhang, K. K. Lai, and S.-Y. Wang, “A new approach for crude oil price analysis based on Empirical Mode Decomposition,” *Energy Economics*, vol. 30, no. 3, pp. 905-918, May 2008.

[62] L. Yu, S. Wang, and K. K. Lai, “Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm,” *Energy Economics*, vol. 30, no. 5, pp. 2623-2635, Sep. 2008.

[63] W. Wang, H. Zhao, Q. Li, and Z. Liu, “A Novel Hybrid Intelligent Model for Financial Time Series Forecasting and Its Application,” *2009 International Conference on Business Intelligence and Financial Engineering*, pp. 279-282, Jul. 2009.

[64] C. Fu, “Forecasting exchange rate with EMD-based support vector regression,” *IEEE submissions*, pp. 0-3, 2010.

[65] C. Lin, S.-hsiung Chiu, and T. Lin, “Empirical mode decomposition–based least squares support vector regression for foreign exchange rate forecasting,” *Economic Modelling*, vol. 29, no. 6, pp. 2583-2590, 2012.

[66] H. Yang and H. Lin, “Combining Artificial Intelligence with Non-linear Data Processing Techniques for Forecasting Exchange Rate Time Series.,” *International Journal of Digital Content Technology and its Applications*, vol. 6, no. April, pp. 276-284, 2012.

[67] L. Yu, S. Wang, K. Keung, and F. Wen, “A multiscale neural network learning paradigm for financial crisis forecasting,” *Neurocomputing*, vol. 73, no. 4-6, pp. 716-725, 2010.

[68] T. Xiong, Y. Bao, and Z. Hu, “Beyond one-step-ahead forecasting: Evaluation of alternative multi-step-ahead forecasting models for crude oil prices,” *Energy Economics*, vol. 40, pp. 405-415, Nov. 2013.

[69] J. Lawrence, *Introduction to Neural Networks*, 5th ed. Nevada City: California Scientific, 1993.
[70] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford: Oxford University Press, 1995.
[71] A. Fraser and H. Swinney, "Independent coordinates for strange attractors from mutual information.," *Physical review. A*, vol. 33, no. 2. pp. 1134-1140, Feb-1986.
[72] a. Sharma, "Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1 — A strategy for system predictor identification," *Journal of Hydrology*, vol. 239, no. 1-4, pp. 232-239, Dec. 2000.
[73] W. S. Sarle, "AI Neural Nets FAQ," 2002. [Online]. Available: http://www.faqs.org/faqs/ai-faq/neural-nets/. [Accessed: 18-Sep-2014].
[74] J. S. Zirilli, *Financial Prediction using Neural Networks*, 1st ed. London: International Thomson Computer Press, 1997.
[75] B. Frénay, G. Doquire, and M. Verleysen, "Is mutual information adequate for feature selection in regression ?," *Neural Networks*, vol. 48, pp. 1-7, 2013.
[76] L. Cao and A. S. Soofi, "Nonlinear deterministic forecasting of daily dollar exchange rates," *International Journal of Forecasting*, vol. 15, pp. 421-430, 1999.

Appendix A: The Levenberg-Marquardt training algorithm

The Levenberg-Marquardt algorithm is based on the Newton and Gauss-Newton algorithms for updating the weights of the

The Levenberg-Marquardt will be derived in four parts: The steepest descent algorithm, Newton's method, Gauss-Neuton's algorithm and finally the Levenberg-Marquardt algorithm [27]. Some of the commonly used indices are first introduced:

- p is the index of patterns from 1 to P, where P is the number of patterns
- m is the index of outputs from 1 to M, where M is the number of patterns
- i and j are the indices of weights, from 1 to N, where n is the number of weights
- k is the index of iterations

The sum squared error (SSE) is used to evaluate the training error, and is calculated by

$$E(\boldsymbol{x},\boldsymbol{w}) = \frac{1}{N}\sum_{p=1}^{P}\sum_{m=1}^{M}\boldsymbol{e}_{p,m}^{2} \qquad (\text{A-1})$$

Where $\boldsymbol{x}$ is the input vector, $\boldsymbol{w}$ is the weight vector and $\boldsymbol{e}_{p,m}$ is the training error of the m'th output and is defined as :

$$\boldsymbol{e}_{p,m} = \boldsymbol{d}_{p,m} - \boldsymbol{o}_{p,m} \qquad (\text{A-2})$$

where $\boldsymbol{d}$ is the desired output vector and $\boldsymbol{o}$ is the actual output vector.

The implementation of the Levenberg-Marquardt algorithm requires the calculation of the Jacobian Matrix. For a three layer multilayer perceptron network the computation of the Jacobian matrix is completed in the following steps:

1) Calculate the net value (net), slopes (y) and outputs (s) for all neurons in the first layer:

$$net_j^1 = \sum_{i=1}^{ni} I_i w_{j,i}^1 + w_{j,0}^1 \qquad (\text{A-3})$$

$$y_j^1 = f_j^1(net_j^1) \qquad (\text{A-4})$$

$$s_j^1 = \frac{\delta f_j^1}{\delta net_j^1} \quad \text{(A-5)}$$

where I_i is the network input at index *i*, the superscript *1* indicates that the first layer is used and *j* is the index of neurons in the first layer

2) Calculate the net values, slopes and outputs for the second layer using the outputs of the first layer as inputs:

$$net_j^2 = \sum_{i=1}^{ni} I_i w_{j,i}^2 + w_{j,0}^2 \quad \text{(A-6)}$$

$$y_j^2 = f_j^2(net_j^2) \quad \text{(A-7)}$$

$$s_j^2 = \frac{\delta f_j^2}{\delta net_j^2} \quad \text{(A-8)}$$

The same notation is used as in step 1.

3) Calculate the net values, slopes and outputs for the third layer using the outputs of the second layer as inputs:

$$net_j^3 = \sum_{i=1}^{ni} I_i w_{j,i}^3 + w_{j,0}^3 \quad \text{(A-9)}$$

$$o_j = f_j^3(net_j^3) \quad \text{(A-10)}$$

$$s_j^3 = \frac{\delta f_j^3}{\delta net_j^3} \quad \text{(A-11)}$$

The same notation is used as in step 1.

The node array ***y*** and the slope array ***s*** can be obtained when steps 1 to 3 are completed.

After the forward computation has been completed, the backward computation can be expressed as:

4) Calculate the error of output *j* and initial δ as the slope of the output

$$e_j = d_j - o_j \quad \text{(A-12)}$$

$$\delta^3_{j,j} = s^3_j \qquad (\text{A-13})$$

$$\delta^3_{j,k} = 0 \qquad (\text{A-14})$$

where δ_j is the desired output of output *j*, o_j is the actual output obtained using forward computation, $d^3_{j,j}$ is the self backpropagation and $d^3_{j,k}$ is the backpropagation of the output neurons.

5) Calculate the backpropagation δ from the inputs of the third layer to the second layer's outputs:

$$\delta^2_{j,k} = w^3_{j,k}\delta^3_{j,j} \qquad (\text{A-15})$$

where k is the index of the neurons on the second layer. This index ranges from 1 to n_2

6) Calculate the backpropagation δ from the outputs of the second layer to the second layer's inputs:

$$\delta^2_{j,k} = \delta^2_{j,k}s^2_k \qquad (\text{A-16})$$

where k is the index of the neurons on the second layer. This index ranges from 1 to n_2

7) Calculate the backpropagation δ from the inputs of the second layer to the first layer's outputs:

$$\delta^1_{j,k} = \sum_{i-1}^{n2} w^2_{j,i}\delta^2_{j,i} \qquad (\text{A-17})$$

where k is the index of the neurons on the first layer. This index ranges from 1 to n_1

8) Calculate the backpropagation δ from the outputs of the first layer to the first layer's inputs:

$$\delta^1_{j,k} = \delta^1_{j,k}s^1_k \qquad (\text{A-18})$$

where k is the index of the neurons on the first layer. This index ranges from 1 to n_1

9) Repeat steps 4 to 8 for the backpropagation of the other outputs.
10) Once the forward and backward computation has been completed the entire $\boldsymbol{\delta}$ and ***y*** array can be obtained.

11) Use the $\boldsymbol{\delta}$ and $\boldsymbol{y}$ arrays to calculate the elements of the Jacobian matrix:

$$\frac{\delta e_{p,m}}{\delta w_{j,i}} = -\delta_{m,j} y_{j,i} \qquad \text{(A-19)}$$

The Newtonian algorithm for weight updating requires the Hessian matrix, which is computationally intensive to calculate due to the requirements of second order derivatives. The Gauss-Newton method simplifies this with the assumption that the Hessian matrix, ***H,*** and the Jacobian matrix, ***J***, share the following relationship:

$$\boldsymbol{H} = \boldsymbol{J}^T \boldsymbol{J} \qquad \text{(A-20)}$$

This approximation decreases the computational complexity, but suffers from another limitation shared by the Hessian matrix, i.e. that the matrix is not invertible. The Levenberg-Marquardt algorithm uses another approximation for the Hessian matrix:

$$\boldsymbol{H} = \boldsymbol{J}^T \boldsymbol{J} + \boldsymbol{\mu I} \qquad \text{(A-21)}$$

where I is the identity matrix and μ is the combination coefficient.

This assumption ensures that the matrix is always invertible. The update rule of the Levenberg-Marquardt training algorithm is an adapted form of the Newton algorithm, and is expressed as:

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \left(\boldsymbol{J}_k^T \boldsymbol{J}_k + \mu \boldsymbol{I}\right)^{-1} \boldsymbol{J}_K \boldsymbol{e}_k \qquad \text{(A-22)}$$

where $\boldsymbol{e}$ is the error vector [27].

The training process using the Levenberg-Marquardt algorithm can be summarised as follows:

7) Generate the total error with randomly generated initial weights.
8) Adjust the weights using equation A-22.
9) Evaluate the total error with the new weights.
10) If the total error is increased retract this steps and repeat step 2 and increase the combination coefficient μ, by a predefined factor.
11) If the total error is decreased then keep the new weights and decrease the combination coefficient by the same factor used in step 4.
12) Repeat from step 2 with the new weights until the total error is smaller than the required threshold value.

Appendix B: Mutual information for input selection

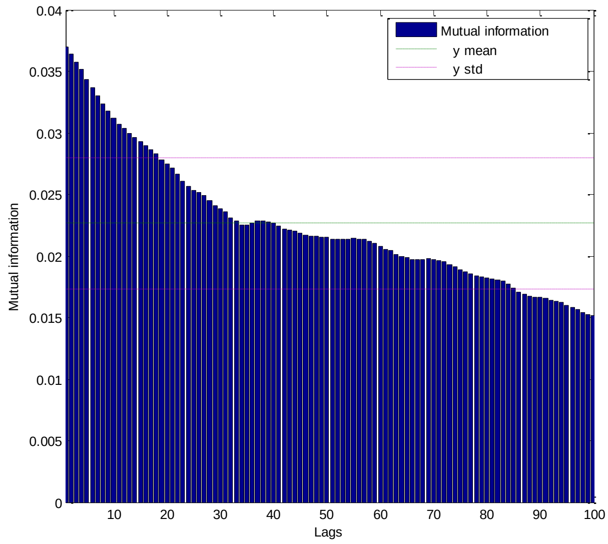

Figure B-1: The average monthly mutual information for 100 lagged values for the EUR/USD log returns sampled at 1 minute with a forecast horizon of 35 minutes

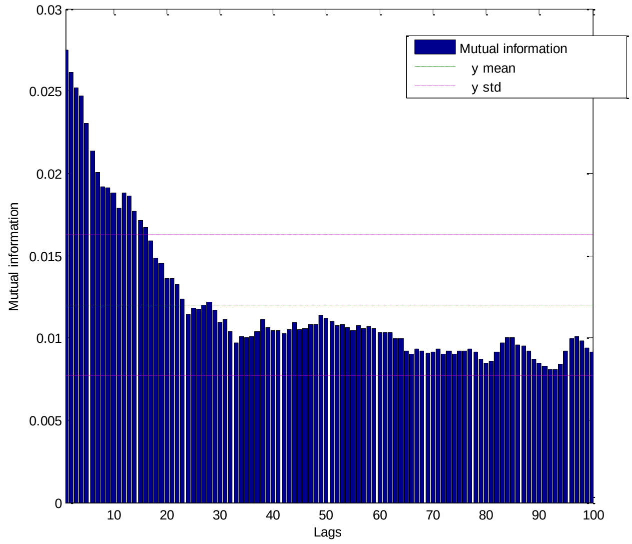

Figure B-2: The average monthly mutual information for 100 lagged values for the EUR/USD log returns sampled at 5 minutes with a forecast horizon of 35 minutes

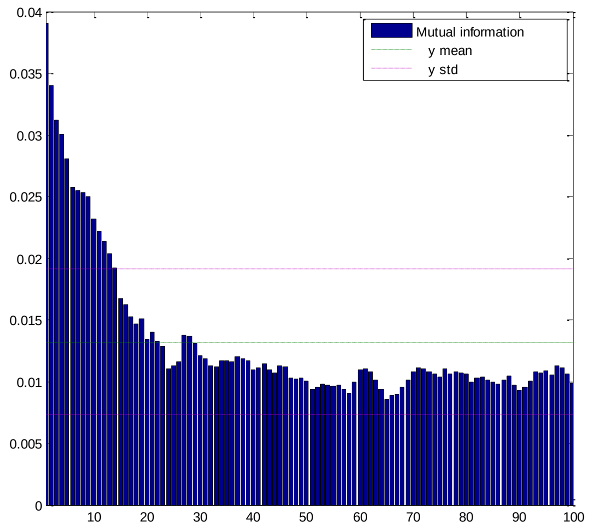

Figure B-3: The average monthly mutual information for 100 lagged values for the EUR/USD log returns sampled at 7 minutes with a forecast horizon of 35 minutes

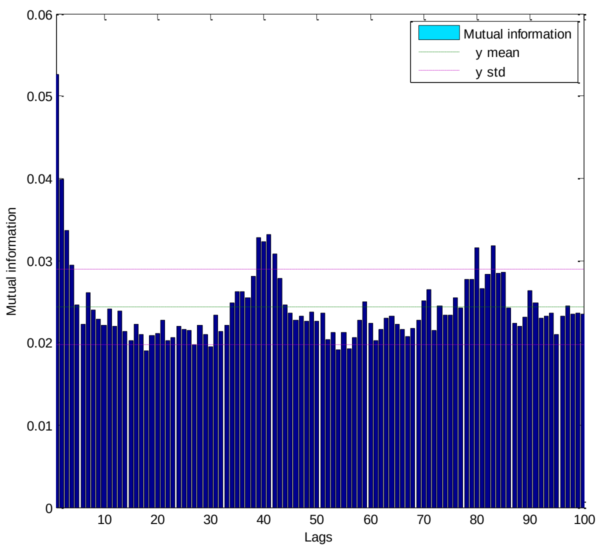

Figure B-4: The average monthly mutual information for 100 lagged values for the EUR/USD log returns sampled at 35 minutes with a forecast horizon of 35 minutes

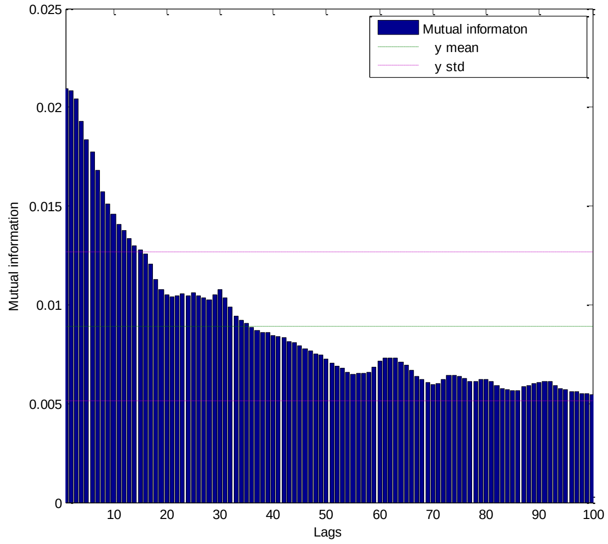

Figure B-5: **The average monthly mutual information for 100 lagged values for the USD/JPY log returns sampled at 1 minute with a forecast horizon of 10 minutes**

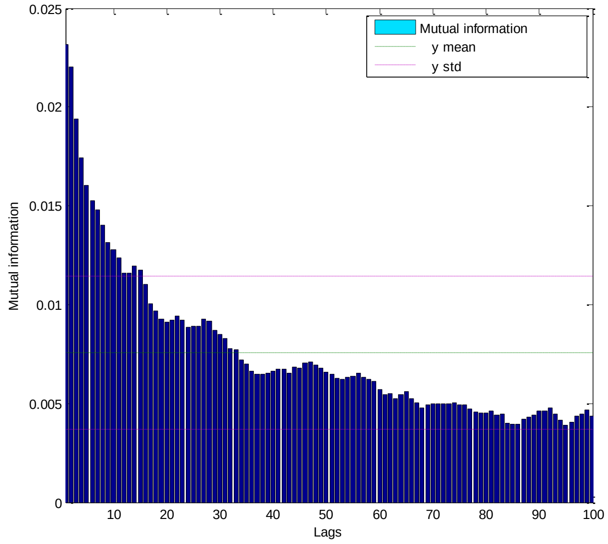

Figure B-6: **The average monthly mutual information for 100 lagged values for the USD/JPY log returns sampled at 2 minutes with a forecast horizon of 10 minutes**

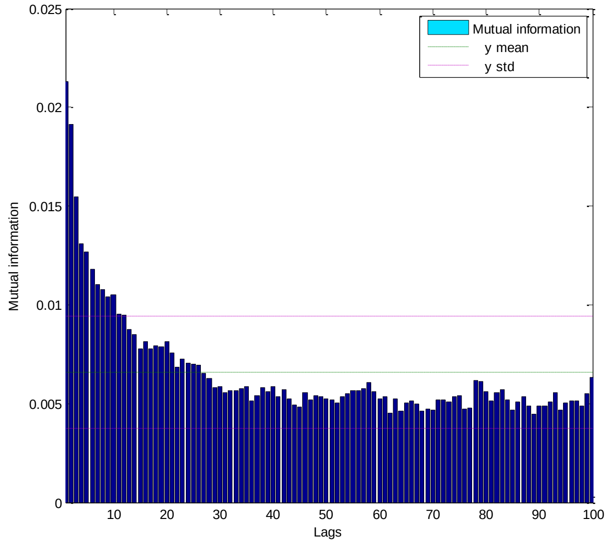

Figure B-7: The average monthly mutual information for 100 lagged values for the USD/JPY log returns sampled at 5 minutes with a forecast horizon of 10 minutes

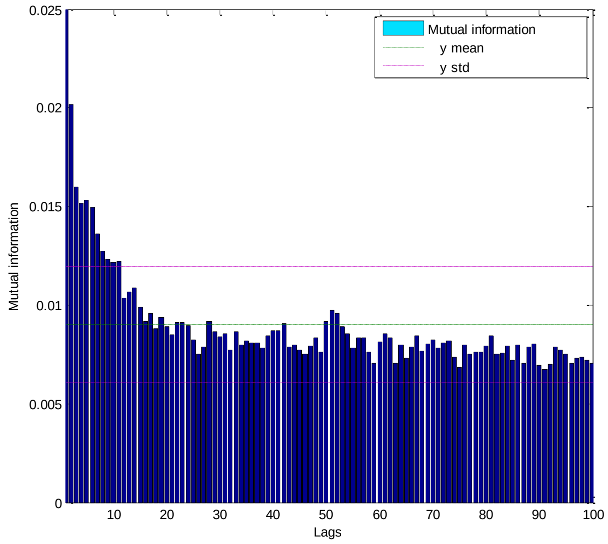

Figure B-8: The average monthly mutual information for 100 lagged values for the USD/JPY log returns sampled at 10 minutes with a forecast horizon of 10 minutes

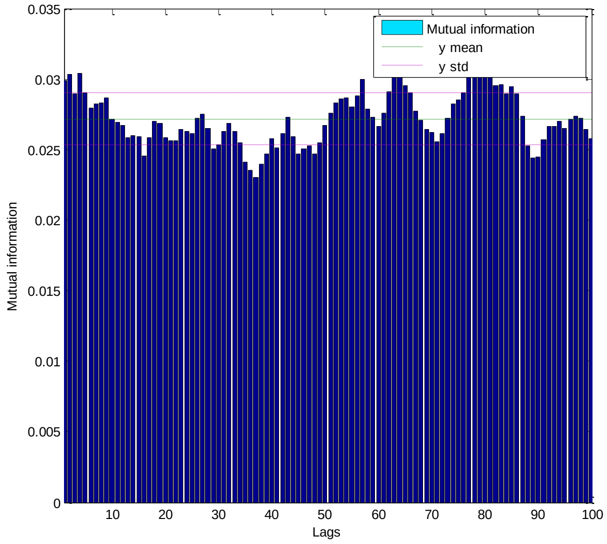

Figure B-9: The average monthly mutual information for 100 lagged values for the USD/ZAR log returns sampled at 15 minutes with a forecast horizon of 180 minutes

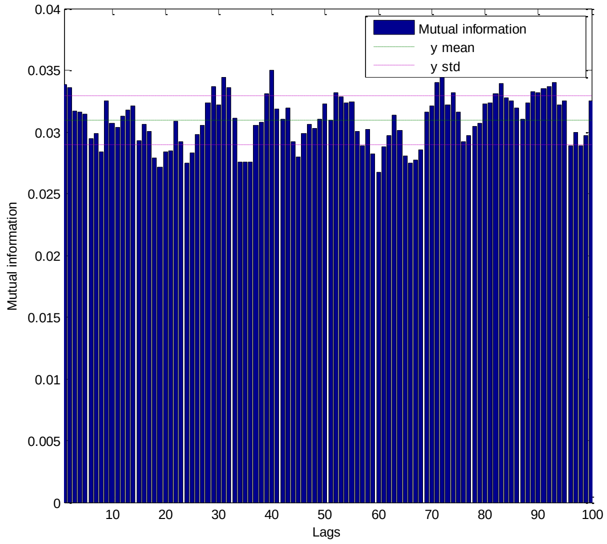

Figure B-10: The average monthly mutual information for 100 lagged values for the USD/ZAR log returns sampled at 30 minutes with a forecast horizon of 180 minutes

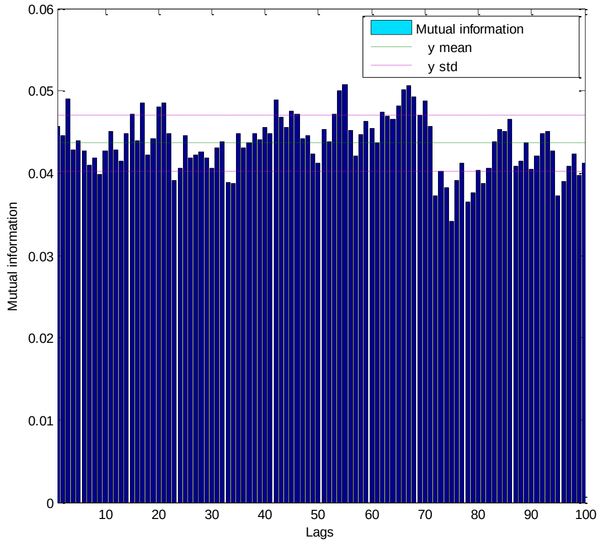

Figure B-11: The average monthly mutual information for 100 lagged values for the USD/ZAR log returns sampled at 60 minutes with a forecast horizon of 180 minutes

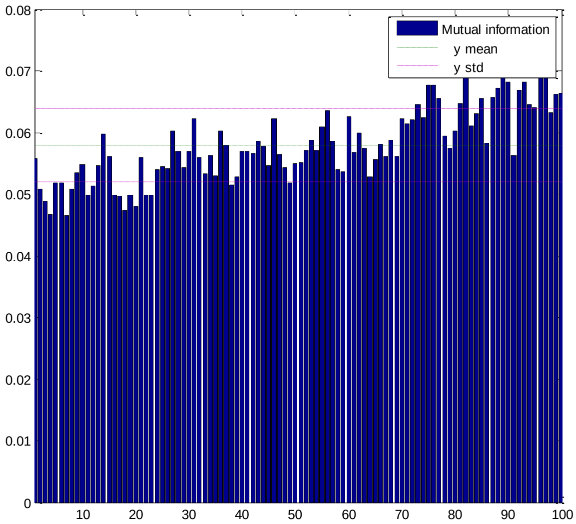

Figure B-12: The average monthly mutual information for 100 lagged values for the USD/ZAR log returns sampled at 90 minutes with a forecast horizon of 180 minutes

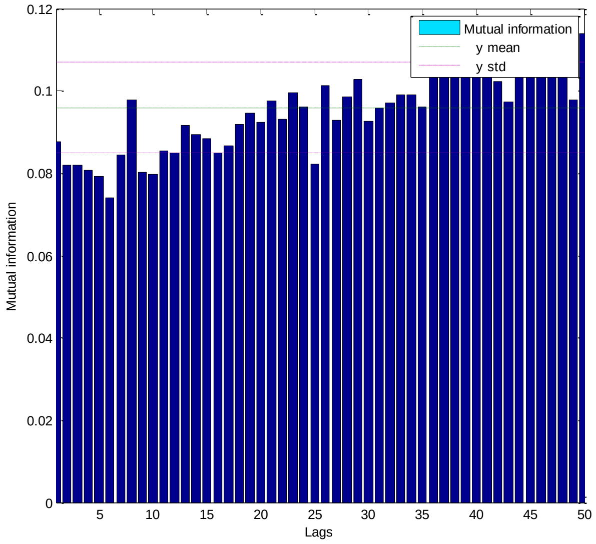

Figure B-13: The average monthly mutual information for 100 lagged values for the USD/ZAR log returns sampled at 180 minutes with a forecast horizon of 180 minutes

Appendix C: Monthly performance evaluation of results

See “Appendix C.pdf” on the enclosed CD.

Appendix C

EUR/USD at the optimal forecast horizon of 35 minutes

Table C-1: EUR/USD monthly forecasting performance for the EMD-filtered ANN using 1 minute samples with a 35 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0007	72.4886	0.6552	30.7262	466	0.0659	0.6567	0.1250	-0.0341
Mar	0.0008	70.4338	0.5439	22.2324	491	0.0453	0.6415	0.1043	-0.0458
Apr	0.0007	67.6941	0.5837	21.4616	506	0.0424	0.6186	0.1007	-0.0391
May	0.0007	65.9817	0.6369	23.8388	478	0.0499	0.6192	0.1098	-0.0359
Jun	0.0006	68.0365	0.5986	22.1740	478	0.0464	0.6088	0.1069	-0.0365
Jul	0.0006	69.0639	0.6952	26.1553	458	0.0571	0.6354	0.1159	-0.0352
Aug	0.0005	70.3196	0.6707	17.0784	471	0.0363	0.6030	0.0825	-0.0275
Sep	0.0005	72.4886	0.6614	17.5934	470	0.0374	0.6489	0.0755	-0.0270
Oct	0.0005	63.6986	0.5844	9.9089	466	0.0213	0.5236	0.0746	-0.0316
Nov	0.0007	63.4703	0.3688	7.8706	475	0.0166	0.5368	0.0741	-0.0421
Dec	0.0005	68.7215	0.6472	12.7372	466	0.0273	0.5815	0.0696	-0.0266

Table C-2: **EUR/USD monthly forecasting performance for the unfiltered ANN using 1 minute samples with a 35 minute forecast horizon**

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0009	55.3653	0.0017	-9.0695	269	-0.0337	0.4349	0.0849	-0.1137
Mar	0.0010	44.8630	-0.0145	-2.7023	38	-0.0711	0.3947	0.2422	-0.2637
Apr	0.0008	58.5616	0.0232	-5.8719	386	-0.0152	0.4301	0.0780	-0.0753
May	0.0009	48.5160	0.0623	-2.7766	169	-0.0164	0.4379	0.1248	-0.1108
Jun	0.0008	59.5890	0.0917	-6.1271	383	-0.0160	0.4178	0.0735	-0.0720
Jul	0.0009	59.0183	0.0006	-5.5784	217	-0.0257	0.4286	0.1095	-0.1128
Aug	0.0007	43.0365	-0.0210	-2.5559	147	-0.0174	0.3197	0.1357	-0.0848
Sep	0.0006	51.4840	0.0169	-7.5407	432	-0.0175	0.3310	0.0558	-0.0490
Oct	0.0006	56.6210	0.0810	-1.5181	225	-0.0067	0.4267	0.0769	-0.0640
Nov	0.0007	42.8082	-0.0586	-8.4625	323	-0.0262	0.3406	0.0665	-0.0694
Dec	0.0006	45.3196	-0.1895	-13.3309	434	-0.0307	0.2535	0.0472	-0.0544

Table C-3: EUR/USD monthly forecasting performance for the random walk model using 1 minute samples with a 35 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0013	34.1324	0.0112	-11.3100	465	-0.0243	0.2602	0.1170	-0.0647
Mar	0.0014	33.7900	-0.0426	-11.4292	454	-0.0252	0.2643	0.0997	-0.0624
Apr	0.0012	32.6484	-0.0240	-12.1904	474	-0.0257	0.2321	0.1152	-0.0605
May	0.0012	30.8219	-0.0499	-13.9217	477	-0.0292	0.2243	0.0999	-0.0599
Jun	0.0011	33.5616	0.0223	-8.9271	466	-0.0192	0.2554	0.1065	-0.0553
Jul	0.0012	32.7626	0.0478	-7.0563	443	-0.0159	0.2664	0.1154	-0.0554
Aug	0.0010	34.2466	-0.0525	-10.1994	461	-0.0221	0.2191	0.0885	-0.0489
Sep	0.0009	33.6758	-0.0252	-9.0690	448	-0.0202	0.2165	0.0858	-0.0458
Oct	0.0008	37.2146	0.0574	-9.0394	444	-0.0204	0.2432	0.0760	-0.0475
Nov	0.0010	32.7626	0.0421	-8.8945	462	-0.0193	0.2662	0.0804	-0.0504
Dec	0.0008	33.5616	-0.0406	-9.1241	451	-0.0202	0.2395	0.0691	-0.0448

Table C-4: **EUR/USD monthly forecasting performance for the EMD-filtered ANN using 5 minute samples with a 35 minute forecast horizon**

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0007	69.9772	0.5886	29.0892	444	0.0655	0.6757	0.1237	-0.0406
Mar	0.0008	66.4384	0.4639	16.2615	489	0.0333	0.5746	0.1019	-0.0465
Apr	0.0007	66.2100	0.5408	23.9218	514	0.0465	0.6031	0.1082	-0.0352
May	0.0006	70.8904	0.6563	23.1849	492	0.0471	0.6220	0.1041	-0.0353
Jun	0.0007	68.7215	0.5366	21.7612	484	0.0450	0.6157	0.1045	-0.0386
Jul	0.0008	67.0091	0.5203	21.7007	448	0.0484	0.6362	0.1069	-0.0426
Aug	0.0006	68.2648	0.6121	14.9268	451	0.0331	0.5987	0.0799	-0.0302
Sep	0.0006	64.2694	0.5118	10.7171	450	0.0238	0.5756	0.0709	-0.0342
Oct	0.0006	62.2146	0.4172	8.5851	510	0.0168	0.5039	0.0707	-0.0324
Nov	0.0007	57.9909	0.2428	5.7925	477	0.0121	0.4717	0.0762	-0.0386
Dec	0.0006	66.5525	0.5712	13.2049	342	0.0386	0.6287	0.0822	-0.0298

Table C-5: EUR/USD monthly forecasting performance for the unfiltered ANN using 5 minute samples with a 35 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0009	52.3973	0.0053	-6.2621	408	-0.0153	0.4216	0.0871	-0.0781
Mar	0.0009	46.9178	0.0226	-12.1398	413	-0.0294	0.3123	0.0844	-0.0733
Apr	0.0009	66.3242	-0.0074	-5.8507	180	-0.0325	0.4500	0.0909	-0.1247
May	0.0008	48.7443	0.0577	-7.5685	322	-0.0235	0.3509	0.1048	-0.0828
Jun	0.0008	58.6758	-0.1192	-8.8100	306	-0.0288	0.3987	0.0720	-0.0880
Jul	0.0008	55.7078	0.0421	-5.1035	415	-0.0123	0.4313	0.0807	-0.0714
Aug	0.0007	43.0365	-0.0423	-1.1681	22	-0.0531	0.2273	0.1654	-0.1559
Sep	0.0006	63.8128	0.0472	-4.4863	280	-0.0160	0.4821	0.0482	-0.0702
Oct	0.0006	57.4201	0.0372	-7.1816	378	-0.0190	0.3651	0.0555	-0.0573
Nov	0.0007	46.5753	-0.0969	-4.7424	190	-0.0250	0.3789	0.0841	-0.0857
Dec	0.0006	55.8219	-0.0092	-6.5603	396	-0.0166	0.3207	0.0534	-0.0464

Table C-6: **EUR/USD monthly forecasting performance for the random walk model using 5 minute samples with a 35 minute forecast horizon**

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0013	34.0183	-0.0250	-10.1524	457	-0.0222	0.2560	0.1245	-0.0635
Mar	0.0013	35.7306	-0.0260	-10.3673	446	-0.0232	0.2982	0.0923	-0.0644
Apr	0.0012	33.3333	-0.0341	-11.7315	480	-0.0244	0.2375	0.1139	-0.0598
May	0.0012	32.4201	-0.0273	-14.8933	480	-0.0310	0.2250	0.0930	-0.0609
Jun	0.0012	32.7626	-0.0339	-10.5005	472	-0.0222	0.2288	0.1163	-0.0563
Jul	0.0011	32.0776	0.0708	-8.2967	470	-0.0177	0.2362	0.1162	-0.0528
Aug	0.0010	34.8174	-0.0435	-10.9963	455	-0.0242	0.2440	0.0757	-0.0520
Sep	0.0009	34.3607	-0.0348	-10.0177	444	-0.0226	0.2365	0.0697	-0.0478
Oct	0.0008	35.3881	0.0199	-9.5791	456	-0.0210	0.2632	0.0716	-0.0501
Nov	0.0011	32.5342	-0.0324	-7.4577	445	-0.0168	0.2697	0.0838	-0.0490
Dec	0.0009	32.4201	-0.1227	-10.2897	455	-0.0226	0.2396	0.0673	-0.0474

Table C-7: EUR/USD monthly forecasting performance for the EMD-filtered ANN using 7 minute samples with a 35 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0008	67.5799	0.5713	24.7621	454	0.0545	0.6300	0.1213	-0.0439
Mar	0.0010	66.4384	0.3413	17.9430	497	0.0361	0.6076	0.0971	-0.0455
Apr	0.0007	64.6119	0.4993	15.5317	512	0.0303	0.5586	0.0971	-0.0422
May	0.0007	64.4977	0.5471	20.4219	456	0.0448	0.6140	0.1080	-0.0429
Jun	0.0007	65.6393	0.4904	18.5319	488	0.0380	0.5840	0.1015	-0.0398
Jul	0.0007	64.9543	0.5891	22.1843	470	0.0472	0.6085	0.1080	-0.0375
Aug	0.0005	66.4384	0.6208	12.9417	443	0.0292	0.5914	0.0793	-0.0358
Sep	0.0006	66.3242	0.2218	6.2240	444	0.0140	0.5135	0.0683	-0.0380
Oct	0.0005	65.2968	0.4904	8.6760	516	0.0168	0.5194	0.0671	-0.0317
Nov	0.0007	58.2192	0.2597	4.3485	475	0.0092	0.4926	0.0694	-0.0424
Dec	0.0006	63.1279	0.4012	4.6303	370	0.0125	0.5270	0.0692	-0.0447

Table C-8: EUR/USD monthly forecasting performance for the unfiltered ANN using 7 minute samples with a 35 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0009	52.7397	-0.0232	-5.7144	109	-0.0524	0.3761	0.1415	-0.1598
Mar	0.0009	48.5160	-0.0116	-8.0759	369	-0.0219	0.3767	0.0850	-0.0775
Apr	0.0008	57.7626	0.0021	-11.3090	462	-0.0245	0.3745	0.0633	-0.0694
May	0.0009	53.1963	0.0210	-4.7373	150	-0.0316	0.4600	0.1101	-0.1418
Jun	0.0008	61.6438	0.0458	-7.1006	439	-0.0162	0.4579	0.0582	-0.0706
Jul	0.0009	53.8813	-0.0520	-8.7371	220	-0.0397	0.3955	0.0909	-0.1175
Aug	0.0007	42.4658	-0.0412	-7.4540	447	-0.0167	0.3199	0.0711	-0.0526
Sep	0.0006	57.8767	-0.0104	-6.6823	398	-0.0168	0.3869	0.0472	-0.0539
Oct	0.0006	56.1644	0.0231	-8.1070	423	-0.0192	0.3712	0.0499	-0.0557
Nov	0.0007	42.9224	0.1044	-8.6898	403	-0.0216	0.3449	0.0625	-0.0607
Dec	0.0006	57.4201	0.0669	-8.4702	473	-0.0179	0.3531	0.0450	-0.0483

Table C-9: EUR/USD monthly forecasting performance for the random walk model using 7 minute samples with a 35 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0013	32.8767	-0.0208	-11.0364	469	-0.0235	0.2516	0.1267	-0.0644
Mar	0.0013	35.0457	-0.0253	-11.4165	446	-0.0256	0.2601	0.1018	-0.0630
Apr	0.0012	35.3881	-0.0022	-10.5204	444	-0.0237	0.2477	0.1109	-0.0606
May	0.0012	31.9635	-0.0281	-14.3003	480	-0.0298	0.2250	0.0979	-0.0605
Jun	0.0012	31.9635	-0.0294	-11.0469	470	-0.0235	0.2468	0.1037	-0.0582
Jul	0.0011	33.5616	0.0499	-5.1927	435	-0.0119	0.2943	0.1111	-0.0547
Aug	0.0010	34.0183	-0.0565	-10.1885	457	-0.0223	0.2210	0.0863	-0.0489
Sep	0.0009	32.9909	0.0004	-9.7654	444	-0.0220	0.2342	0.0714	-0.0473
Oct	0.0008	34.2466	0.0349	-9.7750	458	-0.0213	0.2227	0.0852	-0.0478
Nov	0.0010	33.5616	-0.0079	-9.1152	466	-0.0196	0.2511	0.0869	-0.0500
Dec	0.0009	35.3881	-0.0539	-9.4657	445	-0.0213	0.2382	0.0737	-0.0471

Table C-10: EUR/USD monthly forecasting performance for the EMD-filtered ANN using 35 minute samples with a 35 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0009	51.7714	0.0852	-5.6211	500	-0.0112	0.4140	0.0870	-0.0682
Mar	0.0009	39.5429	0.0382	-4.2137	196	-0.0215	0.3367	0.1345	-0.0937
Apr	0.0009	54.2857	0.0206	-5.8141	438	-0.0133	0.3858	0.0762	-0.0615
May	0.0009	44.9143	0.1327	-2.1859	382	-0.0057	0.4110	0.1028	-0.0698
Jun	0.0008	40.8000	0.0885	-2.1550	114	-0.0189	0.3684	0.1602	-0.1149
Jul	0.0008	46.9714	0.1002	-1.1180	514	-0.0022	0.3716	0.0945	-0.0507
Aug	0.0007	49.6000	-0.0842	-12.7027	414	-0.0307	0.2850	0.0573	-0.0622
Sep	0.0006	47.0857	0.0737	-3.7047	275	-0.0135	0.3527	0.0770	-0.0596
Oct	0.0006	46.6286	0.1822	-5.0055	445	-0.0112	0.3483	0.0658	-0.0474
Nov	0.0007	61.0286	0.0416	-4.0481	312	-0.0130	0.4295	0.0631	-0.0647
Dec	0.0006	49.2571	-0.0419	-1.6761	14	-0.1197	0.5000	0.1016	-0.3894

Table C-11: EUR/USD monthly forecasting performance for the unfiltered ANN using 35 minute samples with a 35 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0009	54.4000	0.0114	-5.7410	189	-0.0304	0.4339	0.1108	-0.1265
Mar	0.0009	44.5714	-0.0245	-6.2238	300	-0.0207	0.3700	0.1066	-0.0851
Apr	0.0009	52.4571	-0.0269	-6.1270	92	-0.0666	0.2935	0.1443	-0.1485
May	0.0009	40.8000	0.0469	-5.1890	132	-0.0393	0.3712	0.1618	-0.1455
Jun	0.0008	43.2000	0.0431	-9.5817	400	-0.0240	0.3325	0.0900	-0.0721
Jul	0.0008	52.0000	0.0331	-1.1746	119	-0.0099	0.5126	0.1296	-0.1409
Aug	0.0007	36.9143	-0.0228	-0.3336	19	-0.0176	0.4737	0.2131	-0.1097
Sep	0.0007	46.9714	-0.0224	-3.6112	232	-0.0156	0.4655	0.0579	-0.0756
Oct	0.0006	56.9143	-0.0145	-4.1799	102	-0.0410	0.4412	0.0900	-0.1390
Nov	0.0007	40.3429	0.0237	-6.9823	376	-0.0186	0.3537	0.0707	-0.0615
Dec	0.0006	35.7714	-0.0727	-3.8668	70	-0.0552	0.2857	0.1335	-0.1332

Table C-12: EUR/USD monthly forecasting performance for the random walk model using 35 minute samples with a 35 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0012	33.1429	0.0219	-10.2638	484	-0.0212	0.2314	0.1392	-0.0600
Mar	0.0013	35.0857	-0.0081	-12.7479	462	-0.0276	0.2489	0.0981	-0.0622
Apr	0.0012	30.8571	-0.0353	-10.3446	457	-0.0226	0.2582	0.1029	-0.0588
May	0.0013	30.6286	-0.0691	-16.9516	490	-0.0346	0.1980	0.0893	-0.0598
Jun	0.0012	31.6571	-0.0375	-11.3852	458	-0.0249	0.2576	0.0986	-0.0607
Jul	0.0011	33.4857	0.0993	-8.4250	450	-0.0187	0.2467	0.1070	-0.0539
Aug	0.0009	33.9429	-0.0237	-10.1560	473	-0.0215	0.2262	0.0806	-0.0473
Sep	0.0009	34.8571	-0.0236	-9.3936	448	-0.0210	0.2612	0.0661	-0.0481
Oct	0.0008	32.0000	0.0208	-9.4199	475	-0.0198	0.2611	0.0689	-0.0471
Nov	0.0010	35.3143	0.0287	-9.2748	460	-0.0202	0.2696	0.0780	-0.0514
Dec	0.0009	33.4857	-0.0740	-8.8124	448	-0.0197	0.2746	0.0651	-0.0480

USD/JPY at the optimal forecast horizon of 10 minutes

Table C-13: USD/JPY monthly forecasting performance for the EMD-filtered ANN using 1 minute samples with a 10 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0006	72.4104	0.6228	89.0570	1687	0.0528	0.6633	0.1185	-0.0330
Mar	0.0004	73.6156	0.6833	66.4492	1646	0.0404	0.6616	0.0888	-0.0278
Apr	0.0006	66.9055	0.4852	76.9220	1650	0.0466	0.6400	0.1241	-0.0395
May	0.0005	69.6743	0.5646	65.4117	1725	0.0379	0.5699	0.1092	-0.0272
Jun	0.0007	71.0098	0.6387	171.4047	1735	0.0988	0.7009	0.2029	-0.0419
Jul	0.0005	71.4007	0.5961	72.4035	1742	0.0416	0.6297	0.1041	-0.0304
Aug	0.0005	73.1922	0.6186	74.9942	1680	0.0446	0.6887	0.0945	-0.0314
Sep	0.0005	70.9772	0.4007	35.5899	1697	0.0210	0.6364	0.0631	-0.0313
Oct	0.0003	70.3257	0.6214	30.2352	1725	0.0175	0.6157	0.0539	-0.0258
Nov	0.0003	67.3616	0.5951	17.9992	1700	0.0106	0.5165	0.0511	-0.0226
Dec	0.0003	70.6515	0.5025	23.1182	1650	0.0140	0.4721	0.0636	-0.0203

Table C-14: USD/JPY monthly forecasting performance for the unfiltered ANN using 1 minute samples with a 10 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0007	53.5831	-0.0569	-20.7561	942	-0.0220	0.4098	0.0839	-0.0722
Mar	0.0006	62.9316	0.0809	-21.1921	1423	-0.0149	0.4308	0.0479	-0.0481
Apr	0.0007	65.0489	0.0945	-20.8741	1394	-0.0150	0.4806	0.0595	-0.0600
May	0.0007	59.1857	0.0184	-23.6042	1327	-0.0178	0.3941	0.0592	-0.0518
Jun	0.0009	60.9121	0.0218	-16.7749	1168	-0.0144	0.4760	0.1098	-0.0789
Jul	0.0006	58.2410	0.0352	-11.7363	594	-0.0198	0.4327	0.0882	-0.0834
Aug	0.0006	50.8795	0.0034	-22.7267	1310	-0.0173	0.4198	0.0599	-0.0551
Sep	0.0005	47.0358	0.0258	-16.1869	962	-0.0168	0.3773	0.0619	-0.0526
Oct	0.0004	68.8274	0.0420	-18.6424	1431	-0.0130	0.4410	0.0292	-0.0391
Nov	0.0004	66.0912	0.0111	-10.2912	809	-0.0127	0.4363	0.0396	-0.0467
Dec	0.0004	60.5863	0.0128	-13.5542	1001	-0.0135	0.3347	0.0484	-0.0384

Table C-15: USD/JPY monthly forecasting performance for the random walk model using 1 minute samples with a 10 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0010	34.3322	-0.0239	-27.6752	1579	-0.0175	0.2780	0.0985	-0.0434
Mar	0.0008	33.8436	-0.0875	-30.6731	1598	-0.0192	0.2578	0.0723	-0.0393
Apr	0.0010	33.4853	0.0030	-29.9966	1583	-0.0189	0.2742	0.0993	-0.0444
May	0.0009	33.8111	0.0163	-28.7243	1610	-0.0178	0.2404	0.0892	-0.0384
Jun	0.0014	32.6710	-0.0312	-33.9254	1610	-0.0211	0.2758	0.1391	-0.0507
Jul	0.0009	33.2899	-0.0270	-33.3349	1650	-0.0202	0.2503	0.0797	-0.0404
Aug	0.0008	35.8632	0.0024	-26.8344	1539	-0.0174	0.2820	0.0795	-0.0413
Sep	0.0007	35.1140	-0.0407	-29.0401	1584	-0.0183	0.2569	0.0613	-0.0367
Oct	0.0006	33.6482	-0.0573	-32.1894	1636	-0.0197	0.2194	0.0507	-0.0334
Nov	0.0006	35.7003	-0.0549	-30.0031	1576	-0.0190	0.1986	0.0534	-0.0317
Dec	0.0006	35.2443	-0.0407	-30.4321	1566	-0.0194	0.1711	0.0605	-0.0309

Table C-16: USD/JPY monthly forecasting performance for the EMD-filtered ANN using 2 minute samples with a 10 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0006	67.5570	0.5055	69.5372	1677	0.0415	0.6225	0.1127	-0.0349
Mar	0.0004	69.3811	0.5926	51.7585	1668	0.0310	0.6337	0.0816	-0.0305
Apr	0.0006	67.2638	0.5499	78.3010	1702	0.0460	0.6340	0.1214	-0.0367
May	0.0006	68.6319	0.5409	60.5168	1711	0.0354	0.5658	0.1062	-0.0280
Jun	0.0008	68.6971	0.5734	137.9591	1765	0.0782	0.6669	0.1868	-0.0436
Jul	0.0006	66.5147	0.4923	50.5328	1658	0.0305	0.5887	0.0973	-0.0334
Aug	0.0005	68.9902	0.4362	59.5286	1672	0.0356	0.6567	0.0898	-0.0343
Sep	0.0005	69.3811	0.3706	34.1780	1625	0.0210	0.6105	0.0664	-0.0300
Oct	0.0004	69.7394	0.5941	26.4785	1662	0.0159	0.5842	0.0560	-0.0261
Nov	0.0004	66.9381	0.5017	12.9146	1732	0.0075	0.5040	0.0474	-0.0234
Dec	0.0004	66.3844	0.3613	9.7516	1764	0.0055	0.4099	0.0598	-0.0225

Table C-17: USD/JPY monthly forecasting performance for the unfiltered ANN using 2 minute samples with a 10 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0007	55.5375	-0.0009	-22.6247	1251	-0.0181	0.4357	0.0696	-0.0621
Mar	0.0005	56.6775	0.0428	-21.1100	1283	-0.0165	0.4014	0.0563	-0.0504
Apr	0.0007	62.8990	-0.0028	-25.6196	1606	-0.0160	0.4502	0.0603	-0.0546
May	0.0007	56.8078	-0.0037	-23.7244	1665	-0.0142	0.3676	0.0622	-0.0425
Jun	0.0009	64.5603	0.0042	-26.4490	1433	-0.0185	0.4934	0.0686	-0.0695
Jul	0.0007	58.2085	0.0253	-13.6261	1033	-0.0132	0.4124	0.0830	-0.0594
Aug	0.0006	54.5928	-0.0386	-24.3883	1418	-0.0172	0.4055	0.0579	-0.0513
Sep	0.0005	58.2085	-0.0004	-28.1036	1657	-0.0170	0.4098	0.0342	-0.0426
Oct	0.0004	62.5733	0.0111	-25.4900	1470	-0.0173	0.3810	0.0294	-0.0396
Nov	0.0004	63.7785	0.0092	-17.8149	1127	-0.0158	0.3833	0.0280	-0.0383
Dec	0.0004	59.3811	-0.0012	-7.7624	326	-0.0238	0.3865	0.0788	-0.0796

Table C-18: **USD/JPY monthly forecasting performance for the random walk model using 2 minute samples with a 10 minute forecast horizon**

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0010	33.6808	0.0043	-24.9287	1593	-0.0156	0.2781	0.1011	-0.0419
Mar	0.0008	33.3876	-0.0684	-29.4325	1572	-0.0187	0.2659	0.0710	-0.0396
Apr	0.0010	34.2671	0.0110	-26.9400	1596	-0.0169	0.2845	0.0981	-0.0433
May	0.0010	33.8436	-0.0181	-31.1729	1626	-0.0192	0.2312	0.0904	-0.0389
Jun	0.0013	33.1270	-0.0054	-29.4382	1637	-0.0180	0.2822	0.1348	-0.0482
Jul	0.0010	32.5081	-0.0521	-33.3436	1652	-0.0202	0.2421	0.0815	-0.0398
Aug	0.0008	33.5179	-0.0018	-28.7283	1573	-0.0183	0.2727	0.0793	-0.0409
Sep	0.0008	35.9609	-0.0525	-29.4517	1590	-0.0185	0.2560	0.0633	-0.0370
Oct	0.0006	34.4300	-0.0620	-30.5972	1624	-0.0188	0.2389	0.0488	-0.0337
Nov	0.0006	35.2117	-0.0476	-30.4824	1586	-0.0192	0.2087	0.0485	-0.0319
Dec	0.0006	35.1466	-0.0199	-31.3129	1614	-0.0194	0.1766	0.0589	-0.0310

Table C-19: USD/JPY monthly forecasting performance for the EMD-filtered ANN using 5 minute samples with a 10 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0007	58.9443	0.3176	24.5464	1641	0.0160	0.5003	0.1073	-0.0412
Mar	0.0005	58.3252	0.2450	6.0994	1720	0.0033	0.4901	0.0661	-0.0364
Apr	0.0007	63.1476	0.3408	35.0325	1874	0.0199	0.5454	0.1000	-0.0390
May	0.0007	58.7162	0.2483	11.8873	1818	0.0082	0.4527	0.0909	-0.0362
Jun	0.0010	56.5331	0.2203	31.8775	1824	0.0158	0.5406	0.1405	-0.0533
Jul	0.0006	61.5184	0.3494	21.9360	1517	0.0127	0.5234	0.0885	-0.0392
Aug	0.0006	60.5409	0.2300	16.0325	1762	0.0088	0.5482	0.0721	-0.0397
Sep	0.0005	57.8038	0.1698	3.4278	1760	0.0017	0.4824	0.0575	-0.0334
Oct	0.0004	56.6634	0.2127	-7.4385	1807	-0.0039	0.4588	0.0421	-0.0321
Nov	0.0004	60.6061	0.2409	-5.2758	1800	-0.0008	0.4300	0.0419	-0.0278
Dec	0.0004	55.2623	0.1545	-13.3860	1375	-0.0081	0.3084	0.0533	-0.0308

Table C-20: USD/JPY monthly forecasting performance for the unfiltered ANN using 5 minute samples with a 10 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0008	58.0319	-0.0106	-14.8532	1069	-0.0139	0.4556	0.0789	-0.0660
Mar	0.0005	54.1870	0.0098	-26.8063	1524	-0.0176	0.3760	0.0503	-0.0454
Apr	0.0007	62.0398	0.0005	-27.5817	1548	-0.0178	0.4289	0.0615	-0.0550
May	0.0007	64.0274	0.0580	-23.8861	1267	-0.0189	0.3717	0.0585	-0.0506
Jun	0.0010	49.2669	-0.0345	-29.1357	1690	-0.0172	0.4272	0.0958	-0.0600
Jul	0.0007	56.0443	-0.0034	-7.1626	275	-0.0260	0.3964	0.1543	-0.1225
Aug	0.0006	54.0241	-0.0145	-26.3057	1626	-0.0162	0.4102	0.0533	-0.0476
Sep	0.0005	53.9915	-0.0392	-6.2910	351	-0.0179	0.4217	0.1013	-0.0904
Oct	0.0004	63.4734	0.0231	-25.9523	1683	-0.0154	0.3928	0.0287	-0.0369
Nov	0.0004	61.6813	0.0435	-17.3937	1212	-0.0144	0.3828	0.0298	-0.0367
Dec	0.0004	56.7937	0.0003	-23.8129	1439	-0.0165	0.2745	0.0418	-0.0331

Table C-21: USD/JPY monthly forecasting performance for the random walk model using 5 minute samples with a 10 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0011	35.9400	-0.0329	-28.0609	1571	-0.0179	0.2521	0.1102	-0.0425
Mar	0.0008	35.0277	-0.0309	-27.7350	1579	-0.0176	0.2755	0.0682	-0.0387
Apr	0.0010	35.0603	-0.0040	-26.8267	1587	-0.0169	0.2930	0.0953	-0.0439
May	0.0009	33.5288	-0.0039	-29.3334	1588	-0.0185	0.2443	0.0928	-0.0400
Jun	0.0014	33.7569	-0.0313	-32.7356	1612	-0.0203	0.2835	0.1377	-0.0508
Jul	0.0009	34.3109	0.0130	-29.3065	1628	-0.0180	0.2635	0.0849	-0.0402
Aug	0.0008	34.6041	-0.0099	-27.6556	1587	-0.0174	0.2710	0.0802	-0.0399
Sep	0.0008	34.5715	-0.0294	-29.2100	1608	-0.0182	0.2407	0.0633	-0.0353
Oct	0.0006	34.1805	-0.0288	-30.8187	1596	-0.0193	0.2343	0.0498	-0.0341
Nov	0.0006	35.6468	0.0108	-28.6637	1581	-0.0181	0.2075	0.0537	-0.0314
Dec	0.0006	35.5165	0.0131	-27.9292	1580	-0.0177	0.1823	0.0629	-0.0302

Table C-22: USD/JPY monthly forecasting performance for the EMD-filtered ANN using 10 minute samples with a 10 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0007	44.9332	0.0729	-17.0075	1641	-0.0104	0.3876	0.0816	-0.0451
Mar	0.0006	45.3242	0.0090	-3.7629	89	-0.0423	0.2921	0.2047	-0.1407
Apr	0.0007	42.5546	0.0950	-17.4636	1537	-0.0114	0.3611	0.0882	-0.0454
May	0.0007	40.9906	0.0356	-10.3834	904	-0.0115	0.3319	0.1117	-0.0545
Jun	0.0009	43.3366	0.0407	-1.8552	695	-0.0027	0.4432	0.1775	-0.0943
Jul	0.0007	42.6849	0.0075	-2.3530	190	-0.0124	0.4211	0.1879	-0.1381
Aug	0.0006	48.8107	0.0608	-17.7939	1314	-0.0135	0.4041	0.0672	-0.0504
Sep	0.0006	43.8579	-0.0195	-22.6294	1438	-0.0157	0.3554	0.0540	-0.0424
Oct	0.0004	56.4353	-0.0132	-8.4818	390	-0.0217	0.3744	0.0677	-0.0683
Nov	0.0004	48.5826	-0.0095	-31.5393	1924	-0.0164	0.2874	0.0326	-0.0306
Dec	0.0004	48.0287	0.0274	-4.4239	351	-0.0126	0.3675	0.0925	-0.0670

Table C-23: USD/JPY monthly forecasting performance for the unfiltered ANN using 10 minute samples with a 10 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0007	52.1017	-0.0264	-23.1051	1177	-0.0196	0.4010	0.0768	-0.0618
Mar	0.0006	55.2949	-0.0162	-12.3294	537	-0.0230	0.3929	0.0882	-0.0795
Apr	0.0007	59.0746	-0.0250	-26.9354	1341	-0.0201	0.4519	0.0616	-0.0636
May	0.0007	53.0792	0.0316	-23.0982	1506	-0.0153	0.3645	0.0644	-0.0451
Jun	0.0009	59.7263	0.0145	-31.1030	1653	-0.0188	0.4549	0.0734	-0.0617
Jul	0.0007	55.1645	0.0079	-11.0661	662	-0.0167	0.4018	0.1066	-0.0776
Aug	0.0006	57.4454	-0.0151	-23.5149	1236	-0.0190	0.4231	0.0560	-0.0573
Sep	0.0005	55.0668	0.0048	-5.6934	322	-0.0177	0.4441	0.0827	-0.0866
Oct	0.0004	60.8016	0.0242	-6.5848	373	-0.0177	0.4531	0.0655	-0.0778
Nov	0.0004	59.8240	0.0040	-2.4034	277	-0.0087	0.3971	0.0898	-0.0666
Dec	0.0004	55.2623	0.0154	-26.1103	1595	-0.0164	0.2502	0.0441	-0.0311

Table C-24: **USD/JPY monthly forecasting performance for the random walk model using 10 minute samples with a 10 minute forecast horizon**

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0010	33.6592	0.0041	-24.9534	1593	-0.0157	0.2775	0.1013	-0.0418
Mar	0.0008	33.3659	-0.0684	-29.4325	1572	-0.0187	0.2659	0.0710	-0.0396
Apr	0.0010	34.2457	0.0110	-26.9244	1596	-0.0169	0.2845	0.0981	-0.0433
May	0.0010	33.8547	-0.0181	-31.1871	1626	-0.0192	0.2312	0.0903	-0.0389
Jun	0.0013	33.1378	-0.0052	-29.3467	1637	-0.0179	0.2822	0.1348	-0.0481
Jul	0.0010	32.4862	-0.0521	-33.3103	1650	-0.0202	0.2430	0.0814	-0.0399
Aug	0.0008	33.4963	-0.0018	-28.7283	1573	-0.0183	0.2727	0.0793	-0.0409
Sep	0.0008	35.9726	-0.0534	-29.5235	1590	-0.0186	0.2553	0.0629	-0.0370
Oct	0.0006	34.4412	-0.0617	-30.5481	1622	-0.0188	0.2392	0.0488	-0.0338
Nov	0.0006	35.2232	-0.0476	-30.4966	1586	-0.0192	0.2081	0.0486	-0.0319
Dec	0.0006	35.1580	-0.0199	-31.2995	1614	-0.0194	0.1766	0.0589	-0.0310

USD/ZAR at the optimal forecast horizon of 180 minutes

Table C-25: USD/ZAR monthly forecasting performance for the EMD-filtered ANN using 15 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0025	63.1944	0.3874	-2.7820	80	-0.0348	0.3250	0.3715	-0.2096
Mar	0.0025	67.3611	0.5157	1.4430	58	0.0249	0.4828	0.3399	-0.2534
Apr	0.0025	65.9722	0.5598	2.9485	83	0.0355	0.4458	0.3790	-0.2081
May	0.0036	72.2222	0.6627	20.6502	83	0.2488	0.5542	0.7277	-0.2580
Jun	0.0038	73.6111	0.4596	4.9852	75	0.0665	0.4933	0.4764	-0.2671
Jul	0.0027	71.5278	0.5927	8.2537	75	0.1100	0.5733	0.3876	-0.2204
Aug	0.0030	68.0556	0.4287	-0.8634	77	-0.0112	0.4026	0.4413	-0.2699
Sep	0.0036	71.5278	0.4007	3.8906	71	0.0548	0.5070	0.4718	-0.3090
Oct	0.0022	64.5833	0.4921	-2.5063	81	-0.0309	0.3457	0.2997	-0.1862
Nov	0.0030	72.2222	0.4268	3.5128	71	0.0495	0.5070	0.3370	-0.2165
Dec	0.0032	61.8056	0.1776	2.5362	68	0.0373	0.4265	0.4089	-0.2291

Table C-26: USD/ZAR monthly forecasting performance for the unfiltered ANN using 15 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0027	43.0556	0.0238	-11.1370	61	-0.1826	0.2623	0.2620	-0.3199
Mar	0.0027	62.5000	0.0397	-0.7784	1	-0.7784	0.0000		0.0000
Apr	0.0030	55.5556	0.0956	-8.5585	52	-0.1646	0.3269	0.3345	-0.3667
May	0.0048	50.6944	-0.0662	5.5486	1	5.5486	0.0000		0.0000
Jun	0.0042	46.5278	-0.0165	0.0242	3	0.0081	0.3333	1.3945	-0.0559
Jul	0.0033	54.8611	0.0976	-5.1050	41	-0.1245	0.3415	0.4072	-0.3476
Aug	0.0034	52.7778	-0.0069	-5.9920	28	-0.2140	0.4286	0.4622	-0.6493
Sep	0.0040	61.1111	-0.0308	-4.1198	13	-0.3169	0.3846	0.7393	-0.9106
Oct	0.0025	44.4444	0.2161	-13.1171	72	-0.1822	0.1806	0.3283	-0.2800
Nov	0.0030	38.8889	-0.1000	-4.2443	19	-0.2234	0.2632	0.6515	-0.4967
Dec	0.0031	48.6111	0.1330	0.5472	17	0.0322	0.3529	0.5461	-0.3289

Table C-27: USD/ZAR monthly forecasting performance for the random walk model using 15 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0039	29.1667	-0.0701	-16.3339	80	-0.2042	0.1625	0.4192	-0.3095
Mar	0.0040	26.3889	-0.0985	-17.5257	76	-0.2306	0.1974	0.2881	-0.3480
Apr	0.0043	36.8056	-0.0678	-12.1607	68	-0.1788	0.2059	0.4915	-0.3236
May	0.0065	36.1111	0.0570	-11.3802	71	-0.1603	0.2676	0.6837	-0.4138
Jun	0.0058	35.4167	-0.0117	-12.8451	69	-0.1862	0.2174	0.6819	-0.3782
Jul	0.0049	33.3333	-0.0672	-13.8417	80	-0.1730	0.1875	0.5072	-0.3041
Aug	0.0051	31.2500	-0.1495	-16.3738	75	-0.2183	0.2000	0.4519	-0.3578
Sep	0.0055	36.8056	0.0073	-13.3559	76	-0.1757	0.1974	0.7010	-0.3486
Oct	0.0037	30.5556	-0.0793	-22.8272	90	-0.2536	0.0778	0.4198	-0.2999
Nov	0.0043	31.2500	-0.0575	-17.5357	89	-0.1970	0.1798	0.3899	-0.3030
Dec	0.0045	34.0278	-0.0487	-14.5559	73	-0.1994	0.1507	0.4889	-0.3128

Table C-28: USD/ZAR monthly forecasting performance for the EMD-filtered ANN using 30 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0027	59.7222	0.2691	-5.2133	85	-0.0613	0.3412	0.3145	-0.2263
Mar	0.0024	71.5278	0.4709	-1.2249	76	-0.0161	0.3684	0.3533	-0.2113
Apr	0.0026	70.8333	0.5336	6.6032	71	0.0930	0.5211	0.4068	-0.2098
May	0.0048	57.6389	-0.0162	5.4715	11	0.4974	0.6364	1.2124	-0.5743
Jun	0.0040	63.1944	0.4030	0.8790	81	0.0109	0.3827	0.5768	-0.2716
Jul	0.0032	68.0556	0.4663	5.7650	75	0.0769	0.4933	0.4237	-0.2204
Aug	0.0030	60.4167	0.3375	0.2203	73	0.0030	0.4110	0.4143	-0.2455
Sep	0.0034	57.6389	0.3996	3.9474	65	0.0607	0.5077	0.4760	-0.3125
Oct	0.0024	65.2778	0.3723	-3.8810	77	-0.0504	0.3896	0.2078	-0.2075
Nov	0.0029	61.1111	0.2518	-3.8453	73	-0.0527	0.3973	0.3376	-0.2808
Dec	0.0028	63.1944	0.4289	7.7028	64	0.1204	0.4375	0.5124	-0.1724

Table C-29: USD/ZAR monthly forecasting performance for the unfiltered ANN using 30 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0027	45.1389	0.0864	-10.5001	76	-0.138	0.237	0.345	-0.269
Mar	0.0027	59.0278	0.0299	-2.5067	17	-0.147	0.353	0.538	-0.553
Apr	0.0030	55.5556	0.0800	-2.2889	0	0.000			
May	0.0048	54.8611	0.0022	-3.5899	26	-0.138	0.385	0.956	-0.714
Jun	0.0043	46.5278	-0.0072	-0.2245	5	-0.045	0.400	1.139	-0.915
Jul	0.0035	47.2222	0.0620	-1.0513	11	-0.096	0.455	0.342	-0.380
Aug	0.0032	58.3333	-0.0718	-14.4714	77	-0.188	0.286	0.264	-0.348
Sep	0.0038	59.0278	0.0151	-5.2814	23	-0.230	0.304	0.601	-0.575
Oct	0.0026	42.3611	-0.0570	-5.3063	41	-0.129	0.317	0.341	-0.323
Nov	0.0029	54.8611	-0.0050	-10.8853	54	-0.202	0.222	0.250	-0.320
Dec	0.0031	63.8889	0.1221	-11.8549	82	-0.145	0.317	0.224	-0.286

Table C-30: USD/ZAR monthly forecasting performance for the random walk model using 30 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0039	32.6389	-0.0571	-15.1568	80	-0.1895	0.2000	0.3939	-0.3170
Mar	0.0040	25.0000	-0.1253	-14.6150	76	-0.1923	0.1842	0.3997	-0.3052
Apr	0.0043	39.5833	-0.0738	-11.9075	72	-0.1654	0.2500	0.3676	-0.3161
May	0.0066	36.8056	0.0439	-13.6561	73	-0.1871	0.2740	0.5775	-0.4371
Jun	0.0060	29.8611	-0.0079	-18.9889	77	-0.2466	0.1948	0.5663	-0.3989
Jul	0.0051	39.5833	-0.0484	-12.6498	70	-0.1807	0.2571	0.4783	-0.3731
Aug	0.0048	31.2500	-0.1484	-17.5216	75	-0.2336	0.2133	0.2944	-0.3567
Sep	0.0051	31.9444	0.0682	-10.0349	68	-0.1476	0.2500	0.5462	-0.3518
Oct	0.0038	33.3333	-0.1248	-21.7495	82	-0.2652	0.0976	0.3288	-0.3183
Nov	0.0042	34.0278	-0.0281	-13.2573	73	-0.1816	0.2740	0.3147	-0.3502
Dec	0.0046	34.0278	-0.0818	-16.5698	75	-0.2209	0.1867	0.3742	-0.3465

Table C-31: **USD/ZAR monthly forecasting performance for the EMD-filtered ANN using 60 minute samples with a 180 minute forecast horizon**

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0028	47.9167	0.1961	-6.3600	69	-0.0922	0.3478	0.3346	-0.2813
Mar	0.0024	64.5833	0.2263	-11.4019	83	-0.1374	0.3012	0.1884	-0.2682
Apr	0.0029	59.7222	0.2451	-2.1817	83	-0.0263	0.4337	0.3094	-0.2516
May	0.0049	56.2500	0.2443	-5.1245	93	-0.0551	0.3656	0.6312	-0.3659
Jun	0.0042	51.3889	0.2886	-5.9939	79	-0.0759	0.3924	0.4383	-0.3402
Jul	0.0033	68.0556	0.3400	-4.3809	77	-0.0569	0.4416	0.3221	-0.3159
Aug	0.0033	53.4722	0.1249	0.1655	59	0.0028	0.4915	0.4432	-0.3533
Sep	0.0037	59.7222	0.0679	-5.9016	45	-0.1311	0.3778	0.4248	-0.4037
Oct	0.0026	48.6111	0.1581	-8.1039	68	-0.1192	0.2941	0.2532	-0.2622
Nov	0.0026	55.5556	0.1640	-10.0093	78	-0.1283	0.2564	0.2604	-0.2488
Dec	0.0030	60.4167	0.1397	-3.7838	74	-0.0511	0.4054	0.3225	-0.2851

Table C-32: USD/ZAR monthly forecasting performance for the unfiltered ANN using 60 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0029	50.6944	0.0360	-13.1626	70	-0.1880	0.2000	0.3437	-0.3091
Mar	0.0025	65.9722	0.1255	-0.7725	7	-0.1104	0.5714	0.7686	-1.3523
Apr	0.0029	53.4722	-0.0278	-14.9928	71	-0.2112	0.2817	0.1831	-0.3499
May	0.0051	50.0000	-0.0034	-3.9284	75	-0.0524	0.3867	0.6690	-0.4197
Jun	0.0043	50.0000	-0.0585	-1.8540	7	-0.2649	0.4286	0.7154	-0.6193
Jul	0.0035	52.7778	0.0159	-12.1266	64	-0.1895	0.2813	0.4093	-0.3891
Aug	0.0033	43.0556	-0.1338	-15.5924	56	-0.2784	0.1964	0.5078	-0.4315
Sep	0.0037	48.6111	0.1880	1.5481	0	0.0000			
Oct	0.0026	34.0278	-0.0373	-2.0247	25	-0.0810	0.2800	0.3757	-0.2803
Nov	0.0027	47.2222	-0.0143	-13.9339	69	-0.2019	0.1884	0.2989	-0.3009
Dec	0.0031	63.1944	-0.0529	-7.7007	34	-0.2265	0.3529	0.2541	-0.4391

Table C-33: USD/ZAR monthly forecasting performance for the random walk model using 60 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0043	30.5556	-0.1973	-17.3277	76	-0.2280	0.1711	0.3466	-0.3341
Mar	0.0035	28.4722	-0.0035	-15.7274	74	-0.2125	0.1892	0.3376	-0.3175
Apr	0.0042	36.8056	-0.0153	-12.6451	70	-0.1806	0.2714	0.2947	-0.3345
May	0.0070	35.4167	0.0417	-16.7878	79	-0.2125	0.2278	0.6191	-0.4085
Jun	0.0062	31.2500	-0.0561	-18.7915	72	-0.2610	0.1528	0.7230	-0.4172
Jul	0.0051	35.4167	-0.0754	-12.7651	71	-0.1798	0.1831	0.6117	-0.3257
Aug	0.0048	29.8611	-0.0637	-19.3680	87	-0.2226	0.1724	0.4449	-0.3331
Sep	0.0051	34.0278	0.0407	-16.1362	81	-0.1992	0.1852	0.5423	-0.3393
Oct	0.0038	34.7222	-0.0651	-18.7020	80	-0.2338	0.1250	0.3389	-0.3034
Nov	0.0035	38.1944	0.0783	-12.8048	75	-0.1707	0.2133	0.3490	-0.2901
Dec	0.0046	29.8611	-0.1145	-15.3494	77	-0.1993	0.1558	0.4694	-0.3123

Table C-34: USD/ZAR monthly forecasting performance for the EMD-filtered ANN using 90 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0028	46.5278	0.1030	-10.8324	76	-0.1425	0.2368	0.3428	-0.2789
Mar	0.0025	53.4722	0.1582	-2.6597	7	-0.3800	0.0000		-0.4987
Apr	0.0029	62.5000	0.0859	-3.1408	14	-0.2243	0.4286	0.6562	-0.7226
May	0.0050	55.5556	0.1651	4.8372	63	0.0768	0.5238	0.6473	-0.4702
Jun	0.0044	65.2778	0.0182	-1.7254	13	-0.1327	0.3077	1.0693	-0.5861
Jul	0.0034	63.1944	0.3207	-4.7198	67	-0.0704	0.3582	0.4419	-0.3168
Aug	0.0032	50.6944	0.1554	-4.7619	72	-0.0661	0.4028	0.3372	-0.2918
Sep	0.0038	61.1111	-0.1279	-1.9064	7	-0.2723	0.2857	0.4897	-0.4886
Oct	0.0027	68.0556	0.0844	-6.8950	75	-0.0919	0.2533	0.2597	-0.1969
Nov	0.0026	59.0278	0.0476	-14.4512	78	-0.1853	0.2564	0.1943	-0.3038
Dec	0.0031	39.5833	0.0128	-3.1303	0	0.0000			

Table C-35: USD/ZAR monthly forecasting performance for the unfiltered ANN using 90 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0028	52.0833	0.0196	-13.9477	75	-0.1860	0.2800	0.2303	-0.3227
Mar	0.0025	56.2500	0.0040	-1.1346	1	-1.1346	0.0000		0.0000
Apr	0.0029	62.5000	0.0216	-10.4661	40	-0.2617	0.3000	0.2867	-0.4684
May	0.0049	59.7222	0.2151	-1.0533	55	-0.0192	0.4727	0.6224	-0.5323
Jun	0.0044	69.4444	0.0289	-0.3435	4	-0.0859	0.5000	0.2750	-0.6066
Jul	0.0035	40.9722	-0.0144	-5.5174	13	-0.4244	0.2308	0.7867	-0.7305
Aug	0.0034	55.5556	-0.0191	-8.8542	46	-0.1925	0.4130	0.3196	-0.4956
Sep	0.0041	49.3056	0.0699	-11.6530	66	-0.1766	0.3333	0.3513	-0.4054
Oct	0.0027	38.1944	-0.0831	-7.8100	20	-0.3905	0.2000	0.3304	-0.5540
Nov	0.0026	41.6667	0.0390	-7.9678	43	-0.1853	0.2093	0.4321	-0.3333
Dec	0.0031	71.5278	0.0804	-13.1251	72	-0.1823	0.3056	0.1490	-0.3158

Table C-36: USD/ZAR monthly forecasting performance for the random walk model using 90 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0043	30.5556	-0.1973	-17.3277	76	-0.2280	0.1711	0.3466	-0.3341
Mar	0.0035	28.4722	-0.0035	-15.7274	74	-0.2125	0.1892	0.3376	-0.3175
Apr	0.0042	36.8056	-0.0153	-12.6451	70	-0.1806	0.2714	0.2947	-0.3345
May	0.0070	35.4167	0.0417	-16.7878	79	-0.2125	0.2278	0.6191	-0.4085
Jun	0.0062	31.2500	-0.0561	-18.7915	72	-0.2610	0.1528	0.7230	-0.4172
Jul	0.0051	35.4167	-0.0754	-12.7651	71	-0.1798	0.1831	0.6117	-0.3257
Aug	0.0048	29.8611	-0.0637	-19.3680	87	-0.2226	0.1724	0.4449	-0.3331
Sep	0.0051	34.0278	0.0407	-16.1362	81	-0.1992	0.1852	0.5423	-0.3393
Oct	0.0038	34.7222	-0.0651	-18.7020	80	-0.2338	0.1250	0.3389	-0.3034
Nov	0.0035	38.1944	0.0783	-12.8048	75	-0.1707	0.2133	0.3490	-0.2901
Dec	0.0046	29.8611	-0.1145	-15.3494	77	-0.1993	0.1558	0.4694	-0.3123

Table C-37: USD/ZAR monthly forecasting performance for the EMD-filtered ANN using 180 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0028	51.7483	-0.0719	-12.8117	63	-0.2034	0.2698	0.1987	-0.3316
Mar	0.0025	39.8601	-0.0765	-1.1910	1	-1.1910	0.0000		0.0000
Apr	0.0031	37.7622	-0.1009	-9.7977	36	-0.2722	0.2500	0.4175	-0.4562
May	0.0107	53.1469	-0.0307	-2.1181	31	-0.0683	0.5484	0.5948	-0.6882
Jun	0.0043	55.2448	0.0199	-2.1089	23	-0.0917	0.3478	0.7886	-0.4965
Jul	0.0035	39.1608	-0.0213	-9.5949	27	-0.3554	0.1481	0.3774	-0.4712
Aug	0.0041	56.6434	0.0126	-8.8903	14	-0.6350	0.2143	0.4217	-0.8769
Sep	0.0037	41.9580	0.0917	-5.6921	57	-0.0999	0.3860	0.4396	-0.3660
Oct	0.0027	44.7552	0.0309	-2.8561	19	-0.1503	0.2105	0.5721	-0.3867
Nov	0.0026	57.3427	-0.1567	-10.2672	44	-0.2333	0.2955	0.1311	-0.3907
Dec	0.0031	41.2587	0.0010	-4.0877	8	-0.5110	0.2500	0.0721	-0.4148

Table C-38: USD/ZAR monthly forecasting performance for the unfiltered ANN using 180 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0032	36.3636	-0.1196	-20.2516	87	-0.2328	0.1724	0.1556	-0.3013
Mar	0.0026	72.7273	-0.0324	-7.8233	39	-0.2006	0.3846	0.1467	-0.4172
Apr	0.0030	59.4406	-0.0577	-11.1529	54	-0.2065	0.3889	0.1993	-0.4300
May	0.0052	63.6364	-0.0253	7.3540	20	0.3677	0.5500	1.1978	-0.5702
Jun	0.0045	60.1399	0.0368	0.8526	10	0.0853	0.4000	1.1465	-0.9079
Jul	0.0035	60.1399	-0.0142	-16.2010	85	-0.1906	0.3059	0.1684	-0.3319
Aug	0.0034	53.8462	0.0979	-14.6954	59	-0.2491	0.2881	0.1847	-0.4124
Sep	0.0037	57.3427	0.0177	-4.1599	17	-0.2447	0.5294	0.6469	-1.0584
Oct	0.0027	51.7483	0.0875	-13.1783	72	-0.1830	0.2361	0.2127	-0.2804
Nov	0.0026	51.0490	-0.0455	-10.0117	61	-0.1641	0.2623	0.2859	-0.3163
Dec	0.0031	71.3287	0.0425	-8.2416	62	-0.1329	0.3871	0.1807	-0.3146

Table C-39: USD/ZAR monthly forecasting performance for the random walk model using 180 minute samples with a 180 minute forecast horizon

	RMSE	DS (%)	Corr (%)	Ret (%)	Number of transactions	Mean return per transaction	Fraction of profitable trades	Mean returns per profitable trade	Mean return per losing trade
Feb	0.0043	30.0699	-0.1972	-16.9945	74	-0.2297	0.1757	0.3645	-0.3424
Mar	0.0035	28.6713	-0.0075	-15.7755	74	-0.2132	0.1892	0.3333	-0.3175
Apr	0.0042	36.3636	-0.0137	-12.2539	68	-0.1802	0.2941	0.2995	-0.3490
May	0.0070	34.9650	0.0419	-16.7029	79	-0.2114	0.2278	0.6191	-0.4060
Jun	0.0062	31.4685	-0.0547	-18.6255	72	-0.2587	0.1528	0.7230	-0.4122
Jul	0.0051	34.9650	-0.0795	-12.9875	71	-0.1829	0.1831	0.5764	-0.3257
Aug	0.0048	30.0699	-0.0695	-19.7553	87	-0.2271	0.1724	0.3954	-0.3331
Sep	0.0051	34.2657	0.0436	-16.0118	81	-0.1977	0.1852	0.5423	-0.3358
Oct	0.0038	34.2657	-0.0663	-18.7361	80	-0.2342	0.1250	0.3389	-0.3068
Nov	0.0035	38.4615	0.0781	-13.3953	75	-0.1786	0.2133	0.3043	-0.2901
Dec	0.0046	30.0699	-0.1140	-15.2978	77	-0.1987	0.1558	0.4694	-0.3108