Evaluation and Application of Higher Order Neural Networks in Financial Forecasting, Value at Risk and Option Pricing

Georgios Sermpinis

Ph.D

2009

Evaluation and Application of Higher Order Neural Networks in Financial Forecasting, Value at Risk and Option Pricing

Georgios Sermpinis

Liverpool Business School

A thesis submitted in partial fulfillment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

Declaration

I declare that with the exception of the assistance acknowledged, this dissertation is the result of an original investigation and that it has not been accepted or currently submitted in candidate for any other degree.

Candidate

.....(signed)

Supervisor

.....(signed)

Committee

Supervisory Team

Prof. Christian Dunis (Director of Studies)

Jason Laws (1st Supervisor)

Examination Team

Dr Neil Kellard (External)
(University of Essex, Essex Business School)

Dr Gianluigi Giorgioni (Internal)
(Liverpool John Moores University, Liverpool Business School)

Acknowledgments

I would like to take this opportunity to thank first and foremost, my Director of Studies; Professor Christian Dunis, whose help and guidance have been invaluable, over the past 2 years he has become a good friend.

I would also like to thank my Supervisor and long-time mentor Jason Laws for his help and guidance, especially in giving me the opportunity to start on this road.

Finally, I would like to thank my colleagues and friends Binam Ghimire and Andreas Karathanasopoulos for their moral support over the past years.

Ἓν οἶδα ὅτι οὐδὲν οἶδα
(***Απολογία***)

I know nothing except the fact of my ignorance
(***Apology***)

Socrates (469 BC–399 BC)

Abstract

The aim of this thesis is to provide empirical evidence on the utility of Higher Order Neural Networks (HONNs) as financial forecasting and trading tools. In order to achieve this, we use HONNs in a series of applications and benchmark them not only with some traditional statistical and technical techniques but also with some other state of the art Neural Networks (NNs) designs. Moreover, we test the stability and the robustness of their performance, a crucial property for models like HONNs, whose their modelling is based on trial and error rather than some formal statistical theory.

The evidence shows that, HONNs perform similarly or outperform their NNs and statistical/technical benchmarks as forecasting and trading tools on the EUR/USD exchange rate (see chapters 4 and 5) although they do not seem capable of exploiting the trading strategies applied as for example the GM networks (see chapter 4). Their superiority is more obvious when we feed our NNs models with autoregressive terms rather than having as inputs multivariate series. Also we test and find our forecasts stable and robust through time (see chapter 6). Moreover, HONNs seem capable of providing accurate forecasts of the realised volatility of the gold bullion, brent oil and FTSE 100 futures index (see chapters 7 and 8). This allows us to apply and exploit our forecasts successfully in a Value at Risk and Option Pricing Modelling context.

The above mentioned empirical evidence confirms that HONNs can provide accurate, profitable and robust forecasts. Our results should go some way towards convincing quantitative risk and fund managers to use non-linear alternative models like HONNs as they seem capable of outperforming the

classical statistical/technical algorithms of their toolbox and generate higher return/risk profiles.

Table of Contents

Chapter 5
Modelling and Trading the EUR/USD Exchange Rate at the ECB Fixing

Chapter 6
The Robustness of Neural Networks for Modelling and Trading the EUR/USD Exchange Rate at the ECB Fixing

List of Tables

Chapter 4

Chapter 5

Chapter 6

Chapter 7

Chapter 8

Appendix

List of Figures

Appendix

CHAPTER 1

1.1 Introduction

The development of accurate forecasting techniques is critical to economists, investors and risk analysts. This task is getting more complex as financial markets are getting increasingly interconnected and interdependent. The traditional statistical techniques, on which market forecasters were relying in previous years, seem to fail to capture the moving interrelationship among market variables. This context has led to a continuous search of techniques capable of identifying and capturing the nonlinearities, the discontinuities and the high frequency multi-polynomial components characterizing financial time series today. A class of such techniques that have provided promising results in previous years are Neural Networks.

Artificial neural networks (NNs), which were firstly introduced by McCulloch and Pitts (1943), are mathematical models inspired by the organization and functioning of biological neurons. In the beginning NNs were seen as a way to model the human brain and expectations from its applications were high. However the expectations were not met until the 80's, as the lack of the necessary computing power put limitations on the research. Then, with the rapid growth of computer science and the works of scientists like Hopfield (1982) the interest in NNs was renewed and huge theoretical steps started to be made. Today NNs are applied in almost every aspect of Science including financial forecasting.

Among the numerous advantages of NNs compared to traditional statistical linear techniques is the fact that they are inherently nonlinear, self adaptive data driven (they require few a priori assumptions) and that they can approximate any continuous function to any desired level of accuracy (Hornik *et al.* (1989)). On the other hand, some issues such as their predictive unreliability with outcomes that are overly sensitive to specific training samples, the malicious vector and their absence of formal theoretical rules for the training procedures has created scepticism over their utility as forecasting tools. A family of NNs that seems to overcome the problem of sensitivity to the trainning samples and malicious vector but not the absence of formal theoretical rules is Higher Order Neural Networks (HONNs). HONNs can better approximate complex, non-smooth, often discontinuous training data compared to the classic Multi-Layer Perceptron models (MLPs) (Fulcher *et. al.* (2006)). Moreover, they are capable of extracting higher order coefficients in the data and there is a one to one correspondence between the polynomial coefficients and the network weights. Therefore they can be considered as open box solutions, a much desirable property in financial applications. This thesis studies the forecasting and trading performance of HONNs having as benchmark a wide variety of NNs, statistical and technical models.

This thesis should be of interest to both hedgers and speculators who want to explore the use of alternative non linear models. An accurate prediction of the future stock market pattern will give them a considerable advantage and allow them to generate attractive return/risk profiles. Moreover, the ability to forecast accurately the Value at Risk (VaR) will allow hedge funds and

investors to develop clever and effective hedging strategies while an accurate forecast of the future volatility can help investors to identify mispriced options and develop sophisticated trading strategies. Morevover, this thesis can contribute to the academic studies as it provides empirical evidents over the forecasting and trading abilities of a wide variety of non linear models over the mean of the EUR/USD exchange rate and the volatility of the gold bullion, brent oil and the FTSE 100 futures index. Also all the forecasts were evaluated through financial and trading criteria which makes it differ from most similar academic studies. Furhhermore, this thesis contribute to financial research by introducing a backtesting algorithm to evaluate the Value at Risk (see section 8.4.2.2).

1.2 Background to the Thesis and Motivation

HONNs were firstly introduced by introduced Giles and Maxwell (1987) as a fast learning network with increased learning capabilities. Although their function approximation superiority over the more traditional architectures is well presented in a series of articles (see among others Redding *et al.* (1993), Kosmatopoulos *et al.* (1995) and Psaltis *et al.* (1998)) their use in finance was limited until recently. This changed when scientists started to investigate not only the benefits of NNs against the traditional statistical techniques but also the differences between the different NNs models architectures. Then a wide variety of articles over their practical applications (for example Zhang *et al.* (2000), Dunis *et al.* (2005) and Fulcher *et al.* (2006)) verified their above mentioned advantages by demonstrating their superior forecasting ability and put HONNs in the front line of research in financial forecasting. However,

previous research stopped in the context of mean forecasting and there is no empirical evidence over their usage in demanding areas such as volatility forecasting in an Option Pricing and VaR context.

The motivation of this thesis is to fill this hole in the literature and to provide empirical evidence of the utility of HONNs in financial forecasting and trading applications. In order to achieve this, we benchmark HONNs not only with some traditional statistical and technical techniques but also with some other state-of-the-art NNs designs. Therefore, we will be able to validate if the theoretical advantages of HONNs compared to the more traditional NNs models are translated in more accurate/profitable forecasts. In order to achieve this our forecasts are evaluated through financial terms while in the literature most applications evaluate their financial forecasts only through statistical means. Moreover, we will explore the utility of HONNs if we feed them not only with multivariate but also with autoregressive series as inputs. Therefore we will be able to draw more solid conclusions on the forecasting ability of our models especially against our statistical autoregressive benchmarks as HONNs now will not have any additional knowledge compared to them. Furhtermore, we will examine the robstuness of the forecsting perfomance of HONNs and our other NNs benchmarks while in the literature these feature has not been studied. Moreover, this research aims to provide the first empircal evidents over the forecasting power of HONNs in an Option Pricing and VaR context something that will further dinstiguish our research from previous similar studies and add originality to our application.

1.3 Contribution to the Knowledge

In this dissertation we test and evaluate the forecasting and trading ability of HONNs. We explore the utility and the robustness of their performance in forecasting the mean and the volatility of financial series in an Option Price Modelling and Value at Risk context. More specifically the contributions to knowledge of this dissertation are threefold:

1) Evaluating the forecasting and trading performance of HONNs.

In chapter 4 we test and evaluate the performance of HONNs in forecasting the EUR/USD exchange rate using as inputs multivariate series while in chapter 5 we repeat the same application after we feed HONNs and the benchmark NNs models with autoregressive series. In order to further improve the trading performance of our models we apply trading strategies using confirmation filters and leverage. In chapter 6 we examine the robustness and the stability of the forecasting and trading performance of our models. The previous mentioned applications will allow us to argue with confidence over the forecasting power of HONNs in predicting the return of the EUR/USD exchange rate whether multivariate or autoregressive series are used as inputs. Moreover, we will examine if the mentioned in the literature unpredictability in the NNs forecasts is still the case in HONNs.

2) Evaluating the performance of HONNs in an Option Price Modeling context.

In chapter 7 we forecast the 1-month (21 trading days) ahead volatility of the FTSE 100 futures index with HONNs and 3 other benchmark models. Then

we use our forecasts to indentify mispriced options and to exploit them in a simple trading application. Therefore, we will be able to examine and to provide the first empirical evidents around the utility of HONNs in generating profit through accurate volatility forecasts in an Option Price Modeling context.

3) Evaluating the performance of HONNs in a VaR context.

In chapter 8 we forecast the VaR of brent oil and gold bullion with HONNs and 3 other NN and technical models. Then we evaluate our forecasts with a series of algorithms and backtesting functions, including an average squared magnitude function introduced for the first time in this thesis. Therefore we will contribute to the knowledge by providing the first empirical evidents around the forecasting power of HONNs in a VaR context. Morever, the unique backtesting procedure followed to chapter 8, will add originality in this thesis.

CHAPTER 2

Literature Review

In this charter we present the literature relevent to the NNs models used on this thesis and the applications of NNs in general, in financial forecasting and in an Option Price Modelling and VaR context.

The most popular and well researched NN architecture is the MLPs. In Finance their use is widespread and their forecasting superiority against most linear statistical and technical models, well acknowledged. Yao *et al.* (1996) forecasts the GBP, DEM, JPY, CHF, and AUD against the USD exchange rates, from 1984 to 1995 with a MLP and a ARMA model. Their conclusions were in favour of the MLPs not only in terms of correct directional change (CDC) but also in terms of profitability. Similarly, Yao *et al.* (1997) forecasts the USD/CHF exchange rate from 1983 to 1995 having this time as benchmarks a 'buy and hold' and a 'trend following' strategy. Once more the MLPs perform better and produce more accurate and profitable forecasts. Similarly, Dunis and Williams (2002) forecast the EUR/USD exchange rate from October 1994 to July 2001 with a naïve, a moving average convergence-divergence (MACD), a ARMA, a logit and a MLP model. In their evaluation the MLP outperforms all other strategies not only in statistical terms but also in terms of trading efficiency as it produces the higher annualised return. Moreover, Pan *et. al.* (2003) in a forecasting application, tries to exploit the various dynamical swings and inter-market influences of the Australian stock marker index with MLP. Their model forecasts exhibits up to 80% directional prediction correctness of the Australian stock market returns. Furthermore,

Zhang *et al.* (2005) forecasts with a MLP the buy and sell signs of the Shanghai Composite Index for a seven years period. Their model generates 3 times higher return than a naïve buy and hold strategy. On the other hand, Zhu *et al.* (2008) forecasts accurately the NASDAQ, DJIA and STI stock market indexes with 3-layer MLPs for a 15 years period.

Recurrent Neural networks (RNNs) have an activation feedback which embodies short-term memory allowing them to learn extremely complex temporal patterns. Their superiority against feedfoward networks when performing nonlinear time series prediction is well documented in Connor *et al.* (1993) and Adam *et al.* (1994). In financial applications, Kamijo *et al.* (1990) applied them successfully to the recognition of stock patterns of the Tokyo stock exchange while Tenti (1996) achieved remarkable results using RNNs to forecast the exchange rate of the Deutsche Mark. Tino *et al.* (2001) use them to trade successfully the volatility of the DAX and the FTSE 100 using straddles while Dunis and Huang (2002), using continuous implied volatility data from the currency options market, obtain remarkable results for their GBP/USD and USD/JPY exchange rate volatility trading simulation. Moreover, Versace *et. al.* (2004) combines RNNs with genetic algorithms (GA) and made statistically accurate forecasts of the DIA exchange traded fund from November 2001 to February 2003. Following a similar approach Kim and Shin (2007) use a hybrid Recurrent-GA neural network to make accurate predictions of the Korea Stock Price Index 200 pattern from 1997 to 1999. On the other hand, Lee (2004) introduces a hybrid Recurrent-Radial Basis network which shows promising results in forecasting and trading 33 major Hong-Kong stocks from 1990 to 1999.

HONNs were first introduced by introduced by Giles and Maxwell (1987) as a fast learning network with increased learning capabilities. Practical applications have verified the theoretical advantages of HONNs by demonstrating their superior forecasting ability and put them in the front line of research in financial forecasting. For example Knowles *et. al.* (2005) forecasts with HONNs and MLPs the EUR/USD exchange rate from October 1994 to July 2001. In their trading application HONNs achieve 8% higher annualised return than MLPs. Dunis *et al.* (2006b) use them to forecast successfully the gasoline crack spread while Fulcher *et al.* (2006) apply HONNs to forecast the AUD/USD exchange rate, achieving a 90% accuracy. However, Dunis *et al.* (2006a) show that, in the case of the futures spreads and for the period under review, the MLPs performed better compared with HONNs and recurrent neural networks (RNNs). On the other hand, Dunis and Nathani (2007) forecast and trade the gold and the silver daily returns with HONNs, MLPs, a Nearest Neighbours and a linear ARMA model. In their trading application, HONNs produce the most profitable trades in terms of annualized return and information ratio.

Psi Sigma networks were first introduced as an architecture capable of capturing higher order correlations within the data while avoiding some of the HONNs limitations such as the combinatorial increase in weight numbers. Shin and Ghosh (1991) and Ghosh and Shin (1992) demonstrate these benefits and present empirical evidence on their forecasting ability. For financial applications, Ghazali *et al.* (2006) compare them with HONNs and MLPs on the IBM common stock closing price and the US 10-year

government bond series and prove their forecasting superiority. In a similar paper, Hussain *et al.* (2006) present satisfactory results of the Psi Sigma forecasting power on the EUR/USD, the EUR/GBP and the EUR/JPY exchange rates using univariate series as inputs in their networks.

In the field of exploiting NNs forecasts in an option pricing model context, Hutchinson *et al.* (1994) use NNs to successfully price the S&P 500 futures options. Malliaris and Salchenberger (1996) forecast accurately the Black-Scholes derived implied volatility of the S&P 100 at-the-money call options with a MLP model while Garcia and Gencay (2000) create an option pricing model with feedfoward networks which is providing smaller delta-hedging errors relative to the ones generate from the Black-Scholes model. Yao *et. al.* (2000) forecasts the option prices of the Nikkei 225 index futures with backpropagation NNs. They conclude that although for volatile markets NNs outperform the BS model, the BS model is still good for pricing at-the-money options. Moreover, Meissner and Kawano (2001) use Garch volatility forecasts as inputs to four different NNs models and create option pricing models which present significant better pricing performance than the Black-Scholes model. Furthermore, Gencay and Altay-Salih (2003) prove that for deep out-of-the money options, feedfoward NNs present a substantially better pricing performance than the BS model. On the other hand, Hamid (2004) forecast the volatility of the S&P 500 futures index using a MLP model. He finds that his volatility forecasts are not statistically different from the realised volatility and more accurate than the implied volatility, generated by the Barone-Adesi and Whaley (1987) model, of the S&P 500 index futures

options. Furthermore, Pires and Marwala (2004) forecast successfully the prices of American style call options on the JSE Securities Exchange of South Africa with Bayesian NNs while Andreou *et al.* (2008) forecast with good accuracy the price of European call options on the S&P 500 by combining a MLP model with the Black-Sholes and the Corrado-Su (1996) models. Moreover, Gardovejic *et. al.* (2009) based on the option pricing models of Hutchinson *et al.* (1994) and Garcia and Gencay (2000) creates a modular NN to price the S&P-500 European call options January 1987 to October 1994. Their model is more accurate than the BS model in all cases except in 1987.

In the field of Risk Management, Locarek-Junge and Prinzler (1998) estimate the VaR of a US dollar portfolio using a Mixture Density Network while Bartlmae and Rauscher (2000) using a Neural Network Volatility Mixture model forecast successfully the one day ahead VaR of the German Stock index. Taylor (2000) found NNs as useful alternatives to GARCH for estimating the conditional density of exchange rate returns. Neely and Weller (2002) argue in favour of the use of genetic programming as an alternative to GARCH and RiskMetrics models while Cornalba and Giudici (2004) argue in favour the theoretical advantages of Bayesian NNs in estimating the VaR. Dunis and Chen (2005) demonstrate that NNs Regression models are superior in forecasting the VaR of the EUR/USD exchange rate compared to GARCH and Stochastic Variance models. Furthermore, Liu (2005) by combining historical simulation and a GARCH (1,1) model with NNs, achieve accurate VaR estimates for the S&P 500 and the DJI indexes and the Ford

and IBM stocks. Similarly, Ozun and Cifter (2007) combine various GARCH, historical simulation and Extreme Value Theory models with Neural Networks to provide accurate estimates of the VaR of the Istanbul Stock Exchange.

The aim of the first three applications of this thesis is to provide empirical evidents around the utility of HONNs in forecasting the mean of financial series with autoregressive and multivariate series. This feature will distinguish our research from the previous mentioned papers which fail to examine the robstuness of their models. Moreover, all the applications in NNs around forecasting the volatility in an option pricing model context, evaluate their forecasts with statistical means. In our research we evaluate our forecasts also with financial means and thus providing more solid conclusions around the financial utility of our models. Furhtermore, in the last chapter of our thesis we provide empirical evidents around the forecasting ability of several non linear models in forecasting the one day ahead VaR. In order to evaluate our forecasts we follow an unique methodology using the Christofferesen tests and two different loss functions. On other hand, similar papers in the literature stop the evaluation of their models in the Christoffersen tests.

CHAPTER 3

Neural Network Modelling

The primary forecasting methodologies used on this thesis are that of neural networks. They are used as decision models to forecast the return of the EUR/USD exchange rate and the variance of Gold Bullion, Brent Oil and the FTSE 100 futures index. NNs can take on several different types of architecture and because of this the 4 different neural network designs that are used in this thesis are explained in the following section. We also present the 2 NNs models of Lindemann *et. al.* (2004), Gaussian Mixture and Softmax Cross Entropy, whose performance we use as benchmarks in the fourth chapter.

3.1 The Multi-Layer Perceptron

A standard MLP has at least three layers. The first layer is called the input layer (the number of its nodes corresponds to the number of explanatory variables). The last layer is called the output layer (the number of its nodes corresponds to the number of response variables). An intermediary layer of nodes, the hidden layer, separates the input from the output layer. Its number of nodes defines the amount of complexity the model is capable of fitting. In addition, the input and hidden layer contain an extra node, called the bias node. This node has a fixed value of one and has the same function as the intercept in traditional regression models. Normally, each node of one layer has connections to all the other nodes of the next layer.

The network processes information as follows: the input nodes contain the value of the explanatory variables. Since each node connection represents a weight factor, the information reaches a single hidden layer node as the weighted sum of its inputs. Each node of the hidden layer passes the information through a nonlinear activation function and passes it on to the output layer if the calculated value is above a threshold.

The network architecture of a 'standard' Multi-Layer Perceptron looks as presented in figure 1:

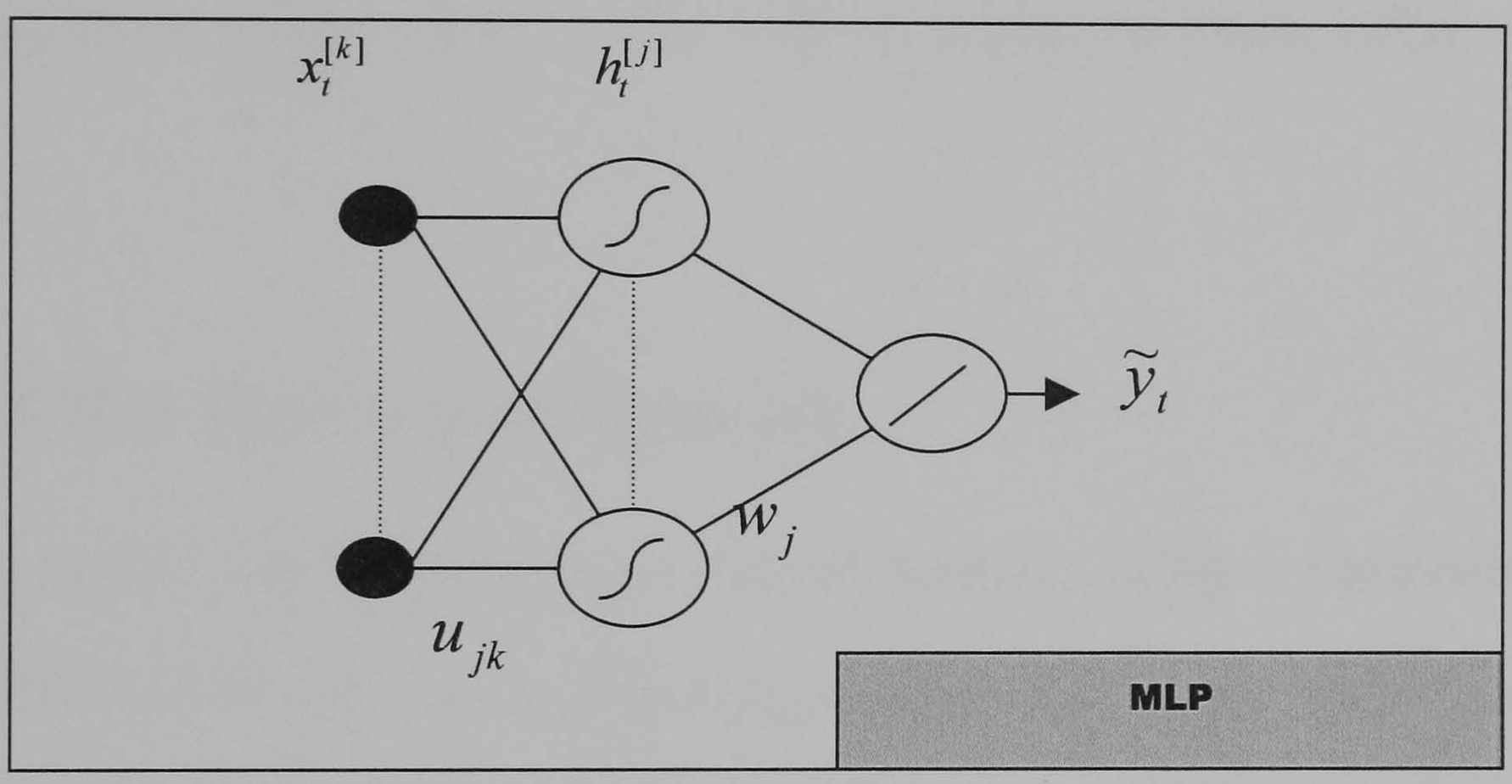


Fig. 1: A single output, fully connected MLP model

where:

$x_t^{[n]}$ $(n = 1,2,\cdots,k+1)$ are the model inputs (including the input bias node) at time t

$h_t^{[m]}$ $(m = 1,2,\ldots, j+1)$ are the hidden nodes outputs (including the hidden bias node)

$\tilde{y}_t$ is the MLP model output

u_{jk} and w_j are the network weights

$\int$ is the transfer sigmoid function: $S(x) = \frac{1}{1+e^{-x}}$, [1]

$/$ is a linear function: $F(x) = \sum_i x_i$ [2]

The error function to be minimised is:

$$E(u_{jk}, w_j) = \frac{1}{T}\sum_{t=1}^{T}(y_t - \tilde{y}_t(u_{jk}, w_j))^2 \text{ , with } y_t \text{ being the target value} \quad [3]$$

3.2 The Recurrent Network

Our next model is the recurrent neural network. While a complete explanation of RNN models is beyond the scope of this thesis, we present below a brief explanation of the significant differences between RNN and MLP architectures. For an exact specification of the recurrent network, see Elman (1990).

A simple recurrent network has activation feedback, which embodies short-term memory. In other words,a recurrent network uses the output of the hidden nodes of period t-1 as inputs to period t. The advantages of using recurrent networks over feedforward networks, for modelling non-linear time series, has been well documented in the past (see among others Elman (1990) and Tenti (1996)). However as described in Tenti (1996) “the main

disadvantage of RNNs is that they require substantially more connections, and more memory in simulation, than standard backpropagation networks" (p.569), thus resulting in a substantial increase in computational time. However having said this RNNs can yield better results in comparison to simple MLPs due to the additional memory inputs.

3.2.1 The RNN Architecture

A simple illustration of the architecture of an Elman RNN is presented below.

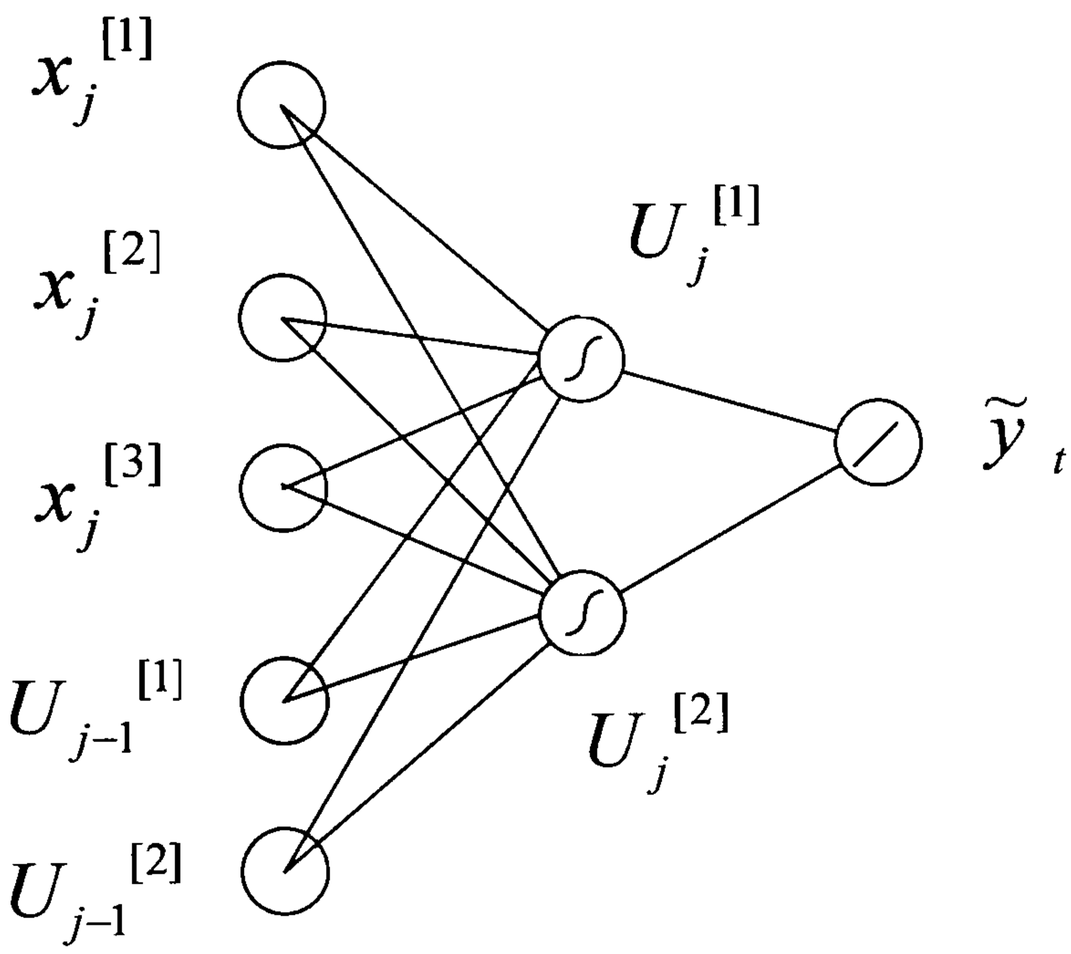


Fig. 2: Elman Recurrent neural network architecture with two nodes on the hidden layer.

where:

$x_t^{[n]}\ (n=1,2,\cdots,k+1),\ u_t^{[1]}, u_t^{[2]}$ are the model inputs (including the input bias node) at time t

$\tilde{y}_t$ is the recurrent model output

$d_t^{[f]}$ $(f=1,2)$ and $w_t^{[n]}$ $(n=1,2,\cdots,k+1)$ are the network weights

$U_t^{[f]}$ $(f=1,2)$ is the output of the hidden nodes at time t

Ⓢ is the transfer sigmoid function: $S(x)=\frac{1}{1+e^{-x}}$, [4]

Ⓛ is the linear output function: $F(x)=\sum_i x_i$ [5]

The error function to be minimised is:

$$E(d_t,w_t)=\frac{1}{T}\sum_{t=1}^{T}(y_t-\tilde{y}_t(d_t,w_t))^2 \quad [6]$$

In short, the RNN architecture can provide more accurate outputs because the inputs are potentially taken from all previous values (see inputs $U_{j-1}^{[1]}$ and $U_{j-1}^{[2]}$ in the figure above).

3.3 Higher Order Neural Networks

Higher Order Neural Networks (HONNs) were first introduced by Giles and Maxwell (1987) and were called "Tensor Networks". For Zhang *et al.* (2002), a significant advantage of HONNs is that "HONN models are able to provide some rationale for the simulations they produce and thus can be regarded as "open box" rather then "black box". Moreover, HONNs are able to simulate

higher frequency, higher order non-linear data, and consequently provide superior simulations compared to those produced by ANNs (Artificial Neural Networks)" (p. 188).

3.3.1 The HONNs Architecture

While they have already experienced some success in the field of pattern recognition and associative recall[1], HONNs have not yet been widely used in finance. The architecture of a three input second order HONN is shown below:

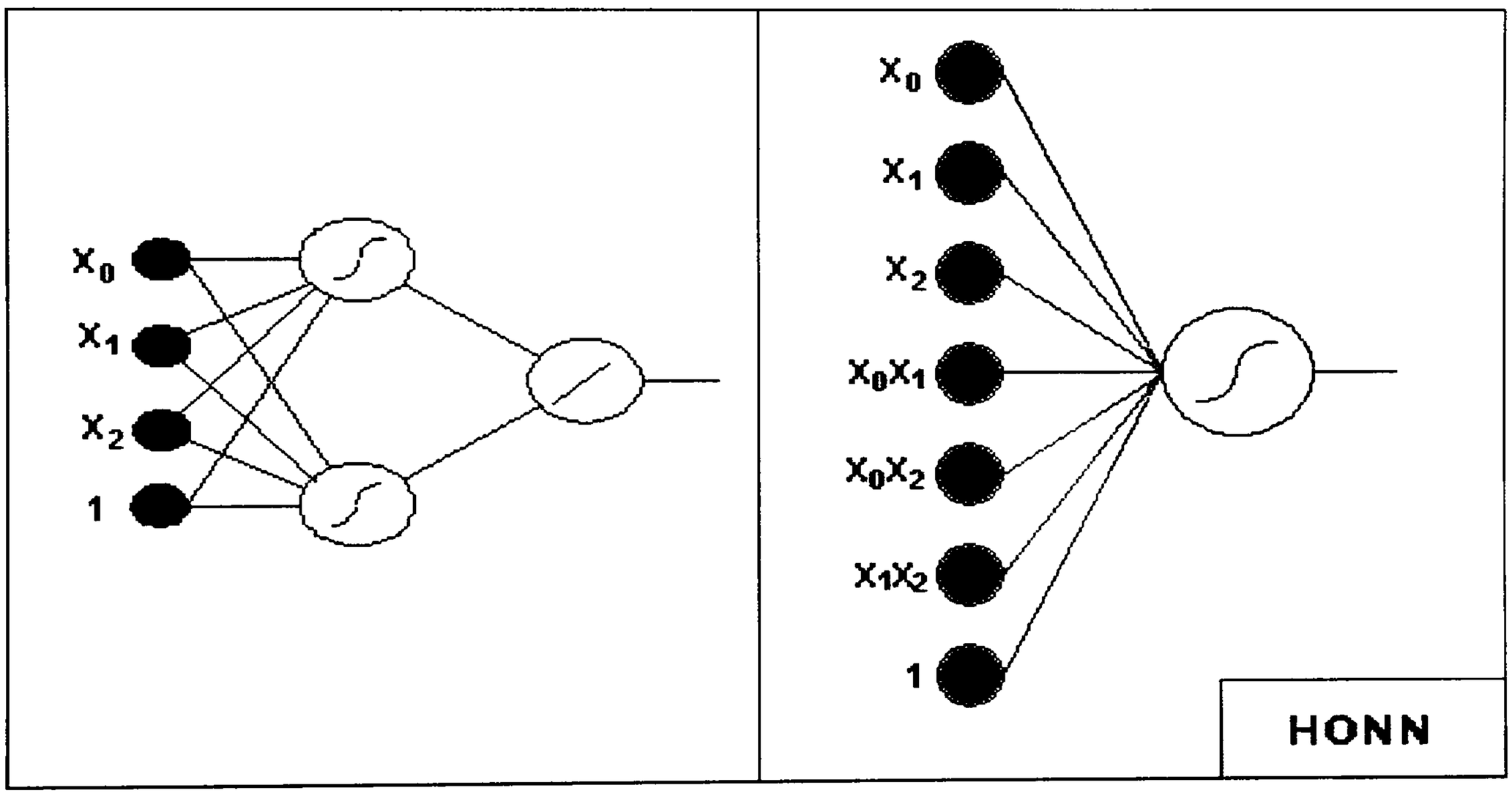


Fig. 3: Left, MLP with three inputs and two hidden nodes; right, second order HONN with three inputs

where:

$x_t^{[n]}$ $(n = 1,2,\cdots,k+1)$ are the model inputs (including the input bias node) at time t

$\tilde{y}_t$ is the HONNs model output

[1] Associative recall is the act of associating two seemingly unrelated entities, such smell and colour. For more information see Karayiannis and Venetsanopoulos (1994).

u_{jk} are the network weights

are the model inputs.

is the transfer sigmoid function: $S(x)=\frac{1}{1+e^{-x}}$, [7]

is a linear function: $F(x)=\sum x_i$ [8]

The error function to be minimised is:

$$E(u_{jk},w_j)=\frac{1}{T}\sum_{t=1}^{T}(y_t-\tilde{y}_t(u_{jk},))^2, \quad \text{with } y_t \text{ being the target value} \qquad [9]$$

HONNs use joint activation functions; this technique reduces the need to establish the relationship between inputs when training. Furthermore this reduces the number of free weights and means that HONNs can be faster to train than MLPs. However, because the number of inputs can be very large for higher order architectures, orders of 4 and over are rarely used.

Another advantage of the reduction of free weights means that the problems of overfitting and local optima affecting the results can be largely avoided, Knowles *et. al.* (2005). For a complete description of HONNs see Giles and Maxwell (1987) while a description of the network training methodology is on chapter 3.7.

.

3.4 The Psi Sigma Network

Psi Sigma networks can be considered as a class of feedfoward fully connected HONNs. First introduced by Ghosh and Shin (1991), the Psi Sigma network utilizes product cells as the output units to indirectly incorporate the capabilities of higher-order networks while using a fewer number of weights and processing units. Their creation was motivated by the need to create a network combining the fast learning property of single layer networks with the powerful mapping capability of HONNs while avoiding the combinatorial increase in the required number of weights. While the order of the more traditional HONN architectures is expressed by the complexity of the inputs, in the context of Psi Sigma, it is represented by the number of hidden nodes.

3.4.1 The Psi Sigma Architecture

In a Psi Sigma network the weights from the hidden to the output layer are fixed to 1 and only the weights from the input to the hidden layer are adjusted, something that greatly reduces the training time. Moreover, the activation function of the nodes in the hidden layer is the summing function while the activation function of the output layer is a sigmoid. The figure below shows a Psi Sigma with one output layer.

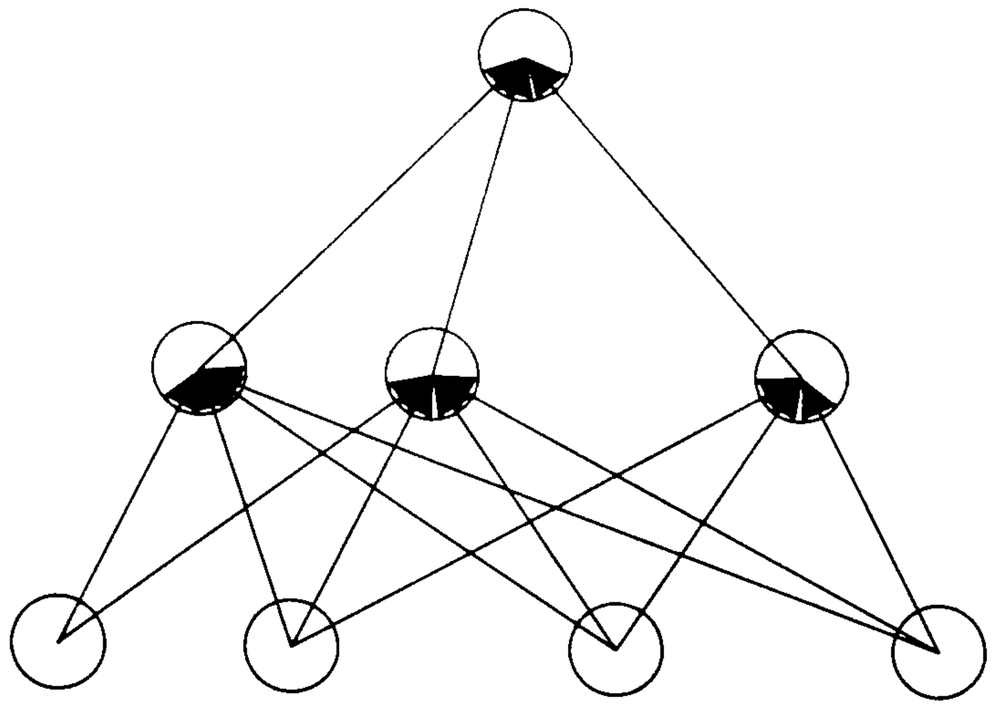


Fig. 4: A Psi Sigma network with one output layer

where:

$x_t \quad (n = 1,2,\cdots,k+1)$ are the model inputs (including the input bias node)

$\widetilde{y}_t$ is the Psi Sigma output

w_j is the adjustable weights

$h(x) = \sum_i x_i$ is the hidden layer activation function [10]

$\sigma(x) = \frac{1}{1+e^{-xc}}$ is the output unit adaptive sigmoid activation function [11]

with c the adjustable term

The error function to be minimised is:

$E(c, w_j) = \frac{1}{T}\sum_{t=1}^{T}(y_t - \widetilde{y}_t(w_k, c))^2$ with y_t being the target value [12]

For example let us consider a Psi Sigma network which is fed with a N+1 dimensional input vector $x = (1, x_1, ..., x_N)^T$ .These inputs are weighted by K weight factors $w_j = (w_{0j}, w_{1j}, ..., w_{Nj})^T$, $j = 1,2,..K$ and summed by a layer of K summing units, where K is the desired order of the network. So the output of

the j-th summing unit, h_j in the hidden layer, is given by: $h_j = w_j^T x = \sum_{k=1}^{N} w_{kj} x_k + w_{oj}$,j=1,2,…, K while the output $\tilde{y}$ of the network is given by $\tilde{y} = \sigma(\prod_{j=1}^{K} h_j)$ (in our case we selected for σ the sigmoid function $\sigma(x) = \frac{1}{1+e^{-xc}}$ [13]). Note that by using products in the output layer we directly incorporate the capabilities of higher order networks with a smaller number of weights and processing units. For example, a k-th degree HONN with d inputs needs $\sum_{i=0}^{k} \frac{(d+i-1)!}{i!*(d+1)!}$ weights if all products of up to k components are to be incorporated while a similar Psi Sigma network needs only (d+1)*k weights. Also note that the sigmoid function is neuron adaptive. As the network is trained not only the weights but also c in [11] is adjusted. This strategy seems to provide better fitting properties and increases the approximation capability of a neural network by introducing an extra variable in the estimation, compared to classical architectures with sigmoidal neurons (Vecci *et al.* (1998)).

3.5 The Softmax Cross Entropy Model

The Softmax cross entropy network (henceforth SCE) is a neural network with a cross entropy cost function and a Softmax activation function at the output nodes. The main idea of this model is to approximate the probability density function for the target value through a histogram representing the probability of the target value being within a range of predefined size. The output value of

a SCE model is therefore a vector with as many elements as there are output nodes, 6 in our case (each node representing one bar of the histogram). The vector elements sum up to unity and represent the density function for the target value while each vector element stands for the probability that the target value lies in the value range the vector element represents.

In order to apply the cross entropy cost function, the target values of the training data set have to be preprocessed so that one gets a target vector (rather than a single target value as with the MLP), where the target vector has as many elements as the SCE model has output nodes. The target vector consists of zeros and a single one. The value 'one' indicates which output node of the network covers the value range where the original target value lies in. Since the network forecasts should be used as a density function, one has to take care that the output vector sums up to unity. This is done by superimposing the Softmax function to the actual network outputs. The Softmax function keeps the internal relationship between the output values but transforms them in a way that their values add up to unity (see equation [16] below).

During the training phase (that is when the network weights are adjusted), the SCE model learns to map the input vector of the training data set to the target vector of the same data set. Since each target vector consists of a single 'one' representing a non-overlapping range of possible output values (while the rest are zeros), the SCE model tries in fact to solve a classification task.

The network might face a situation where the same input vector is related to two different output values (at different times) so that the network has no other chance than to map the input vector to more than one output node. In doing so, the network generates a density function for the target value, while the integrated Softmax function ensures that the probabilities add up to unity.

3.5.1 The SCE network architecture

The difference in architecture with a MLP lies in the multiple output nodes. While the MLP has typically only one output node delivering a level estimation, the SCE network uses several output nodes to represent an approximation of the density function (while being trained on a classification task).

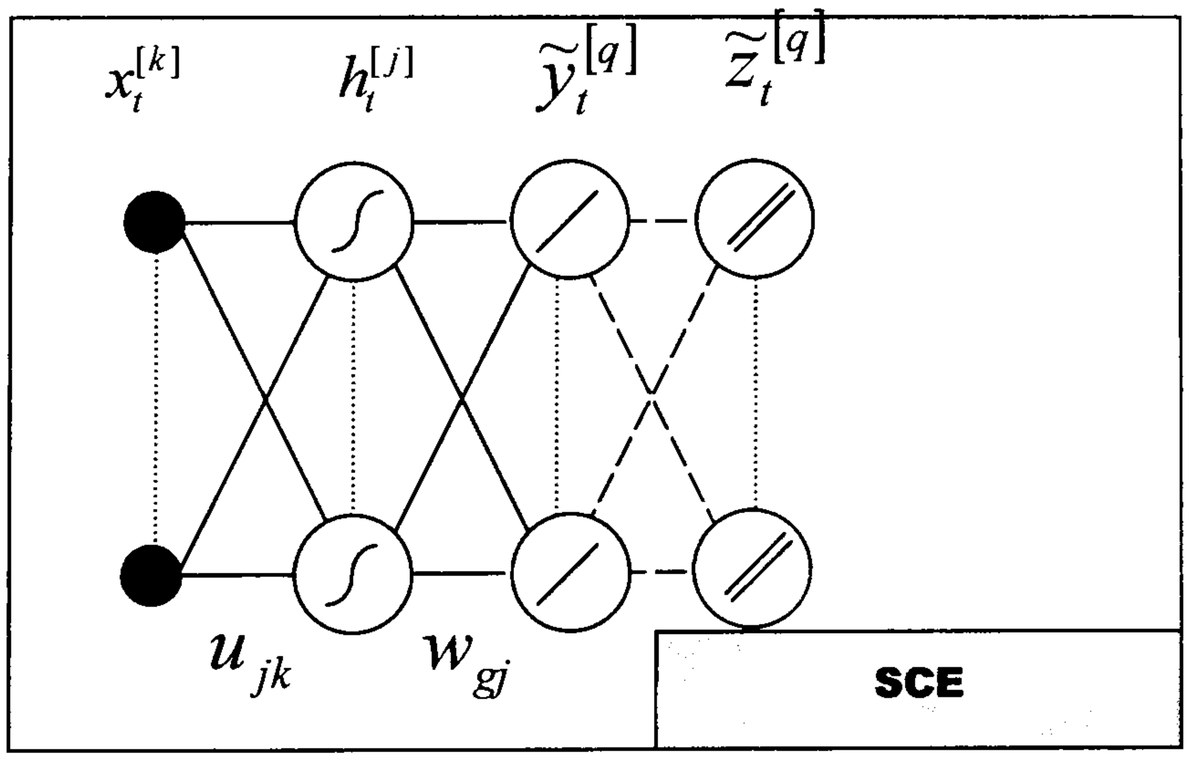


Fig. 5: A single output, fully connected SCE model

where:

$x_t^{[n]}$ $(n = 1,2,\cdots,k+1)$ are the model inputs (including the input bias node) at time t

$h_t^{[m]}$ $(m = 1,2,..., j+1)$ are the hidden nodes outputs (including the hidden bias node)

$\tilde{y}_t^{[g]}$ $(g = 1,2,..., q)$ is the SCE model output before applying the Softmax function

$\tilde{z}_t^{[g]}$ $(g = 1,2,..., q)$ is the network value at the output node *g*

u_{jk} and w_{gj} are the network weights

$\textcircled{S}$ is the transfer sigmoid function: $S(x) = \frac{1}{1+e^{-x}}$, [14]

$\textcircled{/}$ is a linear function: $F(x) = \sum_i x_i$ [15]

$\textcircled{//}$ is the Softmax function $A(g) = \tilde{z}_g = \frac{\exp(\tilde{y}_g)}{\sum_{g_1} \exp(\tilde{y}_{g_1})}$ [16]

with $\tilde{y}_g$ being the output of the linear function

The error function to be minimised is:

$$E(u_{jk}, w_{gj}) = \sum_{t=1}^{T} \sum_{g=1}^{q} y_{tg} \cdot \log\left(\frac{y_{tg}}{\tilde{z}_{tg}(u_{jk}, w_j)}\right), \quad \text{with } y_{tg} \text{ being the target value} \quad [17]$$

3.6 The Gaussian Mixture Model

The GM network was first introduced by Husmeier (1999) and is applied to our EUR/USD time series in Lindemann *et al.* (2004). Additional empirical evident over the GM network forecasting ability in Finance were given by Lindeman *et al.* (2005).

The GM model represents the probability density of the data by a linear combination of a fixed number of normal distributions (where the distribution

width is adapted to the whole set of training data while the locations of the distribution centres depend on the actual input data x_t and the dependent variable y_t). This is done in a hidden layer where each node represents a normal distribution. The actual network output is not the density function itself but the prediction of a single value[2] which is the likelihood of the actual GM model parameters generating the observed value of the dependent variable *y* conditioned on the input data *x*.

To optimise the cost function (that is, to maximise the sum of likelihood values), the weights u_{jk} and w_{ij}, determining the location of the normal distribution centres (μ_t), have to be adapted so that the distance between y_t and μ_t is minimal. Doing so, the centres of the distribution are close to y_t and therefore the likelihood and with it the value of the cost function are high. See figure 5 below to illustrate that working principle.

3.6.1 The GM network architecture

The GM architecture differs in three main ways from the benchmark feedforward MLP network. First, as shown by Husmeier (1999), in order to be a universal approximator at least a second hidden layer is necessary. Second, both the independent and dependent variable (**x**,**y**) are used as input data, since the aim is not to predict **y** but its density distribution $P(\mathbf{y} \mid \mathbf{x})$ respectively

[2] Nevertheless, the whole density distribution can be constructed by varying the value of y over the interesting range of the searched density function.

the corresponding likelihood value. Third, the network uses Gaussian distributions in the second hidden layer.

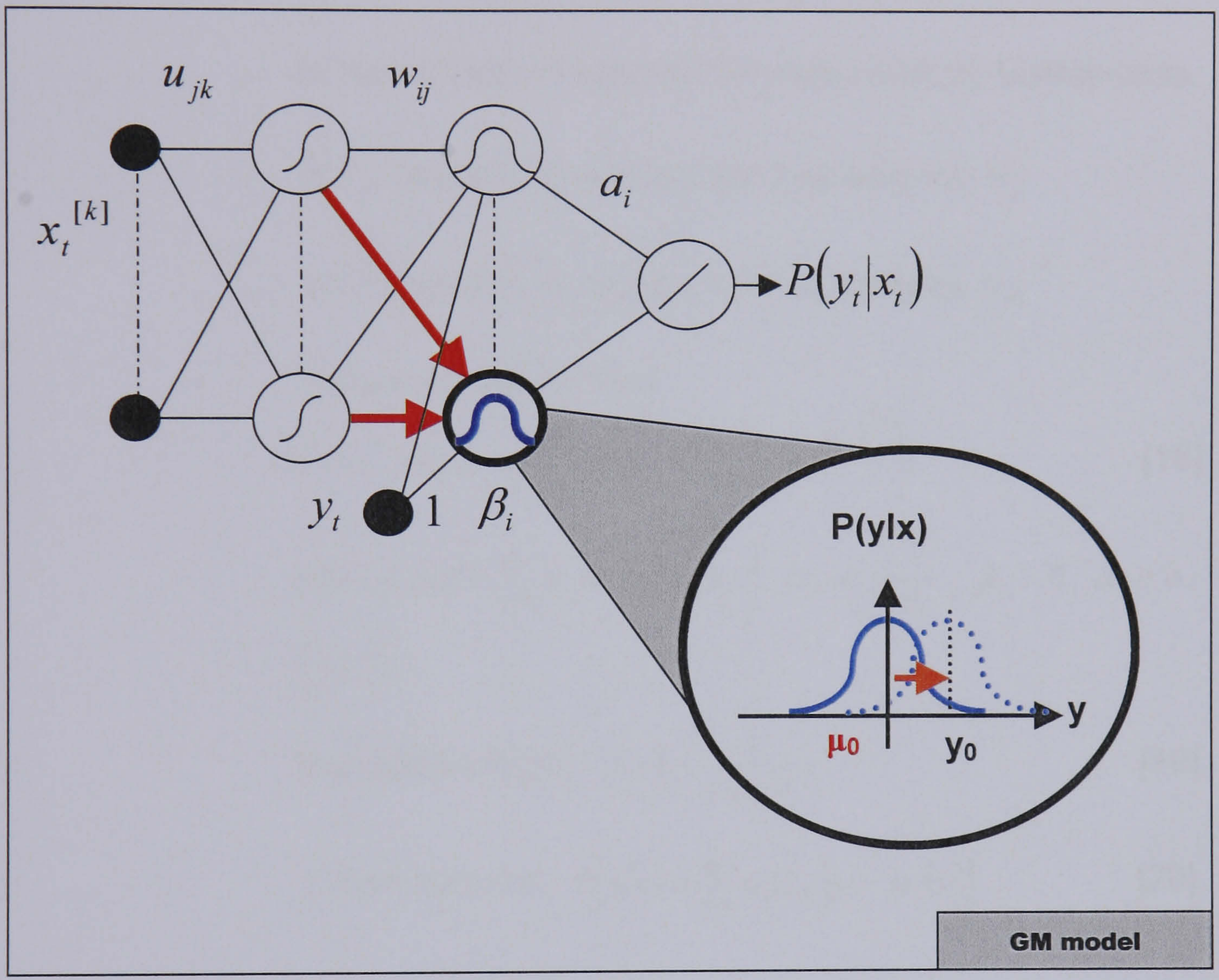


Fig. 6: GM network architecture

The following functions are applied within the GM model:

$x_t^{[n]}$ $(n = 1,2,\cdots,k+1)$ are the model inputs (including the input bias node) at time t

y_t is the argument of the density function conditional on the values of the inputs (note that the weights of y_t are fixed to 1[3])

u_{jk} and w_{ji} are the network weights

β_i define the inverse widths of the Gaussian distributions

a_i are the mixing coefficients, with $\sum_i a_i = 1$

i is the number of applied Gaussian mixture distributions

j is the number of applied network weights w_j

k is the number of applied network weights u_{jk}

Gaussian distribution:

$$G_{\beta_i}(y_t - \mu_i) = \sqrt{\frac{\beta_i}{2\pi}} \exp\left(-\frac{\beta_i \cdot (y_t - \mu_i)^2}{2} \right), \qquad [18]$$

with $\mu_i(x) := \sum_j w_{ij} S\left(\sum_k u_{jk} x_k \right)$, $\sigma_k = \sqrt{\frac{1}{\beta_k}}$, $\beta_i > 0$, $a_i \geq 0$, $\sum_i a_i = 1$

Sigmoid function: $S(x) = \frac{1}{1+e^{-x}}$, [19]

Linear function: $P(y|x) = \sum_i a_i G_{\beta_i}[y - \mu_i(x)]$ [20]

The error function to be minimised is:

$E(u_{jk}, w_{ij}, \beta_i, a_i) = -\frac{1}{T} \sum_{t=1}^{T} \ln\left(P\left(y_t \middle| x_t, u_{jk}, w_{ij}, \beta_i, a_i \right) \right)$, with y_t being the target value [21]

[3] If we would not fix the weight to 1 the network could decrease the cost function not only by adjusting the centres of the Gaussian mixture functions but also by changing the original target value y_t.

It is possible to update the parameters of the GM model by gradient descent, as was done with the MLP network. However this algorithm, due to the architectural complexity of the GM network, is very time consuming.

3.7 Neural Network Training Procedure

The training of the network (which is the adjustment of its weights in the way that the network maps the input value of the training data to the corresponding output value) starts with randomly chosen weights and proceeds by applying learning algorithms based on the backpropagation of errors[4] (Shapiro (2000)). The learning algorithms simply try to find those weights which optimise the error function (normally the sum of all squared differences between target and actual values). Since networks are able to learn the training data (as well as their outliers and their noise) by heart, it is crucial to stop the training procedure at the right time to prevent overfitting (this is called 'early stopping'). This can be achieved by dividing the dataset into 3 subsets respectively called the training and test sets used for simulating the data currently available to fit and tune the model and the validation set used for simulating future values. The network parameters are then estimated by fitting the training data using the above mentioned iterative procedure (backpropagation of errors). The iteration length is optimised by maximising the forecasting accuracy for the test dataset. Finally, the predictive value of the model is evaluated applying it to the validation dataset (out-of-sample dataset).

---

[4] Backpropagation networks are the most common multilayer networks and are the most commonly used type in financial time series forecasting (Kaastra and Boyd (1996).

In the fourth charter we forecast the EUR/USD exchange rate with the networks which present the better statistical performance in-sample. In the fifth and sixth charter, we modify the error function of our models and we use the networks which present the higher financial performance in-sample, in terms of annualised return and Sharpe ratio. In the next charters where we forecast the volatility, we use for each neural network design the average of a committee of 20 networks which presents the better statistical performance in-sample. Our aim is that since the starting point for each network is a set of random weights, forecasts can differ between networks we use the average of a committee in order to eliminate any variance between our neural network forecasts. In all cases the specifications of the NNs models used on this research (number of hidden nodes, number of hidden layers, order of network and number and type of inputs) were chosen based on trial and error in the in-sample period.

CHAPTER 4

Higher Order and Recurrent Neural Architectures for Trading the EUR/USD Exchange Rate[5]

Overview

The motivation for this chapter is to investigate the use of higher order neural network architectures when applied to the task of forecasting and trading the Euro/Dollar (EUR/USD) exchange rate using multivariate series as inputs. This is done by benchmarking three different neural network designs representing a Higher Order Neural Network (HONN), a Psi Sigma Network and a Recurrent Network (RNN) with three successful architectures, the traditional Mutilayer Perceptron (MLP), the Softmax and the Gaussian Mixture (GM) models, as reported in Dunis and Williams (2002, 2003) and Lindemann *et al.* (2004). So in other words, the motivation of this chapter is conduct a forecasting competition between several up to date non linear models and to check if the theoretical advantages of HONNs compared to the traditional MLPs and RNNs are translated to more accurate/profitable forecasts. More specifically, the trading performance of the six models is investigated in a forecast and trading simulation competition on the EUR/USD time series over a period of 8 years. These results are also benchmarked with more traditional models such as a moving average convergence divergence technical model (MACD), an autoregressive moving average model (ARMA) and a logistic regression model (LOGIT).

---

[5] This paper has been presented at the Forecasting Financial Markets 2008 conference in Aix-en-Provence (21 to 23 May 2008) and at the 50th Operational Research Society conference in York (9 to 11 September 2008) and after referees comments is currently in the last stage of the reviewing process for potential publication in '*Quantitative Finance*'.

As it turns out, the MLP, the HONN, the Psi Sigma and the RNN models all do well and outperform the more traditional models in a simple trading simulation exercise. However, when more sophisticated trading strategies using confirmation filters and leverage are applied, the GM network produces remarkable results and outperforms all the other network architectures.

4.1 Introduction

Neural networks are an emergent technology with an increasing number of real-world applications including Finance (Lisboa *et al.* (2000)). However their numerous limitations often create scepticism about their use among practitioners.

The motivation of this chapter is conduct a forecasting competition between several up to date non linear models and to check if the theoretical advantages of HONNs that try to overcome some of the limitations of the traditional NNs, are translated to more accurate/profitable forecasts using multivariate series as inputs. This is done by benchmarking three different neural network architectures representing a Higher Order Neural Network (HONN), a Psi Sgima network and a Recurrent Neural Network (RNN). Their trading performance on the Euro/Dollar (EUR/USD) time series is investigated and is compared with the three best models reported by Dunis and Williams (2002, 2003) and Lindemann *et al.* (2004), the Multi-layer Perceptron (MLP), the Softmax and the Gaussian Mixture (GM) model. In order the competion to be fair, we fed our networks with the same inputs as by Dunis and Williams (2002, 2003) and Lindemann *et al.* (2004) who

computed our benchmark models. We study the forecasting power of our models if we feed them with autoregressive inputs in the next chapter. A direct comparison of the forecasting power of our models between autoregressive and multivariate series is beyond the scope of this chapter.

The results of our three networks can also be compared to the more traditional approaches also studied by Dunis and Williams (2002, 2003), namely a moving average convergence divergence technical model (MACD), an autoregressive moving average model (ARMA) and a logistic regression model (LOGIT).

As it turns out, the MLP, the HONN and the Psi Sigma demonstrate a similar good performance and outperform the more traditional models in a simple trading simulation exercise, while the GM model outperforms all models when more sophisticated trading strategies using confirmation filters and leverage are applied. This might be due to the ability of the GM model to use probability distributions to identify successfully trades with a high Sharpe ratio.

The rest of the chapter is organised as follows. In section 4.2, we describe the dataset used for this research, actually the same as in Dunis and Williams (2002, 2003) and Lindemann *et al.* (2004). Section 4.3 discuss the methodology and gives the empirical results of all the models considered. Section 4.4 investigates the possibility of improving their performance with the application of more sophisticated trading strategies while section 4.5 provides some concluding remarks.

4.2 The EUR/USD Exchange Rate and Related Financial Data

Our benchmark test is to trade the EUR/USD exchange rate based on daily forecasts of its London closing prices[6]. All time series are daily closing data obtained from a historical database provided by Datastream and used in Dunis and Williams (2002, 2003) and Lindemann *et al.* (2004).

Name of period	Trading days	Beginning	End
Total dataset	1749	17 October 1994	03 July 2001
Training dataset	1459	17 October 1994	18 May 2000
Out-of-sample dataset [Validation set]	290	19 May 2000	03 July 2001

Table 1: The EUR/USD dataset

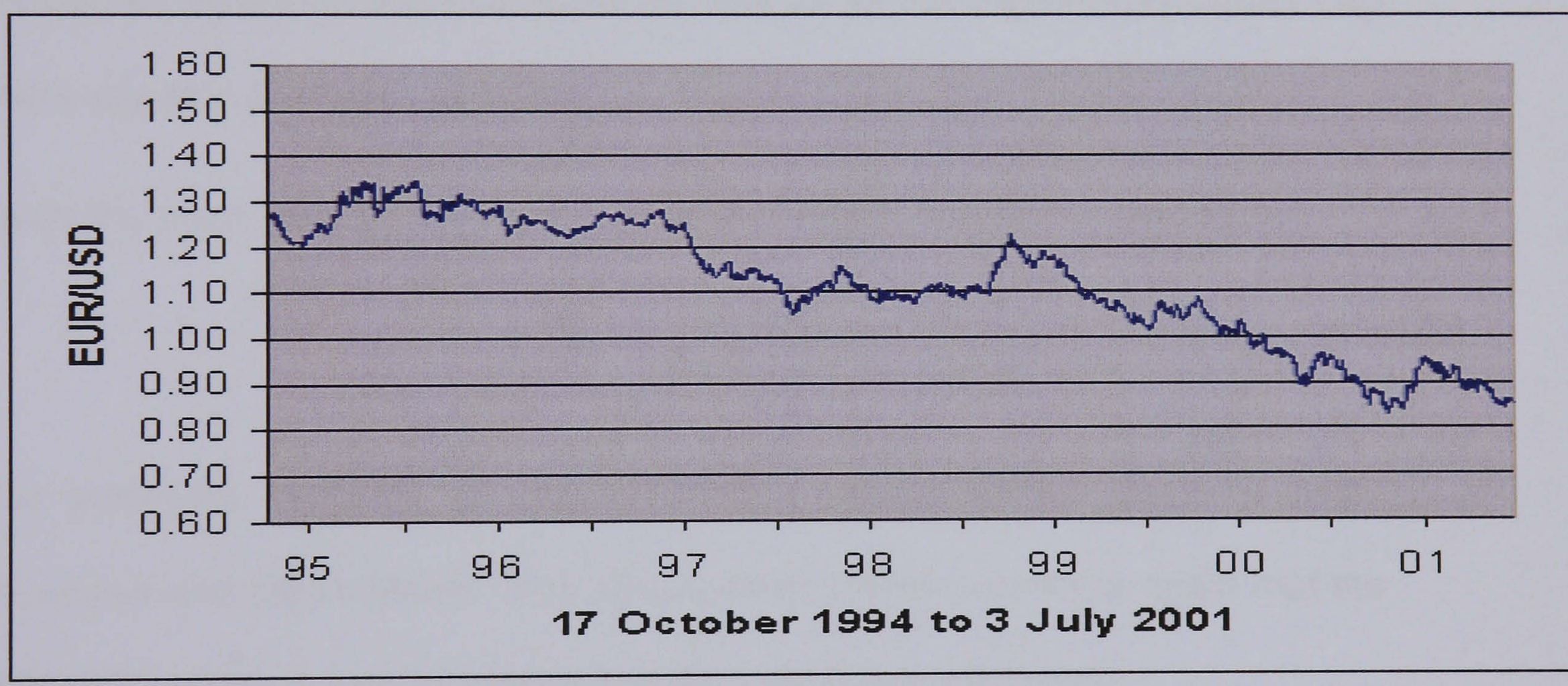


Fig. 7: EUR/USD London daily closing prices (total dataset)

Dunis and Williams (2002, 2003) carried out a variable selection and identified the explanatory variables listed in table 2.

---

[6] EUR/USD is quoted as the number of USD per Euro: for example, a value of 1.2657 is USD1.2657 per Euro. The EUR/USD exchange rate only exists from 4 January 1999: it was extrapolated from 17 October 1994 to 31 December 1998 and a synthetic EUR/USD series was created for that period using the fixed EUR/DEM conversion rate agreed in 1998, combined with the USD/DEM daily market rate.

Number	Variable	Mnemonics	Lag
1	*US $ TO UK £ (WMR) – EXCHANGE RATE*	USDOLLR	12
2	*JAPANESE YEN TO US $ (WMR) – EXCHANGE RATE*	JAPAYE$	1
3	*JAPANESE YEN TO US $ (WMR) – EXCHANGE RATE*	JAPAYE$	10
4	*BRENT CRUDE – Current Month, fob U$/BBL*	OILBREN	1
5	*GOLD BULLION $/TROY OUNCE*	GOLDBLN	19
6	*FRANCE BENCHMARK BOND 10 YR (DS) – RED. YIELD*	FRBRYLD	2
7	*ITALY BENCHMARK BOND 10 YR (DS) – RED. YIELD*	ITBRYLD	6
8	*JAPAN BENCHMARK BOND – RYLD.10 YR (DS) – RED.*	JPBRYLD	9
9	*NIKKEI 225 STOCK AVERAGE – PRICE INDEX*	JAPDOWA	1
10	*NIKKEI 225 STOCK AVERAGE – PRICE INDEX*	JAPDOWA	15

Table 2: Explanatory variables and Datastream mnemonics

The observed EUR/USD time series is non-normal (Jarque-Bera statistics confirmed this at the 99% confidence interval) containing slight skewness and low kurtosis. It is also nonstationary and Dunis and Williams (2002, 2003) decided to transform the EUR/USD as well as all the explanatory series into stationary series of rates of return[7].

Given the price level P_1, P_2,…, P_t, the rate of return at time t is formed by:

$$R_t = \left(\frac{P_t}{P_{t-1}}\right) - 1^{8} \qquad [22]$$

The summary statistics of the EUR/USD returns series reveal a slight skewness and high kurtosis. The Jarque-Bera statistic confirms again that the EUR/USD series is non-normal at the 99% confidence interval.

[7] Confirmation of its stationary property is obtained at the 1% significance level by both the Augmented Dickey Fuller (ADF) and Phillips-Perron (PP) test statistics.
[8] For small returns as in this application, arithmetic and logarithmic returns are almost identical. Moreover, log returns are not linearly additive across portfolio components.

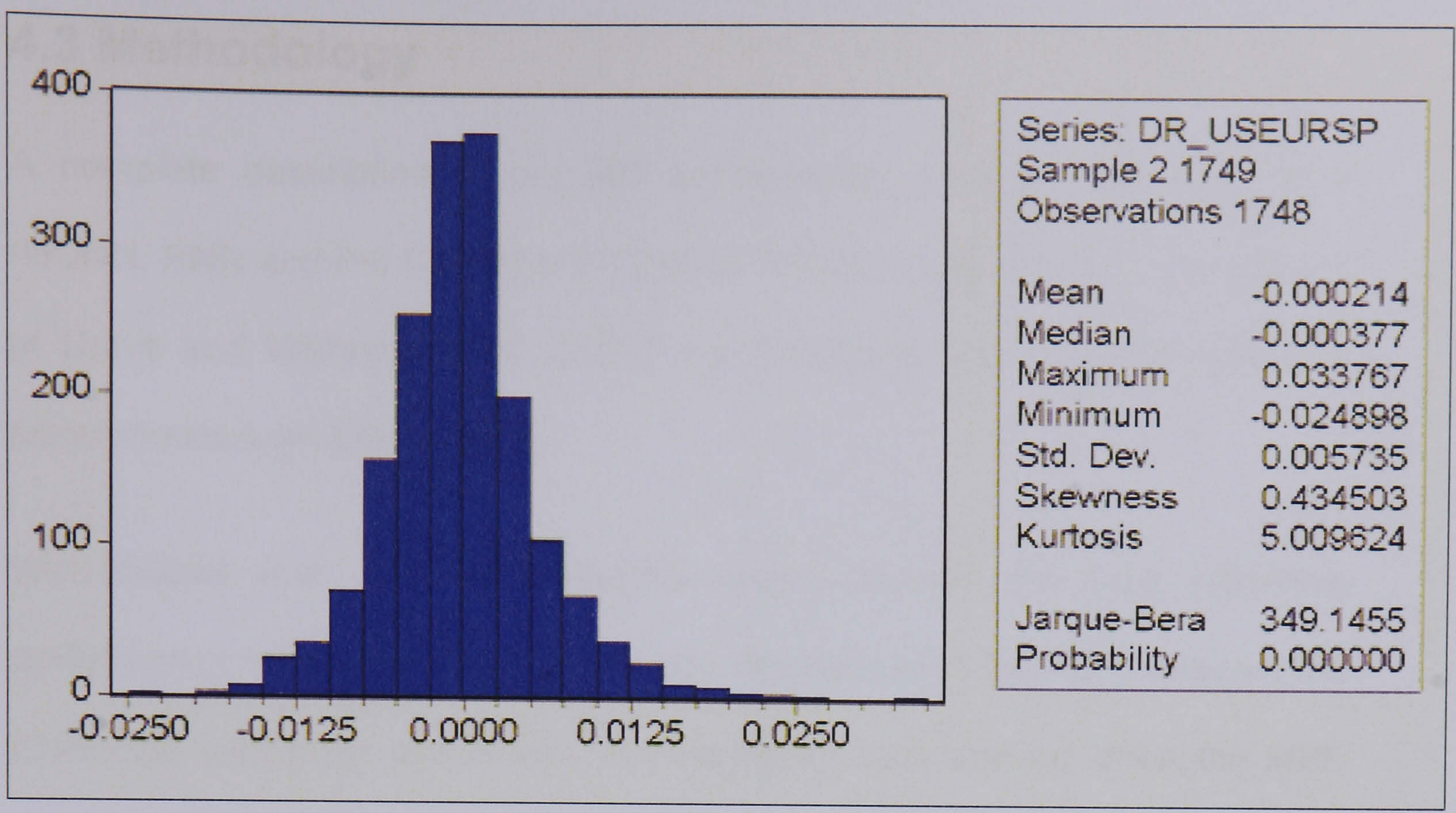


Fig. 8: EUR/USD returns summary statistics (total dataset)

A further transformation includes the creation of interest rates yield curve series, generated by:

$$yc = 10 \text{ year benchmark bond yields} - 3 \text{ month interest rates} \quad [23]$$

Following Dunis and Williams (2002, 2003) and Lindemann *et al.* (2004), we divide our dataset as follows:

Name of period	Trading days	Beginning	End
Total data set	1749	17 October 1994	03 July 2001
Training data set	1169	17 October 1994	08 April 1999
Test data set	290	09 April 1999	18 May 2000
Out-of-sample data set [Validation set]	290	19 May 2000	03 July 2001

Table 3: The neural networks datasets

4.3 Methodology

A complete description of our NN architectures used on this application (HONN, RNN and Psi Sigma) is in chapter 3 while a description of the models of Dunis and Williams (2002, 2003) and Lindemann *et al.* (2004) used as benchmarks is on their papers.

We choose and use the networks which present the best statistical performance in-sample in terms of mean absolute error (MAE) to forecast the EUR/USD exchange rate return. Our networks stop training when the MSE between the actual values and our forecasts in the test sub-period is minimized. Then the predictive value of the model is evaluated applying it to the validation dataset (out-of-sample dataset). The trading strategy applied is simple: go or stay long when the forecast return is above zero and go or stay short when the forecast return is below zero.[9] Our methodology is identical with the one followed by Dunis and Williams (2002, 2003) and Lindemann *et al.* (2004).

In the table 4 below we present the trading performance of our models compared with the models of Dunis and Williams (2002, 2003) and Lindemann *et al.* (2004) who performed best while in Appendix A.1.3 are the characteristics of our networks. Appendix A.1.1 documents the performance measures used while Appendix A.1.2 gives the results of the benchmark models as presented by Lindemann *et al.* (2004).

---

[9] A 'long' EUR/USD position means buying Euros at the current price, while a 'short' position means selling Euros at the current price.

TEXT BOUND INTO

THE SPINE

	MACD	ARMA	LOGIT	Naive	MLP	SCE	GM	RNN	HONN	Psi Sig
Ratio (excluding costs)	0.97	1.10	1.81	1.83	2.57	2.26	2.09	2.57	2.58	2.55
ised Volatility (excluding costs)	11.7%	11.7%	11.6%	11.6%	11.6%	11.6%	11.6%	11.6%	11.6%	11.6%
ised Return (excluding costs)	11.3%	12.9%	21.1%	21.3%	29.7%	26.3%	24.2%	29.8%	29.8%	29.5%
um Drawdown (excluding costs)	-7.8%	-10.1%	-5.8%	-9.1%	-9.1%	-7.8%	-12.4%	-13.8%	-9.2%	-5.9%
ns Taken (annualised)	22	112	123	109	118	143	162	124	129	133

Table 4: Trading performance results

The trading performance of our models (RNN, HONN and Psi Sigma) in terms of annualized return and Sharpe ratio is similar with those obtained by Dunis and Williams (2002, 2003) with a MLP. As the Psi Sigma and HONN models are able to capture higher order correlations, we expected that their performance should be significantly better than the ones for the MLP and RNN models. However, this was not confirmed by our empirical results where HONNs and Psi Sigma present similar performance with the other NN models. However, the other major theoretical advantage of Psi Sigma networks, namely their speed, was clearly confirmed as we achieved about the same results as the HONNs and the RNNs with respectively half and one tenth of their training time[10].

4.4 Trading Costs, Filters and Leverage

Up to now, we have presented the trading results of all our models without considering transaction costs. Since some of our models trade frequently, taking transaction costs into account might change the whole picture.

[10] We needed about 3 minutes to train our Psi Sigma network, about 6 minutes to train our MLP and the HONN and about 30 minutes to train our RNN with an Intel Core 2 Duo T7300 Fujitsu Amilo Laptop.

We therefore introduce transaction costs as well as a filtered trading strategy for each model. The aim is to devise a trading strategy filtering only those trades which have a high probability of being successful. This should help to reduce the negative effect of transaction costs as trades with an expected gain lower than the transaction costs should be omitted.

4.4.1Transaction Costs

The transaction cost for a tradable amount, say USD 5-10 million, is about 3 pips (0.0003 EUR/USD) per trade (one way) between market makers. But, as noted by Dunis and Williams (2002, 2003), since the EUR/USD time series is a series of bid rates, we have to pay the costs only once and not twice per taken position.

With an average exchange rate of EUR/USD of 0.8971 for the out-of-sample period, a cost of 3 pips is equivalent to an average cost of 0.033% per position.

4.4.2 Confirmation Filter Strategies

4.2.2.1 Confirmation Filters

We now introduce trading strategies devised to filter out those trades with expected returns below the 0.033% transaction cost. Due to the architecture of our models, the trading strategy for the MLP, the RNN, the Psi Sigma and the HONN networks consists of one single parameter while the strategy applied to the SCE and GM model uses two parameters. This is because of

the additional available information which the SCE and GM models offer in terms of probability distributions.

Up to now, the trading strategies applied to the models use a zero threshold: they suggest to go long when the forecast is above zero and to go short when the forecast is below zero. In the following, we examine how the models behave if we introduce a threshold *d* around zero (see figure 9) and what happens if we vary that threshold.

The filter rule for the MLP, RNN, HONN and Psi Sigma models is presented in figure 9 below

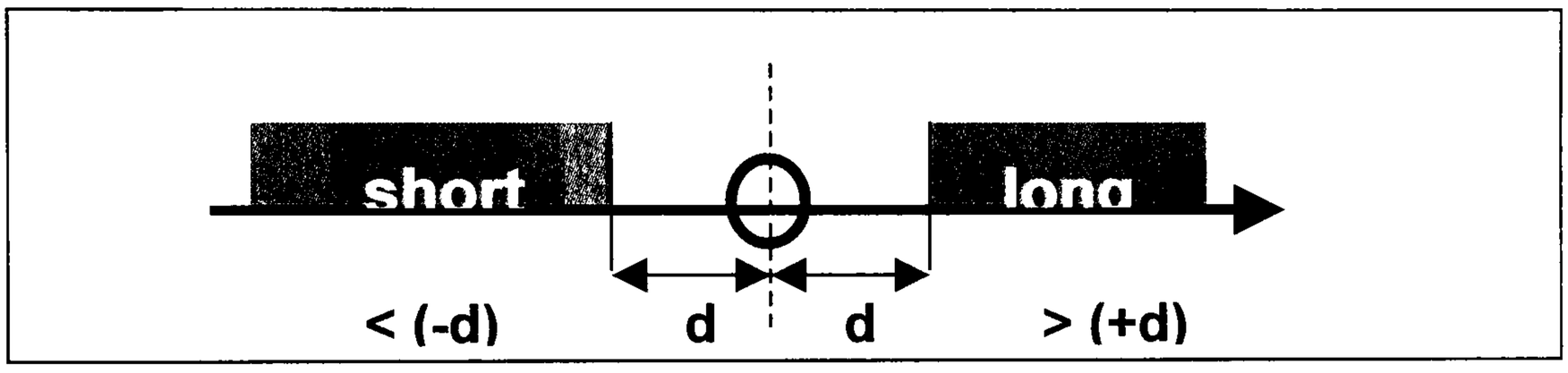


Fig. 9: Filtered trading strategy with one single parameter

Since the forecast of the SCE and GM models provide more information than the other models, we are able to introduce a second parameter for the trading strategy, which is the probability level.

As a result, and following Lindemann *et al.* (2004), all those trading signals are filtered out which are (a) not indicating a price move (in either direction) bigger than the threshold *d* and in addition (b) not indicating a probability higher than x% for the forecast price move (which is the sum of the histogram bars for the SCE model and the space under the density function curve for the GM model). If both conditions are fulfilled at the same time for an up as well

as for a down move, the strategy picks the trading signal with the higher probability.

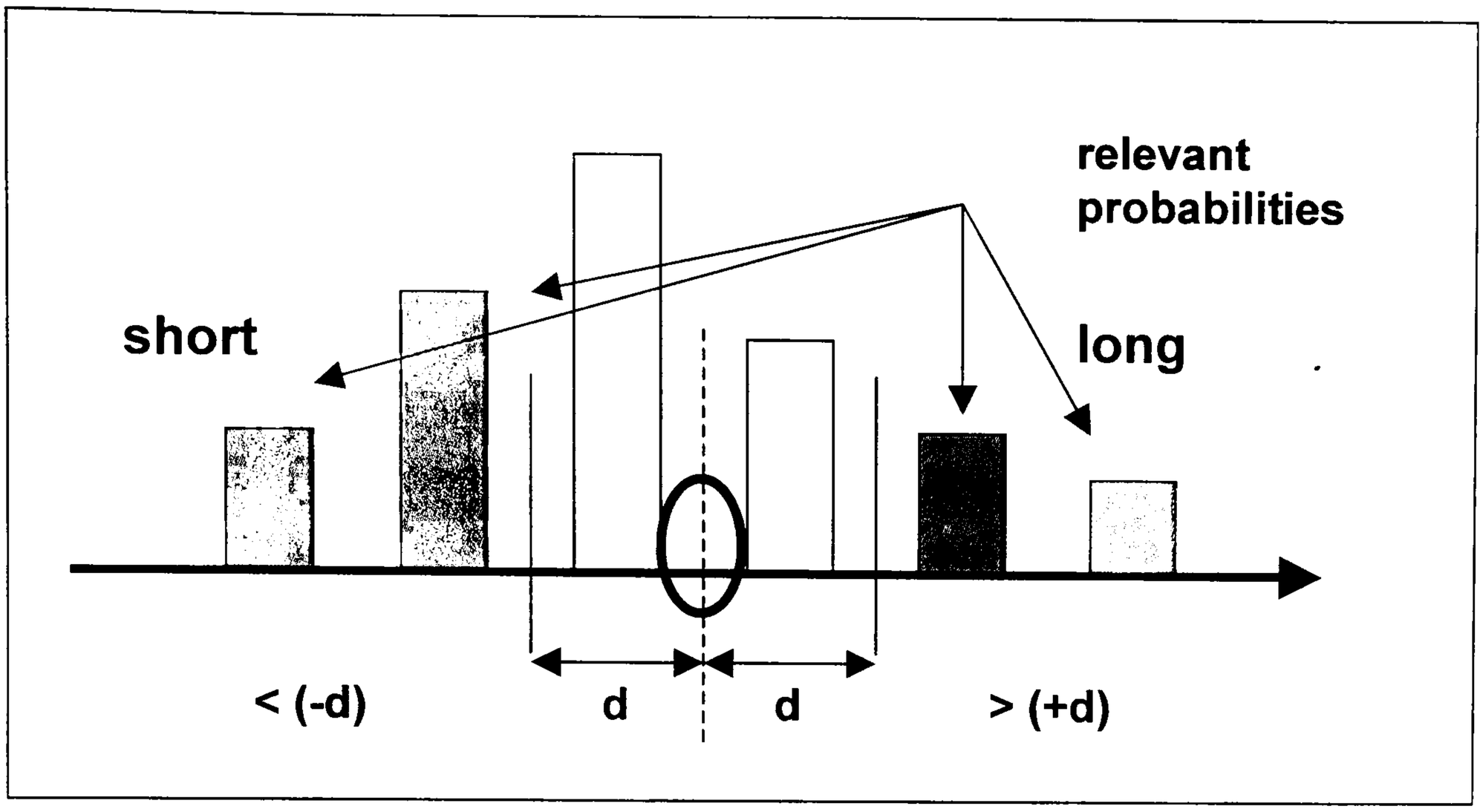


Fig. 10: Filtered trading strategy for the SCE model

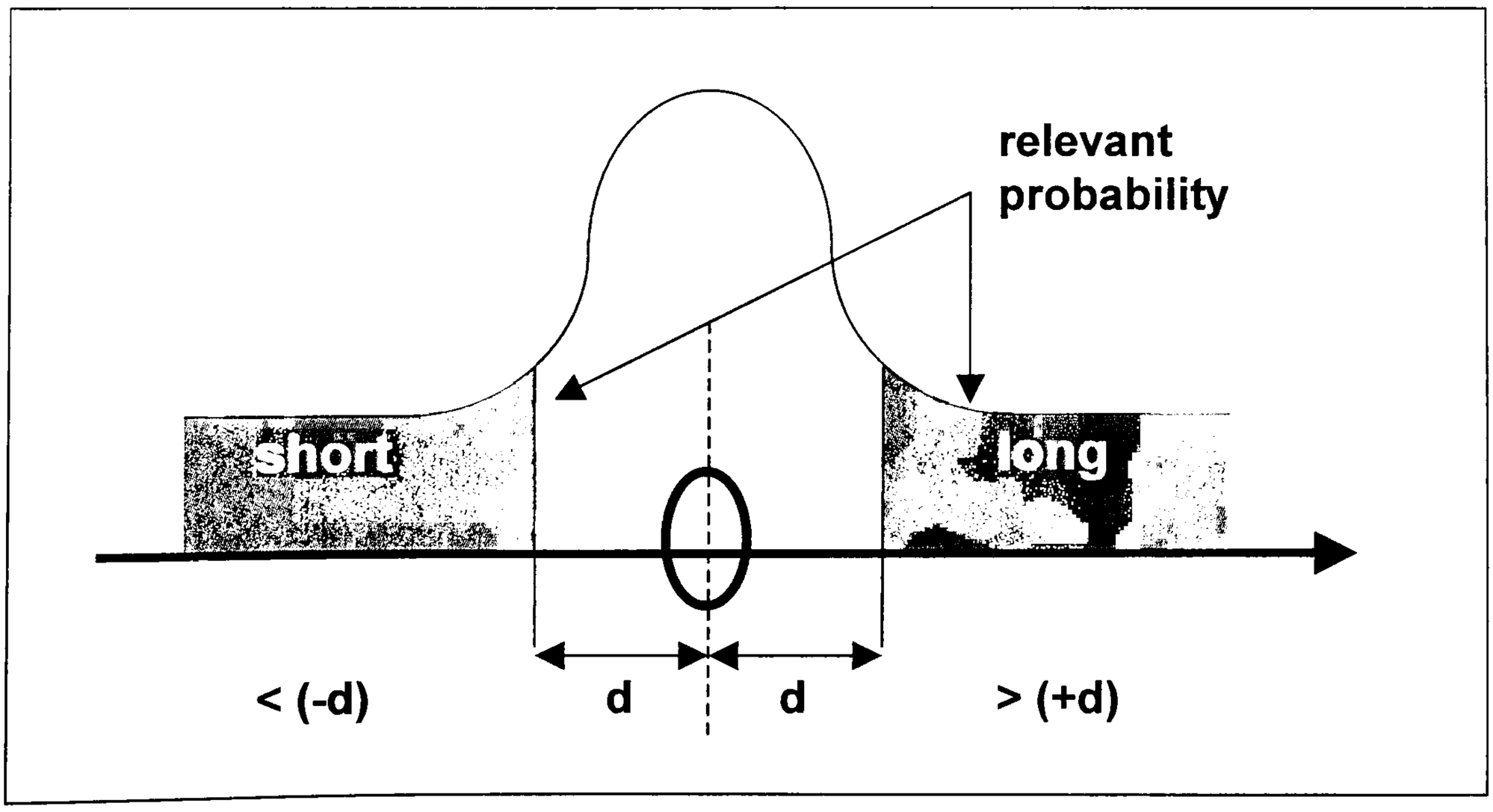


Fig. 11: Filtered trading strategy for the GM model

The thresholds chosen by Lindemann *et al.* (2004) for the GM, the Softmax and the MLP networks are given in table 5 below.

Model	Threshold (d)
MLP	*= 0.00*
SCE	*= 0.25 (move size > |0.3%|)*
GM	*= 0.00 (probability > 0.0%)*

Table 5: Chosen parameters in Lindemann *et al.* (2004)

4.2.2.2 Empirical Results of the RNN, HONN and Psi Sigma models

Following the methodology of Lindemann *et al.* (2004), we proceed with the selection of the optimal thresholds. Taking the test period results, we choose the threshold that gives the higher return and Sharpe ratio. Our chosen parameters are presented in the table below while the detailed results leading to their choice are documented in Appendix A.1.4.

Model	Threshold (d)
RNN	*= 0.05*
HONN	*= 0.00*
Psi Sigma	*= 0.00*

Table 6: Chosen parameters for each trading strategy

For all networks, we leave the threshold at zero (d=0.0) since the profit on the test dataset is largest at this value. The value of d=0.1 looks promising in the case of Psi Sigma from a Sharpe ratio point of view but the lower level of profit deterred us from choosing it as a threshold. We stick therefore to d=0.0 in all cases.

A summary of the out-of-sample trading performance of our three models benchmarked against the Naïve, the MLP, the SCE and the GM networks using the selected thresholds as reported by Dunis and Williams (2002, 2003) and Lindemann *et al.* (2004) is presented in table 7 below.

We can see that the MLP, the RNN, the HONN and the Psi Sigma networks show about the same performance based on the annualised return and the Sharpe ratio. However, it is worth mentioning that the time used to obtain these results with the Psi Sigma network is half that needed with HONNs and one tenth that needed with RNNs.

	NAIVE	MLP	SCE	GM	RNN	HONN	Psi Sigma
Sharpe Ratio (excluding costs)	1.83	2.57	2.67	2.09	2.57	2.58	2.55
Annualised Volatility (excluding costs)	11.6%	11.6%	8.5%	11.6%	11.6%	11.6%	11.6%
Annualised Return (excluding costs)	21.3%	29.7%	22.7%	24.2%	29.8%	29.8%	29.5%
Maximum Drawdown (excluding costs)	-9.1%	-9.1%	-5.7%	-12.4%	-13.8%	-9.2%	-5.9%
Positions Taken (annualised)	109	118	120	162	124	129	133
Transaction costs	3.6%	3.9%	3.9%	5.3%	4.0%	4.3%	4.4%
Annualised Return (including costs)	17.7%	25.8%	18.8%	18.9%	25.7%	25.6%	25.1%

Table 7: Out-of-sample results for the chosen parameters

4.4.3 Leverage to Exploit High Sharpe Ratios

As we have seen, the application of a filtered trading strategy does not improve the results in this case, since all 3 models stick to a threshold of zero. The question then is whether we can gain higher risk-adjusted profits by using leverage.

The leverage factors applied are calculated in such a way that each model has a common volatility of 10%[11] on the test data set.

Since we now have additional information (which is the leveraged trading results based on the test dataset), we can rethink our former choice of thresholds. The thresholds that we select in the end are presented in the table below while an insight about our selection process can been found in Appendix A.1.4.

Model	**Threshold (d)**
RNN	*= 0.05*
HONN	*= 0.00*
Psi Sigma	*= 0.00*

Table 8: Parameters for the leveraged trading strategies

For the HONN and the Psi Sigma network we leave the threshold at 0 as the profit is maximized for this value on the test dataset. For the RNN we choose d = 0.05 which gives the highest profit and Sharpe ratio on the test dataset and filters out only 3 trades per year.

The thresholds reported by Lindemann *et al.* (2004) who follow the same methodology are presented in table 9 below.

Model	**Threshold (d)**	
MLP	*= 0.05*	
SCE	*= 0.25*	*(move size >|0.3%|)*
GM	*= 0.35*	*(probability = 0.25)*

Table 9: Parameters for the leveraged trading strategies

The transaction costs are calculated by taking 0.033% per position into account, while the costs of leverage (interest payments for the additional

[11] Since most of the models (using a threshold of zero) have a volatility of about 10%, we have chosen this level as our basis. The leverage factors retained are given in table 10 below.

capital) are calculated with 4% p.a. (that is 0.016% per trading day[12]). Our final results are presented in table 10 below.

	NAIVE	MLP	SCE	GM	RNN	HONN	Psi Sigma
Sharpe Ratio (excluding costs)[13]	1.83	2.30	2.67	3.80	2.57	2.58	2.55
Annualised Volatility (excluding costs)	11.9%	13.4%	12.5%	12.2%	11.9%	12.3%	11.9%
Annualised Return (excluding costs)	21.8%	30.8%	33.2%	46.4%	30.7%	31.7%	30.4%
Maximum Drawdown (excluding costs)	-9.3%	-10.3%	-8.5%	-11.3%	-14.3%	-9.8%	-6.1%
Leverage Factor	1.03	1.62	1.46	3.99	1.03	1.03	1.03
Positions Taken (annualised)	109	89	120	68[14]	121	129	133
Transaction and leverage costs	3.7%	6.1%	7.1%	12.5%	4.0%	4.3%	4.6%
Annualised Return (including costs)	18.1%	24.7%	26.1%	33.9%	26.7%	27.0%	25.8%

Table 10: Trading performance - final results[15]

As can be seen from table 10, GM networks are able to take advantage of the combination of a confirmation filter and leverage and to deliver higher Sharpe

---

[12] The interest costs are calculated by considering a 4% interest rate p.a. divided by 252 trading days. In reality, leverage costs also apply during non-trading days so that we should calculate the interest costs using 360 days per year. But for the sake of simplicity, we use the approximation of 252 trading days to spread the leverage costs of non-trading days equally over the trading days. This approximation prevents us from keeping track of how many non-trading days we hold a position.

[13] The calculation is done without transaction and leverage costs due to a better comparability to other published numbers (which are generally calculated in this way).

[14] The SCE and GM committees have actually taken more trades than reported in the table above (e.g. the GM model has actually taken 134 positions). The reason why Lindemann *et al.* (2004) report a smaller number of trades is that SCE and GM committees are able to invest less than 100% of their total capital per position (this is due to the fact that the position size is determined by the average number of committee members generating a trading signal). Since our transaction costs of 0.033% per position are based on the assumption of 100% of invested total capital, we have to recalculate the 134 positions of partially invested total capital into the equivalent number of positions with 100% of invested capital (which are the above shown 68 positions).

[15] Not taken into account are the following effects:

a) The interest that could be earned during times where the capital is not traded [non-trading days] and could therefore be invested;
b) The SCE and GM committees are not forced to use 100% of their capital when trading (leaving out a leverage factor <1), since the amount is determined by the average forecast of the 30 models. If the committees invest therefore only a few per cent of the capital available but apply the leverage factor (>1), the additional capital has not to be borrowed (since there is still own money available) and therefore leverage costs would not be incurred. Those 'savings' are not taken into account here.

ratios and returns. The ability of the GM networks to capture the probability density of the data by a linear combination of a fixed number of normal distributions seems to have allowed them to exploit better the trading strategies applied. HONNs achieve the highest annualised return net of transaction costs among the other five competing models while the Psi Sigma, the RNN and the SCE models achieve similar performances. It seems that the ability of HONNs and Psi Sigma to capture higher order correlations within our dataset and the ability of the RNN to embody short term memory does not help them to exploit the leverage and the confirmation filter and to achieve higher trading performance. Overall, our three models perform well (see table 7), however they do not manage to take advantage of more sophisticated trading strategies using confirmation filters and leverage contrary to density distribution networks (see table 10).

4.5 Concluding Remarks

In this chapter, we apply Recurrent, Higher Order and Psi Sigma neural networks to a one-day-ahead forecasting and trading task of the EUR/USD time series. We develop these different prediction models over the period October 1994 - May 2000 and validate their out-of-sample trading efficiency over the following period from May 2000 through July 2001. Our results are benchmarked against those of the Gaussian Mixture, the Softmax Entropy and the Multi-layer Perceptron models presented by Dunis and Williams (2002, 2003) and Lindemann *et al.* (2004) who study the same series over the same time period.

Trading strategies that should filter out potentially unsuccessful trades by using a confirmation threshold have not worked out and the Psi Sigma, HONN and RNN models fail to exploit leverage for the asset and time period under review.

Nevertheless, the trading results of the Psi Sigma, HONN and RNN models are similar to the best model of Dunis and Williams (2002, 2003), the MLP, when applied without confirmation filter and leverage. When more sophisticated trading strategies are applied, our results are not improved significantly although HONNs still perform remarkably.

CHAPTER 5

Modelling and Trading the EUR/USD Exchange Rate at the ECB Fixing[16]

Overview

The motivation for this chapter is to investigate the use of HONNs when applied to the task of forecasting and trading the Euro/Dollar (EUR/USD) exchange rate using the European Central Bank (ECB) fixing series with only autoregressive terms as inputs. This is done by benchmarking HONNs with three other different neural network designs representing the classic MLP, a Psi Sigma network and a RNN and some traditional techniques, either statistical such as a an autoregressive moving average model (ARMA), or technical such as a moving average concergence/divergence model (MACD), plus a naïve strategy.[17] More specifically, the trading performance of all models is investigated in a forecast and trading simulation on the EUR/USD ECB fixing time series over the period 1999-2007 using the last one and half year for out-of-sample testing, a original feature of this chapter. We use the EUR/USD daily fixing by the ECB as many financial institutions are ready to trade at this level and it is therefore possible to leave orders with a bank for business to be transacted on that basis.

---

[16] This paper has been presented at the Forecasting Financial Markets 2008 conference in Aix-en-Provence (21 to 23 May 2008) and has been accepted for publication in the 'European Journal of Finance'.

[17] In this chaper we do not apply the GM network as the study of probalistic models is beyond the scope of this chapter. In chapter 4, these models were treated as benchmarks and presented as reported in Lindemann *et. al.* (2004).

As it turns out, the MLP does remarkably well and outperforms all other models (even the more sophisticated HONNs and Psi Sigma) in a simple trading simulation exercise. However, when more sophisticated trading strategies using confirmation filters and leverage are applied, the HONN network produces better results and outperforms all other neural network and traditional statistical models in terms of annualised return.

5.1 Introduction

The motivation for this chapter is to investigate the forecasting performance of HONN using as inputs autoregressive terms. In order to achieve this, the trading performance of HONN, Psi Sigma, RNN and MLP networks on the Euro/Dollar (EUR/USD) time series is investigated and is compared with some traditional statistical or technical methods such as an autoregressive moving average (ARMA) model and a moving average convergence/divergence (MACD) model, and a naïve strategy.

The conclusions of this chapter can supplement those of chapter 4 where we conduct a forecasting competition over the Euro/Dollar (EUR/USD) London closing time series for a period of ten years (October 1994 to July 2001) using about the same networks but with multivariate series as inputs.

The main reason behind our decision to use the EUR/USD daily fixing by the ECB is that is possible to leave orders with a bank and trade on that basis. It is therefore a tradable quantity while for example the London closing time

series used in the literature and in the previous chapter are not, as there maybe slide when we come to transact. Moreover, to the best of our knowledge the use of ECB fixing series is an original feature.

As it turns out, the MLP demonstrates a remarkable performance and outperforms all other models in a simple trading simulation exercise. On the other hand, when more sophisticated trading strategies using confirmation filters and leverage are applied, HONNs outperform all models in terms of annualised return. Our conclusion corroborates those of Lindemann *et al.* (2004) and of the previous chapter where HONNs also demonstrate a forecasting superiority on the EUR/USD series over more traditional techniques such as a MACD and a naïve strategy. However, the RNN which in the previous chapter performed remarkably well, show a poor performance in this research: this may be due to their inability to provide good enough results when only autoregressive terms are used as inputs.

The rest of the chapter is organised as follows. In section 5.2 we describe the dataset used for this research and its characteristics. An overview of the statistical techniques is given in section 5.3 while section 5.4 gives the empirical results of all the models considered and investigates the possibility of improving their performance with the application of more sophisticated trading strategies. Section 5.5 provides some concluding remarks.

5.2 The EUR/USD Exchange Rate and Related Financial Data

The European Central Bank (ECB) publishes a daily fixing for selected EUR exchange rates: these reference mid-rates are based on a daily concertation procedure between central banks within and outside the European System of Central Banks, which normally takes place at 2.15 p.m. ECB time. The reference exchange rates are published both by electronic market information providers and on the ECB's website shortly after the concertation procedure has been completed. Although only a reference rate, many financial institutions are ready to trade at the EUR fixing and it is therefore possible to leave orders with a bank for business to be transacted at this level.

The ECB daily fixing of the EUR/USD is therefore a tradable level which makes using it a more realistic alternative to, say, London closing prices and this is the series that we investigate in this chapter[18].

We examined the ECB daily fixing of the EUR/USD since its first trading day on 4 January 1999 until 31 December 2007. The data period is partioned as follows.

Name of period	Trading Days	Beginning	End
Total dataset	2304	4 January 1999	31 December 2007
Training dataset	1921	4 January 1999	30 June 2006
Out-of-sample dataset [Validation set]	383	3 July 2006	31 December 2007

Table 11: The EUR/USD dataset

[18] EUR/USD is quoted as the number of USD per Euro: for example, a value of 1.2657 is USD1.2657 per 1 Euro. We examine the EUR/USD since its first trading day on 4 January 1999, and until 31 December 2007.

The graph below shows the total dataset for the EUR/USD and its upward trend since early 2006.

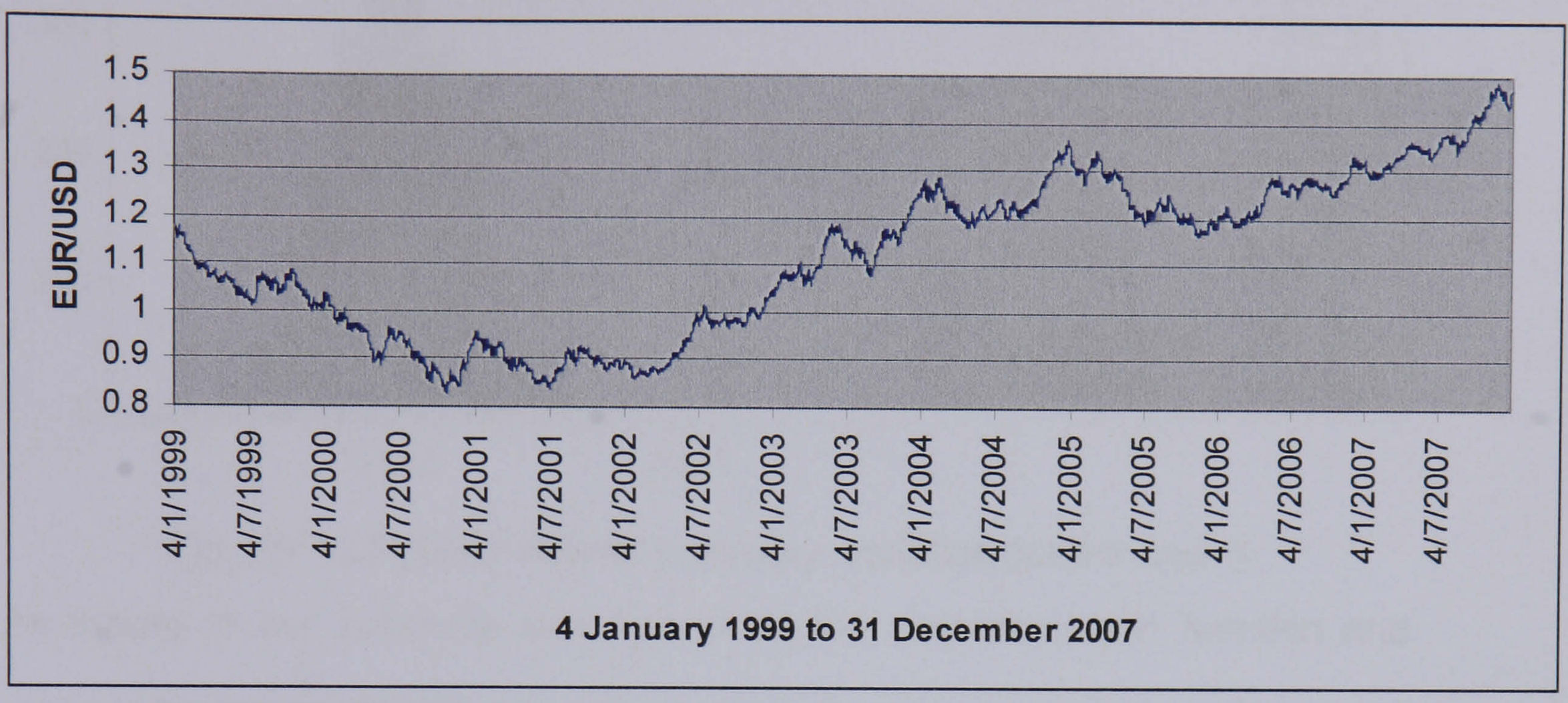
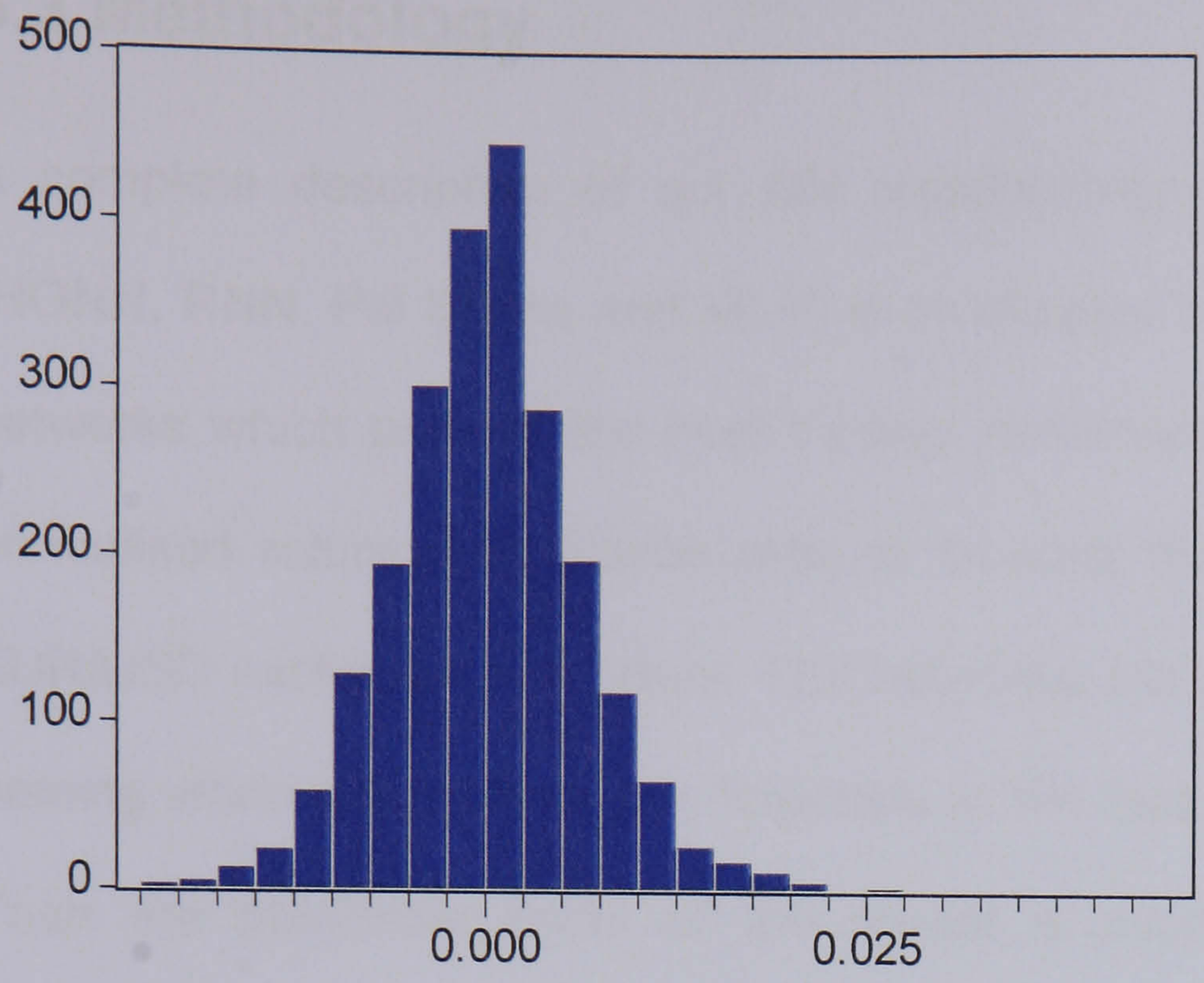


Fig. 12: EUR/USD Frankfurt daily closing prices (total dataset)

The observed EUR/USD time series is non-normal (Jarque-Bera statistics confirmed this at the 99% confidence interval) containing slight skewness and high kurtosis. It is also nonstationary and hence we decided to transform the EUR/USD series into stationary daily series of rates of return[19] using formula [22].

The summary statistics of the EUR/USD returns series reveal a slight skewness and high kurtosis. The Jarque-Bera statistic confirms again that the EUR/USD series is non-normal at the 99% confidence interval.

[19] Confirmation of its stationary property is obtained at the 1% significance level by both the Augmented Dickey Fuller (ADF) and Phillips-Perron (PP) test statistics.



Series: RETURNS
Sample 1 2303
Observations 2303

Mean	0.000115
Median	7.60e-05
Maximum	0.042938
Minimum	-0.022269
Std. Dev.	0.006177
Skewness	0.286813
Kurtosis	4.622358
Jarque-Bera	284.1414
Probability	0.000000

Fig. 13: EUR/USD returns summary statistics (total dataset)

As inputs to our networks and based on the autocorrelation function and some ARMA experiments we selected a set of autoregressive and moving average terms of the EUR/USD exchange rate returns and the 1-day Riskmetrics volatility series.

Number	Variable	Lag
1	EUR/USD exchange rate return	1
2	EUR/USD exchange rate return	2
3	EUR/USD exchange rate return	3
4	EUR/USD exchange rate return	7
5	EUR/USD exchange rate return	11
6	EUR/USD exchange rate return	12
7	EUR/USD exchange rate return	14
8	EUR/USD exchange rate return	15
9	EUR/USD exchange rate return	16
10	Moving Average of the EUR/USD exchange rate return	15
11	Moving Average of the EUR/USD exchange rate return	20
12	1-day Riskmetrics Volatility	1

Table 12: Explanatory variables

In order to train our neural networks we further divided our dataset as follows:

Name of period	Trading days	Beginning	End
Total data set	2304	4 January 1999	31 December 2007
Training data set	1537	4 January 1999	31 December 2004
Test data set	384	3 January 2005	30 June 2006
Out-of-sample data set [Validation set]	383	3 July 2006	31 December 2007

Table 13: The neural networks datasets

5.3 Methodology

A complete description of our NN architectures used on this application (HONN, RNN, Psi Sigma and MLP) is on chapter 3. We choose and use the networks which present the best trading performance in-sample in terms of annualised return and Sharpe ratio to forecast the ECB daily fixing of the EUR/USD exchange rate return. Our networks are designed so they will stop training when the profit of our forecasts in the test sub-period is maximized. Then the predictive value of the model is evaluated applying it to the validation dataset (out-of-sample dataset). The characteristics of the networks used on this chapter are on Appendix A.2.1.

The trading strategy applied is simple and identical for all our models: go or stay long when the forecast return is above zero and go or stay short when the forecast return is below zero. Below there is a brief description of our statistical and technical benchmarks namely an autoregressive moving average model (ARMA), a moving average convergence/divergence technical model (MACD) and a naïve strategy.

5.3.1 Benchmark Models

5.3.1.1 Naïve Strategy

The naïve strategy simply takes the most recent period change as the best prediction of the future change, i.e. a simple random walk. The model is defined by:

$$\hat{Y}_{t+1} = Y_t \qquad [24]$$

where Y_t is the actual rate of return at period t

$\hat{Y}_{t+1}$ is the forecast rate of return for the next period

The performance of the strategy is evaluated in terms of trading performance via a simulated trading strategy.

5.3.1.2 Moving Average Convergence/Divergence

The moving average model is defined as:

$$M_t = \frac{\left(Y_t + Y_{t-1} + Y_{t-2} + \ldots + Y_{t-n+1}\right)}{n} \qquad [25]$$

where M_t is the moving average at time t

n is the number of terms in the moving average

Y_t is the actual rate of return at period t

The MACD strategy used is quite simple. Two moving average series are created with different moving average lengths. The decision rule for taking positions in the market is straightforward. Positions are taken if the moving averages intersect. If the short-term moving average intersects the long-term moving average from below a 'long' position is taken. Conversely, if the long-term moving average is intersected from above a 'short' position is taken.

The forecaster must use judgement when determining the number of periods n on which to base the moving averages. The combination that performed best over the in sample sub-period was retained for out-of-sample evaluation.

The model selected was a combination of the EUR/USD and its 24-day moving average, namely n = 1 and 24 respectively or a (1,24) combination. The performance of this strategy is evaluated solely in terms of trading performance.

5.3.1.3 ARMA Model

Autoregressive moving average models (ARMA) assume that the value of a time series depends on its previous values (the autoregressive component) and on previous residual values (the moving average component)[20].

The ARMA model takes the form:

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \varepsilon_t - w_1 \varepsilon_{t-1} - w_2 \varepsilon_{t-2} - \ldots - w_q \varepsilon_{t-q} \quad [26]$$

where

Y_t	is the dependent variable at time t
Y_{t-1}, Y_{t-2}, and Y_{t-p}	are the lagged dependent variable
ϕ_0, ϕ_1, ϕ_2, and ϕ_p	are regression coefficients
ε_t	is the residual term
ε_{t-1}, ε_{t-2}, and ε_{t-p}	are previous values of the residual
w_1, w_2, and w_q	are weights.

Using as a guide the correlogram in the training and the test sub-periods we have chosen a restricted ARMA (6,6) model. All of its coefficients are significant at the 95% confidence interval. The null hypothesis that all coefficients (except the constant) are not significantly different from zero is rejected at the 95% confidence interval (see Appendix A.2.2).

The selected ARMA model takes the form:

---

[20] For a full discussion on the procedure, refer to Box et al., (1994) or Pindyck and Rubinfeld (1998).

$$Y_t = 6.35 \cdot 10^{-5} - 0.688Y_{t-1} - 0.369Y_{t-2} - 0.219Y_{t-7} - 0.400Y_{t-11} - 0.540Y_{t-12} + 0.693\varepsilon_{t-1} + 0.350\varepsilon_{t-2} + 0.249\varepsilon_{t-7} + 0.398\varepsilon_{t-11} + 0.584\varepsilon_{t-12} \quad [27]$$

The model selected was retained for out-of-sample estimation. The performance of the strategy is evaluated in terms of trading performance.

5.3.2 Results

In table 14 below we present the trading performance of our models for the out-of-sample period while Appendix A.2.3 documents our in-sample results. The performance measures are presented in Appendix A.1.1.

		NAIVE	MACD	ARMA	MLP	RNN	HONN	Psi Sigma
Sharpe Ratio	*(excluding costs)*	0.03	0.70	-0.40	1.88	0.60	0.99	1.18
Annualised Volatility	*(excluding costs)*	6.34%	6.38%	6.38%	6.34%	6.34%	6.37%	6.36%
Annualised Return	*(excluding costs)*	0.16%	4.44%	-2.53%	11.91%	3.82%	6.29%	7.53%
Maximum Drawdown	*(excluding costs)*	-7.32%	-4.73%	-10.12%	-5.05%	-4.64%	-4.37%	-4.86%
Positions Taken	*(annualised)*	132	20	189	109	167	80	117

Table 14: Trading performance results

As can been seen the MLPs outperforms all other statistical, technical and neural network in terms of annualised return and Sharpe ratio. Moreover, the major theoretical advantage of Psi Sigma networks, namely their speed, was clearly confirmed as we achieved our results in one half of the time needed to train the MLPs and the HONNs and one tenth of the time needed for the RNNs.

5.4 Trading Costs, Filters and Leverage

Up to now, we have presented the trading results of all our models without considering transaction costs. Since some of our models trade quite often, taking transaction costs into account might change the whole picture.

We therefore introduce transaction costs as well as a filtered trading strategy for each model. The aim is to devise a trading strategy filtering only those trades which have a high probability of being successful. This should help to reduce the negative effect of transaction costs as trades with an expected gain lower than the transaction costs should be omitted.

5.4.1 Transaction Costs

The transaction costs for a tradable amount, say USD 5-10 million, are about 1 pip (0.0001 EUR/USD) per trade (one way) between market makers. But as the EUR/USD time series considered here is a series of middle rates, the transaction cost is one spread per round trip.

With an average exchange rate of EUR/USD of 1.341 for the out-of-sample period, a cost of 1 pip is equivalent to an average cost of 0.008% per position.

5.4.2 Confirmation Filter Strategies

5.4.2.1 Confirmation Filters

We now introduce trading strategies devised to filter out those trades with expected returns below the 0.008% transaction cost. Due to the architecture of all our models, the trading strategy will consist by one single parameter.

Up to now, the trading strategies applied to the models use a zero threshold: they suggest to go long when the forecast is above zero and to go short when the forecast is below zero. In the following, we examine how the models behave if we introduce a threshold *d* around zero (see figure 14) and what happens if we vary that threshold. The filter rule for all our models is presented in figure 14 below.

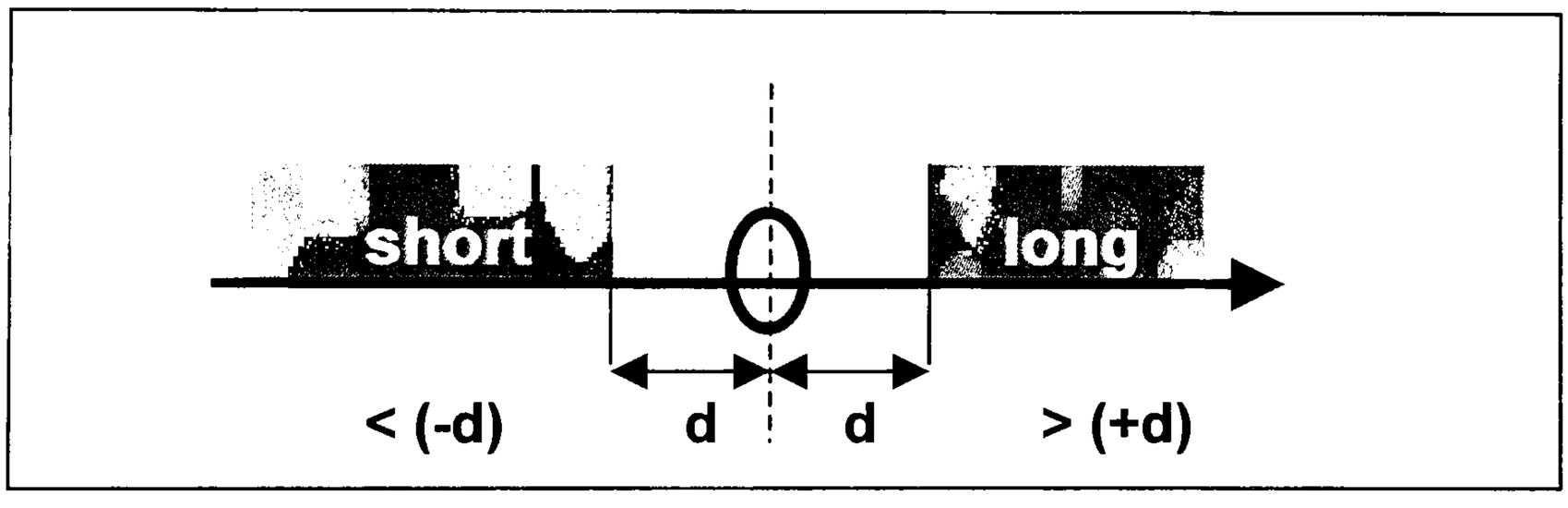


Fig. 14: Filtered trading strategy with one single parameter

5.4.2.2 Empirical Results

The methodology that we follow for the selection of the optimal thresholds is simple. Taking the test period results, we choose the threshold that gives the higher return and Sharpe ratio. Our chosen parameters are presented in the

table below while the detailed results leading to their choice are documented in Appendix A.2.4.

Model	Threshold (d)
Naive	*=0.35*
MA	*=0.00*
ARMA	*=0.10*
MLP	*=0.20*
RNN	*= 0.00*
HONN	*= 0.00*
Psi Sigma	*= 0.30*

Table 15: Chosen parameters for each trading strategy

For the MACD, MLP and the HONN strategies we leave the threshold at zero (d=0.0) since the profit on the test dataset is largest at this value. On the other hand, we selected as threshold the values of 0.35, 0.10, 0.20 and 0.30 for the Naïve, ARMA, RNN and the Psi Sigma models respectively as in these cases the profit in the test sub-period is maximized.

A summary of the out-of-sample trading performance of all models using the selected thresholds and taking into account the transaction costs is on the table below.

	NAIVE	MACD	ARMA	MLP	RNN	HONN	Psi Sigma
Sharpe Ratio (excluding costs)	-0.70	0.70	0.41	1.00	0.60	0.99	0.70
Annualised Volatility (excluding costs)	4.12%	6.38%	3.43%	4.03%	6.34%	6.37%	5.22%
Annualised Return (excluding costs)	-2.90%	4.44%	1.39%	4.02%	3.82%	6.29%	3.63%
Maximum Drawdown (excluding costs)	-7.23%	-4.73%	-3.14%	-2.45%	-4.64%	-4.37%	-5.60%
Positions Taken (annualised)	79	20	80	83	167	80	91
Transaction costs	0.63%	0.16%	0.64%	0.66%	1.33%	0.64%	0.73%
Annualised Return (including costs)	-3.53%	4.28%	0.75%	3.36%	2.48%	5.65%	2.90%

Table 16: Out-of-sample results for the chosen parameters

From the final row of table 16 we can see that, after transaction costs, the HONN network outperforms all the other strategies based on the annualised return. The MACD strategy also performs well and presents the second best performance in terms of annualized return. On the other hand, the MLP networks which performed best without trading filter seem to be unable to fully exploit the introduction of the modified trading strategy. The Psi Sigma which also performed well before the introduction of the trading strategy seems also incapable of exploiting it. However, it is worth mentioning that the time used to derive these results with the Psi Sigma network is half that needed with HONNs and the MLPs and one tenth of that needed with RNNs.

5.4.3 Leverage to Exploit Low Volatility

In order to further improve the trading performance of our models we introduce a "level of confidence" to our forecasts, i.e. a leverage based on the test sub-period. For the naïve model, which presents a negative return we do not apply leverage. The leverage factors applied are calculated in such a way that each model has a common volatility of 10%[21] on the test data set.

The transaction costs are calculated by taking 0.008% per position into account, while the cost of leverage (interest payments for the additional capital) is calculated at 4% p.a. (that is 0.016% per trading day[22]). Our final results are presented in table 17 below.

---

[21] Since most of the models have a volatility of about 10%, we have chosen this level as our basis. The leverage factors retained are given in table 11 below.
[22] The interest costs calculation is in chapter 4 (see footnote 9).

	NAIVE	MACD	ARMA	MLP	RNN	HONN	Psi Sigma
Sharpe Ratio (excluding costs)	-0.70	0.70	0.41	1.00	0.60	0.99	0.70
Annualised Volatility (excluding costs)	4.12%	7.27%	9.67%	5.5%	7.23%	7.26%	8.30%
Annualised Return (excluding costs)	-2.90%	5.06%	3.93%	5.82%	4.35%	7.17%	5.78%
Maximum Drawdown (excluding costs)	-7.23%	-5.39%	-3.58%	-3.55%	-5.30%	-4.98%	-8.90%
Leverage Factor	-	1.14	2.82	1.45	1.14	1.14	1.59
Positions Taken (annualised)	79	20	80	83	167	80	91
Transaction and leverage costs	0.63%	1.02%	11.79%	3.42%	2.19%	1.50%	4.34%
Annualised Return (including costs)	-3.53%	4.04%	-7.86%	2.40%	2.16%	5.67%	1.44%

Table 17: Trading performance - final results[23]

As can be see from the last row of table 17, HONNs continue to demonstrate a superior trading performance. The MACD strategy also performs well and presents the second higher annualised return. In general, we observe that all models expect the HONNs, show an inability to gain any extra profit from the leverage as the increased transaction costs seems to counter any benefits. Again it is worth mentioning, that the time needed to train the Psi Sigma network was considerably shorter compared with that needed for the MLP, the RNN and the HONN networks.

5.5 Concluding Remarks

In this chapter, we apply Multi-layer Perceptron, Recurrent, Higher Order and Psi Sigma neural networks to a one-day-ahead forecasting and trading task of the Euro/Dollar (EUR/USD) exchange rate using the European Central Bank

[23] Not taken into account the interest that could be earned during times where the capital is not traded (non-trading days) and could therefore be invested.

(ECB) fixing series with only autoregressive terms as inputs. We use a naïve, a MACD and an ARMA model as benchmarks. We develop these different prediction models over the period January 1999 - June 2006 and validate their out-of-sample trading efficiency over the period from July 2006 through December 2007.

The MLPs demonstrated the higher trading performance in terms of annualised return and Sharpe ratio before transaction costs and elaborate trading strategies are applied. When refined trading strategies are applied and transaction costs are considered the HONNs manage to outperform all other models achieving the highest annualised return. On the other hand, the RNNs and the Psi Sigma models, which in previous applications over the EUR/USD dollar exchange rate performed remarkably well in the previous chapter seem to have a difficulty in providing good forecasts when autoregressive series are only used as inputs.

CHAPTER 6

The Robustness of Neural Networks for Modelling and Trading the EUR/USD Exchange Rate at the ECB Fixing[24]

Overview

The motivation for this chapter is to investigate the use, the stability and the robustness of HONNs when applied to the task of forecasting and trading the Euro/Dollar (EUR/USD) exchange rate using the European Central Bank (ECB) fixing series with only autoregressive terms as inputs. This is done by benchmarking the forecasting performance of HONNs with two different neural network designs representing a Recurrent Network (RNN) and the classic Multilayer Percepton (MLP) and with some traditional techniques, either statistical such as a an autoregressive moving average model (ARMA), or technical such as a moving average convergence/divergence model (MACD), plus a naïve strategy. More specifically, the trading performance of all models is investigated in a forecast and trading simulation on the EUR/USD ECB fixing time series over the period January 1999-August 2008 using the last eight months for out-of-sample testing. Our results in terms of their robustness and stability are compared with our empirical work in chapter 5 where we apply the same models and follow the same methodology forecasting the same series, using as out-of-sample the period from July 2006 to December 2007.

---

[24] This paper has been accepted for publication in the '*Journal of Derivatives & Hedge Funds*'.

As it turns out, the HONN and the MLP networks present a robust performance and do remarkably well outperforming all other models in a simple trading simulation exercise in both chapters. Moreover, when transaction costs are considered and leverage is applied, the same networks continue to outperform all other neural network and traditional statistical models in terms of annualised return, a robust and stable result as it is identical to the one obtained from the authors in their previous work, studying a different period for the series.

6.1 Introduction

The motivation of this chapter is to investigate the stability and the robustness of the forecasting performance of HONNs. This is done by benchmarking their performance with two different neural network architectures representing a Multilayer Percepton (MLP) and a Recurrent Neural Network (RNN).[25] Their trading performance on the Euro/Dollar (EUR/USD) time series is investigated and is compared with some traditional statistical or technical methods such as an autoregressive moving average (ARMA) model or a moving average convergence/divergence (MACD) model, and a naïve strategy. In terms of the stability and robustness of our findings, we compare them with our conclusions of chapter 5 where we apply the same models and follow the same methodology to forecast the same series, using however a different out-of-sample period. Concerning the inputs of our neural networks, we use the exact same selection of inputs and lags with chapter 5. Similarly, our MACD

[25] As Psi Sigma present a similar or lower performance than HONNs in the previous chapters and seem to have a difficulty with autoregressive series as inputs we decided not to use them in this application.

model is identical with the ones used in the previous chapter while on the other hand our ARMA model is different, as we need all coefficients to be significant in the new in-sample period.

As it turns out, the MLP and HONN demonstrate a remarkable performance and outperform the other models in a simple trading simulation exercise. Moreover, when transaction costs are considered and leverage is applied the MLP and HONN models continue to outperform all other neural network and traditional statistical models in terms of annualised return. As these results are identical to those of chapter 5, we can argue that the forecasting superiority of the HONN and the MLP is stable and robust over time. In terms of the RNN, their poor performance in this research may be due to their inability to provide good enough results when only autoregressive terms are used as inputs.

The rest of the chapter is organised as follows. In section 6.2 we describe the extended dataset used for this research and its characteristics. An overview of the methodology and the new ARMA model is given in section 6.3 while section 6.4 gives the empirical results of all the models considered and investigates the possibility of improving their performance with the application of more sophisticated trading strategies. Section 6.5 provides some concluding remarks.

6.2 The EUR/USD Exchange Rate and Related Financial Data

The series under study is the ECB fixing of the EUR/USD exchange rate as described in chapter 5.2. The data period is partioned as follows.

Name of period	Trading days	Beginning	End
Total dataset	2474	4 January 1999	29 August 2008
In-sample dataset	2304	4 January 1999	31 December 2007
Out-of-sample dataset [Validation set]	170	2 January 2008	29 August 2008

Table 18: The EUR/USD dataset

The graph below shows the total dataset for the EUR/USD and its downward trend for the beginning of 2008.

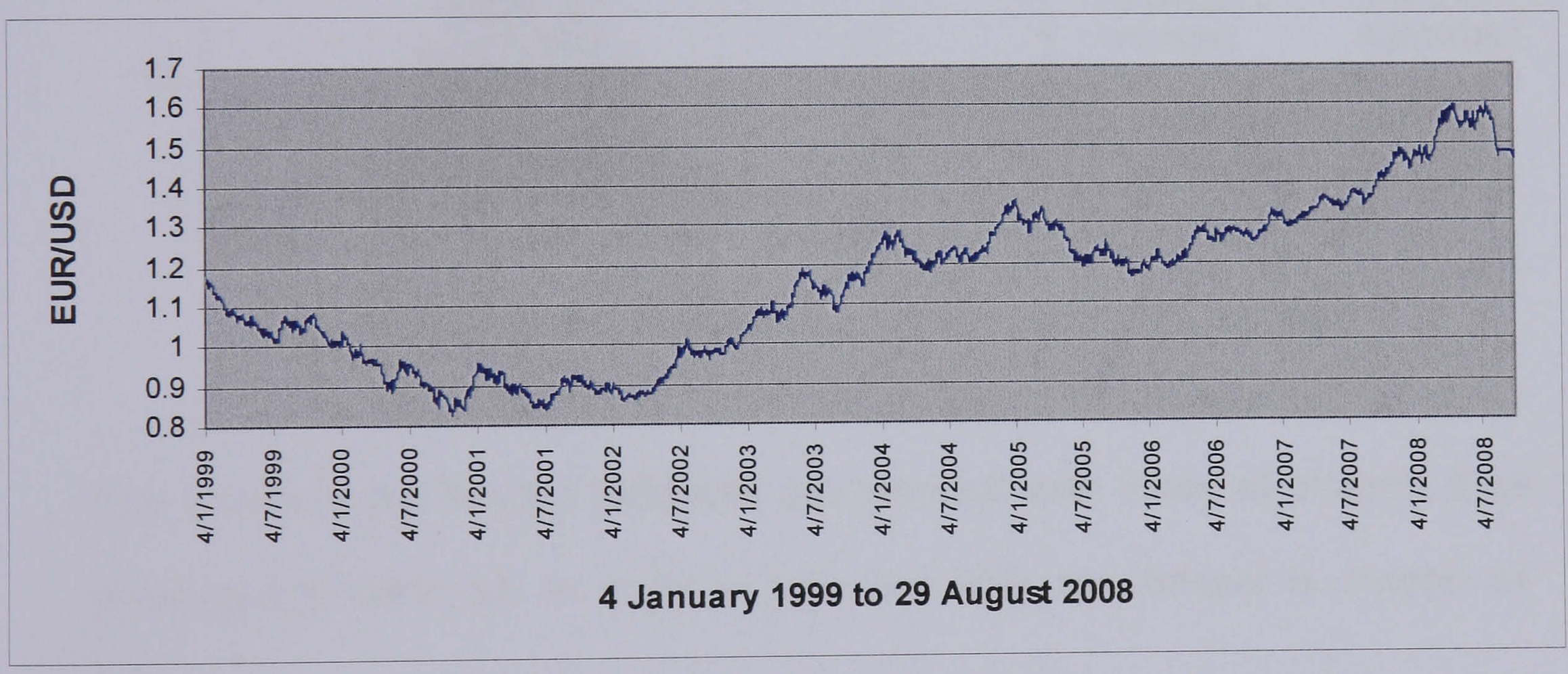


Fig. 15: EUR/USD Frankfurt ECB fixing prices (total dataset)

The observed EUR/USD time series is non-normal (Jarque-Bera statistics confirm this at the 99% confidence interval) containing slight skewness and high kurtosis. It is also nonstationary and hence we decided to transform the

EUR/USD series into a stationary daily series of rates of return[26] using formula [22].

The summary statistics of the EUR/USD returns series reveal a slight skewness and high kurtosis. The Jarque-Bera statistic confirms again that the EUR/USD series is non-normal at the 99% confidence interval.

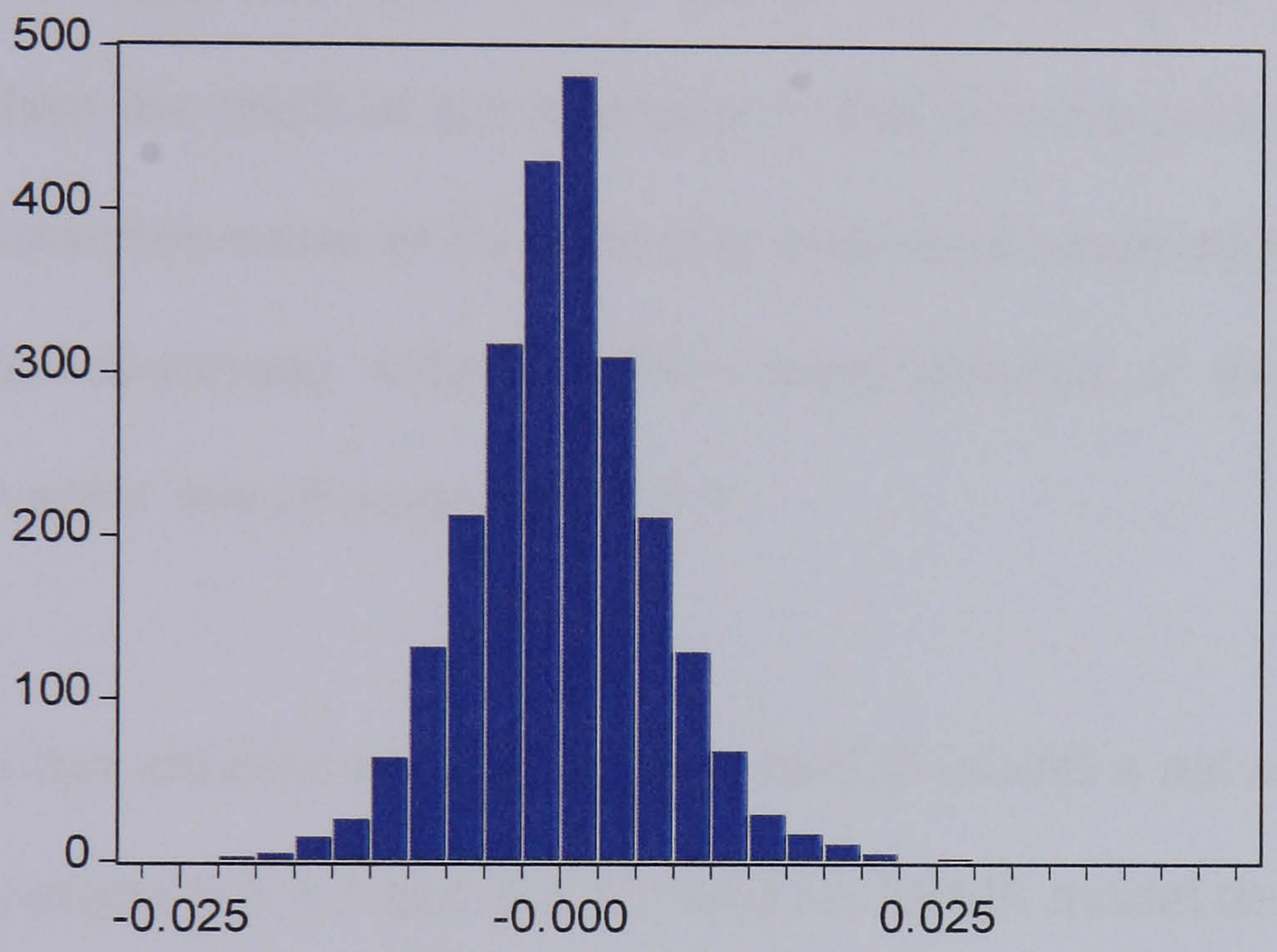


Fig. 16: EUR/USD returns summary statistics (total dataset)

The inputs of our Neural Networks are identical with those of chapter 5 as presented in table 12. In order to train our NNs are dataset is divided as follows.

Name of period	Trading days	Beginning	End
Total data set	2474	4 January 1999	29 August 2008
Training data set	1794	4 January 1999	31 December 2005
Test data set	510	2 January 2006	31 December 2007
Out-of-sample data set [Validation set]	170	2 January 2008	29 August 2008

Table 19: The neural networks datasets

[26] Confirmation of its stationary property is obtained at the 1% significance level by both the Augmented Dickey Fuller (ADF) and Phillips-Perron (PP) test statistics.

6.3 Methodology

A complete description of our NN architectures used on this application (HONN, RNN and MLP) is on charter 3. We choose and use the networks which present the best trading performance in-sample in terms of annualised return and Shapre ratio to forecast the ECB daily fixing of the EUR/USD exchange rate return. Our networks are designed so they will stop training when the profit of our forecasts in the test sub-period is maximized. Then the predictive value of the model is evaluated applying it to the validation dataset (out-of-sample dataset). The characteristics of the networks used on this chapter are on Appendix A.3.1.

In this chapter we also apply a MACD model a naive strategy as described in sections 5.3.1.1 and 5.3.1.2 and an ARMA model described below.

The trading strategy applied is simple and identical for all our models: go or stay long when the forecast return is above zero and go or stay short when the forecast return is below zero.

6.3.1 Benchmark Models

6.3.1.1 ARMA

Autoregressive moving average models (ARMA) assume that the value of a time series depends on its previous values (the autoregressive component) and on previous residual values (the moving average component).

The ARMA model takes the form:

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \varepsilon_t - w_1 \varepsilon_{t-1} - w_2 \varepsilon_{t-2} - \ldots - w_q \varepsilon_{t-q} \quad [26]$$

where

Y_t	is the dependent variable at time t
Y_{t-1}, Y_{t-2}, and Y_{t-p}	are the lagged dependent variable
ϕ_0, ϕ_1, ϕ_2, and ϕ_p	are regression coefficients
ε_t	is the residual term
ε_{t-1}, ε_{t-2}, and ε_{t-p}	are previous values of the residual
w_1, w_2, and w_q	are weights.

Using as a guide the correlogram in the training and the test sub-periods we have chosen a restricted ARMA (11,11) model. All of its coefficients are significant at the 95% confidence interval. The null hypothesis that all coefficients (except the constant) are not significantly different from zero is rejected at the 95% confidence interval (see Appendix A.3.2).

The selected ARMA model takes the form:

$$\begin{aligned} Y_t = {} & 12.8 \cdot 10^{-5} - 1.217 Y_{t-1} - 0.478 Y_{t-2} - 0.140 Y_{t-7} + 0.197 Y_{t-11} \\ & + 1.214 \varepsilon_{t-1} + 0.474 \varepsilon_{t-2} + 0.152 \varepsilon_{t-7} - 0.218 \varepsilon_{t-11} \end{aligned} \quad [28]$$

The model selected was retained for out-of-sample estimation. The performance of the strategy is evaluated in terms of trading performance.

6.3.2 Results

In the table below we present the trading performance of our models for the out-of-sample period while Appendix A.3.3 documents our in-sample results. The performance measures are presented in Appendix A.1.1.

		NAIVE	MACD	ARMA	MLP	RNN	HONN
Sharpe Ratio	*(excluding costs)*	1.28	-0.02	-0.45	0.74	0.40	0.86
Annualised Volatility	*(excluding costs)*	9.51%	9.54%	9.54%	9.18%	9.54%	9.49%
Annualised Return	*(excluding costs)*	12.14%	-0.19%	-4.32%	6.82%	3.83%	8.16%
Maximum Drawdown	*(excluding costs)*	-3.69%	-9.85%	-11.39%	-5.78%	-5.91%	-7.21%
Positions Taken	*(annualised)*	108	28	185	107	93	170

Table 20: Trading performance results

As can been seen the naïve strategy outperforms all other models by far. These results are surprising not only because the simplicity of the naïve model but also based on the training sub-period results (see Appendix A.3.c). There the naïve strategy presents an annualised return of -1.07% and a Sharpe ratio of -0.10. Moreover, with a closer look over the returns in our out-of-sample period we observe that the positive and the negative returns are clustered. In order to verify that the sequence of signs of the returns in the validation period is not random, we conduct the Wald-Wolfowitz or runs test for randomness. The test confirms that the sequence of signs is not random at the 99% and 95% confidence interval. So as the naïve strategy is only using as a forecast for tomorrow today's return, it is able to exploit this phenomenon and present a remarkable performance. This anomaly was not present in previous years and that is why the performance of the naïve strategy in-sample is so much worse. In the circumstances, this phenomenon is accidental and we have no reasons to believe that it should continue in the future and we thus discard the results of the naïve strategy from our conclusions.

Concerning the rest models, we can see that the HONN performs significantly better than the RNN and the traditional MLP models.

6. 4 Trading Costs, Leverage and Robustness

Next we introduce transaction costs as well as a leverage for each of our models. Moreover, we examine the robustness of our models by examining and comparing our results with the conclusions of chapter 5 where we studied the same series with the same models but over a different period. The out-of-sample period here is 2 January 2008 to 29 August 2008 while in the previous charter was 3 July 2006 to 31 December 2007. The purpose of this test is to validate the robustness of our models through time and to provide concrete empirical evidence of the forecasting power of our models.

6.4.1 Transaction Costs

The transaction costs for a tradable amount, say USD 5-10 million, are about 1 pip (0.0001 EUR/USD) per trade (one way) between market makers. But as the EUR/USD time series considered here is a series of middle rates, the transaction cost is one spread per round trip.

With an average exchange rate of EUR/USD of 1.341 for the out-of-sample period, a cost of 1 pip is equivalent to an average cost of 0.008% per position.

In the table below we present the performance of our models after transaction costs are considered.

		MACD	ARMA	MLP	RNN	HONN
Sharpe Ratio	*(excluding costs)*	-0.02	-0.45	0.74	0.40	0.86
Annualised Volatility	*(excluding costs)*	9.54%	9.54%	9.18%	9.54%	9.49%
Annualised Return	*(excluding costs)*	-0.19%	-4.32%	6.82%	3.83%	8.16%
Maximum Drawdown	*(excluding costs)*	-9.85%	-11.39%	-5.78%	-5.91%	-7.21%
Positions Taken	*(annualised)*	28	185	107	93	170
Transaction Costs		0.20%	1.30%	0.75%	0.65%	1.19%
Annualised Return	*(including costs)*	-0.39%	-5.62%	6.07%	3.18%	6.97%

Table 21: Out-of-sample trading performance results with transaction costs (02/01/08-29/08/08)

We observe that although the HONN model presents higher transaction costs, it continues to outperform the other models in terms of annualised return. The MLP comes second while the RNN demonstrates a third best performance. On the other hand, the ARMA and MACD models have a rather disappointing performance as they both present negative annualised returns.

In the table below we present the empirical results of our models for the out-of-sample period studied in the previous charter, 3 July 2006 to 31 December 2007.

		MACD	ARMA	MLP	RNN	HONNs
Sharpe Ratio	*(excluding costs)*	0.70	-0.40	1.88	0.60	0.99
Annualised Volatility	*(excluding costs)*	6.38%	6.38%	6.34%	6.34%	6.37%
Annualised Return	*(excluding costs)*	4.44%	-2.53%	11.91%	3.82%	6.29%
Maximum Drawdown	*(excluding costs)*	-4.73%	-10.12%	-5.05%	-4.64%	-4.37%
Positions Taken	*(annualised)*	20	189	109	167	80
Transaction Costs		0.16%	1.51%	0.87%	1.33%	0.64%
Annualised Return	*(including costs)*	4.28%	-4.04%	11.04%	2.48%	5.65%

Table 22: Out-of-sample trading performance results with transaction costs (03/07/06-31/12/07)[27]

We observe that in both periods the MLP and HONN models clearly outperform the other strategies. In the latest period the HONN model has a better performance while one and a half year before the MLP presented better

[27] The forecasted series presented here are the same used in charter 5. Because in both our previous applications confirmation filters deteriorate the trading performance of our models we decided not to apply them in this study.

results. On the other hand, the ARMA model in both periods presents a rather disappointing trading performance. This empirical evidence allows us to argue that HONNs and MLPs have a consistent and better performance than the RNN, MACD and ARMA models in forecasting the ECB daily fixing of the EUR/USD.

6.4.2 Leverage to Exploit Low Volatility

In order to further improve the trading performance of our models we introduce a "level of confidence" to our forecasts, i.e. a leverage based on the test sub-period that takes into account the low volatility of the trading performance of our models. For the ARMA and the MACD models, which show a negative return we do not apply leverage. The leverage factors applied are calculated with the same way as in the previous chapter.

The transaction costs are calculated by taking 0.007% per position into account, while the cost of leverage (interest payments for the additional capital) is calculated at 4% p.a. (that is 0.016% per trading day). Our results are presented on the table 23 below.

		MACD	ARMA	MLP	RNN	HONN
Sharpe Ratio	*(excluding costs)*	-0.02	-0.45	0.74	0.40	0.86
Annualised Volatility	*(excluding costs)*	9.54%	9.54%	10.01%	10.02%	9.96%
Annualised Return	*(excluding costs)*	-0.19%	-4.32%	7.43 %	4.02%	8.56%
Maximum Drawdown	*(excluding costs)*	-9.85%	-11.39%	-6.30%	-6.21%	-7.57%
Positions Taken	*(annualised)*	28	185	107	93	170
Leverage		-	-	1.09	1.05	1.05
Transaction Costs		0.20%	1.30%	0.99%	0.79%	1.33%
Annualised Return	*(including costs)*	-0.39%	-5.62%	6.44%	3.23%	7.23%

Table 23: Trading performance - final results[28] (02/01/08-29/08/08)

[28] Not taken into account the interest that could be earned during times where the capital is not traded (non-trading days) and could therefore be invested

As can be seen from the last row of table 23, the HONN model continues to demonstrate a superior trading performance. Similarly, the MLP and the RNN continue to perform well and present the second and the third highest annualised return respectively. In general, we observe that all models where leverage was applied were able to exploit it and increase their trading performance despite the higher transaction costs.

The performance of our models for a different out-of-sample period, 3 July 2006 to 31 December 2007, is given in table 24 below.

		MACD	**ARMA**	**MLP**	**RNN**	**HONN**
Sharpe Ratio	*(excluding costs)*	0.70	-0.40	1.88	0.60	0.99
Annualised Volatility	*(excluding costs)*	7.27%	6.38%	7.16%	7.23%	7.26%
Annualised Return	*(excluding costs)*	5.06%	-2.53%	13.45%	4.35%	7.17%
Maximum Drawdown	*(excluding costs)*	-5.39%	-10.12%	-5.70%	-5.30%	-4.98%
Positions Taken	*(annualised)*	20	189	109	167	80
Leverage		1.14	-	1.13	1.14	1.14
Transaction Costs		1.02%	1.51%	1.67%	2.19%	1.50%
Annualised Return	*(including costs)*	4.04%	-4.04%	11.78%	2.16%	5.67%

Table 24: Trading performance - final results[29] (03/07/06-31/12/07)

We note that in both periods the MLP and the HONN models continue to outperform the other models as they are able to exploit the leverage and present an increased trading performance in both out-of-sample periods. So even if the ranking of these two models is different in the two out-of-sample forecasting periods retained, they clearly outperform all other models in all cases something that allows us to argue with confidence about their forecasting superiority and their stability and robustness through time. On the

[29] Not taken into account the interest that could be earned during times where the capital is not traded (non-trading days) and could therefore be invested.

other hand, the RNN model was not able to exploit the extra memory inputs in their architecture and presents rather disappointing results. Moreover, the time spent to derive the RNN results is five times longer than the time needed with the HONN and the MLP models. Similarly, the MACD and the ARMA models present a very weak forecasting power even though their training sub-period performance was promising (see Appendix A.3.3).

6.5 Concluding Remarks

In this chapter, we apply Multi-layer Perceptron, Recurrent, and Higher Order neural networks to a one-day-ahead forecasting and trading task of the EUR/USD exchange rate using the European Central Bank (ECB) fixing series with only autoregressive terms as inputs. We use a naïve, a MACD and an ARMA model as benchmarks. Our aim is not only to examine the forecasting and trading performance of our models but also to see if this performance is stable and robust through time. In order to do so, we develop these different prediction models over the period January 1999 - December 2007 and validate their out-of-sample trading efficiency over the following period from January 2008 through August 2008. To examine the robustness and the stability of our models we compare our results with those from chapter 5 where using the same models and the exact same selection of autoregressive terms as inputs to the neural networks, but with an out-of-sample period between July 2006 and December 2007.

As it turns out, the MLP and HONN models clearly outperform the other models in both out-of-sample periods in terms of annualised return. Our conclusions are the same even after we introduced transaction costs and a leverage to exploit the low volatility of the trading performance of those models. This enables us to conclude with confidence over their forecasting superiority and their stability and robustness through time. On other hand, the RNN model seems to have a difficulty in providing good forecasts when only autoregressive series are used as inputs. Similarly, the ARMA and the MACD models present low or even negative annualised returns in this application despite their satisfactory training sub-period performance.

CHAPTER 7

Modelling and Trading the Realised Volatility of the FTSE100 Futures with Higher Order Neural Networks[30]

Overview

The motivation for this chapter is to investigate the use of Higher Order Neural Networks (HONNs), when applied to the task of forecasting and trading the 21-day ahead realised volatility of the FTSE 100 futures index. This is done by benchmarking their results with those of two different neural network designs, the Multi-Layer Percepton (MLP) and the Recurrent Neural Network (RNN), along with a traditional technique, Riskmetrics. More specifically, the forecasting and trading performance of all models is examined over the eight FTSE 100 futures maturities of the period 2007-2008 using the realised volatility of the last 21 trading days of each maturity as our out-of-sample target. The statistical evaluation of our models is done by using a series of measures such as the the mean absolute error (MAE), the mean absolute percentage error (MAPE), the root mean squared error and the Theil-U statistic. Then we apply a simple trading strategy to exploit our forecasts based on trading at-the-money (ATM) calls options on FTSE 100 futures.

As it turns out, HONNs demonstrate a remarkable performance and outperform all other models not only in terms of statistical accuracy but also in terms of trading efficiency. We also note that both the RNNs and MLPs

---

[30] This paper has been presented at the Forecasting Financial Markets 2009 conference in Luxemburg (27 to 29 May 2009) and is currently being reviewed by '*Studies in Nonlinear Dynamics & Econometrics*'.

provide sufficient results in the trading application in terms of cumulative profit and average profit per trade.

7.1 Introduction

The motivation for this chapter is to investigate the use of Higher Order Neural Networks (HONNs), when applied to the task of forecasting and trading the 21-day ahead annualised volatility of the FTSE 100 futures index. This is done by benchmarking their results with those of two different neural network designs, the Multi-Layer Percepton (MLP) and the Recurrent Neural Network (RNN), along with a traditional technique, RiskMetrics.

In order to evaluate statistically our models we compute the mean absolute error (MAE), the mean absolute percentage error (MAPE), the root mean squared error and the Theil-U statistic. Then we compare our forecasts with the relevant Black-Scholes (1973) market derived implied volatilities and we apply a simple trading strategy based on trading at-the-money (ATM) calls options on FTSE 100 futures. As it turns out, HONNs demonstrate a remarkable performance and outperform all other models not only in terms of statistical accuracy but also in terms of trading efficiency. We also note that both the RNNs and MLPs provide satisfactory results in the trading application in terms of cumulative profit and average profit per trade.

The rest of the chapter is organised as follows. In section 7.2, we describe the dataset used for this research and its characteristics. An overview of the RiskMetrics volatility is given in section 7.3. Section 7.4 gives the empirical

results of all the models in terms of their statistical accuracy and trading efficiency while section 6 provides some concluding remarks.

7.2 The FTSE 100 Futures and Related Financial Data

Our benchmark test is to forecast the 21-day ahead realised volatility of the FTSE 100 futures returns. For the FTSE 100 futures there are four delivery months: March, June, September and December. Trading ceases on the third Friday of the delivery month of the contract as soon as reasonably practicable after 10:15 (London time) once the Expiry Value of the Index has been determined. In our application we are examining the 8 different futures contracts which are expiring in 2007 and 2008. For each of the 8 contracts we have their closing prices for almost a year before their expiration.

More specifically we are examining the 8 different FTSE100 futures contracts presented on the table below.

Delivery month of the contract	Available trading days from Datastream
March 2007	260
June 2007	265
September 2007	265
December 2007	264
March 2008	264
June 2008	265
September 2008	260
December 2008	264

Table 25: FTSE 100 futures contracts

In the figure below we present the FTSE 100 index level for 1 January 2006 to 31 December 2008 while on Appendix A.4.1 we present for the same period the 21-days realised standard deviations of the FTSE 100 returns.

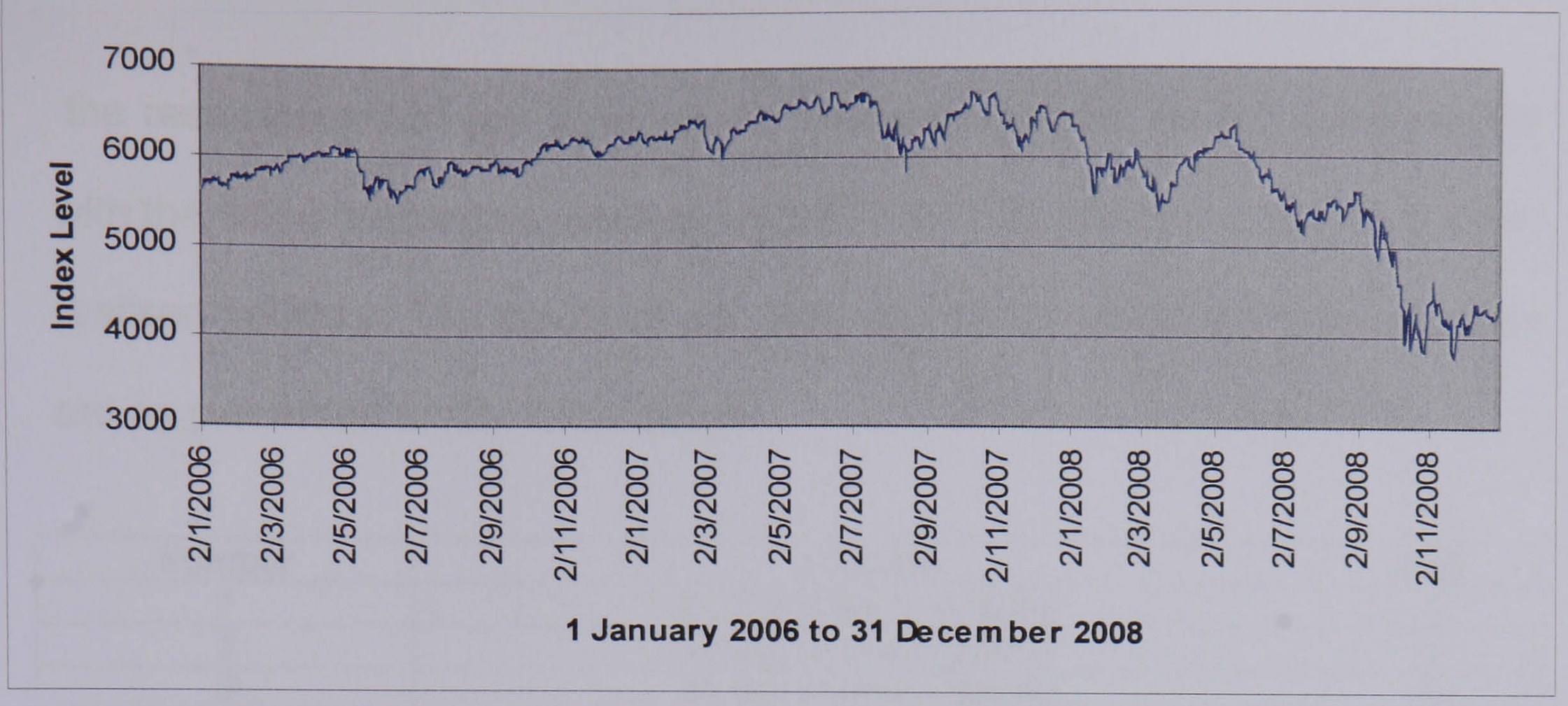


Fig. 17: FTSE 100 index closing prices in pounds

In all cases, we used the last 21 days as out-of-sample dataset and the rest of the days as in-sample dataset. All the eight series are non-normal (Jarque-Bera statistics confirm this at the 99% confidence interval) containing skewness and high kurtosis and are nonstationary. For the purpose of our research we transform them to into stationary series of returns using formula [22].

The summary statistics of the eight futures returns series reveal that they contain a slight skewness and high kurtosis. The Jarque-Bera statistic confirms again that all series are not normally distributed at the 99% confidence interval.

In the absence of any formal theory behind the selection of the inputs of a neural network and due

the restrictions that you have on our available dataset we fed our networks with the first 4 autoregressive lags of the FTSE 100 returns 21-day annualised realised volatility. The inputs of our NNs models for each of the 8 maturities are as presented on the table below.

Number	Variable	Lag
1	21-day realised volatility	*1*
2	21-day realised volatility	*2*
3	21-day realised volatility	*3*
4	21-day realised volatility	*4*

Table 26: Explanatory variables for the MLPs, RNNs and HONNs models

7.3 Methodology

In this chapter we benchmark HONNs with two more traditional NNs designs, a MLP and RNN model, and a statistical technique such as the RiskMetrics volatility in the task of forecasting the 21-day ahead annualised volatility of the FTSE 100 futures index. An estimation of the realised 21-day ahead annualised volatility, which we are interested in forecasting as accurately as possible, can be given by the expression:

$\sigma_{t+1} = \frac{1}{21}\sum_{t-20}^{t}\sqrt{252}|R_t|$ [29] where R_t is the realised daily return of the FTSE 100 futures index.

A complete description of our NN architectures used on this application (HONN, RNN and MLP) is on chapter 3. Since the starting point for each network is a set of random weights, forecasts can differ slightly between networks. In order to eliminate any variance between our NNs forecasts, we used the average of a committee of 20 NNs which presents the better statistical performance in-sample. The characteristics of the NNs used in this

chapter are presented in Appendix A.4.2. The target value in all our networks is the 21-day ahead estimated annualised volatility of our series as defined from equation [29]. In the section below we present our Riskmetrics volatility benchmark.

7.3.1 RiskMetrics Volatility

The RiskMetrics volatility model is treated as a benchmark model owing to its simplicity and popularity in volatility forecast. Derived from the GARCH(1,1) model, but with fixed coefficients, RiskMetrics volatility is calculated using the standard formula:

$RMVOL_t^2 = b\sigma_{t-1}^2 + (1-b)R_t^2$ [30] where σ_t^2 is the futures index variance at time t, R_t^2 is the futures index squared return at time t and b=0.94 for daily data. In this chapter we use RiskMetrics volatility to forecast 21-day ahead annualised volatility for the out-of-sample period. The RiskMetrics volatility is calculated from equation [30] and then we use equation [31] below to calculate the 21-step ahead annualised volatility forecast: $\hat{\sigma}_{t+1} = RMVOL_t\sqrt{252}$ [31].

7.4 Empirical Results

7.4.1 Statistical Performance

As it is standard in the literature, in order to evaluate statistically our forecasts, the RMSE, the MAE, the MAPE and the Theil-U statistics are computed. The RMSE and MAE statistics are scale-dependent measures but give a basis to compare volatility forecasts with the realised volatility while the MAPE and the Theil-U statistics are independent of the scale of the variables. In particular,

the Theil-U statistic is constructed in such a way that it necessarily lies between zero and one, with zero indicating a perfect fit. A more detailed description of these measures can be found on Pindyck and Rubinfeld (1998), Theil (1966) and Dunis and Chen (2005) while their mathematical formulas are on Appendix A.4.3 For all four of the error statistics retained (RMSE, MAE, MAPE and Theil-U) the lower the output, the better the forecasting accuracy of the model concerned. In the table below we present our results for the in-sample period while our results for the out-of-sample period are on Appendix A.4.4.

	RiskMetrics	**MLPs**	**RNNs**	**HONNs**
MAE	0.0730	0.0563	0.0347	0.0291
MAPE	44.43%	30.51%	15.95%	16.49%
RMSE	0.0953	0.0824	0.0706	0.0436
Theil-U	0.2140	0.1919	0.1641	0.0992

Table 27: In-sample statistical performance

As it can be seen from tables 27 and 53 , HONNs outperform all other models and present the most accurate forecasts in statistical terms in both in-sample and out-of-sample periods. It seems that their ability to capture higher order correlations gave them a considerable advantage compared to the other models. On the other hand, RNNs come second and MLPs come third in our statistical evaluation in both periods while the RiskMetrics model presents the least accurate forecasts. Furthermore, we observe that the statistical performance of our NNs is better in-sample than out-of-sample, something that was expected. Moreover, it is worth noting that the time that we need to train our HONNs was less than the time needed for the RNNs and the MLPs.

7.4.2 Out-of-Sample Trading Performance

However, as we are interested to test our models not only in terms of statistical accuracy but also in terms of trading efficiency we apply a realistic trading strategy once our volatility forecasts substantially differ from the implied volatilities of the ATM call options on FTSE 100 futures. As can been seen from the table below, the actual option prices of ATM call options on FTSE 100 futures, 21 days before the expiration of the underlying future contract, considerably differ in some maturities from the Black-Scholes generated option prices if in place of the implied volatility we put our relevant 21–steps ahead volatility forecasts.

	Actual	RiskMetrics	MLPs	RNNs	HONNs
March 2007	64.5	188.5	67	62	67
June 2007	104	73	90	86	87
September 2007	168.5	145	130	187	128
December 2007	179	140	175.5	172	175
March 2008	178	183	172.5	168	164
June 2008	128	115	111	107	99
September 2008	127.5	126	139	145	108
December 2008	288	223	279	225	215

Table 28: Actual and derived option premia in pounds

Our aim is to exploit the differences between the implied volatility and our forecasted volatility by identifying mispriced call options. In order to do so, we are going to compare the Black-Scholes derived implied volatility of ATM call options 21 days before their expiration with our relevant 21-day ahead annualised volatility forecasts[31]. If the difference in absolute terms of the forecasted volatility from the prevailing implied volatility is bigger than a given

[31] We are aware that because of the risk premium the implied volatilities will differ from the relevant realised volatilities at any time point. However, it is beyond the scope of this application to identify and quantify the size and the effect that the risk premium has in our application.

threshold and the forecasted volatility is bigger than the implied, we will buy the ATM call. If the implied is bigger, we will sell the ATM call. In all the other cases that the difference in absolute terms between the forecasted and the implied is smaller than the threshold we will not take a position in the market. In our application, we consider 4 thresholds: 0.5%, 1%, 1.5% and 2%. In the table below we show the difference between our forecasts and the relevant implied volatilities.

	RiskMetrics	**MLPs**	**RNNs**	**HONNs**
March 2007	17.36%	0.46%	-0.35%	0.38%
June 2007	-4.35%	-0.49%	-1.64%	-1.62%
September 2007	-5.37%	-7.08%	2.67%	-7.37%
December 2007	-5.63%	-0.51%	-0.97%	-0.60%
March 2008	0.86%	-0.84%	-1.47%	-2.07%
June 2008	-1.84%	-1.93%	-3.03%	-4.26%
September 2008	-0.26%	1.90%	2.87%	-3.38%
December 2008	-11.79%	-1.63%	-11.59%	-14.81

Table 29: Difference between forecasted and implied volatilities

In terms of our exit rules, a position in a call is held for 10 days after it is firstly initiated. This is happening, because of the drop in time value during the life of an option. Moreover, as we are trading based on volatility forecasts we are interested during our trading our calls to be as near ATM as possible. Thus avoiding the effect that the underlying market fluctuations has on their price. As an optimum period that a call will continue to be near ATM, we choose the 10 days. The transactions costs for one call are assumed to be 3.40£ per

round trip[32]. The trading performance of our models for each of the 4 different thresholds is presented on the tables below.

	RiskMetrics	MLPs	RNNs	HONNs
Cumulative Profit	-145.92%	-24.81%	-6.49%	-4.23%
Trades	7	6	7	7
Buys/Sells	2/5	1/5	2/5	1/6
Profitable Trades	29%	50.0%	42.9%	57.1%
Average Profit per Trade	-20.85%	-4.13%	-0.93%	-0.60%

Table 30: Trading performance for 0.5% threshold

	RiskMetrics	MLPs	RNNs	HONNs
Cumulative Profit	-113.67%	35.46%	86.03%	88.28%
Trades	6	4	6	6
Buys/Sells	2/4	1/3	2/4	1/5
Profitable Trades	33.33%	50.00%	50.00%	66.67%
Average Profit per Trade	-18.94%	8.87%	14.34%	14.71%

Table 31: Trading performance for 1% threshold

	RiskMetrics	MLPs	RNNs	HONNs
Cumulative Profit	-113.67%	35.46%	53.78%	88.28%
Trades	6	4	5	6
Buys/Sells	2/4	1/3	2/3	1/5
Profitable Trades	33.33%	50%	40%	66.67%
Average Profit per Trade	-18.94%	8.87%	10.76%	14.71%

Table 32: Trading performance for 1.5% threshold

	RiskMetrics	MLPs	RNNs	HONNs
Cumulative Profit	-197.97%	1.13%	33.20%	67.71%
Trades	5	1	4	5
Buys/Sells	2/3	0/1	2/2	1/4
Profitable Trades	20.00%	100.00%	50.0%	60.00%
Average Profit per Trade	-39.59%	1.13%	8.30%	13.54%

Table 33: Trading performance for 2% threshold

We observe that HONNs outperforms all other models as they demonstrate a superior trading performance for each of the four thresholds. Moreover, the

[32] These costs were obtained from brokers (see, for instance, www.interactive-brokers.com).

MLPs and the RNNs demonstrate also a sufficient performance with positive cumulative profits for 3 of the 4 thresholds. On the other hand, the trading results of the RiskMetrics are rather disappointing with negative cumulative profit in all cases. In general, this empirical evidence allows us to argue that NNs were able to successfully identify mispriced options and present a satisfactory trading performance.

7.5 Concluding Remarks

In this chapter, we apply Multi-layer Perceptron, Recurrent, and Higher Order neural networks to a 21-day-ahead forecasting and trading task of the FTSE 100 futures returns realized volatility. As a statistical benchmark we use the RiskMetrics volatility. We evaluate our forecasts not only in terms of statistical accuracy but also in terms of trading efficiency by applying a simple trading application.

Our results indicate that HONNs outperform all other models as they present the more accurate forecasts and the higher trading performance. These results may be attributed to their ability to capture higher order correlations within a dataset. We also note that the RNNs and the MLPs show sufficient results in the trading application in terms of cumulative profit and average profit per trade. On the other hand, the statistical and trading performance of RiskMetrics is rather disappointing as it presents the worst results in both the statistical and trading applications.

CHAPTER 8

Modelling Commodity Value at Risk with Higher Order Neural Networks[33]

Overview

The motivation for this chapter is to investigate the use of a promising class of neural network models, Higher Order Neural Networks (HONNs), when applied to the task of forecasting the one day ahead Value at Risk (VaR) of the oil brent and gold bullion series with only autoregressive terms as inputs. This is done by benchmarking their results with those of a different neural network design, the Multilayer Percepton (MLP), an Extreme Value Theory model (EVT) along with some traditional techniques such as an ARMA-GARCH (1,1) model and the Riskmetrics volatility. In addition to these, we also examine two hybrid Neural Networks-RiskMetrics volatility models. More specifically, the forecasting performance of all models for computing the VaR of the brent oil and the gold bullion is examined over the period 2002-2008 using the last year for out-of-sample testing. The evaluation of our models is done by using a series of backtesting algorithms and two loss functions: a violation ratio calculating when the realised return exceeds the forecast VaR and an firstly introduced average squared violation magnitude function computing the average magnitude of the violations.

As it turns out, the hybrid HONNs-RiskMetrics model does remarkably well and outperforms all other models in forecasting the VaR of gold and oil at both

---

[33] This paper has been presented at the Forecasting Financial Markets 2009 conference in Luxemburg (27 to 29 May 2009) and is currently being reviewed by the '*Journal of Applied Financial Economics*' for potential publication.

the 5% and 1% confidence levels, providing an accurate number of independent violations which also have the the lowest magnitude on average. The pure HONNs and MLPs along with the hybrid MLP-RiskMetrics model give also satisfactory forecasts in most cases.

8.1 Introduction

The motivation for this chapter is to investigate the use in Risk Management of Higher Order Neural Networks (HONNs) which have provided some promising empirical evidence in forecasting and trading stock market patterns. This is done by benchmarking their results with those of a different neural network design, the Multilayer Percepton (MLP), an Extreme Value Theory model (EVT) along with some traditional techniques such as an ARMA-GARCH (1,1) model and the Riskmetrics volatility. Moreover, we are examining two hybrid Neural Networks-RiskMetrics volatility models.

Their forecasting performance over the one day ahead VaR of gold and oil series is evaluated by using a series of back-testing algorithms such as the three tests suggested by Christoffersen (1998) and by implementing two loss functions, such as the violation ratio and an average magnitude regulatory function. We consider two different confidence levels, 5% and 1%. Also in our application we consider both tails of the return distribution (long and short position). In the absence of a formal theory around the selection of inputs on a Neural Network model and for our forecasting competition to be fair with the more traditional techniques, we select only autoregressive terms as inputs to our neural network models.

As it turns out, the hybrid HONNs-RiskMetrics model does remarkably well and outperforms all other models in forecasting the VaR of gold and oil at both the 5% and 1% confidence levels providing an accurate number of independent violations which have also the lowest magnitude on average. In other words the HONNs-RiskMetrics model produces an accurate independent exception process and gives the smallest difference on average between the forecast VaR and the actual return when an exception occurs. The pure HONNs and MLPs along with the hybrid MLP-RiskMetrics model also give satisfactory forecasts in most cases. On the other hand, our EVT model presents a disappointing performance something that can be attributed to the fact that only a few extreme events are present in our dataset.

The rest of the chapter is organised as follows. In section 8.2, we describe the dataset used for this research and its characteristics. An overview of the VaR framework and the statistical/technical models used in this research is given in section 8.3. Section 8.4 gives the empirical results of all the models considered while section 8.5 provides some concluding remarks.

8.2 The Brent Oil and the Gold Bullion Series

Our benchmark test is to forecast the one day ahead VaR of the brent oil and gold bullion based on their closing prices. All series are obtained from a historical dataset provided by Datastream.

We examine our series since 1 April 2002 until 31 March 2008. The data period is partitioned as follows.

Name of period	Trading Days	Beginning	End
Total dataset	1566	1 April 2002	31 March 2007
Training dataset	1305	1 April 2002	30 March 2007
Out-of-sample dataset [Validation set]	261	2 April 2007	31 March 2008

Table 34: The Gold and Oil dataset

The graph below shows the total dataset of gold and its strong upward trend from 2005 to 2007.

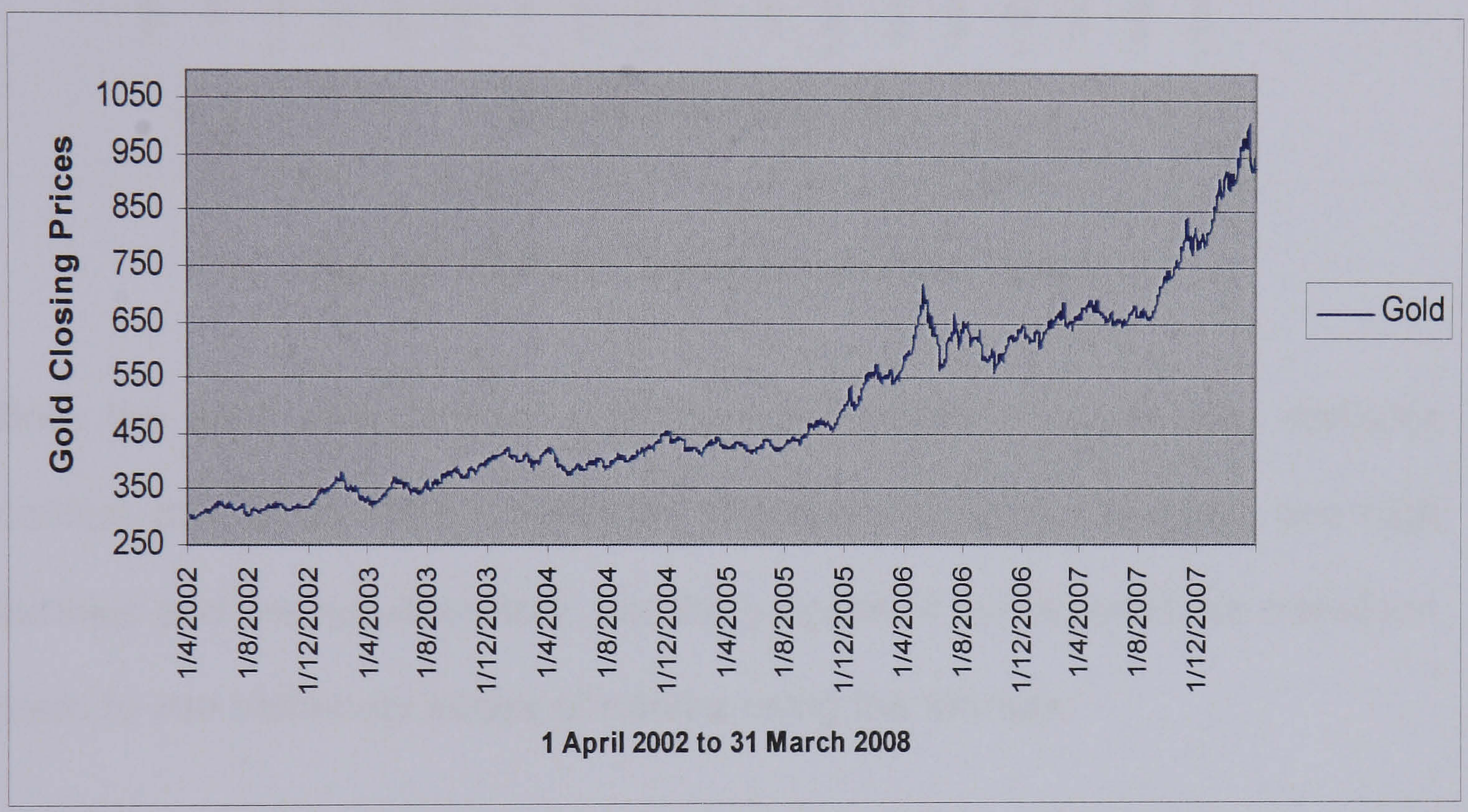


Fig. 18: Gold daily closing prices (total dataset)

Below the graph depicts the trend of oil for the review period.

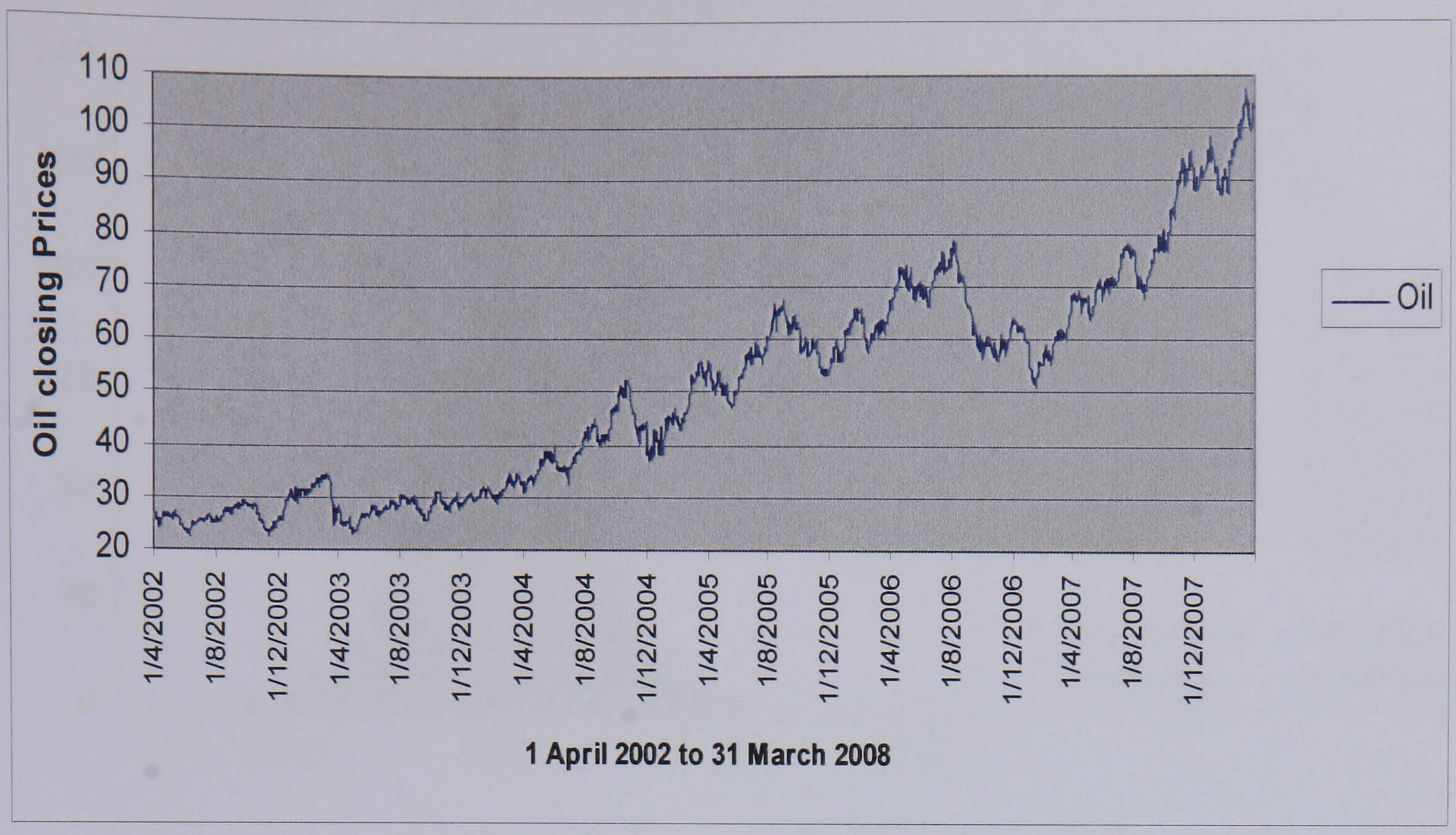


Fig. 19: Oil daily closing prices (total dataset)

Both the gold and oil time series are non-normal (Jarque-Bera statistics confirm this at the 99% confidence interval) containing skewness and high kurtosis and are nonstationary. For the purpose of our research we transform them to into stationary series of returns using the formula:

$$R_t = \left(\frac{P_t}{P_{t-1}} \right) - 1 \qquad [22]$$

Where R_t is the rate of return and P_t is the price level at time t.

The summary statistics of the oil returns series reveal a slight skewness and high kurtosis. The Jarque-Bera statistic confirms again that the oil series is non-normal at the 99% confidence interval.

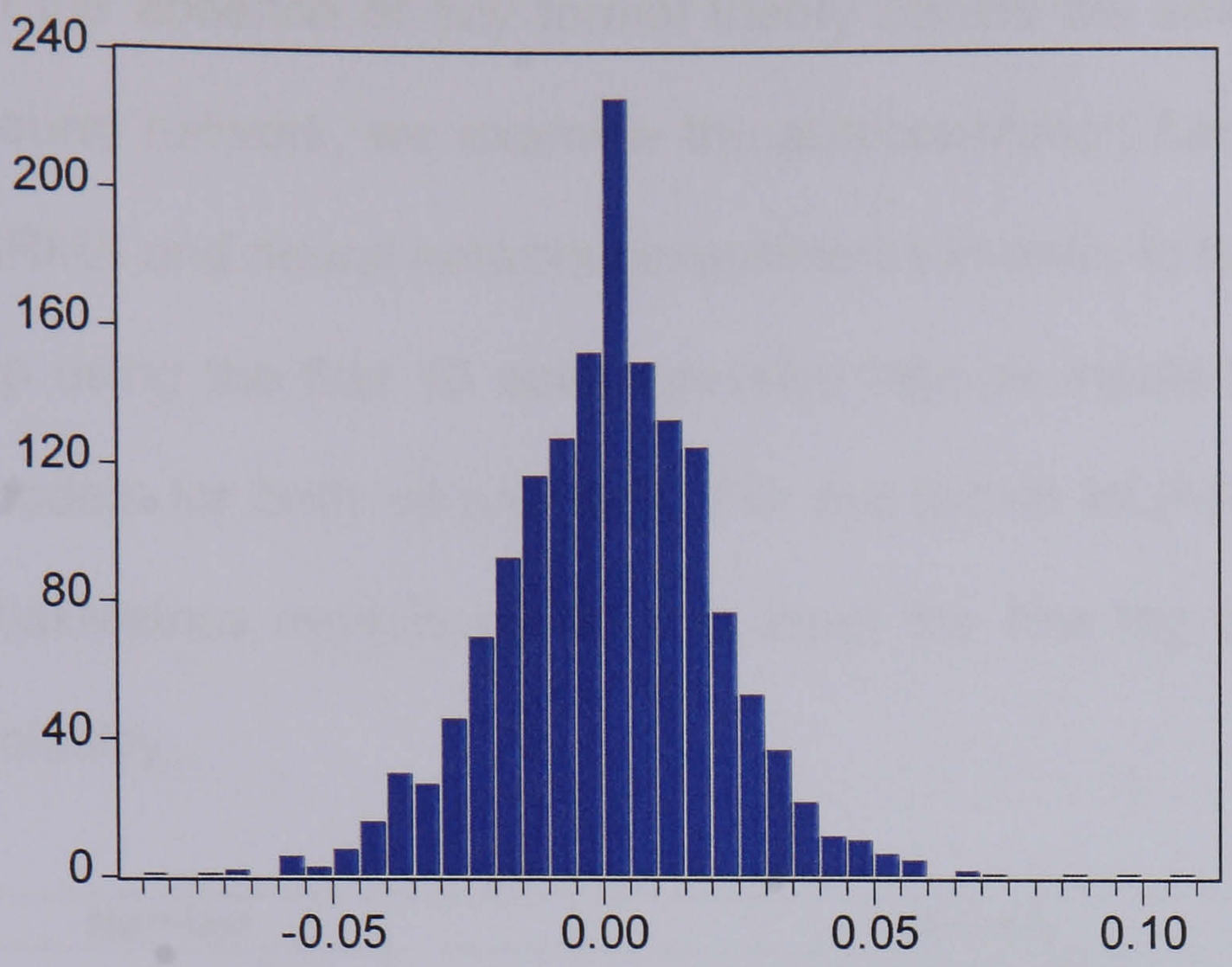


Fig. 20: Oil returns summary statistics (total dataset)

On the other hand, the gold returns series exhibit negative skewness and high kurtosis. Once again, the Jarque-Bera statistic confirms that the gold series is non-normal at the 99% confidence interval.

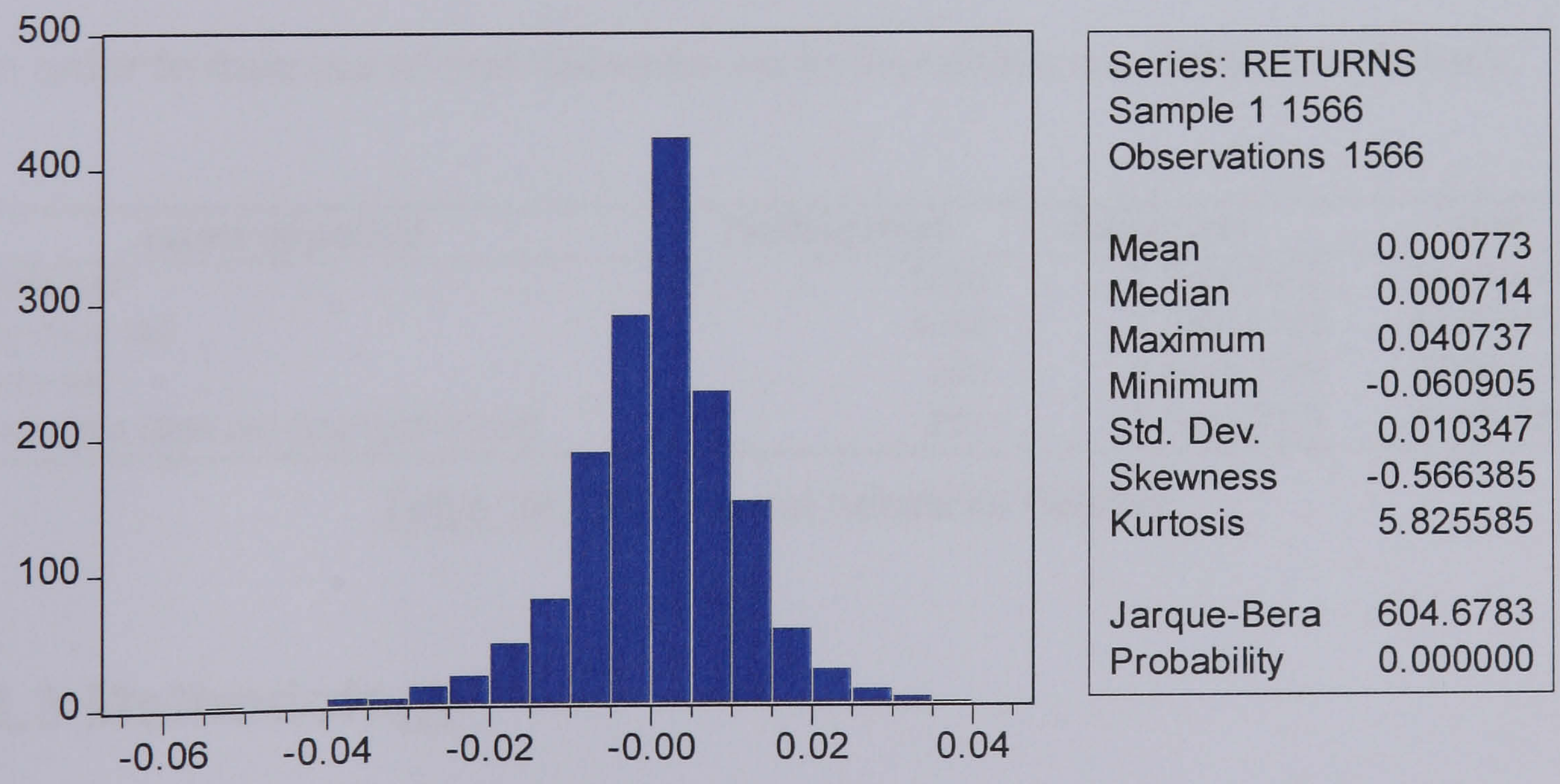


Fig. 21: Gold returns summary statistics (total dataset)

In the absence of any formal theory behind the selection of the inputs of a neural network, we examine the autocorrelation function and conduct some ARMA and neural networks experiments in order to help our decision. We end up using the first 10 autoregressive lags as inputs to the MLP and HONNs models for both oil and gold. For our hybrid MLP-RiskMetrics and HONNs-RiskMetrics models we add as input the first lag of the 1-day RiskMetrics volatility.

Number	Variable	Lag
1	Gold or Oil return	*1*
2	Gold or Oil return	*2*
3	Gold or Oil return	*3*
4	Gold or Oil return	*4*
5	Gold or Oil return	*5*
6	Gold or Oil return	*6*
7	Gold or Oil return	*7*
8	Gold or Oil return	*8*
9	Gold or Oil return	*9*
10	Gold or Oil return	*10*
11	1-day Riskmetrics Volatility	1

Table 35: Explanatory variables for the MLP-RiskMetrics and HONNs-RiskMetrics hybrid models

In order to train our neural networks we further divide our dataset as follows:

Name of period	Trading days	Beginning	End
Total data set	1566	1 April 2002	31 March 2008
Training data set	1045	1 April 2002	31 March 2006
Test data set	260	3 April 2006	30 March2007
Out-of-sample data set [Validation set]	261	2 April 2007	31 March 2008

Table 36: The neural networks dataset

8.3 Methodology

Under a probabilistic framework, at the time t, we are interested in the risk of a financial position for the next h periods. If we define $\Delta V(h)$ to be the asset value change from time t to $t+h$, then this quantity is measured in e.g. dollars

and is a random variable at time *t*. Denote the conditional density function (CDF) of $\Delta V(h)$ by $F_h(x)$. Then we define the VaR of a long position over the time horizon *h* with probability p as:

$$p = \Pr[\Delta V(h) \leq VaR] = F_h(VaR) \qquad [32]$$

For a long position, the loss occurs when $\Delta V(h) \prec 0$ and so the VaR defined in [32] is assumed to have a negative value. On the other hand, for the short position the loss occurs when $\Delta V(h) \succ 0$ and the VaR has a positive value. For long positions, the left tail of $F_h(x)$ is important while for short positions the right tail is important. As an investor can buy or sell an asset we consider both tails of the distribution. Moreover, the asymmetries between the tails of the return distributions of our assets (something that can be observed to some extent from figures 17 and 18) enables one to draw additional conclusions when large discrepancies are observed in our results for long and short positions.

The above equation can be also interpreted as the probability that the holder would encounter a loss greater than or equal to VaR is p. So in other words, VaR is a story of modelling the tail behavior of the CDF $F_h(x)$. However, the CDF is unknown in practice and most VaR forecasting models require an *a priori* definition of the CDF. In our research for our models except the EVT model, we will consider that the unknown CDF is that of the normal distribution. Although we acknowledge that our series do not follow the normal distribution, we decide to take this assumption for the sake of simplicity. This assumption is most common in the literature (see among others Lee and

Saltoglu (2001), Dunis *et al.* (2005), Rau-Bredow (2004) and Liu (2005)). The normality assumption also implies that the VaR modelling will be similar for long and short positions[34]. Estimating the CDF of the series, something that could enable one to use different specifications in the VaR estimation for long and short positions is far beyond the scope of this thesis.

A complete description of our NN architectures used on this application (HONN and MLP) is on chapter 3. Since the starting point for each network is a set of random weights, forecasts can differ slightly between networks. In order to eliminate any variance between our NNs forecasts, we used the average of a committee of 20 NNs which presents the better statistical performance in-sample. The characteristics of the NNs used in this chapter are presented in Appendix A.5.1. The target value in our networks is the one day estimated volatility of our series else the absolute value of the return of our assets. After we forecast the volatility, we compute the VaR using the formula below:

$VaR_{t+1}^{q} = \hat{\sigma}_{t+1} c_q$ [33] where VaR_{t+1}^{q} is the VaR forecast for t+1 period with q% confidence level, $\hat{\sigma}_{t+1}$ is the forecasted volatility for period t+1 from our MLPs and c_q is the critical value of the normal distribution for q% confidence level.

---

[34] We have also considered the case that the CDF follows the Student distribution with 6 degrees of freedom. Our results, which are available upon request, are not given here in order to conserve space and because they are not significantly different from those obtained under the normality assumption. In the relevant literature, other options have been considered as the Weibull distribution or various techniques to identify the CDF of the series under study.

8.3.1 RiskMetrics Volatility

The RiskMetrics volatility model is treated as a benchmark model owing to its popularity in risk measurement. Roughly speaking, RiskMetrics is one of the simplest tools for measuring financial market risk under the VaR framework. Derived from the GARCH(1,1) model, but with fixed coefficients, RiskMetrics volatility is calculated using the standard formula:

$RMVOL_t^2 = b\sigma_{t-1}^2 + (1-b)R_t^2$ [30] where σ_t^2 is the asset variance at time t, R_t^2 is the asset squared return at time t and b=0.94 for daily data. In this chapter we use RiskMetrics volatility to forecast 1-day ahead volatility for the out-of-sample period. The RiskMetrics volatility is calculated from equation [30] and then we use equation [34] below to calculate the 1-step ahead volatility forecast: $\hat{\sigma}_{t+1} = RMVOL_t$ [34]. Then we compute VaR as: $VaR_{t+1}^q = \hat{\sigma}_{t+1} c_q$ [33] where VaR_{t+1}^q is the VaR forecast for t+1 period with q% confidence level, $\hat{\sigma}_{t+1}$ is the forecasted volatility for period t+1 and c_q is the critical value of the normal distribution for q% confidence level.

8.3.2 ARMA-GARCH(1,1)

We also forecast the VaR of our assets, with an ARMA-GARCH(1,1) model. Since the seminal contribution of Engle (1982), ARCH and GARCH type models have become standard tools to model the volatility of financial market data. An ARCH-GARCH model consists of two equations. The first one is an ARMA equation for the returns R_t. In our research we model the returns with a

restricted ARMA(10,10) model where all coefficients are significant at the 95% confidence interval. The second equation is used to model the variance of the underlying series. In our application we model the variance with a GARCH(1,1) equation. Justification for the use of GARCH(1,1) and not of one of its numerous variations can be provided based on the empirical evidence of Corhay and Rad (1994), Vasilellis and Meade (1996) and Walsh and Tsou (1998).

In mathematical terms an ARMA(p,q)-GARCH(1,1) can be expressed as:

$$R_t = c + \sum_{x=1}^{p} \beta_x R_{t-x} + \sum_{x=1}^{q} a_x u_{t-x} + u_t \qquad \text{(Mean equation)} \qquad [35]$$

$$\sigma_t^2 = w + a u_{t-1}^2 + \beta \sigma_{t-1}^2 \qquad \text{(Variance equation)} \qquad [36]$$

Using the above two sets of equations and assuming that the errors u_t follow the normal distribution, we can obtain forecasts of the future return and variance. Our estimation outputs are provided in Appendix A1. We obtain the one day ahead VaR forecast of our series with:

$VaR_{t+1}^q = \hat{R}_{t+1} - c_q \hat{\sigma}_{t+1}$ [37] where VaR_{t+1}^q is the VaR forecast for t+1 period with q% confidence level, $\hat{R}_{t+1}$ and $\hat{\sigma}_{t+1}$ obtained from equations [35] and [36] respectively while c_q is the critical value of the normal distribution for q% confidence level.

8.3.3 Extreme Value Theory Model

Extreme Value Theory (EVT) is a powerful and yet fairly robust framework to study the tail behaviour of a distribution. Even though the theory was primarily applied to climatology and hydrology, recently there has been an increasing

number of extreme value studies around VaR and Risk Management in the literature (see for example Bali (2003), Gencay and Selcuk (2004), Gili and Kellezi (2006) and Samuel (2008)).

There are two main approaches in estimating VaR with EVT, namely the method of block over maxima and the method of peaks over threshold (POT). In our research, based on the empirical evident provided by Gili and Kellezi (2006) who compared the two methods and demonstrated the superiority of POT, we will estimate VaR with the POT approach. Moreover, in our estimation we decided to follow the unconditional approach. In real world environments, the unconditional approach is preferred as it can provide stable estimates through time while avoiding the time consuming computations required by the conditional approach (Gili and Kellezi (2006) and Samuel (2008)). So in these lines and based on the empirical evident of Gili and Kellezi (2006) we follow the POT approach.

The POT method is based on a theorem stated by Picklands (1975) and Balkema and de Haan (1974). According to it, for a large class of underlying distribution functions *F* the conditional excess distribution $F_u(y)$, for a threshold *u* large enough, is well approximated by the generalised Pareto distribution (GPD). In simple words, the implementation of the POT method involves three steps. First we choose a sufficient high threshold to satisfy the above mentioned theorem. In our application we choose our threshold following the methodology employed by Bali (2003). He suggests choosing as threshold the distance of 2 standard deviations from the in-sample mean. For gold, this produces 33 extreme events for long positions and 29 for short

positions (respectively 2.5% and 2.2% of the in-sample observations). For oil, we have 33 extreme events for long positions (2.5% of the in- sample dataset) and 32 for short positions (2.4% of the in-sample dataset). Then based on the excesses over the threshold, we estimate the parameters of the GPD using the method of moments[35]. In the end, we estimate VaR using the formula:

$$VaR^q = u + \frac{\sigma}{\hat{\xi}}[(\frac{n}{Nu}p)^{-\hat{\xi}} - 1] \quad [38]$$ where VaR^q is the VaR estimate for the q% confidence level, u is is the threshold, $\hat{\sigma}$ and $\hat{\xi}$ are the moments estimates of the shape and scaling parameters of the GPD respectively, n is the sample size and Nu is the number of observations above u. More details around the POT method and our VaR estimation can be found at Gili and Kellezi (2006) and Samuel (2008).

8.4 Backtesting

8.4.1 Christoffersen Tests

Christoffersen (1998) introduced a three step VaR evaluation procedure. In a likelihood ratio (LR) testing environment, he introduced a test of correct unconditional coverage, a test of independence and a test for conditional coverage. As a first step in order to evaluate our models we follow this procedure.

---

[35] Moments estimators for the GPD were derived by Hosking and Wallis (1987). According to the empirical evident provided by Singh and Guo (1995) the method of moments seems more appropriate than the maximum likelihood estimation for our dataset.

8.4.1.1 LR Test of Correct Unconditional Coverage

Let us consider a dummy variable d_k^t for model *k* which takes the value of 1 when the return falls behind the VaR forecast estimated from model *k* and 0 in all other cases. Then the indicator sequence d_k^t should follow the binomial distribution with likelihood $L(a) = (1-a)^{n_0} a^{n_1}$ where $a = P(d_k^t = 1)$, n_0 is the number of 0 in the d_k^t sequence and n_1 is the number of 1. In an accurate VaR model with confidence level q%, q should equal α. Else the probability to have a violation should be q%. Christoffersen (1998), under the null that we have a correct violation ratio, formulated all this in the standard LR test presented below[36]:

$LRuc = -2\log[\frac{q^{n_1}(1-q)^{n_o}}{\pi^{n_1}(1-\pi)^{n_0}}] \xrightarrow{d} \chi(1)$ [39] where n_1 is the number of violations, n_0 is the number of non-violations, q is the coverage rate of the VaR model and $\pi = \frac{n_1}{n_0 + n_1}$ is the maximum likelihood estimate of q.

8.4.1.2 LR Test of Independence

The second test is to check whether the indicator sequence d_k^t (else the sequence of violations for our VaR forecasts of model *k*) is serially independent. Christoffersen (1998) was motivated to such a test by noting that a constant VaR given by the unconditional distribution from a GARCH model will have too many exceptions during periods of high volatility and too

---

[36] Kupiec (1995) and McNees (1995) apply similar tests of unconditional coverage.

few during periods of low volatility. Because volatility tends to cluster, failing to adequately model volatility in the VaR will result in serial correlation in the violation sequence.

Under the null hypothesis that the exception sequence is serially independent and the alternative that it is a first order Markov process, the likelihood ratio of independence can be tested by:

$$LRind = -2\log[\frac{(1-\pi_2)^{n_{00}+n_{11}}\pi_2^{n_{01}+n_{11}}}{(1-\pi_{01})^{n_{00}}\pi_{01}^{n_{01}}(1-\pi_{11})^{n_{10}}\pi_{11}^{n_{11}}}] \xrightarrow{d} \chi(1) \quad [40]$$

where n_{ij} is the number that value *i* is followed by *j* in the violations sequence, $\pi_{01} = \frac{n_{01}}{n_{00}+n_{01}}$, $\pi_{11} = \frac{n_{11}}{n_{10}+n_{11}}$ and $\pi_2 = \frac{n_{01}+n_{11}}{n_{00}+n_{01}+n_{11}+n_{10}}$.

8.4.1.3 LR Test of Conditional Coverage

By combining the two tests, the third test for conditional coverage can be constructed. This time the null hypothesis is that that we have an independent exception process with correct violation ratio while the alternative is that we have a first order Markov process with a different transition probability matrix. The likelihood ratio statistic is:

$$LRcc = -2\log[\frac{p^{n_1}(1-p)^{n_0}}{(1-\pi_{01})^{n_{00}}\pi_{01}^{n_{01}}(1-\pi_{11})^{n_{10}}\pi_{11}^{n_{11}}}] \xrightarrow{d} \chi(2) \quad [41]$$

8.4.1.4 Results

The likelihood ratio statistics of the Christoffersen tests for our models are presented in Appendix A.5.3. Our results indicate that all our models except

the EVT model give a correct violation ratio with an independent exception process. In other words, that most our models made accurate forecasts in a VaR framework. Moreover, we observe that there is no discrepancy in the results for the long or short positions. So despite the fact that the modelling of VaR is similar for both tails (which are asymmetric and seem to have different characteristics), our VaR forecasts prove satisfactory in both cases.

These results prevent us from distinguishing among different, but close alternative models. This weakness of Christoffersen tests is also noted in similar applications such in Sarma *et al. (2003)*, Cakici and Foster (2003) and Fantazzini (2007). Moreover, these tests as any other statistical test are subject to Type II errors. Therefore, in order to distinguish our models we follow another approach, the one of loss functions.

8.4.2 Loss Functions

In order to verify the reliability of our models and to distinguish their VaR forecasting performance we apply two different loss functions. The main contribution in this area is the one of Lopez (1998) who defines the general form of those loss functions as: $G = \frac{1}{n}\sum_{i=1}^{n} C_{t+i}$ [42] where $C_{t+i} = f(R_{t+i}, VaR_{t+i})$ if $R_{t+i} \prec VaR_{t+i}$ and $C_{t+i} = g(R_{t+i}, VaR_{t+i})$ if $R_{t+i} \geq VaR_{t+i}$ such that $f(R_{t+i}, VaR_{t+i}) \geq g(R_{t+i}, VaR_{t+i})$.

The Basel Committee on Banking Supervision (1996) indicates that the magnitudes as well as the number of exceptions are a matter of regulatory

concern. In our research we consider one loss function to incorporate the number of exceptions and one for their magnitude.

8.4.2.1 Violation Ratio

The violation ratio (or the hit rate) is simply the percentage occurrence of an actual loss greater than the predicted maximum loss in the VaR framework. In our application this can be formulated as: $G = \frac{1}{261}\sum_{i=1}^{261} Ht + i$ [43] where $H_{t+i} = 1$ if $R_{t+i} \prec VaR_{t+i}$ and $H_{t+i} = 0$ if $R_{t+i} \geq VaR_{t+i}$ with R_{t+i} and VaR_{t+i} to be the actual return and the forecasted VaR from our models for day *t+i* and 261 to be the number of trading days in the out-of-sample period. In our application, we want our models to have a violation ratio as close as it can be to our confidence level. In other words, a prefect model will give a violation ratio of 1% and 5% for the 1% and 5% confidence levels respectively. In table 37 below we present the violation ratios of our models for gold.

		RiskMetrics	ARMA-GARCH	EVT	MLP	HONNS	MLP-RiskMetrics	HONNs-RiskMetrics
5% Confidence Level	Long	4.59%	**4.98%**	7.66%	5.75%	4.59%	5.75%	4.59%
	Short	6.90%	7.66%	8.81%	6.51%	4.21%	3.83%	**4.59%**
1% Confidence Level	Long	2.68%	2.30%	1.92%	1.53%	1.53%	**1.15%**	**1.15%**
	Short	1.92%	1.53%	3.83%	**1.15%**	**1.15%**	1.92%	1.53%

Table 37: Violation Ratios for Gold

Note: The entries in bold represent each time the closest violation ratios to the benchmark.

We observe that the hybrid HONNs-RiskMetrics model outperforms all other models most of the time. However, models such as the HONNs and the hybrid MLP-RiskMetrics give also satisfactory violation ratios. On the other hand, we note that the traditional RiskMetrics and the more sophisticated EVT models present an unsatisfactory performance with large deviations of their ratios from the benchmarks. The violation ratios for oil are presented in the table below.

		RiskMetrics	ARMA-GARCH	EVT	MLP	HONNS	MLP-RiskMetrics	HONNs-RiskMetrics
5% Confidence Level	Long	5.78%	3.83%	0.08%	3.45%	4.30%	3.10%	**4.59%**
	Short	6.13%	3.07%	2.68%	**4.60%**	**4.60%**	**4.60%**	5.75%
1% Confidence Level	Long	2.69%	NA	NA	1.92%	2.54%	2.29%	**1.52%**
	Short	1.52%	NA	NA	NA	1.53%	0.38%	**1.15%**

Table 38: Violation Ratios for Oil

Note: The entries in bold represent each time the closest violation ratios to the benchmark. NA indicates that there were 0 violations and therefore we are unable to assess the model.

We can see that NNs show a more accurate violation ratio compared to the statistical techniques. More specifically in most cases the hybrid HONNs-RiskMetrics model seems to outperform all other models with the MLP and HONNs giving also close ratios to the benchmark. Furthermore, we observe that for oil, the performance of ARMA-GARCH and the EVT models is unsatisfactory with too few or too many violations in the out-of-sample period.

8.4.2.2 Average Squared Magnitude Function

Next in order to examine the magnitude of our models violations, we consider an average magnitude loss function. We want the magnitude of violations of

our models to be as small as possible. This is in order to take into account not only the risk but also the amount of possible default in the position. Different kinds of magnitude loss functions have already been proposed by Lopez (1998) and Sarma *et al.* (2003). However, all functions proposed in these papers depend crucially on the number of exceptions. The fewer the exceptions, the smaller are the function results. This deficiency is crucial as the theoretical framework of these functions suggests accepting as best the model that gives us the smaller realization for the magnitude function. So we may reject a correctly specified model with an accurate number of exceptions because it produces a higher loss function than a more conservative model. For example let us assume that we are studying an asset at the 95% confidence level. A very conservative model with 1 violation (0.04% violation ratio for our out-of-sample dataset) will usually be preferred with the Lopez (1998) function to a model with 13 violations (4.98% violation ratio) irrespective of the magnitude of the violations of both models. In other words, a conservative model has always an advantage even on more accurate specified models. This disadvantage has already been noted by Caporin (2003). To overcome this problem, we measure the average squared cost of exceptions with a separate loss function, independently from the number of exceptions, and not jointly as in Lopez (1998) and Sarma *et al.* (2003), avoiding thus the previous mentioned possible misspecifications. For that reason we consider the function below: $E = \frac{1}{V}\sum_{i=1}^{261} T_i$ [44] with *V* the number of violations of our model, $T_i = (R_{t+i} - VaR_{t+i})^2$ when $R_{t+i} \prec VaR_{t+i}$ and $T_i = 0$ when $R_{t+i} \geq VaR_{t+i}$. A model which minimises [44] is preferred over alternative

models. We use the average magnitude function to further discriminate between models with similar or identical hit rates. In the table below we present the average magnitude of violations of our models for gold.

		RiskMetrics	ARMA-GARCH	EVT	MLP	HONNS	MLP-RiskMetrics	HONNs-RiskMetrics
5% Confidence Level	Long	0.0247%	0.0251%	0.0331%	0.0231%	0.0227%	**0.0218%**	0.0236%
	Short	**0.0040%**	0.0045%	0.0104%	0.0123%	0.0069%	0.0053%	0.0056%
1% Confidence Level	Long	0.0230%	0.0299%	0.0166%	0.0187%	0.0169%	0.0172%	**0.0165%**
	Short	0.0031%	0.0182%	0.0033%	0.0076%	**0.0029%**	0.0051%	0.0993%

Table 39: Average squared magnitude of violations for gold
Note: The entries in bold represent each time the smallest average squared magnitude

We generally observe that the magnitude of violations for short positions is smaller than the one for long positions. Furthermore, we note that for the 5% confidence level our results are inconclusive as the models giving the best violation ratio (see table 37 above) are giving us the larger magnitudes and vice versa. On the other hand, for the 1% level the results seem clearer: the hybrid HONNs-RiskMetrics and the HONNs models have the best violation ratio (see table 37 above) and the smallest average violation magnitude for long and short positions respectively. This allows one to argue that for these particular cases the HONNs-RiskMetrics and the HONNs models are more accurate forecasters of VaR for gold. Table 40 below shows the results of the average squared magnitude function for oil.

		RiskMetrics	ARMA-GARCH	EVT	MLP	HONNS	MLP-RiskMetrics	HONNs-RiskMetrics
5% Confidence Level	Long	0.0085%	0.0047%	0.0097%	0.0052%	0.0048%	0.0057%	**0.0042%**
	Short	0.0058%	0.0038%	0.0127%	0.0096%	0.0024%	0.0031%	**0.0022%**
1% Confidence Level	Long	**0.0013%**	NA	NA	0.0035%	0.0027%	0.0043%	0.0019%
	Short	**0.0008%**	NA	NA	NA	0.0015%	0.0019%	0.0010%

Table 40: Average squared magnitude of violations for oil

Note: The entries in bold represent each time the smallest average squared magnitude. NA indicates that there were 0 violations and therefore we are unable to assess the model.

For the 5% level we observe that for long positions the hybrid HONNs-RiskMetrics model has not only the best violation ratio (see table 38 above) but also the smallest average magnitude for its violations. For short positions, we note that the HONNs-RiskMetrics model continues to have the smallest average magnitude on for its violations and that the HONNs model which has the best hit rate gives the second smallest realization for the magnitude function. For the 1% confidence level the RiskMetrics model has in both cases the smallest magnitude for its violations. However, it has also the worst violation ratios (see table 38 above). On the other hand, the hybrid HONNs-RiskMetrics model has the second smallest realization in results for the magnitude function and the best hit rate for both long and short positions.

8.5. Concluding Remarks

In this chapter, we apply Higher Order Neural Networks to a one-day-ahead forecasting task of the Value at Risk of gold bullion and Brent oil. This is done by benchmarking the HONNs results with those of a Multilayer Percepton

(MLP) network, an Extreme Value Theory model (EVT) along with some traditional techniques such as an ARMA-GARCH (1,1) model and the RiskMetrics volatility of the series. In addition to these, we also examine two hybrid Neural Networks-RiskMetrics volatility models. We develop these different prediction models over the period April 2002 to March 2007 and validate their out-of-sample efficiency over the following period from April 2007 through March 2008. The evaluation of our models is done by using the three tests procedure suggested by Christoffersen (1998) and two loss functions, such as the violation ratio calculating when the realised return exceeds the forecast VaR and an average squared magnitude function, firstly introduced in this application, which focuses on the average magnitude of these violations.

Our VaR estimation was computed based on the assumption that our assets follow the normal distribution. Although our models provided accurate VaR forecasts we aknowledge that a different distribution or methodology to compute the VaR may have led to beter estimations.

The hybrid HONNs-RiskMetrics model demonstrates a better forecasting performance providing an accurate number of independent violations at the 5% and 1% confidence levels for both long and short positions. The HONNs, the MLP and the hybrid MLP-RiskMetrics model also demonstrate a good performance in most cases. On the other hand, the EVT model produces disappointing forecasts, something that can be attributed to the fact that only a few extreme events are present in our dataset. In the circumstances, our

results should go some way towards convincing a growing number of quantitative risk managers to experiment beyond the bounds of the more traditional risk models. Moreover, our unique methodology to estimate the VaR through NNs should lead to more expiriments around the utility of NNs in financial research.

CHAPTER 9

General Conclusions and Future Work

The general motivation of this thesis was to provide empirical evidence on the utility of HONNs in financial forecasting and trading applications. In order to achieve this, we benchmarked HONNs not only with some traditional statistical and technical techniques but also with some other state-of-the-art NNs designs. Therefore, we were able to validate if the theoretical advantages of HONNs compared to the more traditional NNs models are translated into more accurate and hence profitable forecasts. Moreover, we explored the utility of HONNs if we feed them not only with multivariate inputs but also with autoregressive series as inputs. Thus we are able to draw more solid conclusions on the forecasting ability of our models especially against our statistical autoregressive benchmarks as HONNs incorporated no additional knowledge compared to them.

In chapters 4 and 5 we demonstrated the forecasting and trading superiority of HONNs in the task of forecasting the returns of the EUR/USD exchange rate using multivariate and autoregressive series as inputs. Although HONNs with multivariate series as inputs provided only slightly better results than our NNs benchmarks, with autoregressive series their trading performance was significantly better than that of MLP, RNN and Psi Sigma networks. We also note that all our models failed to exploit confirmation strategies using filters and leverage. They thus failed to further improve on their original trading results. Moreover, we observe that for the period and the series under study the RNNs and the Psi Sigma networks seem to have a difficulty in providing

good forecasts when only autoregressive series are used as inputs. Furthermore, in chapter 6 we demonstrate that the forecasting and trading performance of HONNs is stable and robust over time. This is an essential property in real world applications for models like HONNs whose modelling is based on trial and error rather than on some formal statistical theory.

In chapter 7 we examine the use of HONNs when applied to the task of forecasting and trading the 21-day ahead realised volatility of the FTSE 100 futures index. We evaluated their performance not only in terms of statistical accuracy but also through a simple trading application that integrates trasanction costs. HONNs demonstrate a remarkable performance and outperform MLP, RNN and the RiskMetrics volatility models not only in terms of statistical accuracy but also in terms of trading efficiency.

In chapter 8, we test the ability of HONNs to forecast accurately the one day ahead VaR of the brent oil and gold bullion. This time we used the MLP and the RNN networks, an EVT model along with an ARMA-GARCH (1,1) model and the Riskmetrics volatility as benchmarks. We also examine a hybrid HONNs-RiskMetrics model where we use the Riskmetrics volatility as an input to HONNs network. As it turns out, the hybrid HONNs-RiskMetrics model does remarkably well and outperforms all other models in forecasting the VaR of gold and oil at both the 5% and 1% confidence levels, providing an accurate number of independent violations which also have the lowest magnitude on average.

The above mentioned empirical evidence allows us to argue with confidence that HONNs can provide accurate, profitable and robust forecasts. Their performance seems superior to that of the MLP and RNN models and of the linear ARMA, MACD and RiskMetrics volatility techniques. Compared to Psi Sigma they seem to have the same forecasting accuracy when we feed them with multivariate series and a better one when autoregressive series are used as inputs. Moreover, we note that the time needed to train HONNs was far less than the time needed for RNN networks. In general, our results should go some way towards convincing quantitative risk and fund managers to use to alternative non-linear techniques such as HONNs as they generate higher return/risk profiles.

Moreover, this research can be extended and contribute to more fields in financial research. A direct comparison in the forecasting perofmance of HONNs when multivariate and autoregressive series are used as inputs will offer a more complete view around their forecasting abilities and limitations. Moreover, a study over their sensitivity to changes in the training period will further examine the robstuness of their performance while chapter 8 can be extended if we consider other CDF for our assets such as the Weibull distribution or other models such as a conditional EVT model. In the end, a study over the statistical significance of our forecasts.

References

Adam, O., Zarader, L. and Milgram, M. (1994), 'Identification and Prediction of Non-Linear Models with Recurrent Neural Networks', *Laboratoire de Robotique de Paris*.

Andreou, P., Charalambous, C. and Matzoukos, S. (2008), 'Pricing and Trading European Options by Combining Neural Networks with Parametric Models with Implied Parameters', *European Journal of Operational Research*, 185, 1415-1433.

Bali, T. (2003), 'An Extreme Value Approach to Estimating Volatility and Value at Risk', *Journal of Business*, 76, 1, 83-108.

Balkema, A. and de Haan, L. (1974), 'Residual Life Time at Great Age', *Annals of Probability*, 2, 792-804.

Barone-Adesi, G. and Whaley, E. (1987), 'Efficient Analytic Approximation of American Option Values', *Journal of Finance*, 42, 301-320.

Bartlmae, K. and Rauscher, F. (2000), 'Measuring DAX Market Risk: A Neural Network Volatility Mixture Approach', Presentation at the FFM2000 Conference, London, 31 May-2 June.

Basel (1996), '*Overview of the Amendment to the Capital Accord to Incorporate Market Risk*', Basel Committee on Banking Supervision, Basel.

Black, F. and Scholes, M. (1973), 'The Pricing of Options and Corporate Liabilities', *Journal Political Economy*, 81, 637-654.

Box, G., Jenkins, G. and Gregory, G. (1994), *Time Series Analysis: Forecasting and Control*, Prentice-Hall, New Jersey.

Cakici, N. and Foster, K. (2003),' Value at Risk for Interest Rate-Dependent Securities', *Journal of Fixed Income*, 12, 4, 81-96.

Caporin, M. (2003), 'Evaluating Value-at-Risk Measures in Presence of Long Memory Conditional Volatility', *GRETA working paper n. 05.03.*

Christoffersen, P. (1998), 'Evaluating Interval Forecasts', *International Economic Review*, 39, 4, 841-864.

Connor, J. and Atlas, L. (1993), 'Recurrent Neural Networks and Time Series Prediction', *Proceedings of the International Joint Conference on Neural Networks*, 301-306.

Corhay, A. and Rad, T. (1994), 'Statistical Properties of Daily Returns: Evidence from European Stock Markets', *Journal of Business Finance and Accounting,* 21, 271-282.

Cornalba, C and Giudici, P. (2004), 'Statistical models for operational risk management', *Physica A: Statistical Mechanics and its Applications*, 338, 166-172.

Corrado, J. and Su, T. (1996), 'Skewness and Kurtosis in S&P 500 Index Returns Implied by Option Prices', Journal of Financial Research, 19, 2, 175–192.

Dunis, C. and Chen, Y. (2005), 'Alternative Volatility Models for Risk Management and Trading: Application to the EUR/USD and USD/JPY Rates', *Derivatives Use, Trading & Regulation*, 11, 2, 126-156.

Dunis, C. and Huang, X. (2002), 'Forecasting and Trading Currency Volatility: An Application of Recurrent Neural Regression and Model Combination', *Journal of Forecasting*, 21, 5, 317-354.

Dunis, C., Laws, J. and Evans B. (2006a), 'Trading Futures Spreads: An Application of Correlation and Threshold Filters', *Applied Financial Economics*, 16, 1-12.

Dunis, C., Laws, J. and Evans B. (2006b), 'Modelling and Trading the Gasoline Crack Spread: A Non-Linear Story', *Derivatives Use, Trading and Regulation*, 12, 126-145.

Dunis, C. and Nathani, A. (2007), 'Quantitative Trading of Gold and Silver Using Nonlinear Models', *Neural Network World*, 93-111.

Dunis, C. and Williams, M. (2002), 'Modelling and Trading the EUR/USD Exchange Rate: Do Neural Network Models Perform Better?', *Derivatives Use, Trading and Regulation*, 8, 3, 211-239.

Dunis, C. and Williams, M. (2003), 'Applications of Advanced Regression Analysis for Trading and Investment', *Applied Quantitative Methods for Trading and Investment*, John Wiley, Chichester.

Elman, J. L. (1990), 'Finding Structure in Time', *Cognitive Science*, 14, 179-211.

Engle, R. (1982), 'Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation', *Econometrica,* 50, 4, 987-1007.

Fantazzini, D. (2007), 'Dynamic Copula Modelling for Value at Risk', forthcoming in *Frontiers in Finance and Economics*.

Fulcher, J., Zhang, M. and Xu, S. (2006), 'The Application of Higher-Order Neural Networks to Financial Time Series', *Artificial Neural Networks in Finance and Manufacturing*, Hershey, PA: Idea Group, 80-108, London.

Garcia, R. and Gencay, R. (2000), 'Pricing and Hedging Derivative Securities with Neural Networks and a Homogeneity Hint', *Journal of Econometrics*, 94, 93-115.

Gardovejic, N., Gencay, R. and Kukolj, D. (2009),'Option Pricing with Modular Neural Networks', forthcoming in *IEEE Transactions on Neural Networks*.

Gencay, R and A. Altay-Salih (2003), 'Degree of Mispricing with the Black-Scholes Model and Nonparametric Cures', *Annals of Economics and Finance*, 4, 73–101.

Gencay, R. and Selcuk, F. (2004), 'Extreme Value Theory and Value-at-Risk: Relative Performance in Emerging Markets', *International Journal of Forecasting*, 20, 2, 287-303.

Ghazali, R., Hussain, A. and Merabti, M. (2006), 'Higher Order Neural Networks for Financial Time Series Prediction', *The 10th IASTED International Conference on Artificial Intelligence and Soft Computin*g, Palma de Mallorca, Spain, 119-124.

Ghosh, J. and Shin, Y. (1992), 'Efficient Higher-Order Neural Networks for Classification and Function Approximation', *International Journal of Neural Systems*, 3, 4, 323-350.

Giles, L. and Maxwell, T. (1987). 'Learning, Invariance and Generalization in Higher Order Neural Networks', *Applied Optics*, 26, 4972-4978.

Gilli, M. and Kellezi, E. (2006), 'An Application of Extreme Value Theory for Measuring Financial Risk', *Computational Economics*, 27, 1, 1-23.

Hamid, S. (2004), 'Primer on Using Neural Networks for Forecasting Market Variables', *Southern New Hampshire University Working Papers*, 2004-03.

Hopfield, J. (1982), 'Neural networks and physical systems with emergent collective computational abilities', *Proceedings of the National Academy of Sciences of the USA*, Vol. 79, pp. 2554-2558

Hornik. K., Stinchcombe, M and White, H (1989), 'Multilayer Feedforward Networks are Universal Approximators', *Neural Networks*, Vol. 2, Issue 5, pp. 359-366

Hosking, J. and Wallis, J. (1987), 'Parameter and Quantile Estimation for the Generalized Pareto Distribution', *Technometrics*, 29, 3, 339-349.

Hussain, A., Ghazali, R and Al-Jumeily D. (2006), 'Dynamic Ridge Polynomial Neural Network for Financial Time Series Prediction', IEEE International conference on Innovation in Information Technology, IIT06, Dubai.

Husmeier, D. (1999), *Neural Networks for Conditional Probability Estimation - Forecasting Beyond Point Predictions (Perspectives in Neural Computing)*, Springer, London.

Hutchinson, M., Lo, W. and Poggio, T. (1994), 'A Nonparametric Approach to Pricing and Hedging Derivative Securities via Learning Networks', *The Journal of Finance*, 49, 851-889.

Kaastra, I. and Boyd, M. (1996), 'Designing a Neural Network for Forecasting Financial and Economic Time Series', *Neurocomputing*, 10, 215-236.

Kamijo, K. and Tanigawa,T. (1990), 'Stock Price Pattern Recognition: A Recurrent Neural Network Approach', *In Proceedings of the International Joint Conference on Neural Networks*, 1215-1221.

Karayiannis, N. and Venetsanopoulos, A. (1994), 'On The Training and Performance of High-Order Neural Networks', *Mathematical Biosciences*, 129, 143-168.

Kim, H. and Shin, K. (2007), 'A Hybrid Approach Based on Neural Networks and Genetic Algorithms for Detecting Temporal Patterns in Stock Markets', *Applied Soft Computing*, 7, 2, 569-576.

Knowles, A., Hussein, A., Deredy, W., Lisboa, P. and Dunis, C. L. (2005), 'Higher-Order Neural Networks with Bayesian Confidence Measure for Prediction of EUR/USD Exchange Rate', *CIBEF Working Papers*. Available at www.cibef.com.

Kosmatopoulos, E., Polycarpou, M., Christodoulou, M. and Ioannou, P. (1995), 'High-Order Neural Network Structures for Identification of Dynamical Systems', *IEEE Transactions on Neural Networks,* 6, 422-431.

Kupiec, P. (1995), 'Techniques for Verifying the Accuracy of Risk Measurement Models', *Journal of Derivatives*, 2, 173–18.

Lee, T. and Slatoglu, B. (2001), 'Evaluating the Predictive Performance of Value-at-Risk Models in Emerging Markets: A Reality Check', *Mimeo, University of California, Riverside*.

Lee, R. (2004), 'iJADE Stock Advisor: An Intelligent Agent Based Stock Prediction System using Hybrid RBF Recurrent Network', *IEEE Transactions on Systems, Man and Cybernetics*, 34, 3, 421-428.

Lindemann, A., Dunis, C., and Lisboa P. (2004), 'Level Estimation, Classification and Probability Distribution Architectures for Trading the EUR/USD Exchange Rate'. *Neural Network Computing & Applications,* 14, 3, 256-271.

Lindemann, A., Dunis, C. and Lisboa, P. (2005), 'Probability Distributions and Leveraged Trading Strategies: An Application of Gaussian Mixture Models to the Morgan Stanley Technology Index Tracking Fund', *Quantitative Finance*, 5, 5, 459-474.

Lisboa, P.. and Vellido, A. (2000), 'Business Applications of Neural Networks', *Business Applications of Neural Networks: The State-of-the-Art of Real-World Applications,* World Scientific, Singapore.

Liu, Y. (2005), 'Value-at-Risk Model Combination Using Artificial Neural Networks', *Ermory University Working Papers.*

Locarek-Junge, H. and Prinzler, R. (1998), 'Estimating Value-at-Risk Using Neural Networks', *Application of Machine Learning and Data Mining in Finance*, ECML'98 Workshop Notes, Chemnitz.

Lopez, J. (1998), 'Methods for Evaluating Value-At-Risk Estimates', *Federal Reserve Bank of New York Research Paper n. 9802.*

Malliaris, M. and Salchenberger, L. (1996), 'Using Neural Networks to Forecast the S&P 100 Implied Volatility',*Neurocomputing*, 10, 183-195.

McCulloch,W. and Pitts,W. (1943) 'A logical calculus of ideas immanent in nervous activity', *Bulletin of Mathematical Biophysics*, Volume 5, pp. 115-133.

McNees, S. (1995), 'Forecast Uncertainty: Can It Be Measured?', *Mimeo, Federal Reserve Bank of Boston*.

Meissner, G. and Kawano, N. (2001), 'Capturing the Volatility Smile of Options on High-Tech Stocks-A Combined Garch-Neural Network Approach', *Journal of Economics and Finance*, 25, 3, 276-291.

Neely, C. and Weller, P. (2002), 'Predicting Exchange Rate Volatility: Genetic Programming versus GARCH and RiskMetrics™', *Federal Reserve Bank of St. Louis*, 84, 3, 43–54.

Ozun, A. and Cifter, A. (2007), 'Nonlinear Combination of Financial Forecasts with Genetic Algorithm', *MPRA Paper 2488*, University Library of Munich, Germany.

Pan, H., Tilakaratne,C. and Yearwood,J. (2003), 'Predicting the Australian Stock Market Index Using Neural Networks Exploiting Dynamical Swings and Intermarket Influences', *AI 2003: Advances in Artificial Intelligence*, Spinger, Berlin.

Picklands, J. (1975), 'Statistical Inference Using Extreme Value Order Statistics', *Annals of Statististics*, 3, 119-131.

Pindyck, R. and Rubinfeld, D. (1998), *Econometric Models and Economic Forecasts*, 4th edition, McGraw-Hill, New York.

Pires, M. and Marwala, T. (2004), 'Option Pricing Using Bayesian Neural Networks', *In Proceedings of the Annual Symposium of the Pattern Recognition Association of South Africa*, Cape Town, 161-166.

Psaltis, D., Park, C. and Hong, J. (1988), 'Higher Order Associative Memories and their Optical Implementations.', *Neural Networks,* 1, 149-163.

Rau-Bredow,H. (2004), 'Value at Risk, Expected Shortfall, and Marginal Risk Contribution', *Risk Measures for the 21st Century*, Wiley Finance, Chichester.

Redding, N., Kowalczyk, A. and Downs, T. (1993), 'Constructive Higher-Order Network Algorithm that is Polynomial Time', *Neural Networks*, 6, 997-1010.

Samuel, Z. (2008), 'Value at Risk and Conditional Extreme Value Theory via Markov Regime Switching Models', *The Journal of Futures Markets*, 28, 2, 155-181.

Sarma, M., Thomas, S. and Shah, A. (2003), 'Selection of Value-at-Risk Models', *Journal of Forecasting*, 22, 4, 337-358.

Shapiro, A. F. (2000), 'A Hitchhiker's Guide to the Techniques of Adaptive Nonlinear Models', *Insurance, Mathematics and Economics*, 26, 119-132.

Shin, Y. and Ghosh, J. (1991) 'The Pi-Sigma Network: An Efficient Higher-Order Neural Network for Pattern Classification and Function Approximation', *Proceedings IJCNN*, Seattle, July, 13-18.

Singh, V. and Guo, H. (1995), 'Parameter Estimation for 3-Parameter Generalized Pareto Distribution by the Principle of Maximum Entropy (POME)', *Journal des Sciences Hydrologiques,* 40, 2, 165-181.

Taylor, J. (2000), 'A Quantile Regression Neural Network Approach to Estimating the Conditional Density of Multiperiod Returns', *Journal of Forecasting*, 19, 4, 299 – 311.

Tenti, P. (1996), 'Forecasting Foreign Exchange Rates Using Recurrent Neural Networks', *Applied Artificial Intelligence*, 10, 567-581.

Theil, H. (1966), *Applied Economic Forecasting*, North-Holland, Amsterdam, Netherlands.

Tino, P., Schittenkopf, C. and Doffner, G. (2001), 'Financial Volatility Trading Using Recurrent Networks', *IEEE Transactions in Neural Networks*, 12, 4, 865-874.

Vasillelis, G. and Meade, N. (1996), 'Forecasting Volatility for Portfolio Selection', *Journal of Business Finance and Accounting*, 23,125-143.

Vecci, L., Piazza, F. and Uncini, A. (1998), 'Learning and Approximation Capabilities of Adaptive Spline Activation Neural Networks', *Neural Networks*, 11, 259-270.

Versace, M., Bhatt, R., Hinds. O. and Shiffer, M. (2004), 'Predicting the Exchange Traded Fund DIA with a Combination of Genetic Algorithms and Neural Networks', *Expert Systems with Applications*, 27, 417-425.

Walsh, D. and Tsou, G. (1998), 'Forecasting Index Volatility: Sampling interval and Non-Trading Effects', *Applied Financial Economics*, 8, 477-485.

Yao, J., Poh, H. and Jasic,T. (1996), 'Foreign Exchange Rates Forecasting with Neural Networks', *International Conference on Neural Information Processing*, Hong Kong, 754-759.

Yao, J., Li,Y. and Tan,C. (1997), 'Forecasting the Exchange Rates of CHF vs USD Using Neural networks', *Journal of Computational Intelligence in Finance*, 5, 2, 7-13.

Yao, J., Li Y and Tan, C. (2000), 'Option price forecasting using neural networks',*Omega*, 28, 455-466.

Zhang, D., Jiang, Q. and Li,X. (2005), 'A Hybrid Mining Model Based on Neural Network and Kernel Smoothing Technique', *Computational Science – ICCS 2005*, Spinger, Berlin.

Zhang, M., Zhang, J., and Fulcher, J. (2000). 'Higher-order neural network group models for data approximation', *International Journal of Neural Systems,* Vol 10, pp. 123-142.

Zhang, M., Xu, S., X. and Fulcher, J. (2002), 'Neuron-Adaptive Higher Order Neural-Network Models for Automated Financial Data Modelling', *IEEE Transactions on Neural Networks*,13,1, 188-204.

Zhu, X., Wang,H., Xu,L. and Li,H. (2008), 'Predicting Stock Index Increments by Neural Networks: The role of trading volume under different horizons', *Expert Systems with Applications*, 34, 3043–3054.

APPENDIX

A.1.1 Performance Measures

The performance measures are calculated as follows:

Performance Measure	Description	
Annualised Return	$R^A = 252 * \frac{1}{N}\sum_{t=1}^{N} R_t$ *with* R_t *being the daily return*	[45]
Cumulative Return	$R^C = \sum_{t=1}^{N} R_t$	[46]
Annualised Volatility	$\sigma^A = \sqrt{252} * \sqrt{\frac{1}{N-1} * \sum_{t=1}^{N}\left(R_t - \overline{R}\right)^2}$	[47]
Sharpe Ratio	$SR = \frac{R^A}{\sigma^A}$	[48]
Maximum Drawdown	*Maximum negative value of* $\sum\left(R_t\right)$ *over the period* $MD = \underset{i=1,\cdots,t;t=1,\cdots,N}{Min}\left(\sum_{j=i}^{t} R_j\right)$	[49]

Table 41: Trading simulation performance measures

A.1.2 Results of Alternative Benchmark Models (Chapter 4)

		NAIVE	MACD	ARMA	LOGIT	MLP
Sharpe Ratio	*(excluding costs)*	1.83	0.97	1.10	1.81	2.57
Annualised Volatility	*(excluding costs)*	11.6%	11.7%	11.7%	11.6%	11.6%
Annualised Return	*(excluding costs)*	21.3%	11.3%	12.9%	21.1%	29.7%
Maximum Drawdown	*(excluding costs)*	-9.1%	-7.8%	-10.1%	-5.8%	-9.1%
Positions Taken	*(annualised)*	109	22	112	123	118

Table 42: Out-of-sample trading performance results for traditional models as reported by Dunis and Williams (2003, table 1.20, p. 35)

A.1.3 Networks Characteristics (Chapter 4)

Below are presented the characteristics of the networks for the different architectures that presented the best statistical performance on the training and on the test sub-period and that we used on this chapter.

	Reccurent	**HONNs**	**Psi Sigma**
Learning algorithm	*Gradient descent*	*Gradient descent*	*Gradient descent*
Learning rate	*0.001*	*0.001*	*0.5*
Momentum	*0.003*	*0.003*	*0.5*
Iteration steps	*500*	*500*	*500*
Initialisation of weights	*N(0,1)*	*N(0,1)*	*N(0,1)*
Input nodes	*10*	*10*	*10*
Hidden nodes (1layer)	*5*	*N.A*	*5*
Order	*N.A*	*3*	*4*
Output node	*1*	*1*	*1*

Table 43: Network characteristics

A.1.4 Empirical Results (Chapter 4)

The table below shows the results of the filtered trading strategy applied to the test dataset for different values of *d*. We choose the threshold that gives the highest return.

Selection of the optimal threshold based on the test period	Threshold									
	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45
RNN	**28.4% (2.93)**	27.8% (2.39)	12.9% (2.87)	4.36% (1.81)	1.43% (0.98)	-0.6% (-1.1)	-0.5% (-0.9)	-0.1% (-0.2)	0.2% (1.28)	0.3% (1.29)
HONN	**22.5% (2.31)**	17.0% (1.94)	13.9% (1.76)	5.99% (0.88)	-4.8% (-0.9)	-0.1% (-0.1)	-0.2% (-0.1)	0.5% (0.2)	-0.9% (-0.4)	0.6% (0.37)
Psi Sigma	**24.9% (2.51)**	21.8% (2.12)	14.1% (3.74)	4.3% (2.37)	1.44% (1.3)	0.52% (0.92)	0.51% (0.92)	0.52% (0.92)	0.00% (0.00)	0.00% (0.00)

Table 44: Results for alternative threshold values

Note: The entries represent the annualized return values while the values in parenthesis represent the Sharpe ratio.

The table below shows the results of the filtered trading strategy applied to the test dataset for different values of *d* after taking leverage into account. We choose the threshold that gives the highest return.

Selection of the optimal threshold based on the test period	Threshold									
	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45
RNN	29.2% (2.93)	**29.7% (4.3)**	13.2% (2.87)	4.5% (1.81)	1.47% (0.98)	-0.6% (-1)	-0.5% (-0.9)	-0.1% (-0.2)	0.3% (1.3)	0.27% (1.3)
HONN	**23.2% (2.31)**	17.5% (1.94)	14.3% (1.76)	6.17% (0.88)	-4.9% (-0.3)	-0.1% (-0.2)	-0.2% (-0.1)	0.52% (0.2)	-0.9% (-0.4)	0.7% (0.4)
Psi Sigma	**25.1% (2.5)**	22.4% (2.1)	14.3% (3.74)	4.45% (2.5)	1.48% (1.3)	0.52% (0.93)	0.52% (0.93)	0.52% (0.93)	0.00% (0.00)	0.00% (0.00)

Table 45: Results for alternative threshold values

Note: The entries represent the annualized return values while the values in parenthesis represent the Sharpe ratio.

A.2.1 Networks Characteristics (Chapter 5)

We present below the characteristics of the networks used in chapter 5.

	MLP	Reccurent	HONNs	Psi Sigma
Learning algorithm	*Gradient descent*	*Gradient descent*	*Gradient descent*	*Gradient descent*
Learning rate	*0.001*	*0.001*	*0.001*	*0.5*
Momentum	*0.003*	*0.003*	*0.003*	*0.5*
Iteration steps	*1000*	*1000*	*1000*	*500*
Initialisation of weights	*N(0,1)*	*N(0,1)*	*N(0,1)*	*N(0,1)*
Input nodes	*12*	*12*	*12*	*12*
Hidden nodes (1layer)	*7*	*5*	*NA*	*6*
Order	*NA*	*NA*	*3*	*4*
Output node	*1*	*1*	*1*	*1*

Table 46: Network characteristics

A.2.2 ARMA Model (Chapter 5)

The output of the ARMA model used in chapter 5 is presented below.

Dependent Variable: RETURNS
Method: Least Squares
Date: 02/29/08 Time: 12:38
Sample (adjusted): 13 1920
Included observations: 1908 after adjustments
Convergence achieved after 12 iterations
Backcast: 1 12

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	6.35E-05	0.000151	0.420579	0.6741
AR(1)	-0.688152	0.040492	-16.99463	0.0000
AR(2)	-0.369020	0.067865	-5.437592	0.0000
AR(7)	-0.218734	0.073635	-2.970535	0.0030
AR(11)	-0.400372	0.043749	-9.151508	0.0000
AR(12)	-0.539713	0.052040	-10.37107	0.0000
MA(1)	0.692697	0.034799	19.90592	0.0000
MA(2)	0.349884	0.064745	5.404045	0.0000
MA(7)	0.248779	0.073280	3.394927	0.0007
MA(11)	0.397565	0.039306	10.11460	0.0000
MA(12)	0.584116	0.052393	11.14877	0.0000

R-squared	0.014757	Mean dependent var		7.04E-05
Adjusted R-squared	0.009563	S.D. dependent var		0.006518
S.E. of regression	0.006487	Akaike info criterion		-7.232214
Sum squared resid	0.079834	Schwarz criterion		-7.200196
Log likelihood	6910.533	F-statistic		2.841359
Durbin-Watson stat	2.006369	Prob(F-statistic)		0.001639

Inverted AR Roots	.89-.28i	.89+.28i	.61+.70i	.61-.70i
	.14+.98i	.14-.98i	-.37+.89i	-.37-.89i
	-.73+.67i	-.73-.67i	-.89+.16i	-.89-.16i
Inverted MA Roots	.90-.28i	.90+.28i	.62+.70i	.62-.70i
	.14+.98i	.14-.98i	-.37+.89i	-.37-.89i
	-.73-.68i	-.73+.68i	-.90+.16i	-.90-.16i

A.2.3 Empirical Results in the Training and Test Sub-Periods (Chapter 5)

		NAIVE	MACD	ARMA	MLP	RNN	HONN	Psi Sig
Sharpe Ratio	*(excluding costs)*	-0.22	2.49	1.20	0.41	0.34	0.51	0.54
Annualised Volatility	*(excluding costs)*	10.28%	10.24%	10.33%	10.36%	10.35%	10.36%	10.35
Annualised Return	*(excluding costs)*	-2.25%	25.45%	12.40%	4.27%	3.51%	5.31%	5.55
Maximum Drawdown	*(excluding costs)*	-29.39%	-5.96%	-9.69%	-21.42%	-21.37%	-29.79%	-20.31
Positions Taken	*(annualised)*	77	10	53	77	97	52	75

Table 47: In-sample trading performance

A.2.4 Threshold Selection (Chapter 5)

The table below shows the results of the filtered trading strategy applied to the test dataset for different values of *d*. We choose the threshold that gives the highest return.

Selection of the optimal threshold based on the test period	Threshold									
	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45
Naive	-1.1% (-0.1)	-3.7% (-0.4)	-2.2% (-0.3)	-1.7% (-0.2)	-2.5% (-0.4)	-1.2% (-0.2)	1.1% (0.2)	**4.5% (0.8)**	2.0% (0.4)	4.0% (0.8)
MACD	**5.3% (0.6)**	-0.7% (-0.1)	-2.5% (-0.5)	3.7% (1.1)	0.3% (0.1)	-0.2% (-0.8)	0.0% (0.0)	0.0% (0.0)	0.0% (0.0)	0.0% (0.0)
ARMA	6.6% (0.8)	3.9% (0.7)	**7.0% (2.0)**	0.5% (0.3)	0.0% (0.0)	0.0% (0.0)	0.0% (0.0)	0.0% (0.0)	0.0% (0.0)	0.0% (0.0)
MLP	6.3% (0.7)	8.7% (1.1)	9.2% (1.2)	1.7% (1.5)	**12.2% (1.8)**	11.1% (1.7)	6.2% (1.0)	5.4% (1.0)	3.9% (0.9)	-0.1% (-0.1)
RNN	**10.1% (1.1)**	8.1% (0.9)	5.6% (0.7)	4.9% (0.6)	4.2% (0.5)	5.4% (0.7)	4.6% (0.6)	7.5% (1.0)	6.6% (0.9)	8.7% (1.2)
HONN	**6.9% (0.8)**	4.2% (0.6)	2.9% (0.5)	3.6% (0.9)	1.3% (0.5)	1.7% (0.8)	0.4% (0.6)	0.2% (0.8)	0.2% (0.8)	0.0% (0.0)
Psi Sigma	6.3% (0.7)	5.0% (0.6)	5.2% (0.7)	5.6% (0.8)	10.4% (1.5)	10.1% (1.5)	**10.5% (1.6)**	9.7% (1.7)	7.7% (1.5)	5.3% (1.2)

Table 48: Results for alternative threshold values

Note: The entries represent the annualized return values while the values in parenthesis represent the Sharpe ratio.

A.3.1 Networks Characteristics (Chapter 6)

We present below the characteristics of the networks used in chapter 6.

	MLP	RNN	HONN
Learning algorithm	*Gradient descent*	*Gradient descent*	*Gradient descent*
Learning rate	*0.001*	*0.001*	*0.001*
Momentum	*0.003*	*0.003*	*0.003*
Iteration steps	*20000*	*20000*	*10000*
Initialisation of weights	*N(0,1)*	*N(0,1)*	*N(0,1)*
Input nodes	*12*	*12*	*12*
Hidden nodes (1layer)	*7*	*5*	*NA*
Order	*NA*	*NA*	*3*
Output node	*1*	*1*	*1*

Table 49: Network characteristics

A.2.2 ARMA Model (Chapter 6)

The output of the ARMA model used in chapter 6 is presented below.

Dependent Variable: RETURNS
Method: Least Squares
Date: 09/03/08 Time: 17:28
Sample (adjusted): 12 2303
Included observations: 2292 after adjustments
Convergence achieved after 59 iterations
Backcast: ? 0

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.000128	0.000128	1.002067	0.3164
AR(1)	-1.216599	0.042416	-28.68222	0.0000
AR(2)	-0.475940	0.081259	-5.857069	0.0000
AR(7)	-0.139565	0.046558	-2.997679	0.0027
AR(11)	0.197421	0.055779	3.539363	0.0004
MA(1)	1.213517	0.039315	30.86659	0.0000
MA(2)	0.473609	0.077725	6.093399	0.0000
MA(7)	0.152391	0.044815	3.400464	0.0007
MA(11)	-0.217830	0.054607	-3.989048	0.0001

R-squared	0.008541	Mean dependent var	0.000122
Adjusted R-squared	0.005067	S.D. dependent var	0.006169
S.E. of regression	0.006153	Akaike info criterion	-7.339687
Sum squared resid	0.086446	Schwarz criterion	-7.317159
Log likelihood	8420.282	F-statistic	2.458312
Durbin-Watson stat	1.997623	Prob(F-statistic)	0.011925

Inverted AR Roots	.75	.65+.46i	.65-.46i	.23+.77i
	.23-.77i	-.21-.82i	-.21+.82i	-.73-.67i
	-.73+.67i	-.94-.20i	-.94+.20i	
Inverted MA Roots	.76	.66+.46i	.66-.46i	.24+.78i
	.24-.78i	-.21+.83i	-.21-.83i	-.73-.68i
	-.73+.68i	-.94+.20i	-.94-.20i	

A.3.3 Empirical Results in the Training and Test Sub-period (Chapter 6)

		NAIVE	MACD	ARMA	MLP	RNN	HONN
Sharpe Ratio	*(excluding costs)*	-0.10	0.51	1.17	0.46	0.16	0.48
Annualised Volatility	*(excluding costs)*	10.36%	10.44%	10.42%	9.97%	10.38%	10.40%
Annualised Return	*(excluding costs)*	-1.07%	5.31%	12.19%	4.62%	1.67%	4.95%
Maximum Drawdown	*(excluding costs)*	-29.39%	-15.35%	-12.91%	-17.63%	-23.82%	-18.20%
Positions Taken	*(annualised)*	79	11	113	75	63	106

Table 50: Training sub-period trading performance

A.4.1 21-day FTSE 100 Volatilities (Chapter 7)

In the figure below we present the 21-day rolling annualised volatilities of the FTSE 100 returns from 1 January 2006 to 31 December 2008.

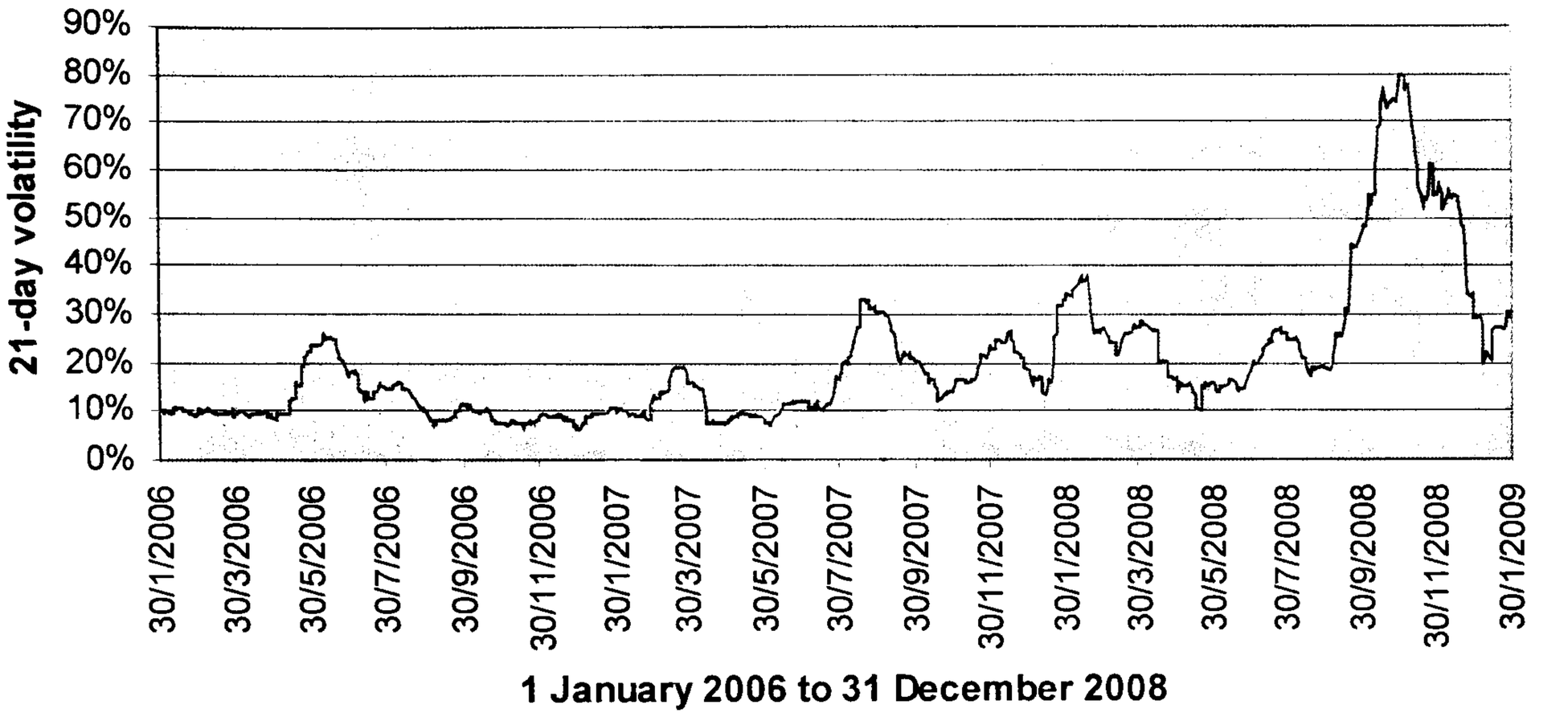


Fig. 22: 21-day annualised volatilities

We observe that there is a peak in the 21-day rolling volatilities as we approach the end of a FTSE 100 futures maturity month. We also note that the volatility during the last 6 months of 2008 is substantially higher than the period before. All this phenomena can lead to misspecifications to our NNs estimations as we are forced to use 1 year's data for each maturity in order to train sufficiently our models. However, as can been seen from table 12 our models seem robust to this anomaly and present statistically accurate forecasts.

A.4.2 Networks Characteristics (Chapter 7)

We present below the characteristics of the networks used in chapter 7.

	MLP	RNN	HONN
Learning algorithm	*Gradient descent*	*Gradient descent*	*Gradient descent*
Learning rate	*0.001*	*0.001*	*0.001*
Momentum	*0.003*	*0.003*	*0.003*
Iteration steps	*10000*	*5000*	*4000*
Initialisation of weights	*N(0,1)*	*N(0,1)*	*N(0,1)*
Input nodes	*4*	*4*	*4*
Hidden nodes (1layer)	*3*	*5*	*NA*
Order	*NA*	*NA*	*3*
Output node	*1*	*1*	*1*

Table 51: Network characteristics

A.4.3 Statistical Measures (Chapter 7)

The statistical measures are calculated as follows:

Performance Measure	Description	
Mean Absolute Error	$MAE = (\frac{1}{n}) \sum_{\tau=t+1}^{t+n} \lvert \hat{\sigma}_\tau - \sigma_\tau \rvert$	[50]
	with σ_τ being the actual volatility and $\hat{\sigma}_\tau$ the forecasted value	
Mean Absolute Percentage Error	$MAPE = \frac{1}{n} \sum_{\tau=t+1}^{t+n} \left\lvert \frac{\sigma_\tau - \hat{\sigma}_\tau}{\sigma_\tau} \right\rvert$	[51]
Root Mean Squared Error	$RMSE = \sqrt{\frac{1}{n} \sum_{\tau=t+1}^{t+n} (\hat{\sigma}_\tau - \sigma_\tau)^2}$	[52]
Theil-U	$Theil - U = \frac{\sqrt{(\frac{1}{n} \sum_{\tau=t+1}^{t+n} (\hat{\sigma}_\tau - \sigma_\tau)^2}}{\sqrt{\frac{1}{n} \sum_{\tau=t+1}^{t+n} \hat{\sigma}_\tau^2} + \sqrt{\frac{1}{n} \sum_{\tau=t+1}^{t+n} \sigma_\tau^2}}$	[53]

Table 52: Statistical measures

A.4.4 Out-of-sample Statistical Performance (Chapter 7)

In the table below we present the out-of-sample statistical performance of our models.

	RiskMetrics	MLPs	RNNs	HONNs
MAE	0.0663	0.0677	0.0498	0.0463
MAPE	26.27%	23.31%	17.84%	16.89%
RMSE	0.0937	0.0940	0.0765	0.0742
Theil-U	0.1607	0.1590	0.1339	0.1343

Table 53: Out-of-sample statistical performance

A.5.1 ARMA-GARCH(1,1) Models (Chapter 8)

Below is the output of the ARMA-GARCH(1,1) model for gold.

Dependent Variable: RETURNS
Method: ML - ARCH
Date: 06/27/08 Time: 16:07
Sample (adjusted): 11 1305
Included observations: 1295 after adjustments
Convergence achieved after 39 iterations
Bollerslev-Wooldrige robust standard errors & covariance
MA backcast: ? 0, Variance backcast: ON
GARCH = C(16) + C(17)*RESID(-1)^2 + C(18)*GARCH(-1)

	Coefficient	Std. Error	z-Statistic	Prob.
C	0.000649	0.000269	2.413482	0.0158
AR(1)	-0.741380	0.115455	-6.421403	0.0000
AR(2)	-0.493408	0.118689	-4.157155	0.0000
AR(3)	-0.482961	0.116673	-4.139425	0.0000
AR(4)	-0.459740	0.180691	-2.544340	0.0109
AR(5)	0.607182	0.187913	3.231196	0.0012
AR(9)	-0.176388	0.069070	-2.553755	0.0107
AR(10)	-0.615690	0.091550	-6.725173	0.0000
MA(1)	0.763123	0.117246	6.508705	0.0000
MA(2)	0.537017	0.117484	4.570988	0.0000
MA(3)	0.543289	0.120257	4.517746	0.0000
MA(4)	0.547401	0.185787	2.946393	0.0032
MA(5)	-0.570967	0.195965	-2.913620	0.0036
MA(9)	0.171212	0.070241	2.437485	0.0148
MA(10)	0.640706	0.097483	6.572484	0.0000
Variance Equation				
C	1.22E-06	3.62E-07	3.360921	0.0008
RESID(-1)^2	0.032150	0.009824	3.272623	0.0011
GARCH(-1)	0.955908	0.010019	95.40560	0.0000

R-squared	0.026836	Mean dependent var	0.000665
Adjusted R-squared	0.013881	S.D. dependent var	0.010226
S.E. of regression	0.010155	Akaike info criterion	-6.450380
Sum squared resid	0.131684	Schwarz criterion	-6.378571
Log likelihood	4194.621	F-statistic	2.071428
Durbin-Watson stat	2.042341	Prob(F-statistic)	0.006334

Inverted AR Roots	.81+.30i	.81-.30i	.41-.89i	.41+.89i
	.03-.95i	.03+.95i	-.74+.63i	-.74-.63i
	-.88-.44i	-.88+.44i		
Inverted MA Roots	.81+.30i	.81-.30i	.42-.91i	.42+.91i
	.03-.95i	.03+.95i	-.75-.64i	-.75+.64i
	-.89+.44i	-.89-.44i		

The output of the ARMA-GARCH(1,1) model for oil used in chapter 8 is presented below.

Dependent Variable: RETURNS
Method: ML - ARCH
Date: 06/29/08 Time: 12:05
Sample (adjusted): 3 1305
Included observations: 1303 after adjustments
Convergence not achieved after 500 iterations
Bollerslev-Wooldrige robust standard errors & covariance
MA backcast: ? 0, Variance backcast: ON
GARCH = C(6) + C(7)*RESID(-1)^2 + C(8)*GARCH(-1)

	Coefficient	Std. Error	z-Statistic	Prob.
C	0.001122	0.000542	2.068413	0.0386
AR(1)	0.400771	0.172971	2.316989	0.0205
AR(2)	-0.645944	0.168823	-3.826161	0.0001
MA(1)	-0.462364	0.163894	-2.821120	0.0048
MA(2)	0.689704	0.160640	4.293469	0.0000

Variance Equation

C	1.78E-05	7.95E-06	2.235393	0.0254
RESID(-1)^2	0.040061	0.017154	2.335444	0.0195
GARCH(-1)	0.917790	0.027824	32.98527	0.0000

R-squared	0.008869	Mean dependent var	0.000939
Adjusted R-squared	0.003511	S.D. dependent var	0.020854
S.E. of regression	0.020817	Akaike info criterion	-4.946019
Sum squared resid	0.561186	Schwarz criterion	-4.914262
Log likelihood	3230.332	F-statistic	1.655356
Durbin-Watson stat	2.023904	Prob(F-statistic)	0.116056

Inverted AR Roots	.20-.78i	.20+.78i
Inverted MA Roots	.23+.80i	.23-.80i

A.5.2 Networks specifications (Chapter 8)

Below we present the characteristics of the networks used in chapter 8. They are identical for both gold and oil.

	MLP	**HONNs**
Learning algorithm	*Gradient descent*	*Gradient descent*
Learning rate	*0.001*	*0.001*
Momentum	*0.003*	*0.003*
Iteration steps	*50000*	*30000*
Initialisation of weights	*N(0,1)*	*N(0,1)*
Input nodes	*10 (11 for the hybrid)*	*10 (11 for the hybrid)*
Hidden nodes (1layer)	*7*	*NA*
Order	*NA*	*3*
Output node	*1*	*1*

Table 54: Network characteristics

A.5.3 Christoffersen Test Results (Chapter 8)

		RiskMetrics	ARMA-GARCH	EVT	MLP	HONNS	MLP-RiskMetrics	HONNs-RiskMetrics
5% Confidence Level	*LRuc*	0.0913	0.0002	3.3744	0.2932	0.0913	0.2932	0.0913
	LRind	1.2560	0.2845	0.3104	0.1412	2.7288	3.1051	0.2845
	LRcc	1.3473	0.2847	3.6848	0.4344	2.8200	3.3988	0.3758
1% Confidence Level	*LRuc*	5.1068	3.2536	1.7430	0.6430	0.6430	0.0561	0.0561
	LRind	0.4419	0.3301	0.2349	0.1559	0.1559	0.0932	0.0932
	LRcc	5.5487	3.5837	1.9779	0.7989	0.7989	0.1494	0.1494

Table 55: Likelihood ratio statistics of gold for long positions

Note: The entries in bold represent rejection of the null hypothesis. In all other cases the null hypothesis is not rejected.

		RiskMetrics	ARMA-GARCH	EVT	MLP	HONNS	MLP-RiskMetrics	HONNs-RiskMetrics
5% Confidence Level	*LRuc*	1.7765	1.7765	**6.5732**	1.1537	0.3572	0.8133	0.8133
	LRind	0.2028	0.2028	0.9389	0.1482	1.0585	0.8785	0.8785
	LRcc	1.9763	1.9763	**7.5121**	1.3019	1.4158	1.6918	1.6918
1% Confidence Level	*LRuc*	1.7430	0.6430	**12.2981**	0.0561	0.0561	0.6430	1.7430
	LRind	0.2349	0.1559	0.8784	0.0932	0.0932	0.1559	0.2349
	LRcc	1.9779	0.7990	**13.1766**	0.1493	0.1494	0.7990	1.9779

Table 56: Likelihood ratio statistics of gold for short positions

Note: The entries in bold represent rejection of the null hypothesis. In all other cases the null hypothesis is not rejected.

		RiskMetrics	ARMA-GARCH	EVT	MLP	HONNS	MLP-RiskMetrics	HONNs-RiskMetrics
5% Confidence Level	*LRuc*	0.2932	0.8133	**15.082**	1.4776	0.3573	2.3726	0.0913
	LRind	1.9565	0.8785	0.0464	0.7159	1.0585	0.5704	1.2560
	LRcc	2.2497	1.6917	**15.129**	2.1935	1.4158	2.9430	1.3473
1% Confidence Level	*LRuc*	5.1068	NA	NA	1.7430	5.1069	3.2536	0.64304
	LRind	0.4419	NA	NA	0.2349	0.4419	0.3301	0.1559
	LRcc	5.5487	NA	NA	1.9779	5.5487	3.5837	0.7989

Table 57: Likelihood ratio statistics of oil for long position

Note: Entries in bold represent rejection of the null hypothesis. In all other cases the null hypothesis is not rejected. NA indicates that we had 0 violations and therefore we are unable to assess our model.

		RiskMetrics	ARMA-GARCH	EVT	MLP	HONNS	MLP-RiskMetrics	HONNs-RiskMetrics
5% Confidence Level	*LRuc*	0.6569	2.3726	**3.5261**	0.0913	0.0913	0.0913	0.2932
	LRind	1.0756	1.5122	1.9733	0.4198	1.2560	0.4198	1.9565
	LRcc	1.7325	3.8848	**5.4995**	0.5111	1.2560	0.5111	2.2497
1% Confidence Level	*LRuc*	0.6430	NA	NA	NA	0.6430	1.3113	0.0561
	LRind	0.1560	NA	NA	NA	0.1559	0.0154	0.0932
	LRcc	0.7990	NA	NA	NA	0.7990	1.3267	0.1494

Table 58: Likelihood ratio statistics of for with short position

Note: Entries in bold represent rejection of the null hypothesis. In all other cases the null hypothesis is not rejected. NA indicates that we had 0 violations and therefore we are unable to assess our model.