THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Multi-step Forecasts of Complex Dynamical Systems Using Soft-computing Tools, with Application to Crude Oil Returns

Imad HAIDAR

Bachelor of Economics
Master in Financial Markets
Master of Computing (Research)

A thesis submitted for the degree of Doctor of Philosophy at

The University of Queensland in 2014

Sustainable Minerals Institute
WH Bryan Mining and Geology Research Centre

Abstract

It is well known that crude oil plays a vital role in economic development. However, crude oil prices are sensitive to a large number of exogenous factors (such as speculation and OPEC behaviour) which result in short-term volatility shocks. This in turn makes it very difficult to forecast the price movement even in the short-term.

This thesis aims to build tools to determine the direction of forecast crude oil returns, multi-steps ahead. The goal is to exploit domain knowledge of the crude oil market dynamics, and incorporate them into a black-box model to improve forecast accuracy and to increase the forecasting horizon. This research is driven by inadequacies in current forecast methods, and their adverse economic impacts. Our investigation begins by running a battery of tests to understand the underlying structure of crude oil prices and returns. For non-linearity in the structure of these series, we use an established test for independence, the BDS test. The Fuzzy Classifier System for non-linearity (FCS) proposed by Kaboudan (1999) and a time-domain test for non-linearity introduced by Barnett and Wolff (2005) are also used. Finally, we estimate the Lyapunov exponents to establish the existence of chaotic dependence in crude oil prices and returns. Our tests consistently show that the dynamic forces driving crude oil prices and returns are non-linear, and possibly of low dimension. Moreover, the FCS test shows evidence of high noise levels, with smoothing or noise reduction being necessary for achieving improved forecast accuracy. We conclude that it is possible to forecast the crude oil price using non-linear models providing noise control measures are applied; the best hit rate achieved for out-of sample was 61%. In addition, we present a number of constraints on the objective function to act as a direct form of domain knowledge and to guide the learning process of the model.

A further problem facing short-term (daily and weekly) crude oil price forecasting is that most of the fundamental variables, such as supply, demand, inventory and GDP, are recorded on monthly or quarterly bases. This leaves us with a limited number of potential explanatory variables. This process would benefit from the incorporation of additional information hints to aid the forecasting process. We show several methods to create and assimilate new time series to act as supplementary information in the learning process for neural networks. These methods include: (i) using non-financial data from the search index information from Google Insight for Search for inclusion within the soft-computing model, (ii) creating a time series from OPEC meeting announcements using dummy variables and wavelet analysis, and (iii) using technical analysis transformation as domain-specific knowledge. Our results show the effectiveness of these methods, with some caveats.

Finally, we propose a novel multi-agent model for the crude oil market. The goal of this model is to generate hints that can be used to aid the training of traditional ANN. Therefore, we test whether the output of an artificial market can generate useful information to improve the learning process of traditional neural networks. The best hit rate we achieved using the forecast of these agents as additional input to ANN was 58%.

This thesis contributes to the body of literature by narrowing the gap between three interrelated fields: (i) energy economics, (ii) time-series econometrics, and (iii) soft-computing closer together in one structure.

Declaration by the author

This thesis is composed of my original work, and contains no material previously published or written by another person except where due reference has been made in the text. I have clearly stated the contribution by others to jointly-authored works that I have included in my thesis.

I have clearly stated the contribution of others to my thesis as a whole, including statistical assistance, survey design, data analysis, significant technical procedures, professional editorial advice, and any other original research work used or reported in my thesis. The content of my thesis is the result of work I have carried out since the commencement of my research higher degree candidature and does not include a substantial part of work that has been submitted to qualify for the award of any other degree or diploma in any university or other tertiary institution. I have clearly stated which parts of my thesis, if any, have been submitted to qualify for another award.

I acknowledge that an electronic copy of my thesis must be lodged with the University Library and, subject to the General Award Rules of The University of Queensland, immediately made available for research and study in accordance with the Copyright Act 1968.

I acknowledge that copyright of all material contained in my thesis resides with the copyright holder(s) of that material. Where appropriate I have obtained copyright permission from the copyright holder to reproduce material in this thesis.

Publications during candidature

In this section we list a number of publications and work in progress (current and anticipated) as an outcome of this research.

- Haidar, I., & Wolff, R. C., (2011). Forecasting Crude Oil Price (Revisited). In the Proceedings of the 30th USAEE/IAEE, 9-12 October 2011, Washington DC.
- Haidar, I., & Wolff, R. C., (2012). Forecasting crude oil price using soft-computing methods and Google Insight for Search. In the Proceedings of the 35th Annual IAEE International Conference, Perth, WA, 24-27 June 2012.
- Wolff, R. C., Haidar, I. & Weatherley, D., (2012) Improving geological estimation using soft-computing models. In the Proceedings of the 34th International Geological Congress (IGC), Brisbane, 5-10 August 2012.
- Haidar, I., & Wolff, R., C., (2013). Multi-agents model for crude oil market. **Submitted** to *Expert System with Application*.

Publications included in this thesis

Chapter 4: Forecasting Crude Oil Price (Revisited) and Forecasting Crude Oil Price: A New Look at Non-linear Dynamics.

Contributor	Statement of contribution
Imad Haidar (Candidate)	Designed experiments (100%) Wrote the paper (90%)
Rodney Wolff (Supervisor)	Designed experiments (0%) Wrote and edited paper (10%)

Chapter 6 Section 6.1: Forecasting crude oil price using soft-computing methods and Google Insight for Search.

Contributor	Statement of contribution
Imad Haidar (Candidate)	Designed experiments (100%) Wrote the paper (90%)
Rodney Wolff (Supervisor)	Designed experiments (0%) Wrote and edited paper (10%)

Chapter 7: Multi-agents model for crude oil market

Contributor	Statement of contribution
Imad Haidar (Candidate)	Designed experiments (100%) Wrote the paper (90%)
Rodney Wolff (Supervisor)	Designed experiments (0%) Wrote and edited paper (10%)

Contributions by others to the thesis

Professor Rodney Wolff, as Principal Supervisor of this thesis, provided normal supervisory intellectual input into this thesis [including suggestions for lines of enquiry and reviewing research as it progressed], and likewise for my Associate Supervisors.

I received editorial comments including grammar corrections for this thesis from a professional editors (Mrs Julie Martyn, Grammarfun: www.grammarfun.com.au and Mr. Carl Smith).

Statement of parts of the thesis submitted to qualify for the award of another degree

The literature review chapter (Chapter 2) contains a limited amount of material which was previously reviewed in my Master's thesis at The University of Ballarat 2008 (see Haidar, 2008). The same material also appeared in a number of publications of which I was the author/co-author.

Except for the above, and where an explicit reference was made in the body of the thesis, this work represents my original research and findings for the current PhD program.

Acknowledgments

I am very grateful, first and foremost, to my Principal Supervisor, Professor Rodney Wolff for providing phenomenal supervision and support in every aspect of my candidature. I learnt many things from Professor Wolff which transformed me from a student into a researcher. I am also grateful for the role Professor Wolff played on my behalf to obtain scholarships throughout my candidature; without these scholarships I would not be able to complete my PhD. I would like to thank my Associate Supervisors, Dr Radhika Lahiri and Dr Marcus Gallagher, for their time and all the meaningful comments and feedbacks they provided. I extend my thanks to Dr Suzanne Morris, the Postgraduate Coordinator at our institute for her support and the significant role she played organizing my milestones after I moved to this university. My thanks and appreciation go to the University of Queensland for the UQI Scholarship. I am also grateful to Queensland University of Technology for the fee-waiver scholarship during the time I was enrolled there. I would also like to thank the International Association for Energy Economy for granting me the OFID/IAEE Support Fund to present my paper at the 35th IAEE International Conference in Perth. Finally, my sincere thanks and appreciation to my former Associate Supervisors, Professor Heping Pan and Associate Professor Shlomo Giva, for the important roles they played during the first year of my candidature.

Keywords

Crude oil price, time series, non-linear models, soft-computing, energy market, forecasting, multi-agents model, supervised learning, reinforcement learning.

Australian and New Zealand Standard Research Classifications (ANZSRC)

ANZSRC code: 080108 Neural, Evolutionary and Fuzzy Computation, 60%

ANZSRC code: 140303 Economic Models and Forecasting, 20%

ANZSRC code: 140205 Environment and Resource Economics, 20%

Fields of Research (FoR) Classification

FoR code: 0801 Artificial Intelligence and Image Processing, 60%

FoR code: 1403 Econometrics, 20%

FoR code: 1402 Applied Economics, 20%

Table of contents

List of Tables

List of Figures

List of frequently used abbreviations and acronyms

AIC	Akaike Information Criterion
ANN	Artificial Neural Networks
ARIMA	Auto-Regression Integrated Moving Average
BIC	Bayesian Information Criterion
BP	Back Propagation
DK	Domain Knowledge
EA	Evolutionary Algorithms
EIA	The Energy Information Administration
EMF	The Energy Modelling Forum
Eq	Equation
FF-ANN	Feed-Forward Artificial Neural Networks
GA	Genetic Algorithms
GARCH	Generalized Autoregressive Conditional Heteroscedastic
GDP	Gross Domestic Product
GFC	Global Financial Crisis
IC	Information Coefficient
KB	Knowledge Base
KBANN	Knowledge Base Artificial Neural Networks
MA	Moving Average
MLP	Multi-Layers Perceptron
NARX	Non-linear Auto-Regressive Exogenous variable
NEAT	NeuroEvolution of Augmenting Topologies
OECD	The Organisation for Economic Co-operation and Development
OPEC	The Organization of the Petroleum Exporting Countries
RNN	Recurrent Neural Network
SVM	Support Vector Machine
VAR	Vector Auto-Regression (not used as Value at Risk)
VC	Vapnik–Chervonenkis dimension
WTI	West Texas Intermediate

CHAPTER 1: Introduction

1.1 Introduction

Crude oil plays a vital role in economic development, and forecasting prices is important for hedging against fluctuations in the market. However, the crude oil market is volatile and sensitive to a large number of factors, which cause the price to move sharply over short periods. The high volatility makes it difficult to predict the crude oil price direction even for the short-term. The body of literature concerned with crude oil forecasting and modelling is substantial. Previous traditional models which dealt with long-term forecasting did not achieve acceptable results, while the majority of the recent literature concentrates on short-term forecasts. We argue that there is a gap between soft-computing time-series modelling and statistical modelling. Soft-computing models, mainly Artificial Neural Networks (ANN) and Support Vector Machines (SVM), which are complex universal function approximations, are often applied directly (with the exception of a few studies) with the implicit assumption that whatever the underlying dynamics of the series are, these models will fit them well.

1.2 Aims and Objectives

This research aims to build tools to forecast crude oil daily returns for the short-term. Particularly, we want to utilise knowledge of the system dynamics to improve forecast accuracy. The goal is to provide a reliable prediction of crude oil price directions and to test for how long a reliable forecast can be achieved.

Forecasting the crude oil price is critically important for a variety of reasons. The price of crude oil at any given time could affect the price of other oil products (petrol and diesel, amongst others) and, to some extent, the price of crude oil derivative products such as natural gas. Therefore, predicting the movements in the crude oil price should help policy makers, energy market participants, and small and medium companies like petrol retailers and other groups to hedge their positions. The motivation of this research is driven by the potential impact of crude oil price prediction on the economy.

1.3 Research Problem

We aim to formulate a model to forecast noisy, complex time series for multi-steps ahead using soft-computing methods. The noise of the series is likely due to speculation in the market, OPEC cartel behaviour and other external events. Moreover, the useful financial and commodity time series are often limited, non-stationary and have a variety of patterns (Abu-Mostafa, 1995a). When trying to forecast such a series using artificial neural networks, which is the scope of this research, one is faced with the problem of poor generalisation (poor out-of-sample forecasts). This is because for noisy data, there may be a large number of functions which seem appropriate but which are not (Abu-Mostafa, 1995a; Weigend, Zimmermann, & Neuneier, 1995). This issue raises the need to constrain the network during the learning process: early stopping, regularisation, pruning, and *hints* are some of the methods that are frequently used for this purpose (Abu-Mostafa, 1995a; Weigend, Zimmermann, & Neuneier, 1995).

1.4 Research Scope

It is important to note that our goal in this research is not to test whether model a is superior to model b in its general form (e.g., whether SVM is superior to ANN), but rather to test if we are able to use the knowledge about the crude oil market, soft-computing methods and statistical inferences to improve short-term time-series prediction. Hence, model selection in its generic form will make little difference to our research question, providing that the model is theoretically suitable for the problem at hand. To explain this issue we fall back upon the *"No free lunch theorem"* from Wolpert and Macready (1997). The authors argue that there is no one optimisation algorithm that could perform superbly across all classes of problems. Therefore, on average, the performance of any given algorithm over all classes of problems should be constant (Wolpert & Macready, 1997). This can be explained by the following equation which compares two models, $a_1$and a_2 (Wolpert & Macready, 1997, p. 67):

$$\sum_f P(d_m|f,m,a_1) = \sum_f P(d_m|f,m,a_2)\,, \tag{1.1}$$

where P is the averaged performance, m is the number of algorithm iterations, d_m is the time-ordered set of m distinct points visited, and f is the combinatorial optimisation problem (Wolpert & Macready, 1997, p. 67). As such, for a given performance metric $\partial(d_m)$, the average performance output of all functions f of $P(\partial(d_m)|f,m,a)$ will be independent from a (Wolpert & Macready, 1997, p. 67). Therefore, it could be concluded that in order to give any black-box model an edge over another, i.e., to achieve better results, domain knowledge of the specific problem needs to be effectively exploited and embedded within the context of the soft computing methods employed (Bonissone, Subbu, Eklund, & Kiehl, 2006; Ho & Pepne, 2001).

In addition, the objective of this thesis is not to create a model that generates financial profit; rather, the goal is to provide a proof of concept for new algorithms that can be used for future forecasting problems, both financial and non-financial. Therefore, issues such as transaction costs were omitted from the forecasting models, which follow the majority of published research in this area.

1.5 Definition

There is no consensus in the literature on one definition for *Domain knowledge* (DK) as Chapter 2 of this thesis shows. The literature discussed in Chapter 2 describes various strategies such as DK. In this research we define two types of DK: energy economic and soft-computing. Energy economic DK aims to exploit the behaviour of the energy market and uses time-series dynamics for improving the forecast, while soft-computing DK helps in efficient model construction.

1.5.1 Domain knowledge representation

The next important issue is how to embed DK within the soft-computing model. As discussed in the literature review (Chapter 2), there are several ways to do so; we focus our attention on five main approaches:

1. problem representation
2. non-financial data
3. constraints
4. architectural enhancements
5. artificial markets.

1.5.1.1 Problem representation

Problem representation includes:

- feature selection
- data pre-processing
- input-output representation
- noise control.

Feature selection is an essential issue, regardless of the forecasting tool being used. Above all, theoretical justification needs to be established before including any variables in the model (within the restrictions imposed on us by the availability of these variables). Statistical and evolutionary methods are useful in finding the most informative variables to be used as input. Data pre-processing, on the other hand, involves transforming the input and/or output in order to reduce noise and emphasize behaviours of interest (e.g., for a mean-reverting series) (Azoff, 1994; McNelis, 2005; Neuneir, & Zimmermann, 1998). On the other hand, input-output and feature selection are crucial issues for the learning process. Another issue is how to select the length of the subset for training and testing.

1.5.1.2 Non-financial examples

Another way of embedding domain knowledge into ANN is to use non-financial examples. This approach can supplement the limited amount of useful financial and commodity data and could help ANN during the training process.

1.5.1.3 Constraints

The constraints in this research are presented as a part of the fitness function for the reinforcement learning model.

1.5.1.4 Architectural enhancements

Another way of including DK is by enhancing the network topology in a way that leads to more of the relevant information being processed by the network and more of the irrelevant information being ignored (Grothmann, 2002; Neuneir, & Zimmermann, 1998). This approach is closely related to some of the methods discussed above, such as data pre-processing, outlier management, noise control, amongst others.

1.5.1.5 Hints from artificial agents

We propose a multi-agent model for the crude oil market forecast. This model takes advantage of Grothmann's (2002) multi-agent neural network model and LeBaron, et al.'s, (1999) multi-agent genetic algorithm and combines them with the concept of neuro-evolution, namely NEAT, in order to reach a more realistic representation of the crude oil market. In this model, DK incorporation is accomplished by: (i) the model design, and (ii) the output of the artificial market representing a virtual example for another ANN model. In other words, the outcome of each agent in our model along with the new return series, i.e., the return series created as a result of the interaction of all agents in the first stage, represents a form of DK to train a supervised soft-computing model.

1.6 **Research question**

The research questions to be addressed in this project are:

Can we forecast complex economic systems like crude oil prices and returns, multi-steps ahead, using DK soft-computing models?

This question covers the following:

1. What types of dynamics are governing crude oil prices and returns?

2. From a statistical point of view, have the dynamics of crude oil returns changed significantly during the past twenty years?

3. Do we have strong empirical evidence that crude oil spot returns are predictable in the short-term?

Does domain knowledge expertise improve the prediction output of a soft-computing model of complex economic systems like the crude oil price?

This question can be divided into sub-questions:

1. Can we formulate our knowledge of crude oil market dynamics into constraints for soft-computing models to:

   (a) improve the forecast accuracy and

   (b) increase the forecast horizon?

Can a multi-agent model based on ANN generate a better forecast of the crude oil price?

1. Would the multi-agent model forecast be better than traditional ANN?
2. Would the combined output of all agents be a useful source of hints to train ANN?

The anticipated principal contribution of this thesis is to the field of energy economics, while the secondary contribution of this work will be to the fields of time-series econometrics and soft-computing.

1.7 Contribution to knowledge

We believe this work provides soft-computing strategies to forecast crude oil returns. More specifically, our contribution to knowledge can be summarized as follows:

1 **Discovering structure in crude oil time series which contradicts the existing knowledge**. In this research we present a comprehensive analysis of the crude oil time series. This includes an analysis of the dynamics of crude oil time series through non-linearity testing and testing for chaos, and also through regime-switching models. The effects of OPEC announcements on crude oil price were also investigated. We also find evidence to support the view that crude oil price dynamics are non-linear chaotic, which explains the seemingly random behaviour of crude oil return. We believe that this analysis contributes to the energy market literature with a new perspective.

2 **New techniques to forecast the crude oil price.** On the modelling side (Figure 1-1), our methodology demonstrates that combining different modelling polices has the potential to generate superior results compared to using only one model. This idea is not new by itself; however, to the best of my knowledge, no other research has effectively combined as many strategies as we have achieved. To elaborate, we combine a local search method, namely the gradient descent, with the global search of genetic algorithms. This helps to capitalize on the advantages of each model type and to reduce the effects of limitations. Fuzzy logic and wavelet analysis also have their place within our methodology. On the other hand, we also combine econometric methods (time series) with macro-economic methods (multi-agent models)[1].

3 **Developing a new strategy for hint incorporation**. Although the concept of hints for soft-computing models is already established in the literature (see the literature review section), we show some techniques based on soft-computing methods to create hints for crude oil market prices. Some of these techniques are not limited to crude oil return and can be used with other financial time series.

4 **Development of novel fitness functions**. We introduce new fitness functions (error functions) for training soft-computing models. These fitness functions are designed to strike a balance between the model error, i.e., the goodness of fit, and the risk-adjusted return, namely the Sharpe ratio. We also believe that the way we incorporate the Sharpe ratio into this fitness function is original, as we rely on fuzzy rules (this represents a method of incorporating domain knowledge into the model).

5 **New multi-agent model for the crude oil market.** We introduce a new multi-agent model for the crude oil market that combines the micro-economic side and the macroeconomic side of the market in one framework. To the best of our knowledge, this is the first time the crude oil market has been modelled using the multi-agent approach. We show how the output of this model can be used as supplementary source of input to ANN.

[1] For more about this issue, see Grothmann (2002).

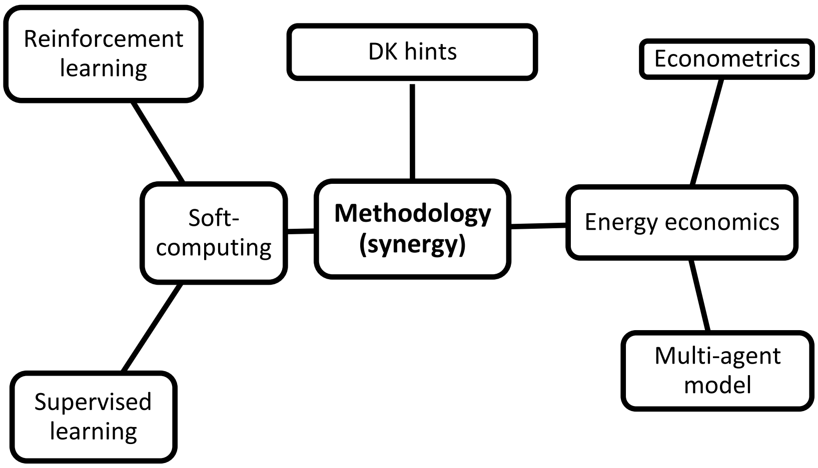

Figure 1-1: A block diagram of our methodology
This figure shows all methods of learning and modelling used in this thesis.

1.8 Organization of the document

This document proceeds as follows. In Chapter 2 we present a comprehensive literature review. The literature review covers previous models in the energy market and also soft-computing models. We also review the knowledge-based model and its application in the literature. In Chapter 3 we introduce Artificial Neural Networks (ANN) and the NeuroEvolution of Augmenting Topology (NEAT), and we also present novel extensions for NEAT. We close the chapter with a description of the performance metrics of the empirical analysis. Chapter 4 starts with a detailed analysis for the non-linear dynamics of crude oil series. Chapter 5 shows how problem representation can be used to improve the generalization of traditional ANN. In Chapter 6, we test two non-financial case studies, one based on data obtained from Google Insights for Search (now known as Google Trends) and the other based on artificial data we constructed to account for OPEC behaviour. Chapter 7 presents a novel multi-agent model for the crude oil market. This model is used to generate supplementary input to train ANN. Finally, the thesis concludes in Chapter 8. Figure 1-2 presents a flow chart to illustrate how each chapter connects within the context of the thesis.

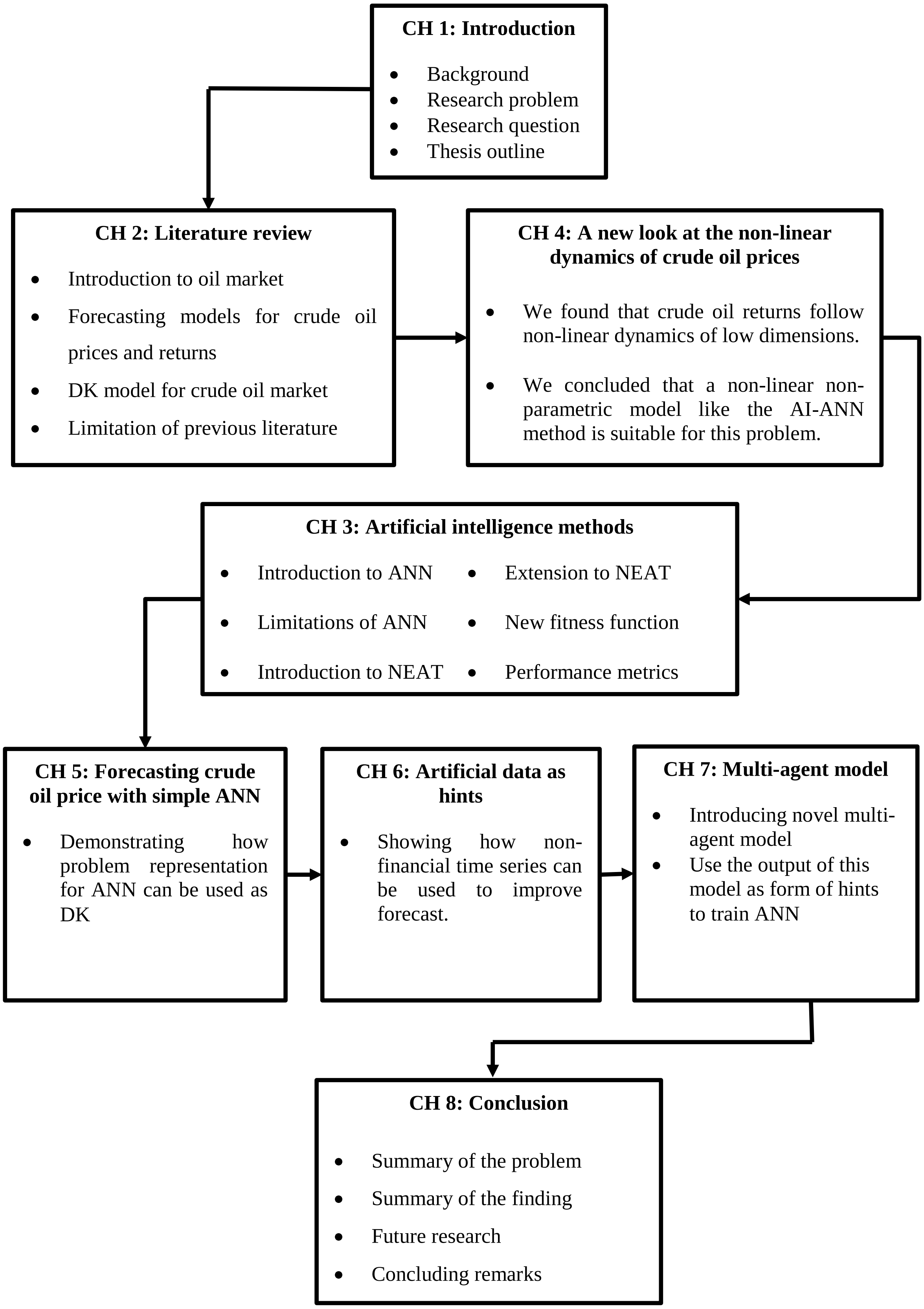

Figure 1-2: A flowchart of the thesis

This chart shows how each chapter connects within the context of the thesis. It shows the main issues studied, key results, and how they connect. The direction of the arrows indicates the logical flow of the thesis.

CHAPTER 2: Literature Review

2.1 Introduction

This thesis is concerned with developing a methodology to improve the forecast accuracy of crude oil returns over multi-time steps, relying on soft-computing models such as ANN. As such, this research falls among three major interrelated disciplines: energy economics, time-series econometrics and soft-computing (Figure 2-1).

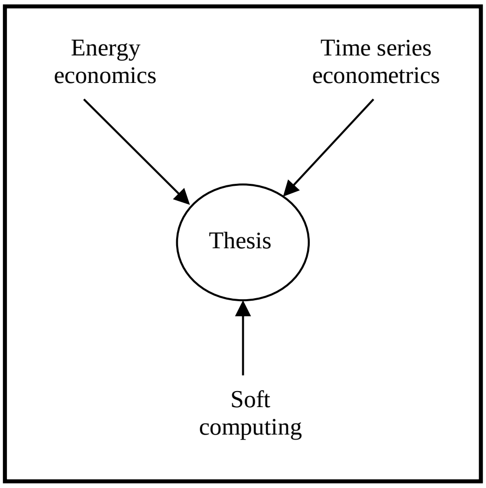

Figure 2-1: Positioning of this research in the context of three major disciplines

What follows is an overview of the major developments related to crude oil price forecasting, focusing on the significant impact of crude oil on the economy, and the usefulness of soft-computing models for time-series forecasting; specifically we concentrate on crude oil forecasting.

2.2 Energy market dynamics

In this section we try to scratch the surface of the energy market dynamics. Mainly we aim to show the significant influence of the crude oil commodity in both the macro- and micro-economy.

2.3 Oil price shocks and the macro-economy

The relation between the sudden change in the crude oil price (oil shock) and economic performance has been studied intensively. A large number of studies have emerged to investigate this issue. Most of these studies were conducted using Vector Auto-Regression (VAR) models. The main question was whether oil shocks caused any of the recessions in the US economy.

An early study by Hamilton (1983) investigated the effect of oil shocks on the US economy between 1948 and 1972[2]. Hamilton (1983) tested the hypothesis of existing linear causality between crude oil price shocks and the US economy as represented by six variables[3], namely: US real GNP, unemployment, implicit price deflator for non-farming business income, hourly compensation per worker, import prices and (M1) financial sector series (Hamilton, 1983). In an attempt to answer this question, Hamilton (1983) used bivariate Granger-causality tests; he then tested if any of the five historical oil shocks caused the economy to slow down. The results showed strong evidence that there was a *systematic relationship* between oil price shocks and each of the US recessions between 1948 and 1972. Moreover, he also established that there was no evidence to conclude that any of the recession episodes included in his study could have been caused by any variable other than oil shocks. The author explained this point by showing that none of the six variables tested above showed any indication of abnormal volatility before the oil shock occurred. Although Hamilton (1983) did not state it overtly, he implied that linear causality existed between each of the oil shocks and the subsequent recessions between 1948 and 1972. Finally, the author concluded that the severity of each of the recessions from 1948 to 1972 would have had to have been less significant if no oil shock had taken place. However, this study failed to explain why, each time, there were lags of three to four quarters between the oil shock and the recession.

In a related study, Mork (1989) noted that all oil shocks examined by Hamilton (1983) were positive oil shocks, in which the price of crude oil increased. Therefore, Mork (1989) tested whether the conclusion of Hamilton (1983) still stood when the data sample was extended to include the oil shock of 1973-74 and the oil price crash of the mid-1980s. As such, Mork (1989) used a similar VAR test, described by Hamilton (1983), but with a larger data set from 1949 to 1988. In addition, Mork (1989) proposed a "*stability*" test to account for the oil price decrease in the mid-1980s. Mork's (1989) results suggested that crude oil price increases have a more significant effect on the US economy than price decreases. This asymmetrical effect of crude oil shocks was later confirmed by the results of Lee and Ni (1995), who applied Mork's (1989) VAR model to even longer data samples. Lee and Ni (1995) also argued that the severity of oil shocks on the economy (namely the GNP and unemployment) was much higher when the shock was preceded by a low price volatility era than a highly volatile price era.

On the other hand, Hooker (1996a; 1996b) could not subscribe to Hamilton's (1983) conclusion and claimed that Granger-causality, concluded by Hamilton's (1983) study, did not hold any more when

[2] The author chose to exclude data from 1972 to 1981 as the pricing system for crude oil changed post-1972 which caused the oil price data to show signs of non-stationarity.
[3] These variables were originally proposed by Sims (1980) as a representation of the US economy.

data post-1972 were included in the model. Moreover, Hooker (1996b) argued that the linear structure of the VAR equation used by Hamilton (1983) and Mork (1989) could not capture the complex relationship between oil price and the macro-economy. Following this further, a more recent study by Hamilton (2003) proposed a flexible non-linear econometric model to re-test the relationship between crude oil price shocks and the US real GDP. The flexibility of Hamilton's new model derives from the notion that the function describing the changes in oil prices can take several forms, and therefore no *a priori* assumption about the shape of this function was made. The results of Hamilton (2003) showed that a sudden increase in oil price (positive price shock) has a more significant impact on the economy than a sudden decrease (negative price shock) of the same magnitude.

In conclusion, the body of evidence in the literature strongly supports the proposition that a positive crude oil price shock (price increase) affects the GDP negatively, while a decrease in oil price seems to have limited effect on economic performance. This conclusion highlights the potential benefit to the economy of accurately forecasting the crude oil price.

2.3.1 **The role of OPEC**

The Organization of Petroleum Exporting Countries (OPEC) was first formed in 1960. The aim was to prevent competition amongst members and to stop the monopoly of the private organisations that were then controlling the market, which were known as "*the seven sisters*" (Frey, Manera, Markandya, & Scarpa, 2009). These comprised five American companies and two of other nationalities, who had almost complete control of the energy market (Frey, et al., 2009). In the following few years the number of OPEC members grew to 15. During the 1960s OPEC's role was not well defined and it was not until the oil crisis of 1973 that OPEC was recognised as an influential cartel in the oil market. In 1970 the USA started to import oil for the first time in its history, because of the rapid growth in the US economy. In 1973 a few OPEC countries led by Saudi Arabia imposed an oil embargo on USA and some European countries as punishment for their support of Israel in the Yom Kippur war of that year (Frey, et al., 2009). Although, the motivation for the embargo is still an arguable issue, nonetheless, the effect of this embargo was devastating as the price of crude oil increased by 400% over just a few months (Crémer & Salehi-Isfahani, 1991; Frey, et al., 2009). As a consequence of the 1973-74 oil price shock and the subsequent recession in the US economy, a significant amount of research emerged to model OPEC's behaviours and influence on the oil market. The main question raised was: Does OPEC function as a cartel (oligopoly) or as a revenue-maximising group of countries?

It could be argued that OPEC also has contributed to price fluctuations, especially while OPEC countries produced about 38% of the world's crude oil. Fattouh (2005) found that although OPEC has an interest in not cutting production, the main problem is to anticipate demand and supply correctly, while there are no accurate data about the level of consumption, production, and stock. Moreover, according to Fattouh (2005), both the differences between OPEC countries' economic conditions and the absence of a penalty system within OPEC for member countries which do not follow the organisation's recommendations, are factors affecting oil prices. De Santis (2003) found that Saudi Arabia's economic welfare benefited from the increase in oil demand. In other words, Saudi Arabia has a great interest in keeping its crude oil prices high which can be achieved by cutting oil production when there is a negative shock and increasing the production when the price is high. However, De Santis (2003) suggested that this has a minor effect on oil prices because the world's oil producers will offset the Saudi oil supply in the short-term; nonetheless it still has an effect on long-term prices.

Some others argued that OPEC could be acting against its best interests as an organisation. As an example, Ruggeri (1983) found that OPEC had the opportunity to increase its revenue much more than it did during the 1973-1978 period. However, OPEC missed this chance because it failed to act as a united organisation. A detailed review and analysis of the role of OPEC and its history can be found in Mabro (1998) while a comprehensive survey about OPEC models can be found in Crémer and Salehi-Isfahni (1991) and Al-Qahtani, Balistreri, and Dahl (2008). Lin and Tamvakis (2010) conducted an empirical investigation to test the effect of OPEC conferences' announcements on crude oil prices. The study included 16 different grades of oil from 1st of May 1982 to the end of December 2008. During this period there were 87 OPEC conferences and 8 OPEC Ministerial Monitoring meetings (Lin & Tamvakis, 2010). Lin and Tamvakis (2010) used an event study methodology to measure the effect of the OPEC announcements on the crude oil daily price by investigating whether these announcements affected the cumulative abnormal return. In other words, after the announcements, if the cumulative abnormal return was significantly different from zero then the announcement was regarded to have had a direct impact on the price. The results of Lin and Tamvakis (2010) were mixed, as the impact of the events was different when OPEC reduced its quota than when OPEC increased it. Moreover, according to the authors, the existent price regime at the time of the announcements also affected the magnitude of the announcements' effect on the price. Finally, the author could not find any difference between the prices of different grades of oil immediately after announcements.

In conclusion, there is no consensus in the energy literature about the role OPEC plays in the oil market, i.e., whether it is a cartel, or whether one member controls the OPEC, namely, Saudi

Arabia. Notwithstanding, OPEC countries produce significant amounts of oil, around 38% of the world's total production, so the behaviour of OPEC in increasing or decreasing production has a direct effect on the crude oil price.

2.4 Modelling and forecasting the crude oil price

A substantial amount of research was directed towards studying several aspects of the energy market. Historically there were two occasions when crude oil price movement was the centre of attention for many scholars. The first one was in the late 1970s and early 1980s subsequent to the first major oil shock of 1973.

Energy market research can be roughly divided into three main categories based on the method of forecasting:

- structural models and econometric models, also including time-series analysis
- lead-lag models which deal with the relation between futures price and spot prices (this area is beyond the scope of this research
- computational/soft-computing models and hybrid models.

What follows is a brief survey on forecasting models for the crude oil market, concentrating mainly on soft-computing approaches.

2.4.1 Structural and econometric models

Following the oil price shocks of the 1970s, a large number of studies emerged to model and forecast crude oil movement in an attempt to isolate the effect of oil shocks. The Energy Modelling Forum (EMF) held at Stanford University (1982) presented one of the early studies in this area[4]. Their aim was to provide a long-term forecast of energy market prices as well as a forecast of the consumption and demand up till 2020. The Forum based their study on ten existing models to generate this forecast. These models came from diverse backgrounds: academic institutions and other government and non-government organisations.[5] These models can be divided into two groups, *"recursive simulation*" and "*intertemporal optimisation models"* (Energy Modeling Forum, 1982, p. 16). According to the Forum, seven out of the ten models were recursive simulation models. This type of model forecast is based on the information available until the time of forecast, i.e., the past and current information (Energy Modeling Forum, 1982). On the other hand, EMF

[4] For a summary of this report, see Gately (1984).

[5]The academic institutions are: NY University, MIT, Stanford University and The University of Texas. The other organizations and governmental institution included: US Department of Energy, US Federal Trade Commission, British Petroleum Corp. and Energy, Christian Michelsen Institute and Energy and Power Sub-committee: US House of Representatives (Energy Modeling Forum, 1982).

(1982) claims that optimisation models take into consideration (at least partially) future changes in the form of assumptions about expected future events for one or more sectors (Energy Modeling Forum, 1982). The forecast was generated under certain assumptions or scenarios about the economic growth level, OPEC supply and price elasticity, amongst others.

Huntington (1994) presented an analysis of the EMF's report (1982), in an attempt to provide a better understanding of those models, strengths and shortfalls. The author concentrated on the forecast outcome of these models for the 1980s and how accurately they predicted price, supply and demand. Generally, and according to Huntington (1994), the forecasts generated by the recursive simulation models were somewhat better than optimisation models. Nonetheless, all the forecasts generated by each and every model were incorrect, and most of the time distinguishably over-predicted the variables. For example when the author compared the forecast price to the actual price, the error was over 200% (Huntington, 1994). Furthermore, the author found that the EMF predictions of oil consumption, at any price level, were extremely high when compared to the actual, while the total consumption forecast figures were very close to the real ones, despite the fact that the price predictions were very far from the real ones. Moreover, the EMF estimations of the oil demand and non-OPEC supply conditions were incorrect, as they over-estimated the former and under-estimated the latter (Huntington, 1994). Finally, Huntington (1994) found that, although the two models developed by the US Department of Energy generated reasonably accurate forecasts of the 1990 levels of world consumption, no individual model performance stood out in predicting all the variables.

Every year the Energy Information Administration (EIA), part of the US Department of Energy, publishes an outlook of the energy market supply consumption and price for US sectors as well as for the world. The EIA publishes two types of projections, short-term and long-term outlooks. The Short-Term Energy Outlook (STEO) model covers the monthly forecast for up to 12 to 24 months ahead. This forecast generated by the Short-Term Integrated Forecasting model (STIFS) which is mainly a national model (for the USA), deals with several energy sectors: fuel, heating oil, electricity and natural gas, amongst others. On the other hand, the long-term projections are generated by the National Energy Modelling System (NEMS). This system is an integrated model which combines computer systems with economic modelling, and produces the projection for about 25 years ahead. This system operates under certain assumptions about the economic growth level, as well as the price level.

Since the EIA energy market outlooks attract widespread attention, a large number of studies investigated the accuracy of this model. Winebrake and Sakva (2006) presented an evaluation of the

EIA's model forecast of energy production and consumption for the period from 1982 to 2003. The authors argued that the main shortfall of previous studies that dealt with the EIA's forecast accuracy is that they assessed the total forecast error for all sectors: transportation, domestic and business amongst others. This could lead to a misleading conclusion about the real performance of the EIA's model forecast (Winebrake & Sakva, 2006). The authors proposed an error decomposing technique that allowed them to measure the extent of error, sector by sector. The authors found that by decomposing the error into sub-errors based on the sectors, a much higher level of error was revealed. The authors explained this as the EIA's forecast overestimated one sector and underestimated another sector by almost the same value, which resulted in a low level of overall error. In a related study, Sanders, Manfredo and Boris, (2009) investigated the accuracy of EIA forecasts for multi-horizons. The goal was to investigate for how many quarters in the future the EIA model provides a meaningful forecast. The authors define "meaningful" as the forecast with the minimum error and at the same time the error is independent and identically distributed (*iid)*, i.e., no serial correlation, absolutely random. The evaluation method used in this study was a multiple regression equation and the authors concluded that the EIA price forecast for crude oil and other related products was meaningful up to three quarters in the future.

Other researchers considered the role of inventory; Ye, Zyren, Blumberg and Shore (2009) (originally in Ye, Zyren, and Shore, 2005, 2006) analysed the relationship between crude oil price with each of the crude oil OECD inventory levels and the oil excess production capacity, employing 3-D graphical representations. The authors identified three different regimes governing the relationship between these variables over the period of 1992 to 2007. Regime 1 extended from 1992 until 1999 in which the market was stable and no significant change in the inventory was recorded. In addition, in Regime 1 there was a surplus of capacity production, and the WTI price was hovering around $20 per barrel (Ye, et al., 2009). According to the authors, from 1999 to 2004 OPEC attempted to gain control over the market, which resulted in a price hike of around $10 per barrel while no significant change in the inventory and the capacity was recorded; the authors called this period Regime 2. Finally, in Regime 3, which ran from 2004 till 2007, the price of WTI increased significantly, and was accompanied by a significant increase in the capacity of production. According to the authors these regimes can be explained by changes in the conditions of the oil market and OPEC policy adjustment, in addition to the change in the crude oil supply and demand needs. Ye, et al., (2009) further explained that from 1990 until early 2000, prediction of the crude oil price using OECD inventory levels (see as an example Ye, Zyren, & Shore, 2006) was possible due to OPEC's surplus in production capacity. In other words, during the period 1990-2003, OPEC had the means to increase the production to meet any sudden increase in the demand

(Ye, et al., 2009). However, the significant increase in demand from 2004 onward reduced OPEC's excess production capacity, and as a consequence, using OECD inventory alone to predict the price was no longer sufficient (Ye, et al., 2009).

Based on this analysis, Ye, et al. (2009) developed a new forecasting model for crude oil WTI monthly prices. The authors used the average nominal crude oil price without reporting any test for the time-series dynamics, e.g., unit root tests, to support their choice of model. In general, the new model of Ye, et al. (2009) was an upgrade of earlier models, Ye, et al. (2006) and Ye, et al. (2005) in which the same relative oil inventory level of OECD countries was used. However, the new variables used in this model were: the deviation of the excess capacity from its critical level CAP and the monthly cumulative excess production capacity over critical level CUMCAP[6] (Ye, et al., 2009). According to the authors and based on the forecast results, these two variables provided a significant improvement to the forecast permanence for in-sample (using dynamical forecast, in which the forecast value was used as input for the next step forecast). The authors also found that the CAP and CUMCAP had also improved the regression statistics estimation. It is important to note however, that the excess production capacity was an estimated figure and not an actual one. Another point is that the authors used the real price, in spite of the existence of a unit root. As such the improvement in the regression statistics recorded by the authors could be attributed to the unit root in the data, as it is well known that using a non-stationary time series can lead to spurious regression.

Other studies have focused on modelling crude oil market volatility, for example, Vo (2009) proposed a hybrid model to predict volatility of the crude oil short-term price. Vo's model was based on a combination of a Markov Switching (MS) model and a stochastic volatility model (SV), which was called MSSV. The author argued that the MSSV model has the ability to describe the volatility behaviour of a time series, especially in uncertain situations, i.e., market shocks. Vo (2009) used the weekly crude oil spot price for WTI from 1986 to 2008; after calculating the log return from the weekly data the author converted it into annual form to perform the estimation. The author fitted the SV, MS and MSSV model to the annual return crude oil data. Moreover, for the SV model the author used Bayesian MCMC as an alternative to the traditional two-step estimation. The goodness of fit results indicated that the MS model was the best to fit the data. Furthermore, the author compared the forecasting power of each of the models using three performance criteria, namely, RMSE, MAE, and Theil-U metrics. According to Vo (2009), the MSSV produced the best

[6] According to Ye et al. there will be no effect on the price if the excess capacity is greater than two million barrels per day (Ye, et al., 2009, p. 46,47).

out-of-sample results for the three forecasting metrics, while for in-sample forecasting the MS model was superior to the SV and MSSV in terms of the mean absolute error metric. Not deterred, the author argued that although the results for in-sample were inconsistent, based on the out-of-sample results, MSSV was clearly and consistently the best model to forecast the volatility for crude oil data. Finally, the author concluded that these results showed strong evidence for regime-switching in the oil market, and MSSV was an adequate model to forecast the volatility in the oil market. In a related study Kang, Kang, and Yoon (2009) investigated the performance of two GARCH type models (fractionally integrated-GARCH, and component-GARCH) to forecast the long-term volatility for Brent, Dubai and WTI crude oil prices. The authors concluded that the above models were superior to the standard GARCH/ IGARCH models in capturing the long-term volatility for the oil market. Cheong (2009) studied and compared the long-term volatility in WTI and Brent blend using an ARCH-type model. In a related study, Cheong (2009) concluded that while there is no evidence of the leverage effect in the WTI series, the Brent series does experience this leverage effect. Moreover, the author claims that for both WTI and Brent, there is evidence of a long lasting effect of the price swings on the volatility.

Choi and Hammoudeh (2009) tested the existence of long-term memory in crude oil spot returns, futures returns, heating oil spot returns and other related series. The test included the absolute and squared returns using both the autocorrelation function and FIGARCH model. Based on the results, the authors claimed that there was strong evidence that crude oil squared returns (for spot and futures) contain long-term memory. The authors went further by forecasting these series for one, five, ten and twenty days ahead using ARMA and Auto Regressive Fractional Integrated Moving Average (ARFIMA). According to Choi and Hammoudeh (2009), the out-of-sample results showed that ARFIMA generated more accurate forecasts than ARMA for all of the time series tested and for all horizons, as it produced lower errors. Moreover, for WTI spot returns, as well as for futures contracts one and two months to maturity, a twenty-steps-ahead forecast has the highest *t*-statistic over other horizons. This could imply the existence of a monthly pattern within the data. The same finding applies to gasoline returns. However, the authors ignored the evidence from the literature that oil and oil-related products series are mostly non-linear series, and therefore, choosing a linear model (such as ARMA) could provide unreliable results.

Sadorsky (2000) studied the relation between each of the monthly crude oil futures price, heating oil No. 2 futures prices and unleaded gasoline futures price with the trade-weighted US exchange rate from 1987 to 1997. Using Johansen's trace test, the author found strong evidence of co-integration[7]

[7] Co-integration analysis is defied in Chapter 6 section 6.2.1.

between each of the energy variables included in the test with the exchange rate, which implied a long-term relationship between these variables (Sadorsky, 2000). Furthermore, the VECM results revealed that for the short-term, the trade-weighted US exchange rate has temporal precedence over the changes in the crude oil futures price, while for the heating oil future price, this result was confirmed for the short and the long term (Sadorsky, 2000). Finally, the author claimed that the exchange rate has the potential to pass on shocks to the energy futures market.

In summary the energy literature is quite rich with several types of forecasting models, simulation and optimisation, amongst others. Nevertheless, several points can be made regarding the structural and econometric models. First of all, structural models should be based on an exact understanding of the problem and the internal dynamics (Refenes, 1995). An effective model should rely on correct assumptions and a clear understanding of the system dynamics, otherwise the model will perform poorly or generate misleading results (Refenes, 1995). This is clearly a very difficult task for a complex commodity like crude oil. Moreover, Labonte (2004), who surveyed the models dealing with the effect of oil shocks on the economy, argued that there are common shortfalls for econometrics methods when dealing with complex issues, namely: "omitted variable bias", "structural misspecification", "problem with endogenity", "Lucas Critique" and "robustness of the results" (Labonte, 2004, p.13-16). According to the author one of the most frequent limitations of such a model is the possibility of unrepresented variables. This occurs when the modeller overlooks some important variables in an attempt to limit the problem's scope, in order to be able to solve it. Another major issue with econometric models is reaching a misleading conclusion; in other words, a statistical relationship between two variables does not always mean a real one exists (Labonte, 2004).

Few studies have investigated the non-linear dynamics nature of the crude oil time series. One of these studies is by Moshiri and Foroutan (2006), in which the authors applied several tests for non-linearity and chaos to crude oil futures prices. The main conclusion of Moshiri and Foroutan (2006)'s analysis was that non-linear dynamics are present in crude oil futures prices; however, the authors could not find evidence of chaos in these series. More importantly, the type of non-linear dynamic was not detected. These results were later challenged by Matilla-Garcia (2007) who found evidence of chaos in crude oil futures prices as well as in the natural gas price and unleaded gasoline futures. A noteworthy point here is that Matilla-Garcia (2007) based a conclusion regarding chaos solely on the direct estimation of the Lyapunov exponent, while Moshiri and

Foroutan (2006) used the Jacobian approach to estimate the largest Lyapunov exponent[8]. The results of Matilla-Garcia (2007) also contradicted an earlier study by Adrangi, et al. (2001) who could not find any evidence of chaos in crude oil futures prices.

It seems there is no consensus in the literature about the dynamics of the crude oil price, so this necessitates re-visiting this topic.

Another point to be made regards the forecast horizon. Most of the models described above used low frequency data, usually monthly or quarterly. This is understandable when a modeller wants to use fundamental variables (supply, demand and storage) or macroeconomic variables such as GDP, because these variables usually are recorded on a low frequency. The implication of this is that the forecasts are derived from a small data size and, consequently, daily forecasting is no longer possible.

2.5 Soft-computing models

Soft-computing models, such as ANN, SVM and Fuzzy Logic, have gained huge momentum in the forecasting community as they have useful characteristics in domains where exact (analytical) solutions are not possible or very hard to obtain. Perhaps the best description of the characteristics of soft-computing models was given by Zadeh (1994) as they:

> "exploit the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, low solution cost and better rapport with reality." (Zadeh, 1994, p.1-2) quoted in (Bonissone, et al., 2006, p. 261).

2.5.1 The application of soft-computing to crude oil forecast[9]

Soft-computing methods have been exploited for financial prediction. These techniques were also applied to the crude oil prediction problem, though not as intensively as for other problems. An early study by Kaboudan (2001) compared the applicability of genetic programming and ANN to trivial predictions of forecasting crude oil prices on a monthly basis. As the scope of this study was monthly price prediction, monthly data were used. In this study, Kaboudan (2001) tested a different set of variables as an input to the model. Specifically the author tested the lagged value of the crude oil price, the world crude oil production, OECD countries' consumption and the US oil inventory,

---

[8] Moshiri and Foroutan (2006) claimed that using the direct approach of estimating the largest Lyapunov exponent in their study did not change the conclusion about chaos. The main issue with the direct approach vs. the Jacobian approach of estimating the Lyapunov exponent is that the latter is more sensitive to the noise in the data, as Matilla-Garcia (2007) pointed out. In principle though, this view is correct as the estimations of the Jacobian matrix will amplify the effect of the noise, yet, in Moshiri and Foroutan (2006) as well as in this thesis (see Chapter 4), noise filters were applied to the data which should be adequate to diminish this limitation.

[9] Important note: some of the studies presented in this section are reviewed in (Haidar, 2008) and also in Pan, Haidar and Kulkarni (2009). Please also note that the literature dealing with the application of soft-computing to the energy market is limited, and this section was updated to include new publications.

amongst others. A very important conclusion was reached: only the lagged value of the crude oil price was useful in providing an acceptable prediction. More specifically, and according to Kaboudan (2001), nine lags of the crude oil monthly price were the best for a one-month prediction. Finally, the author found that genetic programming was superior to ANN as it produced a more accurate forecast. However, the main criticism of this study is that the sample size was very small. Specifically, only 60 observations were used to train both models, which could significantly affect ANN performance.

Some studies examined the effect of decomposing the crude oil time series into multi-series with different frequencies, and then attempted to forecast each one using ANN. For example Yu, Lai, Wang and He (2007) and Yu, Wang and Lai (2008) proposed a model to forecast crude oil prices (WTI and Brent) based on feedforward neural networks. The authors started by decomposing each time series into several sub-series, with each one having different frequencies, using empirical mode decomposition (EMD). The authors argued that by dividing the original time series into sub-series, this will simplify the forecasting task for the neural network, hence improving the forecast outcome. They fed all sub-series to the network as input and trained the network with the original series. Yu, Wang and Lai (2008) compared the results of their proposed model to a forecast generated by feedforward and ARIMA, based on two performance criteria: RMSE and the percentage of direction prediction (hit rate). The authors concluded that the results were superior to the ones obtained by feedforward and ARIMA models without decomposing.

Moshiri and Foroutan (2006) studied the chaos and nonlinearity in crude oil futures prices applying the Lyapunov exponent test and Brock, Dechert and Scheinkman[10] (BDS) amongst others. The author concluded that crude oil futures prices time series are not chaotic; rather, they are stochastic and non-linear. Moreover, the authors compared linear and non-linear models for forecasting crude oil futures prices. Namely, they compared ARMA and GARCH to ANN and found that ANN is superior and produces a statistically significant forecast compared to GARCH and ARMA.

Gaffari and Zare (2009) used a neural fuzzy model to predict the crude oil spot price one day ahead out-of-sample. The model was based on a multi-layer feedforward artificial neural network combined with a fuzzy inference system (Ghaffari & Zare, 2009). The input used for this model was the historical spot price solely, after being smoothed to reduce the noise. The authors claimed that prediction accuracy of the price direction for one day ahead was 68%, which was considered a good result for predicting a complex time series like crude oil. Liu, Bai and Li (2007) also presented a

[10] BDS is a test of independence proposed by Brock et al. (1986). See Chapter 4 of this thesis for details.

hybrid model based on a neural-fuzzy technique to forecast Brent crude oil prices. Three forecasting models were used: radial base networks, Markov chain-based semi-parametric models and wavelet analysis-based forecasting models. The output of the three methods was used as input to the fuzzy neural network, whilst the target was the actual Brent crude oil price. The authors concluded that the non-linear combination out-performed any single model. However, the authors based this conclusion on one performance metric only, the root mean square error.

Xie, Yu, Xu and Wang (2006) proposed a SVM model for monthly crude oil price prediction. The authors claimed that SVM out-performed feedforward ANN with backpropagation (BPNN) and ARIMA for out-of-sample. However, their results were not consistent, as BPNN outperformed SVM for two of the four sub-periods tested. Nonetheless, both BPNN and SVM outperformed ARIMA in all four periods.

Fan, Liang and Wei (2008) proposed a new method for the multi-step forecast of the crude oil price. The proposed method was called generalized pattern matching based on genetic algorithm (GPMGA). According to the authors, the fundamental idea of GPMGA is to detect a pattern from the historical data, not only from the price but also from the price differences. Fan, Liang, and Wei (2008) applied the GPMGA for multi-steps forecast (21 days ahead) of WTI and Brent spot prices and compared the results to the Elman network. They claimed that the proposed model prediction was superior to Elman because it generated a lower error on testing data. While this study was theoretically sensible, the empirical test presented in the paper was not very convincing. First, the authors used only 21 days for out-of-sample testing, which is hardly enough to justify the results. Second, the authors acknowledged that the proposed model did not perform well on the Brent series in terms of goodness of fit. This caused more doubts about the model, especially as it is generally accepted that WTI and Brent oil are co-integrated.

Several points can be made regarding the studies discussed above. First of all, the vast majority of these studies was for very short-term forecasts. This could be attributed to the inherent limitation that time-series models share, which is that the forecast accuracy declines rapidly as the forecast horizon increases. Secondly, it is very difficult to compare the performance of each model with the other. This is because each study used different sample sizes and different performance measures. Furthermore, most of the studies formed their conclusions by comparing the proposed model with the results obtained via a feedforward network and ARMA. However, the majority of these studies did not show any evidence that the feedforward model’s structure (the number of layers and the number of neurons, amongst other factors) was optimal for the data.

The review above shows that, despite the fact that a large number of studies applied soft-computing methods to crude oil forecasting, the vast majority of these models was univariate models which solely used the spot price as input. While using the spot price seems to be an acceptable procedure, it could be useful to use the feedback from economic models when selecting the input.

Most recently a few studies have endeavoured to use a neural network for crude oil forecast, trying to benefit from the specification of this commodity, i.e., to model within the context of the economic framework. In a recent study, Lackes, Börgermann and Dirkmorfeld (2009) presented an ANN model to forecast crude oil trends for 5, 20, and 60 days ahead. Unlike any of the studies reviewed above, this model demonstrated a better consideration of the economic theory in terms of selecting the input for the network. Selecting the input is critically important for the learning process in the neural network paradigm, because feeding the network with poor input will result in poor learning and therefore, unsatisfactory results. Lackes, Börgermann, and Dirkmorfeld (2009) chose several variables as input, for example: crude oil price, futures price (both datasets were filtered), crude oil supply and demand, gasoline supply and demand, and IFO index as an indicator of economic growth, amongst others. The data length used in this study was seven years of daily data (2000 data points) covering the time from 1999 to 2006. The price of crude oil for each time horizon was modified to reflect the trend of the price in five classes, "strong decrease, decrease, constant price, increase, and strong increase" (Lackes, et al., 2009, p.250). These five classes were then used as network outputs. Lackes, et al. (2009) reported significant forecasting accuracy in terms of the hit rate. The authors stated a hit rate of 73% for the short-term, while for the medium term the hit rate was over 95%. However, a major concern about this study is that limiting the assessment criteria to only the hit rate could generate a misleading conclusion, as it ignores the model's goodness of fit. Furthermore, it is not clear what the exact input was that generated the highest hit rate. Also, the author did not clarify if the results were consistent for several trials or not. Most importantly, it was not clear whether changing the frequency of some of the variables from monthly to daily could have skewed the results.

Another attempt to employ soft-computing methods within the economic framework was made by Pan, Haidar, and Kulkarni (2009) who presented a model based on the feedforward artificial neural network to predict the direction of the crude oil price for three days ahead. The main goal of this study was to test the ability of ANN to predict a complex system, such as the crude oil price. The secondary aim was to test if a crude oil futures contract contains newer information about the spot price in the near future by using a non-linear ANN model. Four models were developed: the first one was based on the lagged spot price; the second model was based on spot and futures prices; the third was based on futures solely and the final one was based on spot and market data (gold spot price, S&P 500, US dollar index and heating oil No. 2 spot price). This research distinguishes itself from other studies by paying attention to small details while building and testing the ANN model. Moreover, the selection of inputs and outputs was based on a clear understanding of the economic dimension of the problem. However, the study was limited to feedforward networks only.

In a related study, Khazem (2007) tested the effect of seven different variables to predict the crude oil futures price using artificial neural networks. These variables are: depository institutions lend balance rate at the Federal Reserve Bank, the retail price index (CPI), events (extracted from the internet), crude oil spot price, crude oil futures open interest, natural gas price and heating oil futures price. All variables were secondary time-series variables collected on daily increments, except for the events. According to the author, the event variable was created by gathering information about the expectation of a crisis occurring. This information was gathered from web sites, namely, The Crisis Group website, and then it was converted into numerical values using the Likert scale[11] (Khazem, 2007). The neural network model used in this study was a feedforward backpropagation based on genetic algorithms. Despite the several gaps in this research, acknowledged by the author, Khazem (2007) found that in general, ANN outperformed linear regression. Furthermore, the author also found that CPI, depository institutions' lend balance rate at the Federal Reserve Bank, natural gas price and events have better forecasting power for the crude oil futures price than the crude oil spot price, heating oil futures price and crude oil futures open interest.

In conclusion, the body of literature concerned with crude oil forecasting and modelling is substantial. Previous traditional models which dealt with long-term forecasting did not achieve acceptable results, while the majority of the recent literature concentrated on short-term forecasting. On the other hand, the soft-computing models surveyed above were mainly for one-step forecasting, and the vast majority of these models lacked economic insight.

2.6 Knowledge-based learning

One major limitation of learning from the data approach is poor generalization of the model. This is because in most cases data do not contain all the information about the phenomena under investigation. Moreover, in the case of financial and economic series, data are recorded at different intervals covered with noise, and also the structure usually changes over time, i.e., it is non-

[11] According to Khazem (2007) the crisis information was converted into numerical data using a Likert five points scale by rating the following statement (from 1 no crisis to 5 a strong one): " Is the current world situation in a crisis?" (Khazem, 2007, p.54).

stationary. Therefore, introducing expert knowledge about the system under investigation could provide the answer to these limitations.

The idea of improving artificial neural networks' generalization ability using domain knowledge (DK) was recognised a long time ago. There are two complementary schools of thought on how to embed prior knowledge into ANN: firstly the data-based approach which utilises prior knowledge about the system dynamic in the training phase of the network either as additional inputs, or as additional targets or both; and secondly, through structural change to the network.

Data-based approaches classified these models based on the amount of domain knowledge available. There are three different scenarios for incorporating DK into neural networks: a knowledge-rich environment, an incomplete-knowledge environment and finally, a knowledge-free environment (Hilario & Rida, 1997). In a rich DK environment, one could form rules about the system; hence DK could be introduced in the form of rules, If→Then. On the other extreme, in a knowledge-free environment, one could only rely on a pure data-driven model. However, if we have some incomplete knowledge about the system, hints could be used (Hilario & Rida, 1997). We consider the classification of Hilario and Rida (1997) to be logical; however, in addition to the data-driven approach, the structural modification proposed by Neuneier and Zimmermann (1998) could be used as an alternative. What follows is a brief description of these two methods for incorporating DK into neural networks.

2.6.1 Hints

The term *hints* is used in the literature to refer to different things. Hints were first introduced by Suddarth (1988), and then Suddarth and Kergosien (1990) and Suddarth and Holden (1991). The authors argued that, in the context of neural networks learning, a significant improvement in network convergence speed was recorded when the network was designed to approximate more than one task at once. In a related study, similar evidence was found by Caruana (1997), where the author argued that for many problems it is desirable to model more than one task at a time. According to the author the advantage of approximating several tasks at once is mainly information richness. This finding supports the conclusion of Suddarth and Holden (1991). The key issue here is to have tasks that are related to each other which will lead to this improvement. In other words, if there is no clear association between these tasks, then there is no reason to suppose that an improvement of any capacity will take place.

Abu-Mostafa (1990) was the first to propose the use of hints to aid the soft-computing learning process in a systematic way. The basic idea presented in this paper was if a modeller knew any additional characteristics of the problem, e.g., the function we are trying to approximate is odd, then this information could be used to aid the learning process of a neural network (Abu-Mostafa, 1990). A more crystallised version of this idea was presented in Abu-Mostafa (1995b). Based on Abu-Mostafa's approach, there are two different ways to introduce hints into the network:

1. creating synthetic examples as an additional input

2. selecting examples from the target function itself. These artificial examples will act as constraints on the learning as the network will be trying to minimize the error, not only for the original input but also for artificial examples as well (Abu-Mostafa, 1995b, p.17).

Alternatively *duplicate examples*, i.e., examples drawn from the target function itself which satisfy the content of the hint, could be used as additional training input to the network (Abu-Mostafa, 1995b). Additional improvement was presented by Abu-Mostafa (1995b) in which the number of examples, in a given hint, was selected based on the Vapnik–Chervonenkis (VC) dimension.

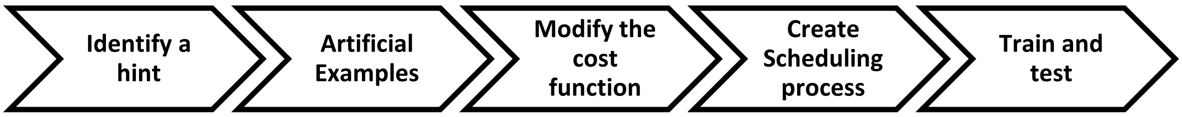

Figure 2-2: Diagram of learning from hint approach
Source: adopted from Abu-Mostafa (1995b)

In addition to this, the author suggested scheduling approaches called "adaptive learning" in order to handle more than one hint in one learning session; Figure 2-2 shows a summary of Aub-Mostafa's hints' approach.

The early work of Abu-Mostafa (1990, 1993, 1994, 1995b, 1995c, 2001) was mainly concerned with image recognition and has showed that including the known knowledge of system dynamics helps to improve the performance of neural networks. Abu-Mostafa (1993) applied one type of hint (the symmetry hint) to foreign exchange market (FX) forecasts. In general, the hint was defined as any known information about the target function that can be included in the *learning from data* process (1993). It should be noted that, while hints are similar to regularization, as they both aim to prevent over-fitting by restricting the learning process, they are not the same. Abu-Mostafa (1995c) argued that it is the "information content" that increases the overall generalisation of the networks. The author described the same results above in more detail (Abu-Mostafa, 1995a). In this publication (only), the author claimed that the input and the output were filtered by a simple filter, without naming the filter overtly (we assume it was a moving average). Secondly, and more importantly, the recorded improvement with the hints was very modest; the daily hit rate merely improved by one percentage point at the best. The majority of improvement was on annualised percentage return.

2.6.2 Network structure modification

The research group in Siemens AG produced a new approach for modelling a financial time series, based on architectural modification and improvement of neural networks, instead of algorithmic modifications. Neuneier and Zimmermann (1998) argued that a standard feedforward network is very well suited to model a financial time series, because the financial data are very noisy, and only limited numbers of observation are available. In contrast, the fundamental idea of this modelling philosophy is that by enhancing the network topology, based on the domain knowledge of the system, more relevant information to the real system dynamics is processed by the network, while irrelevant information is ignored (Grothmann, 2002).

For example, in Neuneier and Zimmermann (1998) the 11 layer network architect for a feedforward network was introduced. This architecture involved creating several building blocks, each block designed to perform a specific task, then all these blocks are combined in one network for the task of foreign exchange forecasting. The authors started with data transformation which reflects the behaviour of the financial time series; basically, in addition to the relative return, the author added the turning point, *the force*. Then they moved to enhancing the network structure; hence, an internal outlier reduction layer was introduced. Moreover, enhancing the network error flow was another issue in this study; by adding a layer with multi-outputs, each output corresponds to forecasting one-step ahead of the previous one. Then when all these blocks are combined, the prior knowledge of the system is exploited (Neuneier & Zimmermann, 1998).

Another approach involves creating an error correction network. Grothmann (2002) and Zimmermann, Grothmann, Schäfer, and Tietz (2005) argued that, for most financial time series, it is often rare to have a complete account of all external factors affecting the market. Therefore, the error of the model itself can be viewed as a measurement of the short-term influence of external forces when used as additional input to the recurrent network (Zimmermann, et al., 2005). According to Grothmann (2002) this concept shares a similarity with two established models, ARIMA models and NARX networks. However, Grothmann (2002) claimed that the recurrent error correction model differentiates itself from the ARIMA by its ability to model non-linear behaviour and from NARX, by its ability to model the long-term horizon (Grothmann, 2002). Weight-sharing and unfolding in time was also part of this approach.

On the same note, Zimmermann, Bertolini, Grothmann, Schäfer and Tietz (2006) argued that recurrent unfolding in time networks (a recurrent network which is based on shared weights over time) is a better approach to modelling a dynamical system than recurrent networks by themselves. This is because the former are better equipped to model cross-influence of a dynamical system, i.e., the effect of other markets or other variables on the system's behaviour. In another study by Schäfer, Udluft and Zimmermann (2008) the same argument was made: that recurrent neural networks with unfolding in time are better frameworks to forecast long-term memory. In contrast, the authors of this paper introduced a slightly modified network called the normalized recurrent network (NRNN). They argued that one issue with training RNN to learn long-term memory is the backpropagation algorithm as the weight differs for each time step, which affects the model's overall generalization. Moreover, according to the authors, training RNN is generally unstable which also affects the model's stability. Therefore, Schäfer, Udluft and Zimmermann (2008) argued for unifying the weight matrix throughout the network into a fixed identity matrix. Hence, all free parameters throughout the network are treated in the same way by the backpropagation algorithm, which in effect solves the instability and the generalization problem of regular RNN (Schäfer, et al., 2008).

Related to this, in his PhD thesis, Schäfer (2008) claimed to produce a link between reinforcement learning and a recurrent neural network. Schäfer's model was an extension to previous work by Zimmermann of the model of unfolding in time and weight-sharing produced by the Siemens AG group. Schäfer (2008) started first with extending the proof of the feedforward network as a universal function approximation by Hornik, Stinchcombe, and White (1989) to include recurrent networks. Using the famous cart-pole problem and also a real world problem (gas turbine control), the author claimed that a modified RNN is able to learn long-term dependence with high accuracy. More importantly, the author argued that when the architecture of the RNN network is modified to include weight-sharing matrices and unfolding in time, it is equivalent to the algorithmic extension of the backpropagation algorithm. This is achieved by combining the overshooting weight-sharing and dynamical forecast in one architecture. The author stressed that this finding contradicted the long standing opinion about the limitation of a neural network with a backpropagation algorithm to generalize for long term horizons.

Although, the network structural modification approach seems to be promising, to the best of our knowledge it was not tested independently outside the Siemens AG's research group surveyed above.

2.6.3 Knowledge incorporation based on genetic algorithms and reinforcement learning

Bonissone, Subbu, Eklund and Kiehl (2006) argued that the combination of evolutionary algorithms (EA) with a soft-computing method (as a platform to implement domain knowledge) is able to handle difficult real world diction and control problems with a high level of accuracy. Citing the *no free lunch theorem* (Wolpert & Macready, 1997), Bonissone, et al. (2006) argued that despite the capability of EAs to handle non-linear complex problems in high-dimensional space, without a tailor-made model it is unlikely for any of these algorithms to find optimal or near-optimal solutions on their own. The author classified the embedding of DK in two broad groups: *implicit* and *explicit*.

According to Bonissone, et al. (2006) implicit knowledge is usually achieved by the problem representation, in other words the way the input and output are represented has significant importance toward effectively solving the problem. Secondly, of importance are encoding methods directly relevant to the problem at hand: this includes binary, real, integer and finite state encoding. Thirdly, both static and dynamic constraints could be used during the training. Static constraints which are constant throughout the training can be embedded by data modification and penalty term on the cost function amongst other factors, while dynamic constraints, such as generation-based penalty term, need to encoded in the model; however, dynamic constraints are very complex to handle (Bonissone, et al., 2006).

Explicit DK representation, according to Bonissone, et al. (2006), can be utilised into the model by selecting a good initial population to start with: this will increase the likelihood for the algorithm to find the global minima (or near global minima) as some guesses are better than others. Also belonging to the explicit DK representation category is the combining of the local search with global search methods, otherwise known as a hybrid genetic algorithm. Memetic algorithms (MAs) are one example of this group and have gained great attention lately. The authors also included several other approaches under the explicit DK representation, which, for the sake of brevity, are not included in this review.

Furthermore, Bonissone, et al. (2006) also showed that hybrid soft-computing methods and EA can produce a much more prosperous environment for solving real world problems. This is because, by definition, soft-computing methods in general and fuzzy logic specifically are flexible enough to deal with uncertainty and incomplete ill-representative data which are the characteristics of real world problems (Bonissone, et al., 2006). The claims of Bonissone, et al. (2006) were verified by these authors through three case studies for real world problems. Each time, the model with DK proved to be best in terms of accuracy and consistency.

Wiewiora, Cottrell and Elkan (2003) presented a method for embedding domain knowledge into reinforcement learning's Q-value paradigm. The new method involves altering the reward system of a reinforcement learning agent to incorporate both the state of the agent and its action. The author presented two variations, the first of which is *look-ahead advice*, an extension to the potential-based shaping approach which works by rewarding the agent for taking the decision that maximizes the Q-value. The second variation is *look-back advice*, in which the agent takes into account not only the potential functions at its present state but also the experience from the previous state.

2.6.4 **Application of DK**

One of the early works in the area of DK neural networks was by Towell and Shavlik (1994). The authors presented the knowledge-based ANN (KBANN) algorithm as a method to combine domain knowledge (an expert system) and learning from example, in a way that improves the overall accuracy of the classification task. Towell and Shavlik (1994) argued that a domain specific system, i.e., tailor- made models, are disadvantageous methods as they assume that the DK about the problem is complete, which is often not a totally accurate assumption. Secondly, in order to acquire the complete DK, one needs to write a large amount of rules. Thirdly, the calibration of this kind of system, i.e., a system with many rules, is a very difficult task. On the other hand, machine-learning methods, according to the authors, depend on a large number of examples in order to converge. Another point is the *context dependency* issue during the classification process, which is hard to deal with in plain machine learning algorithms (Towell & Shavlik, 1994, p. 122).

Therefore, Towell and Shavlik (1994) proposed a hybrid approach in which the theoretical knowledge of a given domain is presented to ANN (feedforward with backpropagation) as propositional logic (a set of rules) and the networks are refined for the final solution. Empirical evidence was offered for the superiority of this approach for several classification problems.

Serpen, Tekkedil, and Orra (2008) provided evidence to support the view that the KBANN of Towell and Shavlik (1994) outperforms general learning from data methods for complex problems. The authors explained that, unlike purely inductive learning methods, KBANN, along with the neuro-fuzzy inference system and Bayesian belief networks, offers an environment for embedding field- specific domain knowledge as opposed to relying on the data and the heuristic algorithm to find the desired solution. In this paper, the goal was to prove that KBANN is a better learning approach for real world non-trivial problems, in this case the diagnosis of pulmonary embolism (PE) disease. Domain-specific rules were developed with the help of a human domain expert. The simulation output of KBANN was compared to the C4.5 decision tree, MLP, Bayesian belief network, naive Bayesian and two other meta-learning methods.

On all occasions the KBANN was superior to all other methods tested, in accurately classifying the out-of-sample data regardless of whether cross-validation or leave-one-out validation was used. Two points are fundamental to evaluating this study. First the problem which Serpen, Tekkedil and Orra (2008) applied to the KBANN has very rich domain knowledge that can be easily translated into if-then rules. This is clearly a great advantage for the methodology used in this research. Second, a human expert was available to monitor this process and also to provide help with calibrating and testing the model. Nevertheless, this paper showed the clear advantage of using KBANN over general inductive methods.

On the same topic, Bose and Nagaraja (2004) examined the performance of KBANN using toy problems and proposed minor improvements by using a different variety of the backpropagation, using regularization and pruning of redundant rules. The author also noted that the improvement in the performance of KBANN was noted when the values of the network's weights were relatively small. Evolutionary algorithms were also used as a medium to implement DK. For example Bonissone (2006) argued that domain knowledge is a vital part of any decision system. Moreover, the author claimed that soft-computing methods offer a fertile and flexible environment for embedding such knowledge. Bonissone (2006) was dealing with prognostics and health management (PHM) problems; however, according to the author, the DK approach should be applicable for any other problem.

Basically, there are two levels of embedding such knowledge: (1) the implicit way which includes using a tailored data structure and applying constraints during the learning process amongst other factors; and (2) explicit methods; imitating the model with a good set of parameters to start the training and combining global and local search methods, amongst other factors (Bonissone, 2006).

DK has also been applied to the SVM model. In his PhD thesis, Yu (2007) presented new methods to implement domain knowledge into kernel-based models, specifically the SVM. The aim of this work was to answer the question of how domain knowledge can be effectively implemented into an "inductive machine learning" model such as SVM. The first method which Yu (2007) used was a rule-based system with IF→THEN logical calculus techniques, designed to help labelling an unlabelled training sample derived from a new release. This method was intended to help in a specific case study about the effect of a news release on a specific firm's stock price movements. The second model was a hybrid SVM and vector quantization (VQ) called (VQSVM) aimed to combine the global and local modelling capability. Two applications of these models were presented for financial markets. The first application, as mentioned above, was forecasting the stock price using the news release. The second application was to improve the auditing of financial corporations.

Finally, Yu (2007) highlighted several issues that make this type of research very difficult. According to the author, collecting domain knowledge is the first obstacle to be overcome. This is because it needs to depend heavily on human experts, which can be unreliable due to confidentiality issues and the communication gap between domain experts and machine-learning experts. Secondly, the knowledge collected is often imperfect and incomplete (Yu, 2007). Thirdly, and most importantly, it is extremely difficult to represent domain knowledge effectively in machine learning algorithms. The chief issue here is machine-learning algorithms deal mostly with numerical data. Hence, as long as the domain knowledge is represented in a numerical form, e.g., via virtual examples, there will be no problem. However, most knowledge delivered by human experts is in a non-numerical form. Finally, there is the difficulty of striking a balance between learning from example and learning from domain knowledge (Yu, 2007).

This issue was also realised some time ago; for example, Haykin (1998) argued that while the effect of adding domain knowledge into machine-learning (ANN, SVM) is important, in the case of SVM, it is mainly done through virtual examples. Likewise, Barakat (2007) argued that domain knowledge can be implemented into an SVM model either by modifying the kernel function, or by using virtual examples. Once again, the author preferred to use a virtual example to introduce the additional expert knowledge for reasons of convenience and flexibility. The objective was to construct virtual examples that convey the correct domain knowledge for a medical diagnoses problem. While the details of these examples are irrelevant for the purpose of our investigation, it worth mentioning that Barakat (2007) found that this approach improved the overall classification power of the SVM model. On the same note, the term ‘domain knowledge’ seems to mean different things to different people.

Bailey and Elkan (1995) showed how prior knowledge embedded into the unsupervised MEME algorithm can improve detecting motifs in DNA or protein sequences. The authors applied Dirichlet mixture priors to provide prior knowledge about the mutual characteristics in amino acids that could be present in a given position of a motif. They showed that this knowledge helped improve the performance of MEME models to detect more motifs in a DNA sequence.

Tilakaratne and Mammadov and Morrise (2008; 2009) and Tilakaratne, Mammadov and Hurst (2006) proposed a method for quantification of inter-market influence, to improve the forecast of the Australian All Ordinaries (AORD) index using a feedforward ANN. The goal was to predict the closing return direction (up, down or steady) one day ahead for the AORD index. The authors argued that inter-market influence could be an additional source of information; however, according to them, the problem is that statistical tests are not well suited to measuring the inter-market influence of a financial series, as the latter are non-stationary and contain a high level of noise. As such, Tilakaratne and Mammadov (2006) proposed to quantify inter-market influence in the form of weights for each market, to be used as additional input to the ANN. These weights were reached by solving a global optimisation problem (initially proposed by Mammadov 2004). These weights (coefficients ξ) represent maximizing the median rank correlation between the target market return one-step ahead and the weighted return ξ_i, $i= 1,2,...,n$) of each of the inputs (inter-markets return and AORD return at time t).The authors claimed that by quantifying the inter-market influences, a profitable strategy can be reached.

Only a few published works relating to crude oil can fall into the category of DK soft-computing. Wang, Yu, and Lai (2005) introduced the TEI@I methodology to forecast the crude oil price. The methodology consists of a combination of three separate components: web mining (gathering information from structured and unstructured web documents), neural networks and auto regression integrated moving average methodology. These three components work individually and then are combined to get the final results. The first component is a web-based text-mining module which collects data from the internet and extracts the information that affects crude oil price variability, then uses the most useful information in building the rule-based system. A rule-based system was built on a knowledge base they developed which takes into consideration all factors that affect the crude oil price, according to the authors. The second part of this system is the use of artificial neural networks to forecast the non-linear aspect of the oil price time series. The third and last component of this system is the ARIMA econometric model which is used to forecast the linear part, the trend of the time series. The authors claim that the forecast generated by TEI@I outperformed any of the individual forecasts generated by ANN or ARIMA.

Another approach is that by Abramson and Finizza (1991) who developed a forecasting model for the crude oil price (quarterly) based on belief networks, which is a category of graphical models belonging to the same family as decision trees (Abramson & Finizza, 1991). The model was macroeconomic in nature as it utilised 140 simple equations to describe the relation between oil supply, demand, OPEC capacity utilisation and GDP, amongst other variables, and the oil price. Since macroeconomic variables were used, the forecasting horizon was quarterly and up to four quarters ahead. The paper concentrated on proving the concept rather than showing the actual prediction; as such, the values of the variables were not presented. The model showed a high level of comprehension as it took into consideration a large number of variables and multi-relationships between them. However, the model made a strong assumption about the relation between the variables, and it also assumed that only linear relations govern the system dynamics. Another point can be made from the practical point of view: the fundamental and macroeconomic variables are not freely available. Furthermore, the author tested the model on only one year of data, i.e., prediction for four quarters, and the results were not updated when Abramson (1994) published the model at a later date.

For crude oil forecasting the only relevant studies (in this update) have been performed by Abramson and Finizza (1991, 1995) who presented a modified version of their Bayesian Belief network to forecast the crude oil average annual price one-step ahead. The authors argued that the failure of their previous model to generate a good forecast out of the sample was due to major changes in the market which followed the Gulf War in 1991 and the fall of the Soviet Union.

The new model proposed by Abramson and Finizza (1995) shared great similarity to their previous model in terms of its concepts, but differed in the variables and the relation between these variables. According to Abramson and Finizza (1995) after the 1991 Gulf War, OPEC capacity utilisation (production/capacity) became an important variable to forecast crude oil development in the short-term. Furthermore, only the behaviour of a few OPEC countries has an impact on the market, in particular, Saudi Arabia (Abramson & Finizza, 1995). In addition, because the elasticity of demand is very low in the short-term and capacity forecasts are exogenous to the model, a price forecast can be reached by forecasting the production level only (Abramson & Finizza, 1995). The new model (ARCO2) uses the WTI lagged price, world oil production both for OPEC and non-OPEC, oil capacity and total world demand market expectation, amongst other factors.

The main advantage of such a model is its ability to include political influences as an input to the model alongside the economic variables. This integration is very important in a commodity in whose production local and international politics play such a big role. The main problem in assessing this model successfully is its incomparability to other models and the inability to rebuild this model as the majority of the variables and the relationship between them were not published. Therefore, we can only rely on the authors' claims, although conceptually, in my opinion, it is one of the best models to describe the crude oil market (Abramson & Finizza, 1995).

Yu, Wang and Lai (2009) proposed a meta-modelling technique to forecast a financial time series based on artificial neural networks. The approach consists of three main steps: first, sampling techniques are used to divide the original data into training validation and testing subset; this was according to the author, a much better approach than user-based division. Second, a large number of ANNs with different architecture are used to forecast the series of interest, as such, each network will converge into a different solution. Third, principal component analysis is applied to these outputs to reduce the size of the output space and also to avoid using inputs that are highly correlated to the main network, and the remaining outputs are fed into the ANNs to generate the final solution. The empirical results presented by Yu, Wang and Lai (2009) suggested that the proposed technique significantly outperformed all the benchmarks, namely ARIMA, single ANN and single SVM for the four time series included in this study. It is worth noting that the authors chose to use raw price and not a return, which could have affected the integrity of the results for one-step-ahead forecasting. This is because the raw price could contain a trend. Also, no comparison was made with the random walk model. In general, however, the approach proposed in this study is a simple, conceptually valid (very similar to the ensemble approach) and apparently effective one.

Kablan (2009) proposed a simple trading system for the EUR-USD rate intraday (5 minutes) forecast based on an adaptive neuro-fuzzy inference system (ANFIS). The ANFIS is considered a hybrid approach that utilises the logic of fuzzy knowledge with the search power of ANN. Kablan (2009) claimed the ANFIS used in his investigation was a standard one already known in the literature. The author acknowledged that the novelty of this approach was only in applying it to high frequency data, specifically 5 minutes data. In general, the study was not robust and the domain knowledge methods for the fuzzy system were not well explained.

The literature shows that several studies have claimed that the combination of domain knowledge and soft computing models (including ANN) was successful in solving several real world problems. However, the term domain knowledge appeared to be rubbery and was used in different studies to describe different things such as: the knowledge of soft-computing models, the knowledge of time-series analysis and the expert knowledge of the specific problem. From the literature survey there appear to be two ways that are consistently used when dealing with domain specific DK: firstly, data- based virtual examples or modified targets, and secondly by-rule-based fuzzy inference hybrid models. On the other hand, new research has emerged that concentrates on combining evolutionary learning (genetic algorithms and memetic algorithms) and soft-computing (mainly fuzzy logic) as methods of improving the overall performance of the model, such as the research by Bonissone (2006) and Bonissone, Subbu, Eklund, and Kiehl (2006). It is important to note that these models are not forecasting models, rather they are decision support ones. Other hybrid models involved global optimisation with soft-computing methods such as the work by Tilakaratne, Mammadov and Morris (2008) to quantify inter-market influence before using ANN to forecast the price direction of the AORD index.

In conclusion there are a large number of methods that claim to incorporate DK into the building process of the models, and further claim that better results were thus achieved. Nonetheless, in the field of financial forecasting, more specifically, for crude oil forecasting, only one of the authors cited above demonstrated the use of field specific knowledge in the modelling process.

2.7 Limitations of the previous work

In conclusion, several points can be concluded from the literature survey presented above. First and foremost, forecasting the crude oil price is a very important task. Because of the vitality of this commodity to the economy, there is hardly anyone not affected by oil prices: governments, industries and even individuals. Secondly, it is evident from the literature that the forecasting record of the crude oil price is generally poor. Forecasting crude oil movements has proven to be very difficult, due to the large number of factors affecting this commodity, the role of speculation and the OPEC monopoly, amongst other factors. Moreover, the absence of reliable data presents an additional challenge for achieving acceptable forecasts. Furthermore, in relation to this thesis's area of focus, soft-computing methods have already been applied to this problem but, in our opinion, with these limitations:

- Some studies were more like a tool looking for an application.
- Most of the studies were for one-step-ahead forecasts only.
- The performance measure did not reflect a high degree of transparency; in other words, it is relatively easy simply to pick a subsample and performance metric to generate high forecast accuracy.
- Fundamental issues such as testing for non-linearity and data transformation (return) were often ignored.
- To our best knowledge, only a few studies attempted to embed domain knowledge into soft-computing models for crude oil forecasting.

In relation to our first research question, "Can we forecast complex economics systems like crude oil price multi-steps ahead, using DK-soft computing models? " we argue that there is a gap among soft-computing, time-series modelling and econometrics modelling. Soft-computing models, mainly ANN and SVM which basically are complex universal function approximations, seem to be applied directly (with the exception of a few studies) with the implicit assumption that whatever the underlying dynamics of the series are, these models will fit it well (Bowden, 2003). However, if the time series follows a linear structure, then there is no justification to applying non-linear complex models, like ANN, which employ a large number of free parameters (Bowden, 2003). Equally important is that any forecast would be compromised if the null hypothesis of independence is not overtly rejected (Kugiumtzis, 2000). Moreover, although non-linearity is always assumed in financial/commodity prices time series, linear models are widely used to forecast these series.

However, if strong evidence of non-linearity in crude oil returns is confirmed, non-linear models are better approaches than linear models. Of course this raises the vexed problem of model choice in the case of non-linearity. Therefore, it is very important to investigate the dynamics of the crude oil prices and returns and use these findings in calibrating our forecasting model. Furthermore, should the dynamics of crude oil pricing be non-linear, it is still important to determine what type of non-linearity characterises this series, i.e., non-linear chaotic, non-linear deterministic or non-linear stochastic. Additionally, if we could find evidence of chaos in crude oil returns, this would provide one explanation of their seemingly random behaviour.

Moreover, if we find evidence of chaos this will have two implications for forecasting expectations, as (i) chaotic systems, in contrast to random systems, are deterministic, hence, they are predictable in principle, and (ii) since chaotic systems are sensitive to initial conditions, long-term forecasting is unlikely to be successful, because the error from each forecasting step will be amplified exponentially (Adrangi, et al., 2001). Therefore, in Chapter 4 we strived to bridge this gap identified above.

The second research question of this thesis is: “Does domain knowledge expertise improve the prediction output of a soft-computing model of complex economics systems like the crude oil price?” We believe there is very limited literature that deals with this problem for financial and commodity prices. Moreover, Abu-Mostafa (1993), the pioneer in this field, argued that the main difficulty in this type of research is to find the expert knowledge. Therefore, in Chapters 5 and 6, we present a number of strategies using well-established econometrics and soft-computing methods to encapsulate DK into soft-computing model.

The third and last research question: “Can a multi-agents model based on ANN produce better use of DK? Or can a combination of different types of DK outperform a single type?” addresses another type of DK which, in our opinion, is under-investigated in the literature. Here we test if we can construct an artificial market that captures crude oil market dynamics, and also whether the output of this artificial market contains new information which can aid the learning process of traditional ANN. In other words, can the output of our multi-agents model act as a hint? We believe this area is under- investigated although it is a very important field for study.

2.8 Significance of this research

This research aims to build a new strategy to forecast crude oil prices over different time horizons based on soft-computing methods. It also aims to embed knowledge of the system dynamics to improve forecast accuracy and time horizon. The goal is to provide a reliable prediction of crude oil price direction, and to test for how long a reliable and meaningful forecast can be achieved. The motivation of this research is driven by the potential impact of crude oil price prediction on the economy.

Forecasting the crude oil price is critically important for a variety of reasons. The price of crude oil at any given time will determine the price of other oil products (petrol and diesel, amongst others) and to some extent, the price of crude oil derivative products such as natural gas. As such, predicting the movements in the crude oil price should help policy makers, energy market participants and small and medium companies like petrol retailers and other groups to hedge their situations.

Part of the contribution of this thesis is to bring these three interrelated fields: (i) energy economics, (ii) time-series econometrics, and (iii) soft-computing, closer together.

CHAPTER 3: Artificial intelligence methods

3.1 Introduction

This chapter provides background information about the main tools used in this research. In particular, we concentrate on ANN and NeuroEvolution of Augmenting Topologies (NEAT) (Stanley and Miikkulainen, 2002). Finally, we discuss the performance metrics used in this thesis to assess forecasting performance.

3.2 Artificial neural networks

Artificial neural networks (ANN) were designed in an attempt to imitate the human brain's functionality (Haykin, 1998; Refenes, 1995). The main idea of ANN is to learn the desirable behaviour from the data with no *a priori* assumptions (Haykin, 1998; Refenes, 1995).

From an econometrics standpoint, ANN falls in the non-linear, non-parametric and multivariate group of models (Grothmann, 2002). This makes it a suitable approach to model non-linear relationship in high-dimensional space (Grothmann, 2002). Hence, ANNs, theoretically speaking, are amongst the candidates to model complex economic systems such as financial/economic time series (Grothmann, 2002).

Just as the basic unit in an animal brain is the biological neuron, similarly, the basic structure of ANN is an artificial neuron (Haykin, 1998). An artificial neuron i can be given by (Haykin, 1998):

$$u_j = b + \sum_{i=1}^{n} x_j w_{ji} \tag{3.1}$$

$$y_i = f(u_j) \tag{3.2}$$

where x are the network inputs $w_{j,i}$ are the network weights, $f(.)$ is the activation function of the neurons, b is the network bias, and y_i is the output of the neuron. When a non-linear neuron is used (which is the closest to its animal counterpart (Haykin, 1998)), a sigmoid or hyperbolic tangent function (Equation (3.3) (Refenes, 1995)) or any other non-linear function can be used, depending on the application; for example:

$$y = f(x) = A \tanh Sx = A \frac{e^{Sx} - e^{-Sx}}{e^{Sx} + e^{-Sx}} = A - \frac{2A}{1 + e^{2Sx}} \tag{3.3}$$

where A and S represent the amplitude and the slope of the function, respectively, and $y = f(x)$ is the output of the neuron.

Therefore, a feedforward network with a *tanh* function and one hidden layer can be given by (Grothmann (2002):

$$y_j = b + \sum_{n=1}^{k} v_{k,n} \; tanh\left(\sum_{i=1}^{z} w_{j,i} x_i\right) \tag{3.4}$$

Where $v_{k,n}$ and $w_{j,i}$ are the network weights. On the other hand, when one or more feedback loops are added to the network, the network is called a recurrent network, as shown in Figure 3-1.

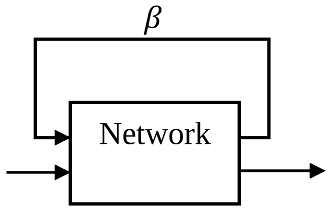

Figure 3-1: Block diagram for a simple recurrent network
Source: (Grothmann, 2002, p. 73)

Mathematically, in its most basic form, the recurrent network in Figure 3-1 can be given by two simple equations—the state transition Equation (3.5), and the output Equation (3.6) (Grothmann, 2002, p. 73):

$$\beta_t = f(\beta_{t-1}, u_t), \tag{3.5}$$

$$y_t = g(\beta_t). \tag{3.6}$$

Training the neural network in the backpropagation paradigm involves continuous change to the values of the network parameters (weights and biases) in the direction that reduces the error between the input and the target, based on some cost function (usually mean squared error), until one of the stopping criteria is met (Refenes, 1995):

$$w_{t+1} = w_t + \Delta w_t \tag{3.7}$$

$$w_{T=} w_0 + \sum_{t=0}^{T-1} \Delta w_t \tag{3.8}$$

$$\Delta w = -\lambda\delta = -\lambda \frac{\partial E}{\partial w}, \tag{3.9}$$

where *w* is the network weights, *t* is the current step, *T* is the number of iterations, λ is the learning rate (step size), δ is the gradients of the error surface, and *E* is the global error.

When feedforward networks are used for time-series forecasting, the problem is transformed into pattern recognition (Refenes, 1995). Therefore, a key issue when using feedforward networks is the representation of the problem. This issue was often overlooked in the econometrics literature that employed ANN. Mostly, using the lagged return solely as an input to the network could lead to a poor fit.

The usefulness of a network is assessed by its ability to generalise, i.e., to learn the real underlying function of the data. It is generally easy for the network to produce a high accuracy in-sample because the network learned the noise rather than the real behaviour (Haykin, 1998; Hinton, 1999; Neuneier & Zimmermann, 1998; Refenes, 1995). The complexity of the model, i.e., the number of hidden neurons/layers along with the training time (number of iterations), has a significant role in over-fitting. Other issues like adding momentum and the size of the learning rate also affect network generalization. There are several methods to reduce over-fitting, such as early stopping, weight pruning, regularization and hints. Another approach to improve ANN fit is to use the average of a number of networks' output, each trained on the same data (Hinton, 1999). Chapter 5 discusses how problem representation can positively affect the network's performance.

For the sake of brevity, other issues like the universal function approximation theorem proved by Hornik, Stinchcombe, and White (1989) and different network topologies are not discussed in this document. Also, the historical development of ANN is beyond the scope of this document.

3.3 **Why ANN?**

Using ANN in this research comes in line with the previous research detailed in Chapter 2 (Lackes, et al., 2009; Khazem, 2007; Moshiri and Foroutan, 2006). Also, Vanstone (2005) as well as Vanstone and Finnie (2010) presented a profitable trading system for the All Ordinaries Index based on a "vanilla" feedforward network. Hence, the use of ANN is justified, from the point of view of econometrics, by its ability to deal with non-linear behaviours. Haykin (1998) emphasized this capability of ANN: "Nonlinearity is a highly important property, particularly if the underlying physical mechanism responsible for generation of the input signal … is inherently nonlinear." (Haykin, 1998, p.3). Zimmermann (2010) stated: "From a mathematical point of view, neural networks allow the construction of models, which are able to handle high-dimensional problems along with a high degree of nonlinearity" (Zimmermann, 2010, p.1). ANN complexity can be easily adjusted depending on the input dimension and problem complexity, either by varying number of neurons and layers or by using a weight-sharing approach. Moreover, the ensemble approach, as an example, aims to reduce the possibility of the network converging at a local minimum.

As highlighted in the scope (subsection 1.4), our aim is not to test whether model *a* is superior to model *b* in its general form (e.g., whether SVM is superior to ANN), but rather to test the information content of the inputs (hints), and so model selection will make little difference to our research question, providing that the model is theoretically suitable for the problem at hand. On the other hand, the "no free lunch theorem" (Wolpert & Macready, 1997) was cited to highlight the importance of using domain knowledge in any given model.

Nevertheless, we acknowledge that ANN has several limitations which are detailed in Section 3.4. We also acknowledge that ANN is not the only model that can be used for this purpose of our research. Therefore, as an alternative learning method we propose another algorithm, NEAT, which is based on reinforcement learning.

ANN was selected in this research because we are dealing with non-linear and very noisy time series. Furthermore, ANN is a flexible modelling method in terms of complexity and structure. The flexibility of ANN is a very important property for our multi-agent model to achieve heterogeneous agents (see Chapter 7for details).

3.4 Beyond traditional ANN

There was a pressing need to move on from the traditional neural networks paradigm (ANN) trained with the backpropagation algorithm (BP). The issue with ANN in relation to my research can be classified into two groups: theoretical issues and technical ones.

3.4.1 Theoretical issues

Training neural networks with the backpropagation algorithm is heavily criticized within the machine- learning community due to several concerns, including how to select the topology and how to justify both the topology and the complexity. Though the BP algorithm has solved the credit-assignment problem (Refenes, 1995), it is still criticized for the way the weights are updated. According to Geoffrey Hinton (one of the contributors to the BP algorithm), there are two main limitations to the BP algorithm.

The first problem is that BP relies solely on the information provided from the targets and ignores the information from the inputs (Hinton, 2007b, 2009). Hence, if this information is not enough or incomplete then it is highly unlikely that ANN will learn the real function of the data (Hinton, 2007b, 2009). Second, in multi-layer networks, the initial selection of the weights plays a crucial role in the learning process (Hinton, 2007b, 2009). In other words, if the initial weights are too small, the error derivatives (during the backpropagation step) will get smaller and smaller as they travel each layer, which means the update to the weight will be very small at each step. This in turn will increase the training time and also the tendency to get stuck in a local minimum (Hinton, 2007b, 2009). On the other hand, according to Hinton (2007b, 2009), if the initial weights are too large then the modeller has already chosen the parameter space for learning the problem, and this could lead to trapping at a local optimum. In other words, in this case the modeller has made strong assumptions about the weight space for the network to search from (Hinton, 2007b, 2009). Another issue with traditional ANN is that the backpropagation algorithm tends to over fit when the network constructed with more than one hidden layer[12] (Hinton, 2007a).

3.4.2 Technical issues

NEAT was selected as a suitable platform to implement our multi-agent model (see Chapter 7). The main issue was to find an algorithm that is theoretically valid and at the same time fits within the research objectives.

3.5 Why NEAT[13]?

Based on the argument in the previous section, we started researching other potential alternatives that were theoretically and technically suitable for our research problem within the realm of neural computing, keeping in mind the need for generating superior models.

NeuroEvolution of Augmenting Topologies (NEAT), developed by Stanley and Miikkulainen (2002) and Stanley (2004), was selected because in theory it solves two core problems. First, because NEAT is based on reinforcement learning and not the BP, it is more likely to find the global optimum, and second, because it provides a systematic way (justification) for finding the network structure (Stanley & Miikkulainen, 2002).

NEAT is an algorithm for optimizing neural network topology and complexity based on a genetic algorithm. The algorithm was developed for solving reinforcement learning problems (pole-cart balancing, automatic car driving and video games, amongst others) where supervised learning is not an option (Stanley, 2004). This is in stark contrast to supervised learning, where the connection weights of the network are updated based directly on the derivative of the error at each step until the final solution is found. Reinforcement learning is based on a reward and penalty approach whereby the system rewards the agents (networks) when they get closer to the solution and penalizes the agents if they drift from the desired solution. However, the parameters (weights) themselves are not directly guided by the error (Miikkulainen, 2010; Miikkulainen et al., 2006; Whiteson, Stone, Stanley, Miikkulainen, & Kohl, 2005).

There is a very small body of literature that deals with reinforcement learning for time-series forecasting (Kuremoto, Obayshi, & Kobayashi, 2005; F. Liu, Quek, & Ng, 2005). In principle, there is no evidence to suggest that reinforcement learning in general or NEAT in particular will not work

[12] Hinton, Osindero and Tech (2006) introduced a new and effective algorithm for training Deep Belief Networks *multi-layer networks*, one layer at a time to model structured data. See also (Mohamed, Hinton, & Penn, 2012; Sarikaya, Hinton, & Deoras, 2014) for some application of Deep Belief Networks.

[13] We used the MATLAB code for NEAT by Christian Mayr; this code was retrieved from http://www.cs.ucf.edu/~kstanley/neat.html. Parts of this code were significantly modified and extended to suit our needs. Also, we wrote additional supplementary MATLAB functions to work within the main code.

for a time-series forecasting problem. Furthermore, the lack of literature in this area makes it more appealing to test this method, as it would be innovative to apply this method to such a problem.

3.6 NEAT in a nutshell

NEAT uses genetic algorithms to solve three main problems: the number of hidden neurons, the network topology and the connection weight. The rest of Section 3.6 is a summary of NEAT from Stanley (2004) and Stanley and Miikkulainen (2002).

According to Stanley (2004), in traditional neuro-evolution algorithms, the initial population usually consists of a set of random topologies. The shortcoming of this approach is that it does not guarantee that the algorithm will find the simplest possible structure, which in turn could affect the network generalization (Stanley, 2004). In contrast, NEAT starts with a uniform initial population of simple networks in which inputs are connected to the output directly, without any hidden nodes. The neurons and the connection links are later evolved as a result of genetic evolution, i.e., as a result of the mutation and crossover process (Stanley, 2004).

The main reason for starting with minimal structure (Figure 3-2), according to Stanley and Miikkulainen (2002, p. 100), is "to minimize the dimensionality of the search space of connection weights" (Stanley & Miikkulainen, 2002, p. 100), which, from the forecasting point of view, should improve the generalization of the network (while for reinforcement learning, the original scope of this algorithm, this improves the speed of finding a solution for a real-time application). Moreover, the system ensures minimizing not only the final solution but also all intermediate ones as well which minimize the search space of the problem (Stanley, 2004).

3.6.1 Genetic encoding and tracking

The connection gene is encoded using direct encoding, which specifies in the genome each connection and node that will appear in the phenotype[14]. Figure 3-3 illustrates the genetic encoding of NEAT. The first row in the connection gene (Figure 3-3 upper panel) represents the connection weight (random weights in the first generation are allocated), the second and third rows define the direction of the connection *From-node →To-node* respectively. The fourth row defines whether the connection is enabled or not, the fifth row defines if the connection is recurrent or not, and finally the sixth row is a global innovation number to track genes throughout the evolution process, which helps (amongst other things) in lining up genes with different lengths to crossover (Stanley, 2004).

[14] On the other hand in the indirect encoding the rules of how connection and nodes should be formed are set without direct specification (Stanley 2004).

The information in the connection genes can be translated into the network structure in Figure 3-3 (bottom panel), where nodes 1 and 2 are the input nodes, nodes 3 and 4 are the hidden nodes and node 5 is the output node.

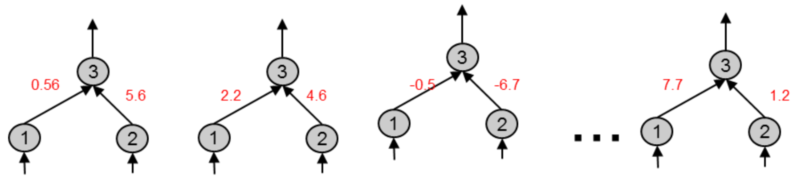

Figure 3-2 A sample of the topology of NEAT initial population
Source: adopted from Stanley (2004)

Connection gene

Weight: 1.2	Weight: -2	Weight: 0.5	Weight: 1.2	Weight: 1.4	Weight: 0.7	Weight: 0.8
From: 1	From: 1	From: 2	From: 4	From: 5	From: 4	From: 3
To: 4	To: 5	To: 5	To: 3	To: 3	To: 5	To: 5
Enable: 1	Enable: 1	Enable: 1	Enable: 1	Enable: 1	Enable: 1	Enable: 1
Recurrent: 0	Recurrent: 0	Recurrent: 0	Recurrent: 0	Recurrent: 0	Recurrent: 0	Recurrent: 1
Innovation : 1	Innovation : 6	Innovation : 2	Innovation : 8	Innovation : 3	Innovation : 4	Innovation : 7

Node gene

input 1	input 2	output 3	hidden 4	hidden 5

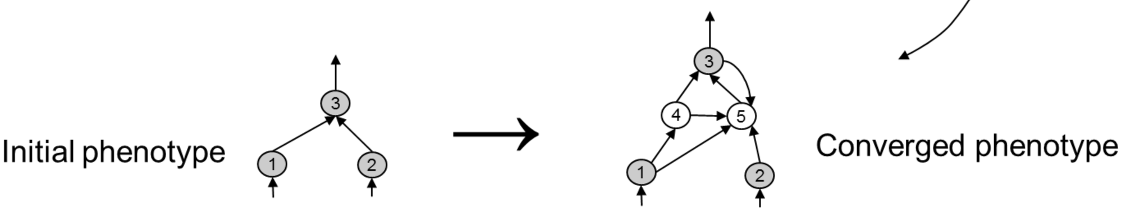

Figure 3-3: Encoding a network connection gene (upper) and node gene (bottom) illustrates the genetic encoding of NEAT.
The first row in the connection gene (Figure 3-3 upper panel) represents the connection weight (random weights in the first generation are allocated), the second and third row define the direction of the connection *From_node →To_node*. The fourth row defines whether the connection is enabled or not, the fifth row defines if the connection is recurrent or not, and finally the sixth row is a global innovation number to track genes throughout the evolution process, which helps (amongst other things) in lining up genes with different lengths to crossover.
Bottom panel: [left] Plots a network from the initial population (phenotype); [right] the network structure after the algorithm has converged. The network biases were omitted from this plot for visual clarity.
Source: (Stanley, 2004, pp. 35-36)

Mutation in NEAT could perform three different tasks (Stanley, 2004):

1. Update the connection weights (the connection weights are perturbed within pre-defined probability and within certain limits based on mutation rate. There is a non-zero probability that the weights are replaced with a totally new weight. The limits and the probability are reached experimentally by Stanley (2004).
2. Add/remove connection link gene to a genome.
3. Add/remove hidden node gene to a genome.

If a connection is divided through mutation, the old part of the connection will maintain its original weight while the new part will have (1) as a connection weight, to guarantee that the new connection does not have any negative effect on any behaviour the network has previously learnt (Buckland, 2002).

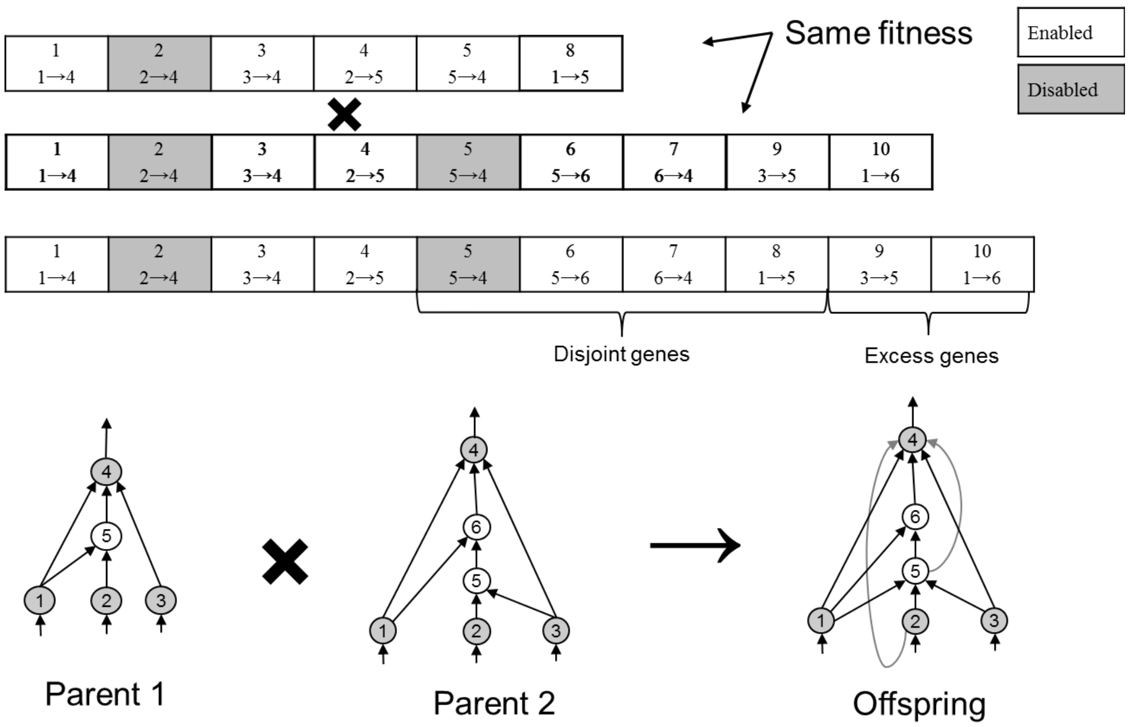

Figure 3-4: Example of network crossover

In this figure the connection genes of two different individuals are lined up for crossover. They were selected because they share some of their genetic material (having similar global innovation numbers). The grey connections in the offspring are disabled ones.

Source: (Stanley 2004, p. 37)

3.6.2 **Speciation**

After the first generation, the initial population is further divided into species, each one containing networks with similar topologies. The idea here is to prevent networks from competing with the entire population, instead making them compete with their niche. This will prevent a large network (or a complex network) taking over the population and will allow more time for a smaller network to be fully evaluated (Stanley, 2004).

To decide the best way for species allocation, the compatibility of networks must be determined. Once again the global innovation number of each gene helps in making the topology of similar networks allocated into a given species (Stanley, 2004).

The number of excess and disjoint genes between two genomes can be used to find their compatibility distance, i.e., how diverse the genomes of two individuals are. The less history two genomes share (if two genomes are more disjoint) the less compatible they are. Moreover, the connection weight is also assessed and the absolute total value is calculated. Subsequently, a compatibility distance δ of a different network structure can be obtained from the number of excess and disjoint genes and the average weight difference of matching genes (Stanley and Miikkulainen, 2002):

$$\delta = \frac{c_{1E}}{N} + \frac{c_{2D}}{N} + c_3.\,\overline{W} \tag{3.10}$$

where E is the number of excess genes and D is the number of disjoint genes, $\overline{W}$ is the average weight difference of matching genes, the coefficients $c_1, c_2,$ and c_3 help in adjusting the significance of each of the three factors above, *N* is the number of genes in the larger genome size (*N* will be 1 if both genomes are smaller than 20 genes—this number was set by Stanley and Miikkulainen (2002) experimentally).

When an individual (network) is allocated in a species it can only crossover within this species. Nevertheless, the species alone does not provide enough protection for new innovation within the population. To prevent any given species from becoming too big, fitness sharing is used. This way each individual within a species must share its fitness within its niche in this, hence, it stops any species from taking over all the population, and allows for various topologies to co-exist. This means that the score of each individual is divided by the size of the species before selection takes place. This effectively acts as a penalty term on larger species. The adjusted fitness f_i' for an individual *i* from every other individual *j* can be found as follows (Stanley, 2004):

$$f_i' = \frac{f_i}{\sum_j^n sh(\delta(i,j))} \tag{3.11}$$

where *sh*=0 when $\delta(i,j) > \delta_t$ and 1 otherwise. Hence, the denominator term of the equation 2 reduces the number of similar individuals in a given species as individual *i*. Moreover if a species did not show an improvement over a certain number of generations (15) then it is terminated, with one exception: where the best performing network is contained within this species it will be allowed to carry on.

Finally, dividing the population into species has another advantage as it reduces the effect of "bloating of genomes" (Stanley, 2004, p. 38). So, as long as the fitness of species is competitive it will survive evolution and it will not be replaced by larger species unnecessarily[15] (Stanley, 2004).

[15] In my personal experiments this is not happening as species are dying out very quickly and being replaced by larger ones. At this stage I am not sure how this has happened.

3.6.3 Structure minimization

One of the main advantages of NEAT is it starts minimally[16] and evolves toward a complex structure through structural mutation and only networks that improve the fitness can survive (Stanley, 2004). As such, the algorithm tries to search in a minimal number of weight dimensions, which reflects on the number of generations that are required to find a solution. In other words, the algorithm is searching a very low-dimensional parameter space with very few connections.

3.7 Extending NEAT

Since NEAT was initially designed to solve benchmark problems and not time-series forecasting ones, it was essential to change the fitness function to a more suitable one. Before continuing, it is important to note that in reinforcement learning algorithms, the fitness function plays a significant role for reaching the solution, as it is the only feedback for the search process on how well the current sets of parameters is performing. In particular, changing the fitness function in NEAT not only affects the optimisation of the problem at hand, but also it will affect the model (network) topology. The evolution process in NEAT, mainly the crossover, is affected by the fitness of individuals (networks). More specifically, if a phenotype (network) did not produce competitive fitness for a number of generations it will not be allowed to reproduce (crossover). As a result, the structure (complexity) across the entire population will be affected. Therefore, modifying the fitness function could have a different effect on the final results of NEAT compared to modifying the cost function in a fixed structure network (this is similar to what has been done in the literature so far). Therefore, it is of utmost importance to handcraft a fitness function suitable for the problem.

Hereafter, we present several cost functions used in the literature, then, building on these functions we introduce several novel fitness functions of our own.

3.7.1 Alternative fitness function

3.7.1.1 New fitness functions with crisp rules

The mean squared error (MSE), has been widely used as a cost function for ANN; however, many scholars have argued that it might not be ideal for financial forecasting problems. For example, refer to Caldwell (1995); Refenes, Bentz, Bunn, Burgess, and Zapranis (1997); and Yao and Tan (2001), especially when forecasting the direction is more important (or more realistic to achieve) than forecasting the value. Some scholars have suggested some modification designed for forecasting financial time series with ANN. Tilakaratne, Mammadov and Morris, (2008; 2009)

[16] It is important to note that a standard NEAT does not start with the most minimal structure possible because all the inputs are connected to the output. This relies on a strong assumption that all the inputs are necessary and useful to find a solution (Whiteson, Stone, Stanley, Miikkulainen and Kohl, 2005).

surveyed the literature dealing with error functions for neural networks and derived new error functions based on the work of Yao and Tan (2001) (Equation 3.12, 3.13 and 3.16) and Refenes, Bentz, Bunn, Burgess, and Zapranis (1997) (Equation 3.14 and 3.15). The error function introduced by Tilakaratne, et al. (2008; 2009) was more suited to trading systems rather than forecasting. In this paper we follow Tilakaratne, et al.'s (2008; 2009) approach and propose a new fitness function more suited to our model. For the sake of completeness we have summarized the previous work directly related to our new function.

Yao and Tan (2001) proposed an error function that penalizes incorrect prediction of the direction (Equation, and) while it rewards the correct prediction of the sign:

$$E_{DP} = \frac{1}{2N}\sum_{t=1}^{N} f_{DP}(t)(tar_i - fore_i)^2, \tag{3.12}$$

where $$d(t) = \begin{cases} c_1 \; if \; (\Delta tar_t - \Delta fore_t) > 0 \; and \; \Delta tar_t \leq \sigma, \\ c_2 \; if \; (\Delta tar_t - \Delta fore_t) > 0 \; and \; \Delta tar_t > \sigma, \\ c_3 \; if \; (\Delta tar_t - \Delta fore_t) > 0 \; and \; \Delta tar_t \leq \sigma, \\ c_4 \; if \; (\Delta tar_t - \Delta fore_t) < 0 \; and \; \Delta tar_t > \sigma. \end{cases} \tag{3.13}$$

$\Delta tar_t = target_t - target_{t-1}, \Delta fore_t = forecast_t - forecast_{t-1} \; and \; \sigma$ is the standard deviation of the target, while c_1, c_2, c_3 and c_4 are positive constants.

Refenes, Bentz, Bunn, Burgess, and Zapranis (1997) proposed another error function that takes the recurrence of the observations. In this function learning is biased towards more recent observations:

$$E_{DLS} = \frac{1}{2N}\sum_{i=1}^{N} \gamma_\tau(t)(tar_t - fore_t)^2 \,, \tag{3.14}$$

where γ_τ is an adjustment for the contribution of the *ith* value of the series and is given by

$$\gamma_\tau(t) = \frac{1}{1 + \exp(\tau - \frac{2\tau t}{N})} \tag{3.15}$$

Yao and Tan (2001) combined the concept of Refenes, et al. (1997) with their own, resulting in a new error function:

$$E_{TDP} = \frac{1}{2N}\sum_{t=1}^{N} f_{TDP}(t)(tar_t - fore_t)^2 \tag{3.16}$$

Based on the above we propose a new fitness function to the one proposed by Stanley[17] (2004) (author of NEAT) by multiplying E_{TDP} (Equation 3.16) by the term δ_t which is derived from the Sharpe ratio:

$$E_s = (10 - E_{TDP}) * \delta_t \tag{3.17}$$

[17] The fitness function defined by Stanley was: $fitness = (4 - error)^2$; so higher fitness reflects a better network.

where
$$\delta_t = \begin{cases} 1.5 \; if \; S \geq 1, \\ 1.1 \; if \; 0 < S < 1, \\ 1 \;\; if \; S = 0, \\ 0.9 \; if \; S < 0, \end{cases} \tag{3.18}$$

and $S = \frac{r_A - r_f}{\sigma_A}$, r_A is the forecasted return (the outcome of the algorithm for each network at each generation), r_f is the risk-free rate of the return (here we use the mean of the actual in-sample return) and σ_A is the volatility measure. Initially, we used the value of the Sharpe ratio outright as a weight; this way a better Sharpe ratio will result in better network fitness while a poor Sharpe ratio will have lower fitness in proportion to its value. However, after many experiments, we find that the fixed weights proposed above are the best combination for reaching consistent results.

The use of the Sharpe ratio as a performance measure is quite common in the financial literature. Nonetheless, there are very few attempts, to the best of our knowledge, to use it as a fitness function. Perhaps the best attempt in this direction was made by Moody, Wu, Liao and Saffell (1998) as they presented a differentiable Sharpe ratio as a fitness function for a reinforcement learning algorithm: this function was optimized in the online mode. In contrast, our approach is, instead of using the Sharpe ratio as the sole fitness function (and ignoring the MSE and sign accuracy), which in our experiment yielded a very unstable fitness function, we make use of this important ratio as a reward and penalty method (as we have done in Equation (3.17) and (3.18)).

3.7.1.2 New fitness functions based on fuzzy logic

We also introduced a new and more robust fitness function which makes use of fuzzy logic. This fitness function is designed to find a balance among: (i) the absolute error between the algorithm output and the actual target, (ii) the sign prediction (hit rate), (iii) the recentness of the error, (iv) the Sharpe ratio, and R^2 from a fitted liner regression model between the algorithm output and the actual target[18]. Only item (iv) depends on a fuzzy logic inference system. Equation (3.18) presents this new fitness function.

$$E_{fuzzy} = (10 - E_{TDP}) \times W, \tag{3.19}$$

where E_{TDP} is presented in Equation (3.16) and the term W in Equation (3.19) is a weight that can take any value between 0 and 2.5. The weight is determined by a set of fuzzy rules which make a balance between the Sharpe ratio ([$S = \frac{r_A - r_f}{\sigma_A}$], r_A is the forecasted return (the outcome of the algorithm for each network at each generation), r_f is the risk free rate of the return, σ_A is the volatility measure and the R^2 forms a fitted linear regression between the network output and the

[18] It could be argued that the use of R squared in this function is redundant, since we are already accounting for the error. However, the R squared is used here to balance the Sharpe ratio in the event that when the model generates a higher Sharpe ratio than usual (outlier), this avoids rewarding poor performance.

actual return for each network at each generation. Figure 3-5 shows the weight surface in relation to the Sharpe ratio and the R^2 based on the shape and values of the membership functions for each rule.

Any weight below (1) represents a penalty on the fitness of the phenotype (network) which in turn will affect the entire species due to the fitness sharing function, while weights greater than one reward the phenotype and hence the species.

The fuzzy rules for this function are:

1. If Sharpe ratio is high AND R^2 is high THEN weight is high.
2. If Sharpe ratio is high OR R^2 is high THEN weight is medium.
3. If Sharpe ratio is low AND R^2 is low THEN weight is low.

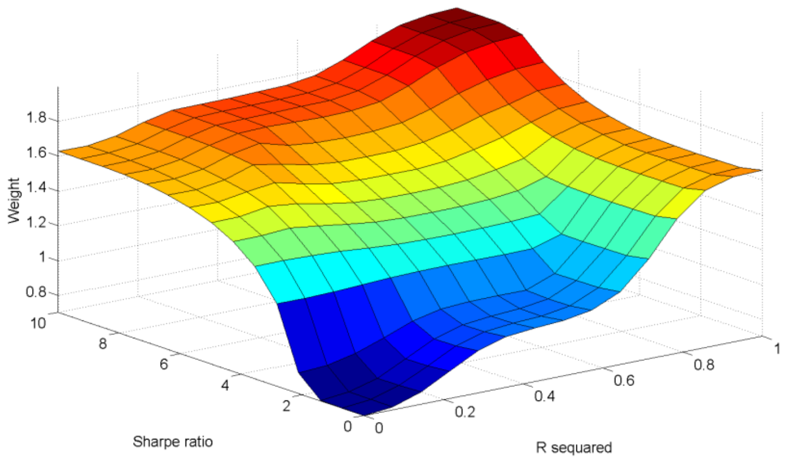

Figure 3-5: [Coloured] A 3D plot of the weight surface in relation to the Sharpe ratio and the R^2
The *x* axis shows the shape of the membership function for the Sharpe ratio; the y axis shows the value of the weight, while the *z* axis is the value of the R squared.

Source: This figure was generated by MATLAB fuzzy logic toolbox based on our membership function

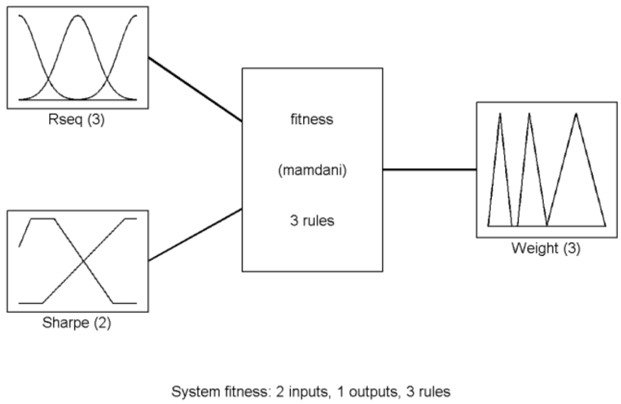

Figure 3-6: A summary of the fuzzy rule system and the memberships function for the inputs and the output
The first two blocks on the left side show the shape of the membership function for R^2 (upper block) and Sharpe ratio (lower block) with the number of rules associated with each; the middle block summarizes the fuzzy system—it shows from top to bottom the system name, the fuzzy inference system we are using (by Ebrahim Mamdani, 1975) and the number of rules. The final block shows the membership function for the system output.
Source: This figure was generated by MATLAB fuzzy logic toolbox based on our model parameters

The membership function for the R^2 is based on Gaussian distribution while a trapezoidal function is used for the Sharpe ratio. Equation (3.19) is subtracted from ten (10) because NEAT deals with positive fitness only, so a greater fitness means a better solution.

The degree of membership is more sensitive to small improvements in the Sharpe ratio than the R^2, since our goal is to achieve a financially rewarding prediction. It is noteworthy that negative values for both inputs are regarded as zero.

3.7.1.3 Simple fuzzy function

The fuzzy fitness function we introduced in the previous section includes many components which could play a role in shaping the final output. To isolate the effect of the fuzzy component we introduced two simple functions and both of the functions use the same fuzzy logic inference system (a point which could be changed in the future).

The first fitness function is given by:

$$E_{fuzzy1} = (10 - e)^2 \times v, \tag{3.20}$$

where e is the error between the network output and the actual output and v is the fuzzy weight generated using the same fuzzy inference system.

The second fitness function is as in Equation (3.21) but the fuzzy weights this time are generated between the Sharpe ratio and the hit rate for each network output at each generation, (instead of the R^2):

$$E_{fuzzy2} = (10 - e)^2 \times j. \tag{3.21}$$

We used a bounded return where outliers were replaced by a minimum and maximum bound of (-0.06 and +0.06) respectively during the testing phase.

3.7.2 Feature Selection NEAT

Whiteson, Stone, Stanley, Miikkulainen and Kohl (2005) presented an extension to the regular NEAT which allowed the algorithm to perform the crucial task of finding the best input/s from the initial input space using the same concept of neuro-evolution, calling their approach feature selection NEAT (FT-NEAT). Under this extension, the algorithm is set to randomly select one of the inputs to be connected in the first generation while the rest of the inputs are kept disconnected. Then as structural mutation takes place, new connections will be added and only a connection that contributes to improving the fitness will survive the evaluation.

This means that the algorithm will start with the minimal structure possible and evolve to a more complex one as needed. Hence, this should reflect positively on the algorithm efficacy, i.e., reaching the solution with the smallest network structure and a smaller number of generations. Moreover, this also will provide domain knowledge as to what input/s are best for a given problem, although this point was not recognized by the authors. The empirical findings of Whiteson, et al. (2005) showed that FT-NEAT is much more efficient than regular NEAT since the complexity of the networks generated by the algorithm tend to level out after a number of generations in contrast to regular NEAT in which it continues to grow.

On the other hand we find that starting with a network in which the crude return $(t-1)$ is connected generates a better outcome than connecting an input randomly. This is because the algorithm can still start minimally, but the addition of some kind of knowledge about the problem and the most likely best input candidate both contribute toward finding a solution. This is done by pre-connecting one of the inputs (by utilizing expert knowledge) and leaving the rest of the inputs unconnected, thus new connections will be added and removed by the evolution process. The main advantage of our approach is even if the initial knowledge was wrong, i.e., the input we initially connected is not the best one, we will end up with FT-NEAT which is, according to Whiteson, et al. (2005), much more efficient than regular NEAT. However, if our knowledge about the input selection is correct then we should have a much more efficient algorithm than both NEAT and FT-NEAT.

3.7.3 Advantage of NEAT in the context of this thesis

1. To the best of my knowledge it is the first time this algorithm is applied to time-series forecasting, and definitely the first time it is applied to a financial problem.

2. There is a potential for extending the algorithms:
    a. To make the algorithm find the effective number of inputs. Thus, instead of starting with all the inputs connected to the output we leave the algorithm to find which input is helping to improve the solution.
    b. To make the algorithm find the optimal length of the input window.
3. By using this algorithm in my research it closes all the gaps related to the methods used (tools) to achieve the forecast.
4. The network topology we will obtain will be the means of providing domain knowledge for the problem. Only then would a rule extraction make sense.
5. It provides a systematic way of selecting the network architecture based on GA (not necessarily the optimal architecture but much more justifiable than trial and error).
6. Weight optimisation (updated) of the network is done through a global search (GA) instead of a local search (back-propagation). This reduces the possibility for the network to become stuck in local minima due to the process of mutation.
7. The algorithm is oriented towards finding the minimal structure; hence, the entire initial population is uniform of input layers directly connected to the output and no hidden layers are used.
8. The system involves local recurrent connection thus it avoids over-parameterisation by employing unnecessary recurrent connections. This is a unique feature of this algorithm.

3.7.4 Outstanding problems with NEAT

There is a very limited body of literature dealing with reinforcement learning for time-series prediction. Since NEAT is based on genetic algorithms, it is computationally intensive. We used parallel computing programming to improve and extricate algorithm convergence time. There is also a consistency issue, because NEAT is a search algorithm and not a learning one, so the final solution depends heavily on the initial weight space, and therefore it is not generating consistent results, especially for the same problem.

3.8 Performance metrics

An important issue in this study is how to measure the performance. For financial and economic forecasts, two types of performance metrics are commonly used: statistical metrics to assess the goodness of fit and economic metrics to assess model profitability. Statistical metrics are the most commonly used in this type of study. However, Satchell and Timmermann (1995) argued that, unlike linear models, when dealing with non-linear models, a large mean squared error (MSE) does not necessarily correspond to a low possibility of high sign prediction. Therefore, MSE is not the best performance metric when assessing the forecast of non-linear economic/financial time series. Instead, the sign prediction, i.e. the hit rate (Equation 1 in Table 3-1) of the returns is more likely to give a better indication of the economic importance of the forecast, even though this does not mean that predicting the sign automatically translates into better economic significance (Satchell & Timmermann, 1995). Related to this, from a trading point of view, a model with low forecast accuracy could still generate profits, if it manages to capture large price shifts (Vanstone, 2005).

The performance metrics used in the numerical experiments are presented in the table below with a brief description of each. The metrics can be grouped into two categories: statistical metrics (Equations 1 to 11) and profitability metrics (Equations 12 to 14). These metrics are gathered from Refenes (1995) and McNelis (2005). The R^2 is used in this research as a metric for the sake of completeness but we believe it is not suitable to assess a model that deals with noisy time series. Therefore, throughout this thesis, we assess the accuracy of the forecasting models based mainly on the hit rate.

Metrics	Explanation	No.
Hit rate	$HR = \frac{1}{n}\sum_{t=1}^{n} I$ $I = 1$ if $y_t . \hat{y}_t > 0$, and 0 otherwise; where: n be the sample size $y_t, \hat{y}_t$, are the value of the target and the output at time t consecutively.	(1)
Root mean squared error	$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(y_t - \hat{y_t})^2}{n}}$	(2)
Mean squared error	$MSE = \frac{\sum_{t=1}^{n}(y_t - \hat{y_t})^2}{n}$	(3)
Mean absolute error	$MAE = \sum_{t=1}^{n} \frac{\lvert y_t - \widehat{y_t} \rvert}{n}$	(4)
Sum squared error	$SSE = \sum_{t=1}^{n}(y_t - \hat{y_t})^2$	(5)
R, and R squared	$R = \frac{\sum_{t=1}^{n}(y_t - \bar{y})(\hat{y}_t - \bar{\hat{y}})}{\sqrt{\sum_{t=1}^{n}(y_t - \bar{y})^2 \sum_{t=1}^{n}(\hat{y}_t - \bar{\hat{y}})^2}},$	(6)

	$$R^2 = \left(\frac{\sum_{t=1}^{n} (y_t - \bar{y})(\hat{y}_t - \bar{\hat{y}})}{\sqrt{\sum_{t=1}^{n} (y_t - \bar{y})^2 \sum_{t=1}^{n} (\hat{y}_t - \bar{\hat{y}})^2}} \right)^2$$	
Information coefficient (IC)	$$IC = \frac{\sqrt{\sum_{t=1}^{n} (\hat{y}_t - y_t)^2}}{\sqrt{\sum_{t=1}^{n} (y_t - y_{t-1})^2}}$$ where y is the predicted value, and $\hat{y}$ is the actual value. This ratio provides an indication of the prediction compared to the trivial predictor based on the random walk where $IC \geq 1$ indicates poor prediction, and $IC < 1$ means the prediction is better than the random walk (Refenes, 1995).	(7)
DA metric by Pesaran-Timmerman (Pesaran & Timmerman, 1992)	$\hat{y}_{n+j}, j = 1, \ldots, n$ $I_j = 1$ if $\hat{y}_{m+j} . y_{m+j} > 0$ and 0 otherwise $HR = \frac{1}{n} \sum_{t=1}^{n} I_j$ $I_j^{true} = 1$ if y_{m+1}>0, and 0 otherwise $I_j^{pred} = 1$ if $\hat{y}_{m+j} > 0$, and 0 otherwise $P = \frac{1}{n} \sum_{j=1}^{m} I_j^{true}, \hat{P} = \frac{1}{n} \sum_{j=1}^{m} I_j^{pred}$ $S = P.\hat{P} + (1-P).(1-\hat{P})$ $var(S) = [\frac{1}{n}(2\hat{P} -)^2 P(1-P)^2 + (2P-1)^2 \hat{P}(1-\hat{P}) +$ $\frac{4}{n} P.\hat{P}(1-P)(1-\hat{P})]$ $DA = \frac{HR-S}{var(HR-var(S)} \overset{a}{\sim} N(0,1).$	(8)
Mean reverting (TU)	$$TU = \frac{\sqrt{\sum_{t=1}^{n} (y_t - x_t)^2}}{\sqrt{\sum_{t=1}^{n} (\bar{x} - x_{t+1})^2}}$$ where y is predicted and x is actual and $\bar{x}$ is mean x. If TU=1 then we are forecasting nearly the mean. If TU<1 our forecast is better than the mean. And any TU>1 indicates the forecast is worse than the mean.	(9)
Akaike information criterion	$$AIC = \frac{1}{n} \sum_{i}^{n} (x_i - y_i)^2 \left[\frac{n+k}{n-k} \right]$$	(10)

Bayesian information criterion	$$BIC = ln\left[\frac{\sum_i^n (x_i - y_i)^2}{n}\right] + \frac{ln[n]}{n}k$$ where k is the number of free parameters of the model and n is the number of observations.	(11)
Net return	$$net\ return = \sum_i^n p_t \times (x_{t+1} - x_t)$$ where $$p_t = \begin{cases} 1\ if\ (\hat{x}_{t+1} - x_t) > 0 \\ -1\ if\ (\hat{x}_{t+1} - x_t) < 0 \\ 0\ if\ (\hat{x}_{t+1} - x_t) = 0 \end{cases}$$ x is the actual value and $\hat{x}$ is the forecasted value. The profitability of the model will be evaluated against the benchmark model.	(12)
The Sharpe ratio	$$s = \frac{r_A - c}{\sigma_A}$$ where c is the risk-free rate; r_A is the return; and σ_A is the volatility (standard deviation). According to Refenes (1995) this ratio has one limitation: in case of increased upside volatility, this will affect the ratio and generate a low value for it.	(13)
The realized potential	$$r_d = \frac{\sum_i^n p_t \times (x_{t+1} - x_t)}{\sum_i^n \lvert x_{t+1} - x_t \rvert}$$ with p_t is as defined is Equation 12. It is important to note that the realized potential value will exaggerate network performance as the transaction cost and the taxes are ignored.	(14)

Table 3-1: The performance metrics used in this research

In this table we show the main performance metrics used in the numerical experiments; we place a significant emphasis on the hit rate as predicting the direction accurately is the objective of this thesis. The R^2 is used in this research as a metric for the sake of completeness but we believe it is not suitable to assess model that deals with noisy time series.

3.9 Summary

In this chapter we presented the main artificial intelligence tools we use in this thesis. The first tool is the traditional artificial neural network trained by a supervised learning approach (backpropagation). The second is a reinforcement learning NEAT model which will be the main platform to implement the multi-agents model (see Chapter 7). Finally, we explain the metrics we used in this thesis to assess our models' performance. In the next chapter we present novel extensions to NEAT in order to make it more appropriate to deal with financial/commodity time-series data.

CHAPTER 4: A new look at the non-linear dynamics of crude oil prices

4.1 Introduction

Forecasting the crude oil price is among the most important issues facing energy economists. Nevertheless, the success in formulating a reliable model to describe the complex dynamics of this commodity is limited. Between 2004 and 2007 the crude oil price increased significantly to reach the highest level ever of 140 U.S. dollars per barrel before crashing to 30 dollars per barrel in the aftermath of the global financial crisis. Since then the market has recovered quickly, especially after the global economy showed evidence of recovery. However, all these events have reignited the question about the actual dynamics of the crude oil price and whether we can predict changes in the crude oil price, or if it is merely a random walk.

This chapter attempts to answer the following questions: what type of dynamics is governing crude oil prices and returns? Specifically, we investigate if there are any non-linear deterministic dynamics (chaos) which could be misspecified as a random walk. Also, do we have strong empirical evidence that crude oil spot returns are predictable in the short-term? From a statistical point of view, have the dynamics of crude oil returns changed significantly during the past twenty years? Moreover, we investigate how we can use this information to select and calibrate a forecasting model, including the forecast accuracy and horizon. We believe that answering these questions will assist in formulating a forecasting model for crude oil prices/returns, and will set more realistic goals for the expected forecast accuracy and horizon which could be obtained using a pure time-series approach.

To answer these questions we start with testing the existence of non-linearity and chaos in a crude oil spot price/return series using a number of tests. Moreover, we attempt to explore what type of dynamics drive crude oil spot price/return using the fuzzy classification test for non-linearity, introduced by Kaboudan (1999). We then turn our attention to formulating a short-term forecasting model for crude oil spot prices/returns.

A significant amount of this chapter was adopted from:

- Haidar, I., & Wolff, R. C., (2011). Forecasting Crude Oil Price (Revisited). In the Proceedings of the 30th USAEE/IAEE, 9-12 October 2011, Washington DC.

4.2 **Methodology**

4.2.1 **Data**

The main series represents crude oil daily spot prices/returns for the West Texas Intermediate (WTI) official closing price from 2 January 1986 to 2 March 2010 (6194 daily observations). The data were retrieved on 11 March 2010 from the Energy Information Administration (EIA) website (http://www.eia.doe.gov/).

4.2.2 **Diagnostic Tests**

Figure 4-1 (upper panel) shows a plot of the crude oil spot price (daily close price) for WTI, from 2 January 1986 to 2 March 2010. As can be seen in this figure, crude oil prices increased significantly between 2007 and 2008, followed by a price crash as an aftermath of the global financial crisis of October 2008. On the other hand, crude oil returns (Figure 4-1 lower panel) show evidence of volatility clustering and some outliers. The interaction between the crude oil price and other events/variables are not the scope of this study; rather, we are interested in finding out about the behaviour of the series itself, which might have been shaped by such events.

The series was divided into two subsections: Return I from 1 January 1986 to the end of December 1998, and Return II from 1 January 1999 to the end of February 2010.

4.2.2.1 Unit root tests

The autocorrelation function and the partial correlation function show that there is weak evidence of serial correlation in the crude oil return. On the other hand the autocorrelation is much more evident in the squared log-returns, especially for Return II, which was significantly over the upper confidence level. This could present evidence of heteroskedasticity. In addition, we test if the crude oil price and returns are weakly stationary. A series can either be stationary or non-stationary: a weakly stationary process has a constant mean, a constant variance, and constant autocovariances for each given lag. Mathematically, the time series must satisfies the three conditions for $t = 1,2,3, \dots, \infty$ (Brooks, 2008, p. 208):

$$E(y_t) = \mu \tag{4.1}$$

$$E(y_t - \mu)(y_t - \mu) = \sigma^2 < \infty \tag{4.2}$$

$$E(y_{t1} - \mu)(y_{t2} - \mu) = \rho_{t2-t1} \quad \forall\, t_1, t_2 \tag{4.3}$$

For a series to be considered strictly stationary, the distribution of its values does not change over time. In this thesis we are referring to weakly stationary processes only. If a non-stationary series is integrated of order d or $\mathrm{I}(d)$, it requires to be differenced d times to become weakly stationary and is conceded to have no unit root. On the other hand, a weakly stationary series has no unit roots or $\mathrm{I}(0)$ (Brooks, 2008; McNelis, 2005).

For econometric models, modelling with a stationary series is important for many reasons— a non-stationary process can strongly influence its behaviour and properties. For example, “shocks” to the system will gradually disappear in a stationary series, but for a non-stationary series it may take a longer time for shocks to fade. Also, modelling with a non-stationary series could lead to *spurious regressions.* Applying standard regression techniques to non-stationary data gives an invalid regression output (a large R^2, even if the data sets are unrelated). Hypothesis tests about the regression parameters may not be valid. If a regression is run on non-stationary data, the standard assumptions for asymptotic analysis (i.e., classical Normal theory for hypothesis testing) cannot be applied (Brooks, 2008).

We applied two tests for a unit root, the augmented Dickey-Fuller (ADF) and the Phillips–Perron test. The ADF test is given in Equation (4.4):

$$\Delta y_t = \rho y_{t-1} + \sum_{j=1}^{P} \alpha_j \Delta y_{t-j} + \varepsilon_t \tag{4.4}$$

where $\Delta y_t = y_t - y_{t-1}, \rho,\ \alpha_j(j = 1 \dots, P)$ are the model coefficients to be estimated, and ε_t is random error. ADF tests for a unit root in a time series by examining the null hypothesis that $\rho = 0$. If the null hypothesis cannot be rejected, then the series is stationary. If the null hypothesis is rejected, then the series contains a unit root. Further hypothesis testing is needed if $\rho \neq 0$, to determine if the series contains one or more unit roots. The Phillips-Perron (PP) test is similar to the ADF but it differentiates itself in the way it corrects for autocorrelation in the residuals (Brooks, 2008; McNelis, 2005).

The results of unit root tests, both the ADF and the Phillips–Perron tests for crude oil prices and returns at 1% significance level are as follows:

- Crude oil price for the whole series from January 1986 to March 2010 is integrated of the first order, or I (1).
- Crude oil price for the first subsection from January 1986 to January 1998 is I (0).
- Crude oil price for the second subsection from end of January 1998 to March 2010 is I (1),
- Thus the returns for all subsections are I (0).

- Return II is from 1 January 1999 to the end of February 2010[19].

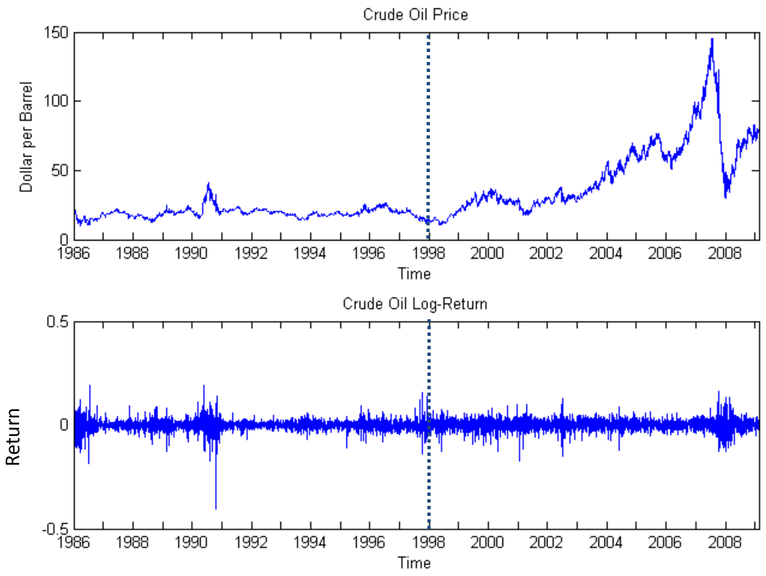

Figure 4-1: Crude oil daily spot price (upper) and crude oil daily log return (lower)
The upper panel shows a plot of the crude oil daily nominal spot price covering the period from 1986 to 2010. The lower panel shows crude oil logarithmic returns for the same period. The vertical dotted line separates period (I) and period (II) in our analysis.

In other words, the price series are non-stationary except the crude oil price from January 1986 to January 1998 and all the returns series are weakly stationary. For brevity we only show the unit root tests for the crude oil spot price in Table 4-1:. Nevertheless, caution needs to be taken with ADF and the PP test as there is a general belief that these tests have low power detecting unit root especially if the series has a structural break, hence, these tests are biased towards the unit root in the presence of structural change.

Structural breaks are characterized by abrupt change in the parterres of the data generating function (Valentinyi-Endresz, 2004). Suppose a time series y_t $\{t_0 = t - N + 1$ to current time, $t > t_0\}$ is generated by an AR(1) model Equation (4-1), a significant change in the intercept or a change in the error volatility would be lead to a structural change in the series (Valentinyi-Endrész, 2004):

$$y_t = \mu + \rho y_t + \varepsilon_t \tag{4.5}$$

$$E[y] = \frac{\mu}{1 - \rho} \tag{4.6}$$

$$Var[y] = \frac{\sigma^2}{1 - \rho^2} \tag{4.7}$$

[19] A general rule of thumb when using ANN for financial forecasting is to use ten years of historical data (for daily data) for training and one year for out-of-sample testing. This way we can avoid using redundant behaviour from old observations but at the same time have long enough data for training.

where $\varepsilon_t \sim \mathrm{idd}(0, \sigma)$ is white noise.

Structural breaks are often trigged by economic influences and major political events. In the case of crude oil, a change in OPEC policies, political factors, or a shift in the supply/demand balance may cause theses breaks. In the current crude oil sample and as can be seen in Figure 4-1 the structural break took place as a consequent to the Global Economic Crisis of 2007-2008.

It is very important to differentiate between information shocks and structural breaks. Information shocks in financial and commodity markets are also caused by economic and political circumstances, which leads to changes in the price level, however, these changes are usually short lived, price change in the case of structural break, however, persist for long time (Timmermann, 2001).

Perron (1989) introduced a modified ADF test to account for single exogenous structure breaks. The main issue with Perron's (1989) test is that a judgment needs to be made about the time of break. Zivot and Andrews (1992) proposed a unit root test that accounts for endogenous structural breaks in the series. The Zivot and Andrews (1992) test transforms Perron's (1989) exogenous unit root test and finds the date for break point based on the t-statistics from the data. In Zivot and Andrews' (1992) model as in Perron's (1989) they have three models: Model A that allows for structural break on the intercept only, i.e., the level of the series (Equation 4.8); model B that allows for structure break on trend only (Equation 4.9) and Model C which allows for structural break on intercept and trend (Equation 4.10).

$$y_t = \alpha^A + \gamma^A DU_t(\lambda) + dDU_t(\lambda) + \beta_t^A + \rho^A y_{t-1} + \sum_{i=1}^{p} \emptyset^A \, \Delta y_{t-1} + e_t \qquad (4.8)$$

$$y_t = \alpha^B + \beta_t^B + \gamma^B DT_t(\lambda) + \rho^B y_{t-1} + \sum_{i=1}^{p} \emptyset^B \, \Delta y_{t-1} + e_t \qquad (4.9)$$

$$y_t = \alpha^C + \alpha_1 DU_t + \gamma^C DU_t(\lambda) + \beta_t^C + \gamma^C DT_t(\lambda) + \rho^C y_{t-1} + \sum_{i=1}^{p} \emptyset^C \, \Delta y_{t-1} + e_t \qquad (4.10)$$

where $(y_t)_1^T$ is a time series, T is the break time, $DU_t(\lambda)$ is the intercept dummy to capture the change in the level and take the value of 1 if $t > T\lambda$ and 0 otherwise. The slope dummy DT_t accounts for the change in the slope of the trend $DT_t = t$ if $t > T$, and 0 otherwise. The break point is selected by finding the values of T that have the t-statistic for when α is minimized.

Table 4-12 shows the results of the Zivot and Andrews (1992) test (Model C) on crude oil price. As can be seen in Table 4-12, the test confirms the conclusion of stationary series for price I at 99%

confidence limit. However, it contradicted results of the ADF and PP for price II, as it shows that price II is also a stationary series after controlling for the structural break in the series. The test identified 16-Jan-1991 as a possible breakpoint in the data for price I; this data coincided with the start of the military operation to liberate Kuwait when the oil price increased significantly over a short period of time.

Price I						
	ADF test			**PP test**		
Lag	**P value**	**T stat**	**C value***	**P value**	**T stat**	**C value***
1	0.0013	-4.5762	-3.9669	0.0013	-4.5762	-3.9669
4	0.0096	-3.9843	-3.9669	0.0031	-4.3859	-3.9669
8	0.0213	-3.7234	-3.9669	0.0057	-4.1465	-3.9669
Price II						
	ADF test			**PP test**		
Lag	**P value**	**T stat**	**C value***	**P value**	**T stat**	**C value***
1	0.2903	-2.6106	-3.9669	0.2903	-2.6106	-3.9669
4	0.-3674	-2.4549	-3.9669	0.3540	-2.4819	-3.9669
8	0.4189	-2.3509	-3.9669	0.3607	-2.4684	-3.9669

Table 4-1: Unit root test for crude oil daily spot price from January 1986 to end of January 1998
The upper section of this table shows the ADF and PP tests for the crude oil price in Period I. The lower section of this table shows the same tests for the crude oil price in Period II. The test is reported only for 1, 4, and 8 lags. The ADF test shows that there is significant statistical evidence to reject the null hypothesis for the unit root for lags from 1 to 5 but not from 6 to 8 at 1% level of significance, while the results of the PP test reject the null hypotheses of the unit root for all lags at a significance level of 1%. For the 5% significance level (results not shown here) both tests reject the null hypotheses of unit root for all lags, i.e., the crude oil spot price for this subsection is I (0).*C value is the critical value, and the null hypotheses of the unit root is accepted when the critical value is greater than the *t* statistic of the test at a given *p* value.

Series	TB	Lags	t	Diction	Corresponding break time
Price I	16-Jan-1991	4	-7.9993***	Stationary	Golf War: start of Military action to liberate Kuwait
Price II	19-Sep-2008	4	-6.1308***	Stationary	During Global Economic Crisis

Table 4-2: Zivot and Andrews (1992) unit root test for crude oil price I and price II
The critical values: 0.01= -5.57 0.05= -5.08 0.1= -4.82, for 99%, 95% and 90% significance level respectively *** indicate 95% level of significance

4.2.3 Testing for non-linearity and chaos

A few studies investigated the dynamic nature of crude oil time series. One of these was by Moshiri and Foroutan (2006), mentioned in the introduction to this thesis, in which the authors applied several tests for non-linearity and chaos to crude oil futures prices. The main conclusion of their analysis was that non-linear dynamics are present in crude oil futures prices; however, the authors could not find evidence of chaos in these series. More importantly, the type of non-linear dynamics was not detected. These results were later challenged by Matilla-Garcia (2007) who found evidence of chaos in crude oil futures prices as well as in the natural gas price and unleaded gasoline futures. A noteworthy point here is that Matilla-Garcia (2007) based their conclusion regarding chaos on direct estimation of the Lyapunov exponent solely, while Moshiri and Foroutan (2006) used the

Jacobian approach to estimate the largest Lyapunov exponent[20]. The results of Matilla-Garcia (2007) also contradicted an earlier study by Adrangi, et al. (2001) who could not find any evidence of chaos in crude oil futures prices.

It seems there is no consensus in the literature about the dynamics of the crude oil price, i.e., whether crude oil follows linear or non-linear dynamics and what is the type of non-linear dynamics—stochastic, deterministic or chaotic—which motivates us to revisit this topic. In this chapter we concentrate only on the crude oil spot price series. For non-linearity, three tests were applied, namely the Brock, Dechert and Scheinkman (BDS) test, the fuzzy classification system (FCS) test and the time domain test for non-linearity. We chose these three non-linearity tests to increase the confidence of a non-linearity conclusion and avoid the limitations of any individual test. To elaborate, Brooks and Henry (2000) showed that the BDS test is not able to find asymmetries present in the data, making it incapable of determining, in some cases, whether the data is independent and identically distributed. As such, our strategy in this chapter is to apply different tests for non-linearity and cross-match the results. Testing for non-linearity is the first step in our methodology because if the data did not show evidence of non-linear behaviour, a linear model would be a better approach. On the other hand, if the data showed evidence of non-linear behaviour, a non-linear model would be superior in forecasting the series. As for the type of non-linearity, we rely on two approaches: an elimination process, in which we test for chaos and try to confirm or eliminate its presence, and secondly a fuzzy classification test proposed by Kaboudan (1999).

4.2.3.1 The BDS Test[21]

The Brock, Dechert and Scheinkman (BDS) test (Brock, et al., 1996)[22] is widely used in economic applications, and is based on the correlation integral. The correlation integral can be described as follows. Let X_i be a time series for $i = 1,2,3, \dots, n$, then the correlation integral measures the number of pairs from X_i and a delayed version of the series X_i for $j = 1,2,3, \dots n - m$ that has a

[20] Moshiri and Foroutan (2006) claimed that using the direct approach of estimating the largest Lyapunov exponent in their study did not change the conclusion about chaos. The main issue with the direct approach vs. the Jacobian approach of estimating the Lyapunov exponent is that the latter is more sensitive to the noise in the data, as Matilla-Garcia (2007) pointed out. In principle, though, this view is correct as the estimations of the Jacobian matrix will amplify the effect of the noise, yet in Moshiri and Foroutan (2006), as well as in this chapter, noise filters were applied to the data which should be adequate to diminish this limitation.

[21] The MATLAB code for BDS test used in this section was written by Ludwig Kanzler, and retrieved from: http://ww61.tiki.ne.jp/~kanzler/, December, 2009.

[22] Originally published in: Brock, W. A ., W. D. Dechert, and J. A. Scheinkman (1987) *A test for independence based on the correlation dimension*, Department of Economics, University of Wisconsin at Madison, University of Houston, and University of Chicago.

distance less than the pre-specified number (Bowden, 2003). In other words, it measures the frequency of a sequential pattern appearing within the data (Brock, et al., 1996):

$$C(m,\varepsilon,n)=\lim_{n\to\infty}\xi\sum_{k,j}\left(\varepsilon-\|X_k-X_j\|\right), \tag{4.11}$$

where ξ is the heaviside function and takes the value of 1 or 0:

$\zeta(z)=1$ if $z>0$ and $\zeta(z)=0$ if $z\leq 0$.

In other words ξ will contribute to the sum in Equation (4.11) only if

$\| X_k - X_j \| < \varepsilon$.

The BDS test is given by (Brocket, et al., 1996) specifically:

where $$W(m,\varepsilon,\mathrm{n})=\frac{B(m,\varepsilon,n)}{\sigma_m} \tag{4.12}$$

$$(m,\varepsilon,n)=\sqrt{n}.\left[C(m,\varepsilon,n)-C(1,\varepsilon,n)^m\right]. \tag{4.13}$$

Although the BDS statistic originally was a measure of independence, it is widely used in the economic literature as a test of non-linearity. After filtering the linear structure from the data, any remaining dependence is attributed to non-linearity.

The original sub-series of crude oil prices and returns were filtered by fitting a linear model. This linear model and the number of lags were selected by the software. There are several different methods for selecting the distance ε and the embedding dimension *m* for the BDS test, (e.g., Wolff (1995; 1994) for selecting the value of ε). The most widely used approach in the econometrics literature, however, is to set the values of the correlation integral's distance ε to be varied at 0.5, 1, 1.5, and 2 of the data standard deviations, and *m* to be from 2 to 8. This is because when selecting ε, one should avoid extreme values (small or large), as the correlation integral (for a given embedding dimension *m*) will capture very few points when ε is too small and too many points when ε is large (Steurer, 1995). One advantage of using this approach is that results are more comparable to what was previously published.

It is evident from the results of the BDS test (Table 4-3) that the data strongly reject the null hypothesis of *iid* for all crude oil prices and returns sub-series as the BDS statistic W was significant for all embedding dimensions tested. Also it appears that the evidence of non-linearity was stronger for larger embedding dimensions, as the BDS statistic W increased with larger embedding dimensions, for price and returns alike. Moreover, to reduce the impact of oil shocks (1991 and 2008) on the results, the BDS test was applied to a sample of the return (III) from 1992-

2007 which was a relatively calm period; the conclusion of non-linearity still holds, as can be seen from Table 4-5.

Crude oil price I								
	ε=0.5		ε=1		ε =1.5		ε =2	
m	W	SIG	W	SIG	W	SIG	W	SIG
2	12.5104	0	14.09	0	14.75712	0	12.98588	0
5	23.64956	0	23.53	0	22.82329	0	19.7774	0
8	40.71834	0	33.85	0	28.61	0	22.58775	0
Crude oil price II								
	ε =0.5		ε =1		ε =1.5		ε =2	
m	W	SIG	W	SIG	W	SIG	W	SIG
2	16.90563	0	16.83	0	17.74887	0	24.22801	0
5	43.82379	0	32.85	0	27.37772	0	34.09951	0
8	115.9072	0	52.94	0	34.27498	0	38.15595	0

Table 4-3 : The BDS test for crude oil price I and II

The upper section of this table shows the results of the BDS test for crude oil prices in Period I. The lower section shows the same test for crude oil price in Period II. *m* is the embedding dimension, *ε* is the distance, and *W* is the BDS statistic, *SIG* is the significance level at which we reject the null hypothesis of *iid*. The test is repeated four times, each time with a different value of *ε* (0.5, 1, 1.5 and 2); the embedding dimensions *m* are shown in the first column.

Return I								
	ε =0.5		ε =1		ε =1.5		ε =2	
m	W	SIG	W	SIG	W	SIG	W	SIG
2	63.28235	0	32.41485	0	28.294	0	24.21126	0
5	277.9648	0	63.43841	0	41.301	0	34.10912	0
8	1569.915	0	113.0694	0	51.747	0	38.16579	0
Return II								
	ε =0.5		ε =1		ε =1.5		ε =2	
m	W	SIG	W	SIG	W	SIG	W	SIG
2	69.901	0	30.065	0	21.751	0	19.484	0
5	370.17	0	80.356	0	37.184	0	30.265	0
8	2574.9	0	194.74	0	57.465	0	36.404	0

Table 4-4: The BDS test for crude oil log return I and II

The upper section of this table shows the results of the BDS test for crude oil returns in Period I. The lower section shows the same test for crude oil returns in Period II. *m* is the embedding dimension, *ε* is the distance, and *W* is the BDS statistic, SIG is the significance level at which we reject the null hypothesis of *iid*. The test is repeated four times, each time with different value of *ε* (0.5, 1, 1.5 and 2); the embedding dimensions *m* are shown in the first column.

Return III		
m	W	SIG
2	16.08305	0
5	30.51802	0
8	47.82609	0

Table 4-5: The BDS test for crude oil return III (excluding oil shocks)

This table shows the results of the BDS for crude oil returns in Period III which excludes the oil shock. *m* is the embedding dimension, *ε* is the distance (in this table *ε* equals the series standard deviation), *W* is the BDS statistic and SIG is the significance level at which we reject the null hypothesis of *iid.*

4.2.3.2 FCS Test

The BDS test is considered to be one of the best tests for *iid*. Notwithstanding, it still has two drawbacks when used as a test for non-linearity: firstly, in order to argue that the residuals of a linear model are purely non-linear, it is assumed that the linear model fitted to the sample is the optimal one, otherwise the dependence captured by the BDS test could be due to linear or non-linear dynamics. Secondly, providing that the linear filter is optimal, the BDS test does not clarify what kind of non-linearity is inherent in the data, i.e., non-linear stochastic, chaos, white noise or any other form. Hence, relying on the BDS test alone could result in a misleading conclusion.

In an attempt to overcome the latter drawback of the BDS test, Kaboudan (1999) introduced another test of non-linearity, the fuzzy classification system (FCS), also based on the correlation integral. This test is based on calculating the value of θ introduced earlier in Kaboudan (1995), defined as:

$$\theta = (M-1)^{-1} \sum_{m=2}^{M} \frac{\ln C_y(m, \varepsilon_1, n) - \ln C_y(m, \varepsilon_2, n)}{\ln C_s(m, \varepsilon_1, n) - \ln C_s(m, \varepsilon_2, n)} \tag{4.14}$$

where y is the original series filtered by a linear model, s is a shuffled version of the series (a bootstrap method was used for the shuffle), m is the embedding dimension at each shuffle, ε represents the distance in the correlation integral equation, where, ε_1 was set to be equal to the standard deviation of the original series, and ε_2 equal to 0.75 of the data standard deviation. And since a bootstrap method was used for the shuffle, the author recommended the data to be shuffled at least 1000 times. In this test the value θ represents a measure of the non-linearity in a time series after removing the linear structure from the series.

According to Kaboudan (1999) the value of R^2, from the fitted linear model, is a good indication of the extent of linear structure in the data, while the value of θ indicates the strength of the non-linear structure in the data. Hence, ten "implication rules[23]" were constructed, four to describe the degree of linearity based on the value of R^2, and six to describe the degree of non-linearity based on the value of θ. Consequently, 42 membership functions were derived from these rules[24]. Finally, the linear and non-linear classifications were combined to obtain the final decision about the series degree of linearity/non-linearity, i.e., the class (Kaboudan, 1999).

[23] The FCS test's rules and classes are (Kaboudan, 1999, p.4):

Rule no	If	Then
1	R^2 is high	The process is SL (strongly linear)
2	R^2 is average	The process is FL (fairly linear)
3	R^2 is low	The process is WL (weakly linear)
4	R^2 is very low	The process is NotL (not linear)
5	θ is very low	The process is NL (non-linear)
6	θ is low	The process is CHT (chaotic)
7	θ is average	The process is NL-LN (non-linear-low noise)
8	θ is above average	The process is NL-MN (non-linear-mid noise)
9	θ is high	The process is NL-HN (non-linear-high noise)
10	θ is very high	The process is WN (white noise)

[24] For a full description of these membership functions see: Kaboudan, (1999) p. 4-5.

We applied the FCS test to linearly filtered data. The FCS requires the value m and the number of shuffles to be set by the user. Kaboudan (1999) recommended that the embedding dimension be set from $m = 2$ to 10, with at least 1000 shuffles for best results.

According to the FCS test (Table 4-6:) the crude oil price series has linear components and a non-linear component with a high level of noise. These results come in line with the fact that the prices series usually contain linear trends, which explains the large value of R^2. On the other hand, the results of the FCS on the returns show that the return series are non-linear with high noise (stochastic) for Return (all) and Return I, whereas Return II is classified by the test as white noise. If the Return II series is indeed white noise then it cannot be forecast.

However, the value of θ presented in Table 4-6: needs to be interpreted with care. This is because θ in Equation (4) is an average of the correlation integral for embedding dimensions from 2 to m at each shuffle from 1-1000, and then the final 1000 values of θ (for each shuffle) are averaged again. Equally important is that there is not much difference between non-linear results with high noise and non-linear results with white noise. Table 4-6: also shows the results of the FCS test for both smoothed Return II (with three days simple moving average), and filtered Return II (using a wavelet filter). The classification changed to non-linear deterministic with a strongly linear component in the case of smoothed Return II, and with a weakly linear component in the case of a wavelet filter. This highlights the significant effect of the noise on the series when trying to formulate a forecasting model.

In order to make a comprehensive analysis of crude oil returns, the returns data were divided into twelve sub-series using a sliding window approach[25]. The aim was to uncover the development of the crude oil returns series over time, and also to determine if the behaviour of this commodity has changed in the past years.

[25] We started with the first ten years of the sample, i.e., Return I, and each time we slid one year head until we covered the rest of the series.

Data set	Fitted ARIMA	R^2	θ	Decision
Oil price (all)	Simple	0.99	0.96	SL-NL-HN
Oil price I	ARIMA(3,1,3)	0.90	0.94	SL-NL-HN
Oil price II	Simple	0.90	0.87	SL-NL-HN
Oil return (all)	ARIMA(0,0,5)	0.005	0.96	NL-NL-HN
Oil return I	ARIMA(3,0,3)	0.01	0.93	NL-NL-HN
Oil return II	ARIMA(2,0,0)	0.02	0.99	NL-WN
3-MA return II*	ARIMA (2,0,6)	0.77	0.00	SL-NL
Wavelet return II**	ARIMA (3, 0, 3)	0.018	0.29	WL-NL

Table 4-6: The FCS test of crude oil prices and returns

This table shows the results of the FCS test for the crude oil prices and returns. The first column shows the name of the tested time series, the second column shows the linear filter applied to the series, followed by the value of R squared, the value of θ and the final column shows the system classification of the series.
SL: strongly linear; NL: non-linear; HN: high noise; WN: white noise.
* is smoothed return II with a simple three days moving average.
** is filtered return II using a wavelet filter.

Data set	Fitted ARIMA	R^2	θ	Decision
Return win 1	(3, 0, 3)	0.016	0.94	NL-HN
Return win 2	(0, 0, 3)	0.017	0.94	NL-HN
Return win 3	(0, 0, 3)	0.017	0.95	NL-HN
Return win 4	(0, 0, 3)	0.017	0.96	NL-HN
Return win 5	(0, 0, 3)	0.015	0.97	NL-HN
Return win 6	(0, 0, 6)	0.007	0.95	NL-HN
Return win 7	Simple	0.01	0.95	NL-HN
Return win 8	(0 , 0, 0)	-4.933E-17	0.99	WN
Return win 9	(0, 0, 6)	0.02	0.97	NL-HN
Return win 10	(0, 0, 6)	0.04	0.98	WN
Return win 11	Simple	.000	0.99	WN
Return win 12	(2, 0, 0)	0.002	0.99	WN

Table 4-7: The FCS test overtime

This table shows the results of the FCS test for each window of crude oil return. The first column shows the name of the tested time series; the second column shows the linear filter applied to the series, followed by the value of R squared, the value of θ and the final column shows the system classification of the series.
SL: strongly linear; NL: non-linear; WL: weakly linear; HN: high noise; MN: medium noise; WN: white noise.
Return win 1 covers the same period as Return I.
Return win 12 covers the same period as Return II.

As can be seen from Table 4-7:, on average the value of θ was generally large. Figure 4-2 presents a 3-D plot of the value of θ at each shuffle for each window. It is clear from Figure 4-2 that the value of θ increased over time, yet it is generally very high for all windows tested, which is consistent with the uncertainty in the oil market consequent to major changes and events in the market, e.g., OPEC's policy changes, the substantial demand increase of oil from the Asian countries (especially China and India) and the global economic crisis, amongst others. In summary, according to the FCS test: (i) the dynamics of crude oil returns are generally non-linear stochastic, which could be attributed to the high level of noise, (ii) these dynamics have not changed significantly in the last twenty years, and (iii) the level of noise in crude oil returns has increased noticeably in the past few years.

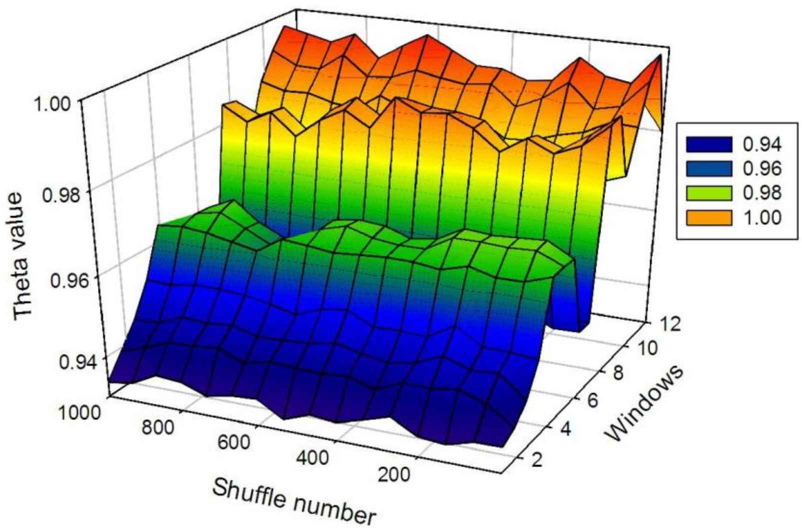

Figure 4-2: [coloured] A 3D plot of the value of θ at each shuffle for each time window*

This figure shows a smoothed 3D plot of the value of θ for each return's window shown in Table 4-7:. The *x* axis represents the time window, the *y* axis is the shuffle number and the *z* axis is the value of θ at each shuffle[26].

4.2.3.3 The time domain test for non-linearity

Thus far, both tests for non-linearity (the BDS and the FCS) in our analysis were based on the correlation integral. To acquire more confidence in our results at this point, another test proposed by Barnett and Wolff (2005) is applied, based on high order spectral analysis, namely the third order moment. Let $\mu(r,s)$ be the third moment of a time series $X_{t=1}^{\infty}$ (*r* and *s* are integers):

$$\mu(r,s) = E(X_t, X_{t+r}, X_{t+s}). \tag{4.15}$$

The Barnett and Wolff (2005) test starts with estimating the asymptotically unbiased third-order moment of the series as[27]:

$$\hat{\mu}(r,s) = \frac{1}{n} \sum_{t=1+|\tau|}^{n-\varphi} X_t\,, X_{t+r}, X_{t+s} \tag{4.16}$$

where $\varphi = \max(0, r, s)$ and *M* represents the truncation values, and takes any value between 1 and *n*. Then the estimated third-order moment in Equation (4.16) is compared to a group of limits obtained from linear stationary phase scrambled bootstrap data. In this way, large differences between the observed third moment and the calculated one are considered as a sign of non-linearity or non-stationary status in the series, as this indicates higher-order interactions in the original series

[26] For visual clarity the original data was smoothed by the Kernel function: $\frac{\mathbf{1}}{\sqrt{\boldsymbol{x}^2+\boldsymbol{y}^2}}$.

[27] According to Barnett and Wolff (2005) because the third order is symmetrical, Equation (4.16) needs only to be solved within the boundaries of $0 \le r < s \le M$, excluding the value at $\mu(0,0)$; as such, the authors defined $\Delta^{\#} = \{(r,s); 0 \le r \le s \le M, r+s > 0\}$, and $T^{\#} = \frac{(M+1)(M+2)}{2} - 1$ is the test cardinality.

(Barnett & Wolff, 2005). Moreover, to avoid reaching incorrect variance from the surrogate series the authors used a second bootstrap adjustment[28].

We set the test parameters in this analysis as recommended by Barnett and Wolff (2005): the embedding dimension we use is 5, the number of bootstrap replications is 1000, and the test significance level is 5%.

Series	Outside	Standardised	Upper limit	P value	H_0
Price (all)	3658.2	2.4477	77.823	0.02	1
Return (all)	1.09E-05	14.283	1.97E-06	0	1
Return I	1.97E-05	12.321	3.02E-06	0	1
Return II	9.32E-06	24.236	1.11E-06	0	1
Return I 3MA	1.73E-06	6.117409	5.31E-07	0.005	1
Return II 3MA	3.38E-06	27.56804	2.24E-07	0	1

Table 4-8: Results of Barnett and Wolff test[29]

This table shows the results of Barnett and Wolff's test for crude oil prices and returns. The first column shows the name of the time series, the last column shows H_0, which is a Boolean variable. H_0 indicates whether we accept or reject the null hypothesis.

The results in Table 4-8 confirm that the generating forces of the crude oil spot prices and returns time series are non-linear.

4.2.3.4 Lyapunov exponent test

The Lyapunov exponents (LE) is a quantitative measure of the sensitivity of a time series to the changes in the initial condition. There are several techniques to estimate LE based on the Jacobian approach: neural networks and non-parametric regression, amongst others. In this chapter we use a Volterra expansion model[30] (neural network models were also tested and similar results were obtained) to approximate the Jacobian matrix to estimate the LE; the embedding domination was set to six. The Jacobian approach for estimating the largest Lyapunov exponents is as follows (Lai & Chen, 1998):

Let $x_{t+1} = f(X_t) + e_t$ be a dynamical system where $X_t : \{x_t, x_{t-1}, \dots, x_{t-d+1}\}, d \geq 1$ are the data, e is a random error and f is a non-linear function. Also, let $J_t = Df(X_t)$ be the Jacobian matrix of f, and $T_m = J_m J_{m-1}, \dots, J_1 = Df^m, m = 1, 2, \dots,$

Lyapunov exponents λ are estimated as:

$$\lambda_i(X) = \lim_{m \to \infty} \frac{1}{m} \ln|a_i(m, X)|, i = 1,2, \dots, d \qquad (4.17)$$

[28] For more details about this test please see: (Barnett & Wolff, 2005).

[29]The MATLAB code of this test was written by A. Barnett, and retrieved from http://www.mathworks.com/matlabcentral/fileexchange/authors/29105, 15 August 2010.

[30]For the MATLAB code of LE we used writing by Mohammadi, S. (2009) which was retrieved from: http://fmwww.bc.edu/repec/bocode/l/lyapexpan.m , 30 September 2010.

where $a_i(m, X)$ represents the i^{th} largest eigenvalue of the Jacobian matrix J_t.

Table 4-9: shows the estimation of the largest Lyapunov exponents for each of the embedding dimensions included in this analysis. As can be seen, the largest λ was positive for all embedding dimensions we tested.

The direct interpretation of the Lyapunov exponent results appearing in the second column of Table 4-9: is that the crude oil spot return series is a non-linear deterministic of low dimensions dynamics, i.e., chaos is governing crude oil returns. However, the lower limit of the 99% confidence level is not significantly different from zero. Therefore, to avoid misleading conclusions, the effect of the noise in the data on the LE results cannot be ignored, and can explain the positive values of λ, as Abhyanker, Copeland and Wong (1997) pointed out. Table 4-9: also shows (lower section) the results of the LE test for the smoothed oil return series with three days moving average. It is evident that smoothing the series generates higher λ in general, nevertheless, the lower limit of the 99% confidence level is still not significantly different from zero[31]. Hence, we cautiously conclude that, providing noise control measures are applied, the dynamics of the crude oil returns series are non-linear deterministic, possibly chaotic. This conclusion contradicts the findings of Moshiri and Foroutan (2006) in which they found no evidence of chaos in crude oil futures price[32].

In summary, the BDS statistics indicate the existence of non-linear behaviour in all crude oil prices and returns sub-series which we tested. This was confirmed by the time domain test. On the other hand, the FCS test suggests that the dynamics of the crude oil series are non-linear stochastic. However, when the data are smoothed with a three day moving average, the classification developed a strong linear component and a non-linear deterministic component. Furthermore, when a wavelet filter was applied on Return II, the FCS classification became non-linear deterministic. Finally, the Lyapunov exponents for crude oil returns (and smoothed returns) highlight the possibility of low dimensional deterministic dynamics, i.e., chaos.

[31] The same conclusion can be made for the 95% confidence level for both the smoothed and unsmoothed return series. Only at a 90% confidence level was the lower limit significantly different from zero.

[32] It is important to note that Moshiri and Foroutan (2006) were testing LE using the raw price of crude oil futures contracts and not the spot return. Also, the LE in their study was also positive, but close to zero.

Return	Lyapunov exponent	99% Confidence level 1000 times bootstrap		Test's bias (Jackknife)
m	λ	Highest	Lowest	
d 1	0.0924	1.45E-18	0	572.391
d 2	0.1951	0.063865	-2.20E-20	1086.21
d 3	3.48E-19	0.078794	-1.10E-19	-38.8648
d 4	0.0204	0.176158	0.021725	100.06
d 5	0.0318	1.875162	-1.30E-18	-336.658
d 6	9.74E-20	0.648368	-0.0502	-737.621
3MA return	**Lyapunov exponent**	**99% Confidence level 1000 time bootstrap**		**Test's bias (Jackknife)**
m	λ_1	Highest	Lowest	
d 1	0.07	1.32E-18	0	-4.34E+02
d 2	0.3405	0.062429	-3.10E-20	-2.11E+03
d 3	0.1101	0.046871	-8.40E-20	-4.41E+02
d 4	0.0766	0.127102	-1.40E-19	1.80E+02
d 5	0.1353	2.08148	-9.30E-19	-5.20E+02
d 6	2.82E-19	0.594717	-0.03211	2.14E+03

Table 4-9: Lyapunov exponent for crude oil returns (all)

The confidence level was established using the bootstrap approach; after estimating the LE based on 1000 times bootstrap the test's bias was established based on the jackknife approach

4.3 Forecasting crude oil pricing using econometrics models

To put the results thus far into perspective, in this section we compare the forecasting performance of two models, $ARIMA_{(p, d, q)}$, $GARCH/EGARCH_{(p, q)}$.

4.3.1 ARIMA models

Based on the non-linearity tests in the previous section, fitting a linear model such as ARIMA is unlikely to be adequate for a crude oil series. Table 4-10: shows the ARIMA parameters fitted for the crude oil series. Table 4-11 presents the performances metrics of the ARIMA (or the exponential smoothing) for each series. It is evident that linear models did not provide a good fit. In the case of the price the R^2 was significant, but on a closer look the model merely fitted a random walk. As for the returns the R^2 was insignificant. Therefore, we did not proceed to forecasting out-of-sample.

Model				Estimate	SE	t	Sig.
Oil spot price	Simple			0.944	0.013	74.384	0
Oil spot price I	$ARIMA_{(3,1,3)}$	AR	Lag 1	-0.541	0.111	-4.867	0
			Lag 3	0.272	0.107	2.548	0.011
		MA	Lag 1	-0.551	0.105	-5.246	0
			Lag 3	0.358	0.1	3.571	0
Oil spot price II	Simple			0.938	0.018	52.275	0
Oil return	$ARIMA_{(0,0,5)}$	MA	Lag 2	0.043	0.013	3.385	0.001
			Lag 3	0.034	0.013	2.7	0.007
			Lag 5	0.051	0.013	3.994	0
Oil return I	$ARIMA_{(3,0,3)}$	AR	Lag 1	-0.541	0.111	-4.867	0
			Lag 3	0.272	0.107	2.548	0.011
		MA	Lag 1	-0.551	0.105	-5.246	0
			Lag 3	0.358	0.1	3.571	0
Oil return II	$ARIMA_{(2,0,0)}$	AR	Lag 2	-0.045	0.018	-2.51	0.012

Table 4-10: ARIMA parameters for crude oil spot prices and returns
This table presents the parameters of the linear model fitted for each time series. The simple model is an exponential smoothing model. SE is the standard error of estimate.

Model	Stationary R^2	R^2	RMSE	MAPE	MAE
Oil spot price	0.003	0.998	1.041	1.78	0.591
Oil spot price I	0.011	0.978	0.541	1.638	0.322
Oil spot price II	0.003	0.997	1.368	1.924	0.86
Oil return	0.005	0.005	0.026	105.722	0.018
Oil return I	0.011	0.011	0.025	106.262	0.016
Oil return II	0.002	0.002	0.027	102.796	0.019

Table 4-11: The performance metrics for the linear models for each series
This table presents the performance measure of the linear model. The first column shows the name of the time series.

4.3.2 GARCH type model

For crude oil Return I (1986-1998) $GARCH_{(1,\ 1)}$ was found to be appropriate in explaining the volatility in this series[33]. On the other hand, for Return II (1998-2010) none of the GARCH type models tested in this chapter was able to explain the volatility in this period, which was characterised by significant movements in the price level (Figure 4-1). It is well known that GARCH models are generally not able to model efficiently under these conditions. Table 4-12: shows the parameters of the $EGARCH_{(2,\ 2)}$, as an example of several models tested for Return II. It can be seen that the ARCH test and Ljung-Box-Pierce Q-test could not reject the null hypotheses and that there is no serial correlation in the fitted model normalised residuals at the 5% significance level.

[33] For brevity the model parameters were not presented.

Parameter	Value	Standard Error	T Statistic
C	0.00211	0.000788	2.6784
AR(1)	-0.90477	0.23544	-3.8429
AR(2)	-0.02402	0.024104	-0.9963
MA(1)	0.86696	0.23457	3.6959
K	-0.19392	0.056157	-3.4531
GARCH(1)	-0.01014	0.005718	-1.7725
GARCH(2)	0.9842	0.005644	174.3848
ARCH(1)	0.079318	0.015262	5.197
ARCH(2)	0.091389	0.015319	5.9657
Leverage(1)	-0.05261	0.010952	-4.804
Leverage(2)	-0.0399	0.011048	-3.6119
DoF	6.1778	0.61763	10.0023
Ljung-Box-Pierce Q-test	P-value	Test Statistic	Critical value
5 lags	0.0001	35.2661	18.307
10 lags	0.0003	40.9236	24.9958
20 lags	0.0006	47.1045	31.4104
ARCH test	P-value	Test Statistic	Critical value
5 lags	0.0002	33.9247	18.307
10 lags	0.001	37.6072	24.9958
20 lags	0.0012	44.6485	31.4104

Table 4-12: EGARCH$_{(2, 2)}$ model parameters for Return II

This table presents the parameters of the EGARCH$_{(2, 2)}$ fitted to return II. The first column C is the constant in the mean equation while K is the constant of the variance equation. Both the ARCH test and the Ljung-Box-Pierce Q-test could not reject the null hypotheses as there is no serial correlation in the normalised residuals of the fitted model at 5% significance level.

4.3.1 ANN and Fundamental variables

In this experiment the explanatory power of fundamental[34] energy variables was tested using a fully connected ANN. The data represent monthly observations (340 data points) of WTI crude oil prices, US oil stocks, US oil supplies, US oil imports and US oil consumption from Jan 1986 to May 2014 inclusive. All series were retrieved from the Energy Information Administration website http://www.eia.doe.gov/. For this experiment only one lag of each variable was used. This is mainly to reserve as many data points as possible for the training and testing process.

As can be seen in Table 4-13, the out-of-sample forecast was significantly better for the network trained with fundamental variables compared to a network trained solely with the crude oil monthly return (for the same time period). Also, the DA statistic was significantly larger for the fundamental variables to reject the null hypothesis of independence. Moreover, the RMSE and R^2 did not improve when adding the exogenous energy variable.

[34] The term “fundamental” is used here to describe variables related to the energy market, and does not have any relation to fundamental variables as financial statements.

Metrics	Benchmark		Fundamental	
	in-sample	out-of-sample	in-sample	out-of-sample
Hit rate	55.17241	49.62963	59.60	58.51
RMSE	0.077838	0.085657	0.077201	0.085378
R^2	0.06193	0.004622	0.0777736	0.015063
R	0.248858	0.067985	0.2788791	0.1227332
IC	0.77737	0.859306	0.7710159	0.8565162
MSE	0.006059	0.007337	0.0059601	0.0072895
MAE	0.060163	0.064169	0.0568912	0.0616337
SSE	1.229927	0.99051	1.2099015	0.9840888
DA	1.0336075	0.249806	1.2116041	1.3998422
P value	0.000	0.401369	0.000	0.08078028

Table 4-13: In-sample and out-of-sample average forecast metrics for the fundamental network vs. a benchmark (price solely). The benchmark is ANN trained only with crude oil returns.

On the other hand, financial market data like the S&P GSCI index daily return was also considered as additional input to the crude oil return. The S&P GSCI index consists of several commodities; crude oil constitutes around 55% of the value of this index, and the other 45% consisted of other commodities (part of which are energy-related, such as heating oil). However, the S&P GSCI was not a good predictor of the crude oil price, as the results in Table 10-11in Appendix II show.

4.4 Discussion

In this section we reflect on the findings of this chapter to connect them to the big picture with crude oil price forecasting. An important question arises: do the results obtained in this chapter lead to a better forecasting model for crude oil returns, and how?

We believe that the answer to the first part of this question is affirmative. To elaborate, although non-linearity is always assumed in financial/commodity prices time series, linear models are widely used to forecast these series. However, the strong evidence of non-linearity in crude oil returns suggests that non-linear models are better approaches than linear ones. Of course this raises the vexed problem of model choice in the case of non-linearity. Additionally, the evidence of chaos explains the seemingly random behaviour of crude oil returns.

This evidence of chaos also has two implications for the forecasting expectations, as (i) chaotic systems, in contrast to random systems, are deterministic, hence, they are predictable in principle, and (ii) since chaotic systems are sensitive to the initial conditions, long-term forecasting is unlikely to be successful, because the error from each forecasting step will be amplified exponentially (Adrangi, et al., 2001). This is supported by the empirical experiments we presented above. On the other hand, should the chaos be of a very low-dimensional nature (with low noise), it can be possible to recover the dynamics using Takens' theorem: see Cheng and Tong (1992).

Furthermore, the existence of chaos in the crude oil series provides an insight into the selection process of the forecasting model. In this chapter a neural network approach was found to be superior to traditional econometrics models. Following this further, a radial basis function neural network (RBF), as an example, is a favourable model for chaotic time-series forecasting as it is able to recreate the phase space of a chaotic system better than other models due to its high memory capacity and sensitivity to dimensionality in the series (Flake, 1998; Tao & Hongfei, 2007).

Although, we estimated ARIMA and GARCH models, both fitted the data rather poorly. However, the best results achieved in this chapter were by using real world financial data. Crude oil fundamental data, namely WTI crude oil prices, US oil stocks, US oil supplies, US oil imports and US oil consumption, generated a hit rate of 58.5% for monthly out-of-sample forecast. The improvement from the fundamental variable could be attributed to the information richness of the input variable. In other words, the network will find it difficult to learn the underlying function in noisy and incomplete input. Nevertheless, we cannot ignore the fact that these results are based on monthly granularity, which is generally less noisy. However, the fact remains here that these results generated are based on low frequency data, which is generally less noisy than daily data. Also, the practical application of the monthly forecast is different from a more granular forecast. Furthermore, the results in Table 4-12 (Benchmark) also shows that the problem of forecasting crude oil is much more complex than just applying a generic soft-computing model as the Benchmark performed poorly with a hit rate of 48%. In the next chapter we test a number of techniques to find how we could improve the forecast of crude oil.

Another important issue is that we could not find any evidence that the dynamics of the crude oil spot price series had changed over time, though the noise seemed to be higher in the more recent years. As such, one could conclude that the major changes in the market, such as the shifts in OPEC policies, increasing Asian demand for oil and the GFC did not have an effect on the dynamics of the crude oil price series; rather, they increased the uncertainty which, in our opinion, led to increasing the level of noise in this series.

Finally, as in all empirical analyses, the findings of this chapter should be interpreted in the light of the length and scale resolution of our data series.

4.5 **Concluding remarks**

In this chapter we aimed to discover what type of dynamics govern crude oil prices/returns and whether these dynamics have changed over time. Our analyses include several tests for non-linearity and chaos, as well as fitting several forecasting models. The BDS test rejects the null hypotheses of *iid* for all crude oil price and return sub-series included in this study. On the other hand the Kaboudan FCS test highlights the presence of noise in the data. In addition to this, according to the FCS test the noise has increased over the last ten years. This is consistent with the major changes in the market, from OPEC policies, increasing Asian demand for oil, to the GFC. This noise is a serious issue in hindering our ability to forecast crude oil returns even for one day ahead, regardless of the model being used. The Lyapunov exponent shows some evidence of chaos, which could explain the random walk-like behaviour of the crude oil returns. Nevertheless the Lyapunov exponents showed some evidence to support the existence of chaos in crude oil prices and returns. Overall, our tests show that the dynamical forces driving crude oil prices and returns are non-linear ones, of possibly low dimensions. What is evident though, is that the noise in the series plays a hindering role in any forecasting or analysis; however, since crude oil dynamics are mostly non-linear, henceforth, we will concentrate our effort on modelling with ANN.

4.6 **Summary**

The recent changes in crude oil price behaviour between 2007 and 2009 revived the question about the underlying dynamics governing crude oil prices. Even more importantly, the outstanding question over whether we can forecast crude oil prices and returns or not needs to be re-addressed. The goal of this chapter was to present an analysis of crude oil spot daily price/returns. The aim was to find if structural changes in the crude oil market have had an effect on the ability to forecast daily returns. Also, we have argued that there is still a gap between computational methods and traditional statistical methods for time-series forecasting; hence, we tried to make an effort to give due consideration to the statistical properties of the time series in the building process of soft-computing models.

As such, our investigation started with testing for non-linearity in the structure of these series using the most trusted test for *iid*, the BDS test. The fuzzy classifier system for non-linearity (FCS) proposed by Kaboudan (1999) and a time-domain test for non-linearity introduced by Barnett and Wolff (2005) were also used. Finally, we estimated the Lyapunov exponents to investigate the existence of chaos in crude oil prices and returns. Our tests showed consistently over time that the dynamical forces driving crude oil prices and returns are non-linear ones, of possibly low dimensions. Moreover, the FCS test shows evidence of high level of noise, which means that smoothing or noise reduction is necessary for achieving any level of forecast accuracy. To forecast the short-term crude oil spot returns we compared the performance of $ARIMA_{(p, d, q)}$, $GARCH_{(p, q)}$ and $EGARCH_{(p, q)}$, while the ANN trained using monthly crude oil fundamental data generated a hit rate of 58%.

CHAPTER 5: Forecasting crude oil price with simple ANN

5.1 **Introduction**

The objective of this chapter is to demonstrate how problem representation, (smoothing, data transformation) can aid the learning process of ANN, resulting in better short-term forecasting. Soft-computing models, such as ANN, SVM and fuzzy logic, have gained huge momentum in the forecasting community as they have useful characteristics in domains where exact (analytical) solutions are not possible or are very hard to obtain. ANN were designed in an attempt to imitate the human brain's functionality: the fundamental idea of ANN is to learn the desirable behaviour from the data with no *a priori* assumptions (Refenes, 1995). In this chapter we present simple ANN models to forecast crude oil returns.

5.2 **Network calibration**

Network calibration is a very important aspect of neural learning, because an ill-designed neural network is most likely to over-fit or under-fit the data (Neuneier & Zimmermann, 1998). There are a number of factors that affect the neural network's ability to learn. The top three issues are:

1. network complexity
2. network topology
3. representation of input and output.

The rest of this chapter presents a brief background and empirical tests for each of three factors above, applied to crude oil data.

5.2.1 **Network complexity**

Network complexity, i.e., the number of hidden neurons, is also a very important issue when modelling with neural networks, as too many hidden neurons will result in over-fitting and too few could result in under-fitting. The goal is to use the least number of neurons which generate the best results for out-of-sample. In this thesis (with traditional ANN) we used a simple approach proposed by Tan (2001) which is to start with a very small number of neurons followed by training and testing the networks to a fixed number of iterations. The hidden neurons increased gradually until the appropriate[35] number of neurons was found. As such, we use fully a connected feedforward network with two hidden layers, 25 and 8 hidden neurons in each hidden layer respectively. The network complexity was chosen based on a number of experiments. The non-linear activation function in each layer is the hyperbolic tangent, with a linear function in the output layer. All networks were trained by the Levenberg Marquardt algorithm[36]. Moreover, since ANN is a non-parametric and general function approximate model, each experiment was repeated 10 times (and some experiments for 1000 times), to ensure consistency in the findings.

Data transformation (not only filtering) is also an important issue that could help in harnessing the capabilities of ANN. Transformation will aid the learning process by emphasising certain structures in the data during the learning process. For example, in addition to the logarithmic return, the turning point of the series was used as additional input. Alternatively, if the series is mean reverting, this structure could be also used (Neuneier & Zimmermann, 1998). The results presented in Table 5-5: show minor improvement when the mean reverting transformation was used, although the DA statistic was not significant enough to reject the null hypothesis.

5.2.2 **Network topology**

Network topology covers the connectivity of the network, i.e., feedforward vs. recurrent, and fully connected vs. partly connected network.

5.2.2.1 Recurrent vs. feedforward

According to Zimmermann, Grothmann, Schäfer, and Tietz (2005), an economic time series, as an open non-linear dynamical system, consists of two parts: an “autonomous” part, i.e., endogenous behaviour which is forecastable, and “external” forces or exogenous events which are difficult to forecast. Following this line of argument, the historical data represent the accumulation of both. Therefore, to forecast a time series based on historical observation of the series itself, it is essential to improve the prediction of the autonomous part (Zimmermann, et al., 2005).

First we start with a feedforward network to put the results of the BDS and FCS into perspective. We attempt to predict one-step ahead with each window (the same sliding windows used in the FCS

[35] We define the appropriate number of neurons as: the number of neurons that generate the best generalisation on the out-of-sample set, based on the performance metrics used in this study.
[36] Other specifications: the learning rate was set to 0.3, and early stopping was used to prevent over-fitting.

test) using a fully connected feedforward network with two hidden layers, 25 and 8 hidden neurons in each hidden layer respectively[37], and 12 lags of each window as follows:

$$input: \begin{bmatrix} y_t^{(i)} & y_{t-1}^{(i)} & \cdots & y_{t-N+12}^{(i)} \\ y_{t-1}^{(i)} & y_{t-2}^{(i)} & \cdots & y_{t-N+11}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ y_{t-11}^{(i)} & y_{t-12}^{(i)} & \cdots & y_{t-N+1}^{(i)} \end{bmatrix}$$

$$target\text{: } [y_{t+1}^{(i)} \quad y_t^{(i)} \quad \ldots \quad y_{t-N+13}^{(i)}]$$

where y is the crude oil return, N is the length of the time series, t is the time index of the series, and $i = 1\ to\ 12$ is an index to the return window.

Table 5-1: presents the forecast results of each window (averaged over ten trials). The results in Table 5-1: show that the forecast was very poor (even worse than tossing a coin). Moreover, there was no significant difference between windows. Although the RMSE was lower for out-of-sample in earlier windows than later ones, the forecast was poor in all other metrics. The hit rate for window 11 on average was 53%; however, when we look at the DA statistic we could not reject the null hypothesis that the sign of the forecast and the sign of the actual were independent.

	Hit rate	RMSE	R^2	DA	P value
Window 1	46.47	0.0196	0.0047	-1.4644	0.8012
Window 2	49.66	0.0338	0.0057	-0.2186	0.5752
Window 3	49.33	0.0269	0.0074	0.2012	0.4340
Window 4	51.51	0.0269	0.0045	0.3518	0.3906
Window 5	47.39	0.0273	0.0023	-0.9544	0.7653
Window 6	49.66	0.0307	0.0164	-0.1701	0.5644
Window 7	46.81	0.0236	0.0035	-1.4257	0.7992
Window 8	48.32	0.0208	0.0034	-0.5513	0.6861
Window 9	47.82	0.0192	0.0058	-0.7774	0.7515
Window 10	51.76	0.0211	0.0056	0.1909	0.4388
Window 11	53.28	0.0502	0.0072	1.1117	0.2502
Window 12	49.33	0.0254	0.0011	-0.5229	0.6608

Table 5-1: Out-of-sample average forecast, one-step ahead of each window

In principle, though, recurrent networks are powerful tools to predict dynamical systems, if and only if the full description of the exogenous forces is clear; this is because recurrent networks could detect the "inter-temporal relationship" (Zimmermann, et al., 2005, p.8).

[37] Throughout this section, the network complexity was determined experimentally based on the out-of-sample performance, taking into consideration the network consistency (similar results each time we train) over a large number of experiments. The number of lags chosen was based on the findings of Haidar (2008).

In an attempt to improve the forecast for the raw return, a recurrent network was employed. The network has two hidden layers with 25 and 8 hidden neurons respectively. Each hidden layer has a feedback loop; however, there is no global feedback loop[38]. Table 5-2: presents the results obtained from this network compared to those obtained from standard feedforward (with only one hidden layer and 12 lags. Also it can be compared to window 12 in Table 5-1: for a 2 hidden layer FF network). The recurrent network produced a higher hit rate than a standard feedforward network, and the DA statistic rejects the null hypothesis that the sign of the forecasts and the sign of the actual are independent.

	Recurrent		**Feedforward**	
	in-sample	**out-of-sample**	**in-sample**	**out-of-sample**
Hit rate	56.34	**55.02**	49.69	48.821
RMSE	0.023272	0.032627	0.025556	0.035563
R^2	0.262595	0.004407	0.028949	0.018154
R	0.51244	0.066388	0.170144	0.134737
IC	0.604679	0.882426	0.687541	0.752927
MSE	0.000542	0.001065	0.000653	0.001265
MAE	0.017472	0.022954	0.018633	0.025279
SSE	1.540801	0.265072	1.812428	0.375617
DA	6.813056	**1.65941**	-0.37257	-0.43037
***p* value**	4.78E-12	0.048517	0.645264	0.666538

Table 5-2: Recurrent vs. feedforward return II

5.3 Smoothing

It is evident from the FCS test and the ANN forecast tests so far that the noise in the data is hindering the effort to forecast over any horizon, even one-step ahead. Therefore, one important issue is how to deal with the noise without affecting the information content.

5.3.1 Wavelet filter

This section presents results using wavelet filters to remove undesired noise from the original series. In contrast to Fourier analysis (Equation 5.2), which breaks the original signal into sine waves, wavelet analysis (Equation 5.1) breaks the signals into shifted and scaled sub-signals from the original signal under investigation; therefore, the time information about the signal is preserved (Misity, Misity, Oppenheim, & Poggi, 1997). This is a very important feature of the wavelet filter that makes it much more useful, especially in financial analysis, as the time index is a very important component (Misity, et al., 1997):

[38] Using a standalone machine, the network took 15 hours and 15min to converge (925 epochs).

$$C(scale, position) = \int_{-\infty}^{\infty} f(t)\xi(scale, postion, t)dt \quad (5.1)$$

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}dt \quad (5.2)$$

where $f(t)$ is the signal, C is the wavelet coefficient (function of scale and portion), ξ is the wavelet function, and $F(\omega)$ the Fourier transformation.

The original signal is transferred into two wavelets, the mother wavelet which included the higher frequency part of the original signal, and a father wavelet for the low frequency component of the signal (Kaboudan, 2005). Several transformation methods are available: Haar wavelet and Daubechies wavelets, amongst others. The Haar wavelet, for example, transforms the series x_t by finding the midpoint of the average (s_1) and differences (d_1) of each successive pair. Here the signal coefficients represent the series of the differences, which will carry the information of the signal fluctuations, whereas, the averages contain the main signal behaviour (Kaboudan, 2005). This process is called the discrete wavelet transformation and can take different level values N which results in different numbers of frequency sub-series $d_1 \ldots, d_N$.

Once the signal is transformed, there are two approaches to incorporating wavelet transformation into a forecasting problem. The first approach is to decompose the original signal s into a group of sub-signals and forecast each one separately: a "divide and conquer approach" (Kaboudan, 2005, p. 3); in the second approach is to decompose the original signal then filter one of these sub-signals, based on some threshold, then reconstruct the original signal.

The thresholding method involves five steps (Kaboudan, 2005):

- Decompose (scaling) the original signal into several sub-signals.
- Determine which frequency of the signal is most contributing to the noise.
- Create a threshold for noise reduction.
- Remove the noise from the sub series.
- Recombine the series without the noise.

A level 3 Daubechies wavelets transformation approach was applied to the crude oil price series. Figure 5-1 shows the father and the mother series, whereas Figure 5-2 presents a comparison between the signal s1 generated from the crude oil price and each of the frequencies, *d1*, *d2*, and *d3*.

The threshold filter was applied to *d1*[39]. Figure 5-2 compares *d1*, *d2* and *d3*. Finally, Figure 5-3 shows the original crude oil series and the filtered version using this approach.

Then, the filtered series was converted into returns and used as input to the network (12 lags), while the target for the network was kept as raw return unfiltered, unchanged for both in-sample and out-of-sample**.** Although, this experiment was applied to the entire series, it could be applied to part of the series (for example, the part that contains the shock).

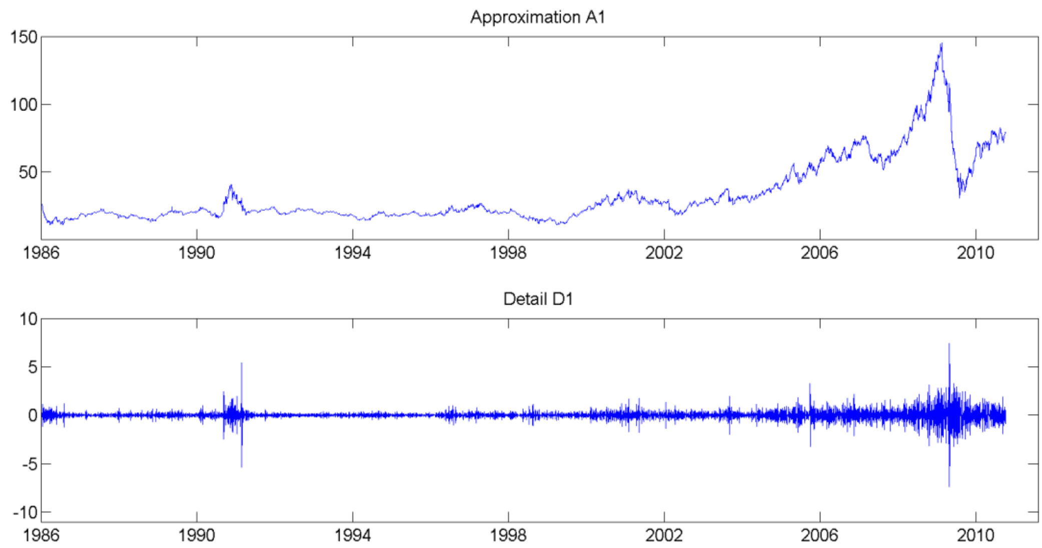

Figure 5-1: Wavelet transformation of crude oil price, father signal (upper) mother signal (bottom)

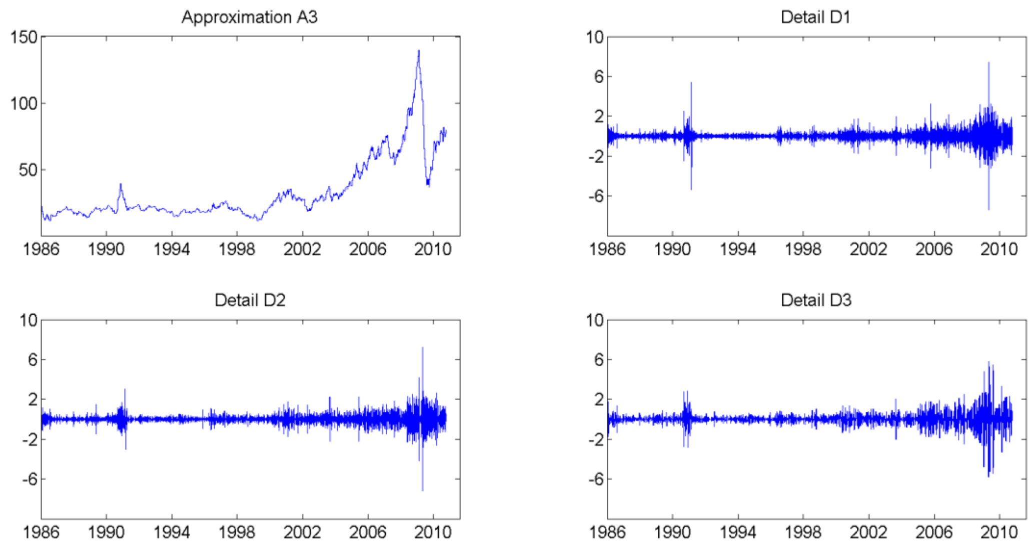

Figure 5-2: Wavelet transformation for the mother signal, s_1, d_1, d_2, and d_3

[39] Deciding which frequencies to filter was based selectively in this experiment; a better approach is to base the decision on some statistical analysis on all sub-series.

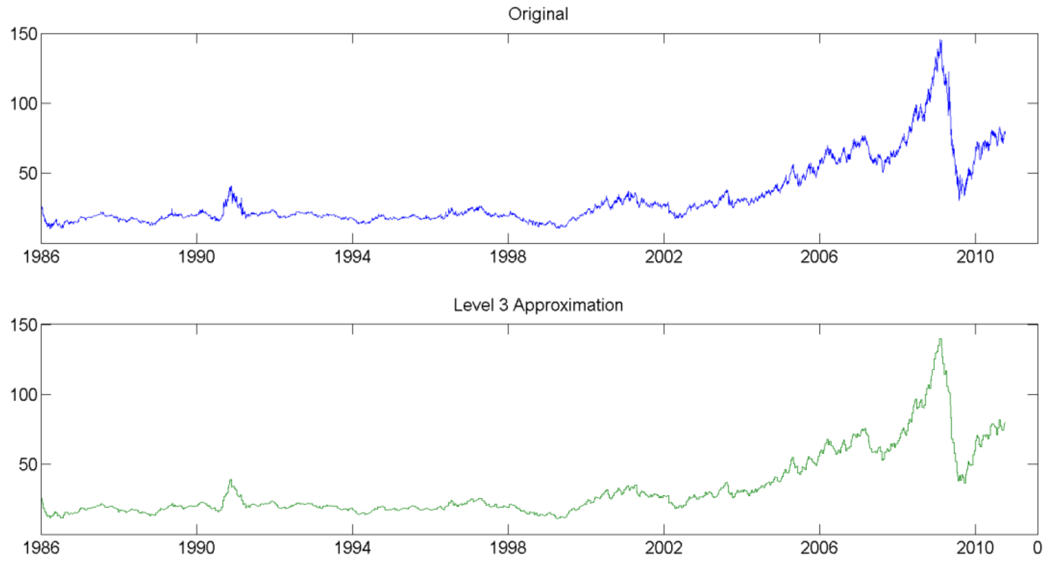

Figure 5-3: A plot of the original un-filtered crude oil price (upper), and the filtered price (bottom)

In Table 5-3: the descriptive statistics are compared between the crude oil raw spot prices/returns and the wavelet filtered ones. In addition we also compare them to a three days moving average filter on the returns. Both filtering approaches seem to change the variance, which could be attributed to the improvements in the forecast.

		Oil Price	Oil Return	Wavelet Price	Wavelet Return*	MA3 return
Range	Statistic	135.06	0.6	133.09	0.45	0.23
Minimum	Statistic	10.25	-0.41	10.76	-0.29	-0.14
Maximum	Statistic	145.31	0.19	143.85	0.15	0.1
Mean	Statistic	32.6715	0.0002	32.6718	0.0002	0.0002
	Std. Error	(0.29574)	(0.00033)	(0.29542)	(0.00014)	(0.00019)
σ	Statistic	23.27324	0.0262	23.24816	0.01114	0.01481
Variance	Statistic	541.644	0.001	540.477	0.000	0.000
Skewness	Statistic	1.947	-0.79	1.947	-3.085	-0.506
	Std. Error	(0.031)	(0.031)	(0.031)	(0.031)	(0.031)
Kurtosis	Statistic	3.789	14.808	3.777	97.32	6.45
	Std. Error	(0.062)	(0.062)	(0.062)	(0.063)	(0.062)

Table 5-3: Descriptive statistics for crude oil price, return, wavelet price and wavelet return, and three days moving average return. *The FCS test for Wavelet return (filtered by ARIMA (3,0,3)): R^2= 0.18, and θ = 0.29; so, the series is WL-NL (weekly Linear- Nonlinear).

Table 5-4: shows the results of the average forecast one-step ahead. The results show a significant improvement in terms of hit rate, while the RMSE was as good as, or better than, the benchmark. However, there was no significant improvement to the R^2. Also, the Pesaran and Timmerman DA test was significant, which increased confidence in the results.

	Wavelet input* (1)		Wavelet input & output** (2)		Benchmark (3)	
	In-sample	**Out-of-sample**	**In-sample**	**Out-of-sample**	**In-sample**	**Out-of-sample**
Hit rate	59.13	**61.99**	83.10	**74.97**	49.69	48.82
RMSE	0.0250	0.0337	0.0090	0.0196	0.0256	0.0356
R^2	0.1143	0.0293	0.3383	0.0837	0.0289	0.0182
R	0.3357	0.1680	0.5801	0.2749	0.1701	0.1347
IC	0.6729	0.7554	0.7517	0.9693	0.6875	0.7529
MSE	0.0006	0.0011	0.0001	0.0004	0.0007	0.0013
MAE	0.0178	0.0233	0.0045	0.0114	0.0186	0.0253
SSE	1.8125	0.3208	0.2278	0.1127	1.8124	0.3756
DA	10.1600	**4.0110**	34.3910	**8.4024**	-0.372	-0.4304
***p* value**	0.0000	0.0008	0.0000	0.0000	0.6453	0.6665

Table 5-4: One-step-ahead forecast using filtered return II and one-layer feedforward network with 8 neurons
*The 965% confidence limit for hit rate over 1000 trials was 58-62%, the mean hit rate was 61%. **The 95% confidence limit for hit rate over 1000 trials was 68-73%, the mean hit rate was 71%.

5.4 Pre-processing

Pre-processing the input/output is another method that could improve the network's performance. So far we explored pre-processing for filtering purposes only. However, pre-processing could also be used to stress certain behaviours in the time series (hints). Neuneier and Zimmermann (1998) argued that the turning point of the series given by Equation (5.3) adds additional information about the time series when used alongside the relative return (or the log return) as an input to the network. On the other hand, if the time series returns quickly to its mean after a shock then Equation (5.4) could be used (Neuneier & Zimmermann, 1998):

$$y_t = \frac{x_t - 2x_{t-n} + x_{t-2n}}{x_{t-n}} \tag{5.3}$$

$$z_t = \frac{x_{t-n} - \frac{1}{2n+1}\sum_{\tau}^{2n} x_{t-\tau}}{x_{t-n}} \tag{5.4}$$

where n is the forecast horizon with $n = 1$ for the current experiment, and τ is the length of the window.

For this experiment feedforward networks with two hidden layers and (25, 8) neurons respectively were used. Twelve lags from each variable (log-return, mean reverting, log-return and turning point, log-return and mean reverting, and all together) were used as input while the raw return was used as an output. Early stopping was used to prevent over-fitting.

The results are presented in Table 5-5:. As can be seen from this table only marginal improvement was recorded for the out-of-sample hit rate when Equation (5.3) solely was used as network input, and also when all variables were used. However, on average the other performance metrics did not improve the network's outcome compared to the benchmark (log-return only). The DA statistic for a network trained with mean reverting input (Equation 5.4) was significantly better than the benchmark, yet not enough to reject the null hypothesis of independence. An important point needs to be stressed: the out-of-sample test for this experiment not only is not filtered but also consists mainly of the two years where the major shock was recorded. Therefore, a more robust test would be to repeat the same experiments with sliding windows in order to exclude the shock as a factor. Also the same test could be repeated on a smoothed dataset, to exclude the effect of noise.

Out-of-sample	Benchmark	Return &Turning point	Mean reverting	Return &Mean reverting	All series
Hit rate	48.82155	50.89767	**52.4237**	51.0772	52.06463
RMSE	0.035563	0.043335	0.03699	0.055316	0.053263
R^2	0.018154	0.002153	0.001823	0.004015	0.012084
R	0.134737	0.008004	0.008378	-0.02876	0.106294
IC	0.752927	0.870221	0.742804	1.110796	1.069579
MSE	0.001265	0.001899	0.001369	0.003399	0.003098
MAE	0.025279	0.028995	0.025914	0.033689	0.035939
SSE	0.375617	1.057783	0.762477	1.893172	1.725341
DA	-0.43037	0.426225	**1.21906**	0.537452	0.981236
***p* value**	0.666538	0.336294	0.14908	0.316984	0.196946

Table 5-5: Out-of-sample results of different inputs

Flake (1998) suggested that when the squared value of a given input is added to the network as additional input this changes the structure of the feedforward to combine the global capability of feedforward with the local capability of the radial basis function (RBF). Following this further, the feedforward reaches a solution by changing the value of its parameters in a way that they respond to the change in the input-output as a whole, i.e., the global response. In contrast, the radial basis network is a local model as it responds to a change within small areas in the input space (Flake, 1998).

In mathematical terms (Flake, 1998, p. 147, 158; Grothmann, 2002, p. 54):

$$y_{ff} = \sum_j \tanh(\sum_j w_{j,i}\ x_i - b_j) \tag{5.5}$$

$$y_{rbf} = \sum_j v_j e^{-\frac{1}{2}\sum_i\left(\frac{x_i-\mu_{j,i}}{\sigma_{j,i}}\right)^2} \tag{5.6}$$

$$y = \sum_j v_j \tanh(\sum_i w_{j,i}x_i + s_{j,i}x_i^2 - b) \tag{5.7}$$

where y_{ff}, y_{rbf} and y are the output from the feedforward neural network, RBF network, and Flake's network respectively; x_i is the network input, s_{ij} are the squared input, v_j, $w_{j,i}$ are the network weights, b_j is the network bias.

The output of a feedforward with hyperbolic tangent can be represented in Equation (5.5). On the other hand the output of the RBF network with Gaussian function is represented in Equation (5.6). While Equation (5.7), combines the output of Equations (5.5) and (5.6), representing both an **approximation** of the RBF through x_i^2 while reserving the original input where x_i represents the output of the feedforward network (Flake, 1998). By so doing, according to Flake (1998) the network is effectively able to combine the global mapping capability of feedforward with the local capability of RBF, which results in improving the results of the network (Flake, 1998; Grothmann, 2002).

We applied this method to Return II by adding 12 lags of squared return to the original input (12 lags of the Return II) using the feedforward network[40]. It is evident from the results in Table 5-6 that a significant improvement for the hit rate was recorded when adding the squared return, compared to the feedforward with a similar structure trained with Return II only and also compared to the RBF network trained with Return II solely. The RMSE was slightly better for in-sample with squared network while it was worse-off for out-of-sample compared to the feedforward network, while the RMSE for RBF was significantly worse for the out-of-sample set.

Metric	**Feedforward**		**RBF**		**Feedforward Squared**	
	in-sample	**out-of-sample**	**in-sample**	**out-of-sample**	**in-sample**	**out-of-sample**
Hit rate	51.88	49.33	75.19407	44.53782	59.28017	55.12605
RMSE	0.0273	0.0254	0.014754	0.108629	0.021133	0.032827
R^2	0.0535	0.0011	0.704411	0.003657	0.38985	0.004447
R	0.2224	0.0235	0.839292	-0.06047	0.620285	0.055196
IC	0.7075	0.7466	0.382785	3.19539	0.548289	0.965628
MSE	0.0007	0.0006	0.000218	0.0118	0.000449	0.001164
MAE	0.0195	0.0196	0.010165	0.050614	0.015507	0.02222
SSE	2.1102	0.1537	0.616901	2.808437	1.273785	0.27694
DA	2.2324	-0.5229	26.82848	-1.78366	9.91519	1.51874
P value	0.1059	0.6608	0	0.96276	3.06E-11	0.091106

Table 5-6: Average of in-sample and out-of-sample forecast for crude oil return II using the squared return as a hint

Another pre-processing approach proposed here is using a sliding window approach of five days; the mean variance, skewness and kurtosis of the crude oil price series was calculated. Figure 5-4 shows a plot test series compared to the log return. Several combinations of these series were tested as input, whereas the network target was the raw log return.

[40] It is important to note that this hint could be implemented in two different yet equivalent ways: the first and the easiest way is as we have done above; another way is to embed this process into the structure of the network (Flake, 1998). However, in my opinion there is no obvious reason for the benefit of the latter approach.

This way the network input and output can be represented as follows:

$$\begin{bmatrix} y_t^{(i)} & y_{t-1}^{(i)} & \cdots & y_{t-N+12}^{(i)} \\ y_{t-1}^{(i)} & y_{t-2}^{(i)} & \cdots & y_{t-N+11}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ y_{t-11}^{(i)} & y_{t-12}^{(i)} & \cdots & y_{t-N+1}^{(i)} \end{bmatrix}$$

$$[y_{t+6}, \quad y_{t+5} + \ , \ldots, y_{t-N+18}]$$

where $y_t^{(i)}$ represents the transformed version of the series, and the index i represents each of the transformation methods. Since a sliding window was used, the target of the network was shifted by the same size of the sliding window to avoid including information in the training that was already available in the target, which would affect the generalisability of the network.

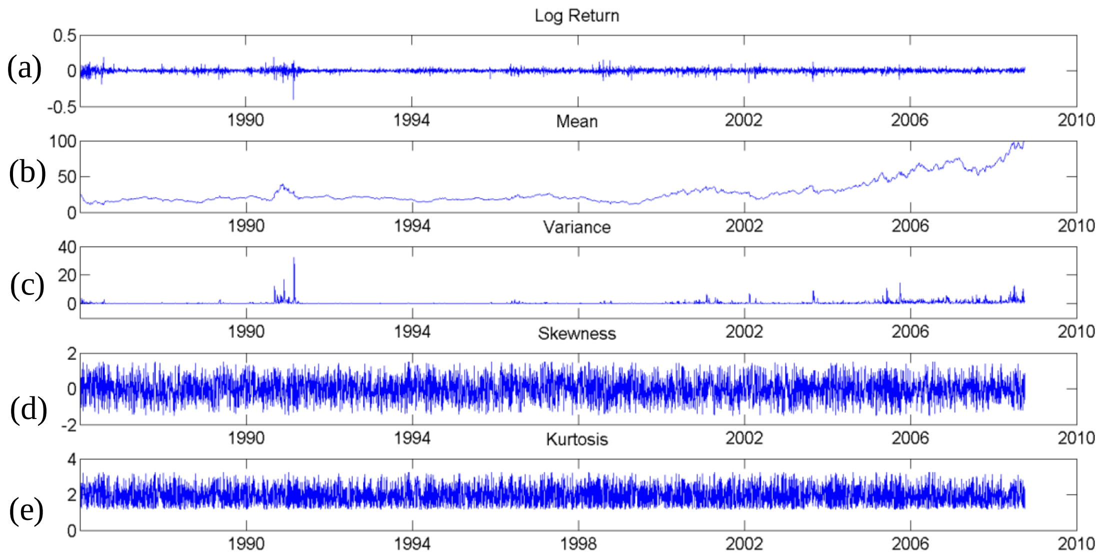

Figure 5-4: A plot of transformed price

(a) is crude oil long-return; (b) is spot price five days sliding window mean; (c) is spot price five days sliding window variance; (d) spot price five days sliding window skewness; (e) is spot price five days sliding window kurtosis.

The results are presented in Table 5-7:. While some of the methods seem to improve the hit rate, no improvement was recorded on the other metrics, especially the RMSE and R^2

Inputs	Hit rate	RMSE	R^2	R	IC	MSE	MAE	SSE	DA	*p* value
ALL	51.55	0.39	0.00	0.05	7.43	0.27	0.15	130.37	0.70	0.31
Return Mean Var Skewness	50.25	0.08	0.00	0.05	1.59	0.01	0.04	5.62	-0.03	0.47
Return Mean Var	50.14	0.14	0.00	0.01	2.76	0.03	0.07	13.36	0.07	0.47
Return Mean	50.76	0.04	0.00	0.04	0.72	0.00	0.03	0.67	0.37	0.36
Var Skewness Kurtosis	50.62	0.05	0.01	0.03	0.94	0.00	0.03	1.31	0.27	0.47
Var	51.72	0.08	0.00	0.05	1.63	0.01	0.05	3.91	0.80	0.24
Skewness	47.93	0.04	0.00	-0.03	0.78	0.00	0.03	0.81	-0.92	0.72
Kurtosis	52.96	0.04	0.00	0.01	0.71	0.00	0.03	0.67	1.39	0.10
Box-Cox return	51.95	0.05	0.01	0.08	3.09	0.00	0.03	1.15	0.87	0.21

Table 5-7: Average results for out-of-sample

The same experiment above was repeated with a sliding window of 25 days. The mean and the skewness seem to improve the generalisation of the network as the results in Table 5-8 suggests.

Inputs	Hit rate	RMSE	R^2	IC	MSE	MAE	SSE	DA	*p* value
ALL	56.67	0.0711	0.0104	1.3664	0.0051	0.0443	2.4952	2.9980	0.0075
Return Mean Kurtosis Var	53.51	0.0864	0.0064	1.6600	0.0089	0.0522	4.3425	1.6524	0.1214
Return Mean Kurtosis	58.28	0.0373	0.0732	0.7167	0.0014	0.0268	0.6809	3.6888	0.0002
Return Mean	55.48	0.0524	0.0418	1.0070	0.0030	0.0368	1.4631	2.4979	0.0184
Var Skewness Kurtosis	55.32	0.0854	0.0188	1.6419	0.0079	0.0485	3.8422	2.4356	0.0245
Var	49.86	0.0692	0.0032	1.3297	0.0050	0.0440	2.4566	-0.054	0.5187
Skewness	56.92	0.0363	0.0392	0.6984	0.0013	0.0258	0.6435	3.0951	0.0054
Kurtosis	50.60	0.0370	0.0014	0.7110	0.0014	0.0264	0.6668	0.4191	0.3635
Mean	55.03	0.0653	0.0097	1.2540	0.0050	0.0435	2.4191	2.2652	0.0170
Mean Skewness	58.03	0.0399	0.0483	0.7674	0.0017	0.0283	0.8045	3.5816	0.0018

Table 5-8: Average out-of-sample results

5.4.1 Normalized output

The goal of these experiments is to examine if multi-task learning is useful in improving the forecasting of the crude oil return. Multi-task learning theoretically has two advantages: first it accelerates the conversion of the network (Suddarth & Kergosien, 1990), and second, it improves the error flow of the network, therefore improving the generalization (Neuneier, & Zimmermann, 1998; Grothmann, 2002). The information flow issue can be explained as the error signal passing through the network in both directions being generated by a single output to supply several inputs, thus making the network unstable during training (Grothmann, 2002). Therefore, by adding more outputs, more error signals are generated, which reflects positively on the stability of the network (Grothmann, 2002).

The key issue here is that the task in each output needs to be related to each other output, otherwise we are risking negative results from a clash of interest between each output trying to pull the weight towards its own direction. Azoff, (1994) and Neuneier & Zimmermann (1998) recommended that each output represents a one-step ahead forecast $(y_{t+1}, y_{t+2}, y_{t+3})$ while Azoff (1994) argued against using multi-outputs for time-series forecasting applications to avoid the clash of interest issue.

In this experiment the raw return was normalised to reflect the change in direction only, which simplifies the learning process as the network only needs to learn the direction change and ignore the magnitude. Therefore, the normalised and raw return was used as input and targets (two output neurons). The results were measured for the actual return only while the normalised one was ignored. The results in Table 5-9: show that this process has improved the out-of-sample hit rate. The DA statistic was significant as well, but improvement was recorded to the goodness of fit measures.

	in-sample	out-of-sample
Hit rate	57.29	53.66
RMSE	0.02458	0.03964
R^2	0.047052	0.00171
IC	0.688212	0.76234
MSE	0.000605	0.00159
MAE	0.016717	0.02724
SSE	3.436007	0.77803
DA	14263072	1.630232
***p* value**	0	0.0715

Table 5-9: Average results for the actual target (target 1)

5.4.2 Cubic spline interpolation

As a possible solution for the lack of daily data, the cubic spline (piecewise-polynomial) interpolation (CS) was tested, to convert weekly and monthly data into daily data. The CS is sometimes used in economic applications, mainly to transfuse quarterly data into monthly data.

Before using this approach for prediction, we tried to apply it to a data set for which daily and weekly observations were available to us (we used weekly crude oil price retrieved from EAI from www.eia.gov), to make sure it will generate a useful series. Figure 5-5 shows that the CS method was successful in capturing the overall behaviour of the series from weekly observations; after controlling for weekends and public holiday in the CS expanded data, both series seems to fit closely with each other. This gives us confidence with the applicability of this approach.

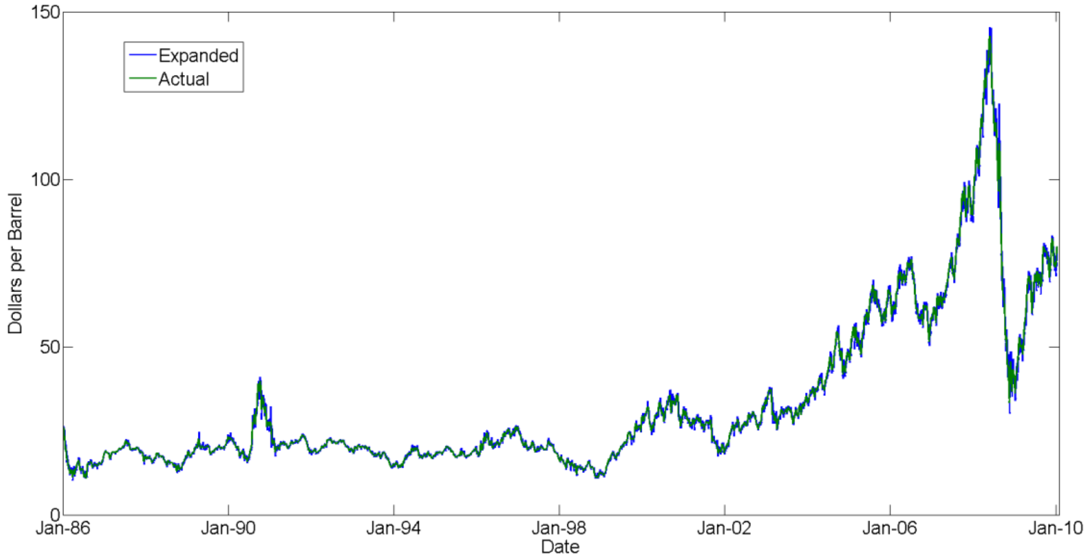

Figure 5-5: A plot of the actual crude oil daily price (green) and the transformed price using CS method (blue)

Daily return

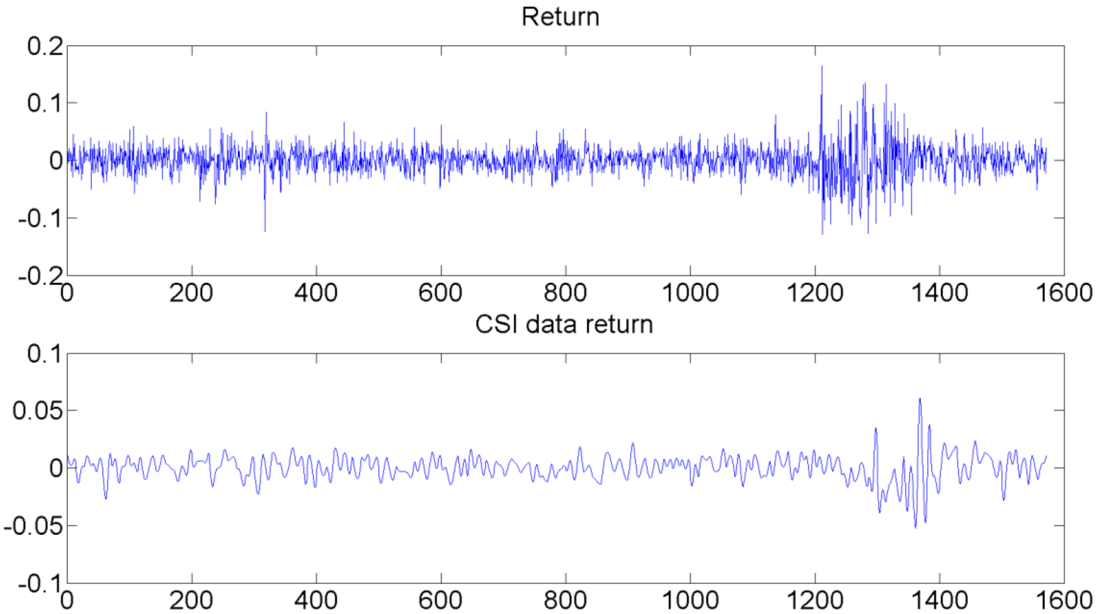

Figure 5-6: A plot of the output of the CS methods on converting weekly data (black) into daily compared to the actual daily observations (blue)

We tested this method using one of the terms in the Google data, "WTI oil price", the original weekly data (319 observations) were transformed into 1574 data points and used as input along with daily Return II. Figure 5-6 shows a plot of the data before and after transformation, while Table 5-10: compares the results obtained from comparing the transformed data with the original one (319 data points). As can be seen, the hit rate for in-sample has decreased with this new test; however, for out-of sample the hit rate was better. Moreover, the RMSE was significantly better for this new method. It is important to acknowledge though, that these tests are not totally comparable as we used feedforward for this test (with a static network) while NARX networks were used for the previous one.

Metric	Oil & WTI without CS		Oil & WTI as CS	
	in-sample	out-of-sample	in-sample	out-of-sample
Hit	60.48	52.53	55.86	**54.16**
RMSE	0.0344	0.0703	**0.0228**	**0.0211**
R^2	0.0716	0.0259	0.3023	0.0051
R	0.1436	-0.1083	0.5490	-0.0175
IC	0.7683	0.7644	0.5816	0.7428
MSE	0.0012	0.0049	0.0005	0.0004
MAE	0.0281	0.0515	0.0173	0.0165
SSE	0.2478	0.4895	0.7231	0.0717
DA	2.9912	0.3470	4.6049	0.8286
P value	0.0014	0.3697	0.0002	0.2208

Table 5-10: Testing the CS transformation on “WTI” term from Google

5.5 Multi-steps forecast

Forecasting the raw return was not successful so far for even one-step ahead, therefore, we attempted to forecast a smoothed return for multi-steps ahead. Our previous experiments on multi-steps forecasting were done on the smoothed input-output until the hit rates were no longer significant, i.e., below 50% (around five steps for filtered data with five days moving average). Nevertheless, our new test revealed an interesting pattern in the data. In this experiment a static multi-step forecast was conducted up to 25 steps in the future, using a feedforward network with two hidden layers. The results showed that the forecast hit rate on out-of-sample became insignificant, i.e., 50% or less, after four steps ahead. We believe that improvement in the results presented in Table 5-11 are not due to information leakage, or including information from the target into the training set (Kaufman & Rosset, 2012). Following this further, the data was smoothed by the five days simple moving average, hence, beyond step five, the input does not include information for the future. Remarkably, the hit rate was significant (around 60%) at steps 18-20.

Step	Hit rate	RMSE	R^2	R	IC	MSE	MAE	SSE	DA	P value
19	**59.44**	0.0103	0.0316	0.1756	1.6924	0.0001	0.0084	0.0229	**2.3282**	0.0335
20	**60.37**	0.0104	0.0249	0.1547	1.7131	0.0001	0.0084	0.0233	**2.6743**	0.0298
21	**55.33**	0.0108	0.0173	0.0777	1.7887	0.0001	0.0087	0.0249	**1.5442**	0.1214
22	**55.12**	0.0106	0.0089	0.0656	1.7663	0.0001	0.0087	0.0242	**1.0719**	0.2401
23	**54.81**	0.0105	0.0130	0.0967	1.7410	0.0001	0.0086	0.0235	0.7199	0.3453
24	53.71	0.0108	0.0135	0.0963	1.7786	0.0001	0.0088	0.0245	**1.2466**	0.1467
25	**56.48**	0.0105	0.0233	0.1511	1.7268	0.0001	0.0086	0.0230	**1.9027**	0.0446

Table 5-11: Average out-of-sample results for multi-step ahead forecast on smoothed input-output with five days moving average

Forecast horizon	Hit rate renege	Confidence limit	Mean hit rate for 1000 tests
19 days	52-60%	95%	56%
20 days	52-60%	95%	56%
21 days	52-59%	95%	56%
22 days	52-58%	95%	55%
23 days	51-57%	95%	54%
24 days	52-58%	95%	54%
25 days	52-58%	95%	55%

Table 5-12: Out-of-sample confidence limit in hit rate for multi-steps forecast

5.6 Discussion

In this chapter we tested a number of techniques to improve the accuracy of the crude oil return forecast. The focus of this chapter was based on traditional FF-ANN. There is a substantial body of computing literature that discusses how problem representation dramatically affects the training of an ANN. However, this issue is not given adequate investigation in most financial and economic literature using ANN.

Design issues like network complexity and topology were extensively tested while maintaining the same input-output. Network topology, namely the recurrent network seemed to improve the forecast performance to some extent; the hit rate for the recurrent network was 55% compared to 48% for feedforward. The pioneering work of Elman (1990) showed that the *memory* capability of the recurrent network enable them to find an internal relation in a time series. However, recurrent networks are very difficult to train when there is “long-term dependency” as the network has to keep track of many time steps form the data during training (Sutskever & Hinton, 2010, p.239) Moreover, the result of the FCS test in Chapter 4 highlighted the high noise in crude oil returns. Hamilton (2009) suggest that the high noise in crude oil returns could be attributed to high volatility, low price elasticity and the sensitivity of supply to interruption (Hamilton, 2009) . Therefore, we tested the potential impact of a filtering and signal processing technique, namely, a simple moving average and wavelet analysis. Initially, we obtained significantly accurate results using the simple moving average approach with different time windows. However, the improvement in results was misleading and attributed to the leakage[41]. Leakage took place as we unintentionally included information from the time series past into future (Kaufman & Rosset, 2012). This means that the supervised learning algorithm input contained information about the future state of the time series. Wavelet analysis, on the other hand, as a local signal transformation approach does not interfere with the time scale of the time series. After decomposing filtering the crude oil price using discrete wavelet analysis, the FCS test showed that the series classification has changed from nonlinear with high noise into weakly linear- Nonlinear (WL-NL). The 95% confidence limit out-of-sample hit rate was 74% as Table 5-4 column 2 shows. However, we believe the results presented in Table 5-4 column 2 might not have any practical application, since both the input and the target were filtered. We believe that the results in Table 5-4 column 1 are much more sensible as only the input were filtered; the 95% confidence limit out-of-sample hit rate was 61%.

Furthermore, for the multi-step forecast we used a simple five days moving average approach. We found that the network was able to predict the direction remarkably well for steps 19-25 ahead (the highest hit rate was 60 % at step 20). We believe that the positive results presented in Table 5-11 are not due to information leakage as discussed in Section 5.5. This is because the data was smoothed by the five days simple moving average; therefore, the input does not include information for the future beyond step five. A possible explanation of these positive results is long memory in the crude oil price. Choi and Hammoudeh (2009) found evidence of long memory (a slow decay in the autocorrelation function) in crude oil spot price and futures contact and other petroleum products. Choi and Hammoudeh (2009) showed that the forecasting accuracy for the crude oil WTI return was better for 20 days ahead than for five days ahead, using a econometric forecasting model. Another plausible explanation would be the existence of a structural break in the series. Granger and Hyung (2004) argued that structural break and regime-switching in financial and economic series can be mistakenly identified as long memory. The authors showed that the number of regimes and the frequency and timing of switching could affect the parameter estimate for the long memory model. A detailed survey on this topic is provided by Banerjee and Urga (2005). In the next chapter we estimate Hamilton's (1989; 1990) regime-switching model to find if crude oil price flow Markov switching process. And if so would this explain the results we obtained in Tables 5-11 and 5-12.

[41] The author is grateful to an examiner for pointing out that the improvement in forecast using the moving average was in fact misleading due to leakages.

5.7 **Conclusion**

This chapter has illustrated how the representation of the problem can be used as a simple yet effective way to improve the forecast performance of ANN. Problem representation refers to the way the input and output are introduced to the model (in this case ANN) and this covers a number of techniques: data transformation, smoothing and averaging methods, amongst others. The goal here was to test how these simple methods affect the forecast accuracy of complex time series. We believe that this issue is often ignored or given low attention in the computational finance literature. We also believe that financial time series require different types of pre-processing than other types of signals. This is because financial and commodities time series have special dynamics (see Chapter 4) and therefore need to be treated accordingly. In theory, problem representation is very important in ANN as it assists the model in finding a better function, e.g., noise reduction from the input makes it easier for the network to find the right relationship between the input and the target (Abou-Mostafa, 1995a). Our approach in this chapter was to build upon knowledge uncovered in Chapter 4, and use it to improve the forecasting accuracy and horizon of crude oil returns. For example, we found in Chapter 4 that crude oil returns are very noisy. To deal with this issue we applied a smoothing model. Moreover, in Section 5.7 we used Flek's (1998) approach to approximate the RBF, which is better equipped to deal with the low dimension dynamics of crude oil returns we discovered in Chapter 4. Our empirical results showed that some of these measures are effective in improving the forecast accuracy. Overall, we found that noise in crude oil returns plays a significant role in hindering the learning process of ANN. The best out-of-sample hit rate achieved in this chapter was 61% after decomposing filtering the return using wavelet analysis. Finally, the results of multi-steps forecast open the question about whether crude oil return has a long memory or follow regime-switching process. In the next chapter we fit Hamilton's (1989, 1990) regime-switching model to address this question.

5.8 **Summary**

In this chapter we demonstrated how problem representation can influence the forecast outcome of ANN. Moreover, in our ANN forecasting model, we have discovered a non-linear pattern in the smoothed crude oil return data. We show that for smoothed data, multi-step forecasting is possible (for 19-25 steps ahead) with reasonable accuracy. Finally, we expect that the analysis presented in this paper may well be useful for researchers and energy economists who have an interest in the prediction of crude oil prices and return. We conclude that it is possible to forecast the crude oil price using non-linear models, provided noise control measures are used.

CHAPTER 6: Case studies

6.1 **Introduction**

This chapter presents three case studies used in an attempt to improve crude oil price forecasting. The motivation of this chapter is to find: (i) Does non-financial data (Google search queries) contain information to improve the short-term forecasting of crude oil returns? (ii) Can we apply our knowledge about the crude oil market and soft-computing to generate variables to aid the training of ANN? and (iii) Would data transformation, inspired from technical analysis, suffice in improving the forecasting of crude oil prices? To test these questions, for the first case study we presented a model based on a non-financial data time series obtained from Google Insight for Search. In the second case study, we created an artificial time series from OPEC meetings' announcements to account for the outcomes of such meetings. Finally in the third case study we used crude oil fundamental data technical analysis and data transformations to improve the forecasts.

A significant amount of section 6.2 was adapted from:

- Haidar, I., & Wolff, R. C., (2012). Forecasting crude oil price using soft-computing methods and Google Insight for Search. In the Proceedings of the 35th Annual IAEE International Conference, Perth, WA, 24-27 June 2012.

6.2 **Google Insight for Search**

6.2.1 **Overview**

One of the problems that face short-term (daily and weekly) crude oil price forecasting is that most of the fundamental variables, such as supply, demand, inventory and GDP, are recorded on monthly or quarterly bases. This leaves us with a very limited number of potential explanatory variables apart from the lagging crude oil price itself. Thus, it would be extremely beneficial if we could identify additional sources of information to act as hints to aid the forecasting process of such an important commodity.

Google Trends and also Google Insight for Search are relatively new services from Google, introduced in 2006 and 2008 respectively (Anvik & Gjelstad, 2010). These services give users a query index representing the following for given geographic locations (H. Choi & Varian, 2009):

$$\frac{Total\ search\ for\ a\ phrase\ at\ time\ (t)}{Total\ search\ in\ Google\ at\ time\ (t)}$$

The query index represents a normalized version of the search index between 0 and 100, where 100 is given to the highest search and zero to the lowest search.

Although Google Trend and Google Insight for Search provide almost the same output, we are interested mainly in the latter as it allows users to download the data into a CSV file (comma separated values). This service can be accessed from: http://www.google.com/insights/search/#.

The question here is: Is this tool useful in aiding the short-term forecast of financial and economic time series? Figure 6-1 shows the search query for the term "cold weather" downloaded from Google Insight for Search. There is a distinctive, clear seasonal pattern in this series. This makes it tempting to use these search queries as explanatory variables. Nevertheless, in order for these variables to help any forecasting effort there must be some temporal presence, i.e., enough users who have done their searches before the actual event (in this case, cold weather) took place.

The first study to investigate this issue was conducted by (H. Choi & Varian, 2009), members of the Google research group; thereafter, a few scholars took up this idea. For example Anvik and Gjelstad (2010) and Askitas and Zimmermann (2009) attempted to forecast unemployment, while Wu and Brynjolfsson (2009) studied the housing market. However, to the best of our knowledge, there is no study to date which explores whether the search activities of Google users could indeed help to forecast complex time series like the crude oil price. Moreover, almost all the studies we reviewed in this area were based on linear models, without even testing the dynamical structure of these series. Therefore, our aim is to bridge this gap, concentrating only on crude oil price forecasts.

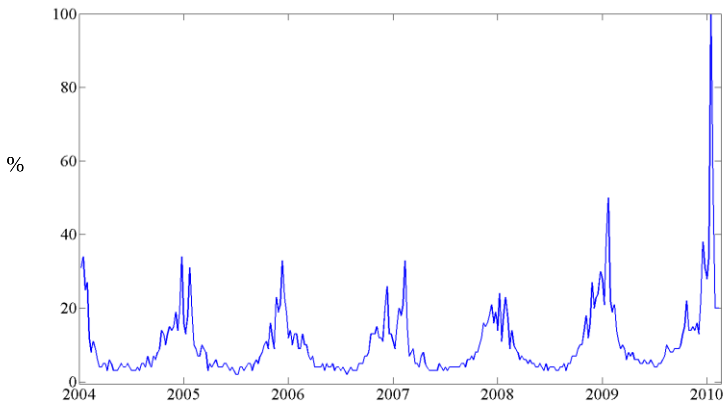

Figure 6-1: Google search index results for the phrase 'cold weather'
In this plot we can clearly see the seasonal pattern in the search index.

6.2.2 **Methodology**

6.2.2.1 Data

The first step was to identify several terms that could have a relation to the crude oil price. In this research the selection process was based on two factors[42]: (a) our knowledge about the crude oil market (b) the availability of these terms on Google Insight for Search [43]. The terms included in this research are: “coal price”, “cold weather”, “crude oil”, “GFC[44]”, “GPD China”, “Iran”, “Iran sanction”, “Iraq”, “Middle East”, “NYMEX future crude”, “OPEC”, “petrol”, “petrol price”, “Saudi”, “speculations”, “supply”, “UK petrol”, “war” and “WTI”. It is worth mentioning that the terms “oil embargo” and “OPEC meeting” yielded monthly data (around 80 data points) and were excluded from the analysis, while other terms did not return any hits. In both cases this indicates that either a low or no search volume was recorded for these phrases (Anvik & Gjelstad, 2010). In addition to this, Google allows users to select the geographic region of the search query. In this research this region was set to “worldwide” to ensure diversity.

Furthermore, some of these series are related to each other; for example, it is generally accepted that there is a relation between the crude oil price and the petrol price, and also between OPEC meetings and the crude oil supply. Therefore, to account for this relation we created a new variable that accumulates (sums) the search query of the following series in a new artificial time series, which we can call A1. It includes: “OPEC”, “oil supply”, “petrol”, “petrol price” and “crude oil price”. The sample size for all the series used is around 328 data points. Each of these series represents the search query of each phrase and covers the period from 4 January 2004 until 6 February 2010 at weekly frequencies. The crude oil weekly spot price for West Texas Intermediate (WTI) was retrieved on February 5, 2010, for the same period, from the Energy Information Administration website: http://www.eia.doe.gov/. All series were pre-processed in order to ensure that the dates matched each other.

However, one problem remains unsolved, as Google data cover a whole week including weekends, while crude oil price series (as for all financial data) includes only business days. Moreover, the official weekly price we are using in this research represents the official end of the week closing

[42] It could be argued that the selection methods of these terms was not systematic; however, the goal of this study is to explore the potential of this concept. A more systematic method could be used in future investigations.

[43] Our initial list of words was much richer than what we are presenting in this research; however, these terms were excluded from this study either because no search results were returned or the search hit retrieved only a monthly frequency, which is not the scope of this research.

[44] The phrase GFC stands for Global Financial Crisis. As expected the phrase returned zero hits until September 2008; hence, the results of the forecast using this term are not reliable.

price. Despite this problem we argue that our tests are still valid, as the opening price at the beginning of the week should not be very far off the closing price at the end of the previous week[45].

6.2.3 Testing for non-linearity

Testing for non-linearity was the first step in our methodology. If the data do not show evidence of non-linear behaviour then a linear model would be a better forecasting approach. On the other hand, if the data show evidence of non-linear behaviour then a non-linear model would be superior in forecasting the series. Of course this raises the vexed problem of model choice in the case of non-linearity. We relied on the fuzzy classification system (FCS) proposed by Kaboudan (1999) to test for the non-linearity, and also to identify the type of non-linearity. The FCS test is described and defined in Chapter 4.

6.2.4 NARX networks

A NARX network (Non-linear AutoregRessive with eXogenous variable) is a recurrent network and well suited for forecasting non-linear time series. It is represented by:

$$\hat{y}_{t+1} = f\big(y_t, y_{t-1}, \dots, y_{t-n(1)}, u_t, u_{t-1}, \dots, u_{t-n(2)}\big), \tag{6.1}$$

where y is the oil price time series, u represents each of the phrases, one at a time, $n_{(1)}$ is the length of the crude oil series and $n_{(2)}$ is the length of each of the phrase series. As with other types of ANN the goal is to approximate the non-linear function $f(.)$.

6.2.5 NARX Network architecture

Generally, there are three main requirements for any successful ANN model (Refenes, 1995, p. 15): (i) in-sample accuracy, (ii) the ability of the model to perform with new data, and (iii) stability and consistency of the network output. Without a doubt, model generalization and its stability are the most important criteria for assessment.

To ensure the above points are successfully met, a large number of considerations need to be taken into account. The most critical issue when dealing with neural networks is to determine the network complexity, i.e., the number of hidden layers and the number of hidden neurons in each layer. Using too many free parameters could result in over-fitting while using too few could result in the network not learning the correct function. The goal is to use the least number of neurons which generates the best results for out-of-sample (Kaastra & Boyd, 1996). There are no formal rules to solve this

[45] As a way forward to solve this problem, we propose that, instead of using the official closing price, one could start with daily data and create a weekly average which includes the opening price after the weekend. This should account for the search activities of Google users during the weekend. However, crude oil opening price data are not reported by the EIA.

dilemma. Heuristics algorithms and evolutionary computing methods are often used. Each of these methods has its advantages and disadvantages.

For NARX networks we select the number of hidden neurons by starting with one hidden neuron and adding one at a time.

As such, we found that, for NARX networks, one hidden layer with eight hidden neurons was enough for this problem. The tapped delay line[46] was set to eight steps (this number was found experimentally). Bayesian regularization was used to prevent over-fitting. For most the networks we trained, around 30 out of the 129 networks' free parameters were effectively used, as indicated by the Bayesian regularization algorithm in MATLAB, which means a network with only two hidden neurons would be enough for the approximation of the same delay used. The activation function used was the hyperbolic tangent; hence, the input was normalized to the [-1, 1] range. All networks were trained with a Levenberg Marquardt algorithm.

6.2.6 **Results and analysis**

6.2.6.1 Diagnostic tests

The correlation coefficients for each of these terms and the crude oil weekly price/return (and squared return) were not significant for all the phrases tested; hence, no evidence of cross-correlation was found (for the sake of brevity, details of the results are available in Appendix II: Google experiment subsection 10-2). Moreover, when the linear regression model was fitted for each of these terms (first difference) and crude oil (first difference) the correlation coefficient was not significant for all phrases. Therefore, no evidence was found of any linear relationship between each variable and crude oil returns.

Although it might seem worthwhile to conduct a co-integration analysis between the crude oil price /return and each of the Google series to find if an extended relation between these variables and the crude oil price exists. Co-integration analysis studies the long term relationship between two time series. In other words, there might be some influencing factors affecting and bounding the moment of these series. As such, if a stationary linear combination can be found, I(0), from two weekly non-stationary time series I(1) they are considered to be co-integrated (Brooks, 2008). However, the unit root tests[47] showed that apart from the crude oil price and "oil price" as a phrase, the test rejected

[46] The tapped delay line is similar to the concept of lags in econometric models.
[47] Both the Augmented Dickey-Fuller (ADF) test and the Phillips-Perron (PP) test were conducted. While ADF results were inconclusive (we reject the null hypothesis of the unit root for some lags and accepted it for others), the PP test constantly rejected the existence of the unit root for all lags (1-12) at 1% significance level with two exceptions: the crude oil price and "crude oil price" as a phrase.

the null hypotheses of a unit root in the data, i.e., the phrases were I(0). The main implication of this conclusion is that a co-integration test is no longer applicable as it requires both series to be I(1).

6.2.7 The FCS test

Table 6-1: presents the results of the FCS test applied to each series after removing the linear structure using a linear filter listed in the second column. In other words, the goal is to remove any linearity inherited within each individual series (this has nothing to do with the cross-correlation between each series and crude oil in section 6.2.6.1). According to the FCS test, most of these series have a linear component (a trend) and a non-linear component as well. Also some of these series have a high level of noise. As such we believe our choice of using ANN for this problem is justified.

Data set	Fitted ARIMA	R^2	θ	Decision
Oil price WTI	(3, 1, 8)	0.98	0.098	SL-WN
War	(0, 1, 2)	0.86	0.94	SL-NL-HN
OPEC	(1, 0, 2)	0.47	0.95	WL-NL-HN
Supply	Simple	0.37	0.88	WL-NL-HN
Iran	(3, 0, 0)	0.58	0.9	FL-NL-HN
Iraq	(0, 1, 1)	0.57	1.07	FL-WN
Saudi	(0, 1, 2)	0.48	0.95	WL-NL_HN
GFC*	(0, 1, 11)	0.94	0.08	SL-NL
Petrol-price	(1, 0, 0)	0.71	0.72	FL-NL-MN
Petrol	(0, 1, 2)	0.64	0.89	FL-NL-MN
Cold weather	(0, 1, 1)	0.55	0.98	FL-WN
Speculation	Simple	0.39	0.92	WL-NL-HN
Middle East	(0, 1, 2)	0.69	0.91	FL-NL-HN
Crude oil (phrase)	(0, 1, 12)	0.86	0.81	SL-NL-HN
Iran sanctions	Simple	0.46	0.28	WL-NL
WTI price	Simple	0.595	0.93	FL-NL-HN
NYMEX oil price	Simple	0.609	0.27	SL-NL
UK petrol price	Simple	0.596	0.79	FL-NL-MN
Growth GDP China	(0, 1, 3)	0.342	0.93	WL-NL-HN
Coal price	Simple	0.632	0.86	FL-NL-HN
A1	Simple	0.70	0.79	FL-NL-MN

Table 6-1: FCS test results for each phrase

6.2.8 Forecasting results

We commenced by creating a benchmark which includes the crude oil spot price solely as an input. We then used each term separately as an exogenous input and compared the results. All things considered, the differences in RMSE amongst all terms were not significant enough to draw a solid conclusion. Moreover, Table 6-2: shows all the performance metrics out-of-sample forecasts, and while the hit rate seems to be significant, the IC reveals that the forecast on average is significantly worse than the random walk. The same conclusion can be made when the data are transformed into returns (logarithmic difference).

	Hit rate	RMSE	R^2	IC	TU	MAE	BIC	AIC
Benchmark	57.96	0.06	0.02	0.74	1.00	0.00	0.05	-24.38
OPEC	55.33	0.06	0.00	0.76	1.02	0.00	0.05	-24.25
Cold weather	52.90	0.07	0.00	0.75	1.01	0.00	-0.04	-31.81
GPD China	53.50	0.07	0.03	0.75	1.02	0.00	-0.04	-31.73
Iran	**58.20**	0.06	0.00	0.75	1.01	0.00	0.05	-24.33
Iraq	54.40	0.07	0.01	0.75	1.01	0.00	-0.04	-31.82
NYMEX future crude	51.13	0.06	0.00	0.75	1.01	0.00	0.05	-24.33
Petrol	49.10	0.07	0.04	0.76	1.03	0.00	-0.04	-31.57
Saudi	56.10	0.07	0.04	0.74	1.00	0.00	-0.04	-31.95
Supply	52.80	0.07	0.00	0.75	1.01	0.00	-0.04	-31.81
War	52.50	0.07	0.00	0.75	1.01	0.00	-0.04	-31.80
WTI	52.10	0.07	0.02	0.75	1.01	0.00	-0.04	-31.78
UK Petrol	55.73	0.06	0.01	0.75	1.01	0.00	0.05	-24.32
Speculations	52.00	0.07	0.01	0.75	1.01	0.00	-0.04	-31.79
Petrol price	56.27	0.06	0.02	0.75	1.01	0.00	0.05	-24.32
Meddle East	55.00	0.06	0.01	0.75	1.01	0.00	0.05	-24.31
Iran sanction	55.00	0.06	0.01	0.75	1.01	0.00	0.05	-24.31
GFC	55.40	0.07	0.00	0.85	1.15	0.00	0.06	-23.21
Crude oil	51.80	0.07	0.01	0.75	1.01	0.00	-0.04	-31.77
Coal price	52.70	0.07	0.04	0.76	1.03	0.00	-0.04	-31.63
A1	53.40	0.07	0.02	0.75	1.02	0.00	-0.04	-31.72

Table 6-2: Out-of-sample one-step forecast for each term and the benchmark as return.
The benchmark is network trained by the crude oil return as an input; all other networks use the return of each term individually as an input while the crude oil return is used as a target.

6.2.9 Preliminary conclusion

In this research we tested whether Google Insight for Search could assist in predicting the crude oil spot price in the short-term. Our preliminary results showed that all the terms selected in this study did not improve the overall forecast error when used as sole input for training NARX networks; as a matter of fact, these terms seem to have a hindering effect compared with the benchmark. Finally, to answer our research question, ‘can Google Insight for Search improve the crude oil price forecast?’ we answer with a cautious affirmative, depending on (i) the selection of terms, (ii) whether sufficient search volume is available in Google for each term, and (iii) the selection of forecasting tool.

6.3 OPEC meeting announcements

6.3.1 Overview

In theory OPEC have one regular meeting each year and as many extraordinary meetings as needed, depending on the global economic status, extraordinary events, change in demand and so on. Each of these meetings ends with an official announcement about the production level of OPEC members with respect to production cuts, no changes, or production increases. Members’ production quota are allocated based on each member’s proved reserves, and it is generally accepted that some members provide over-estimated figures about their proved reserves in order to get a higher quota (Crémer & Salehi-Isfahani, 1991). Moreover, when OPEC declares a production cut, several members do not reduce their production accordingly (Crémer & Salehi-Isfahani, 1991). This point means that an announcement of a production cut could have a different effect on the return than an announcement of a production increase or no change (Crémer & Salehi-Isfahani, 1991). Table 6-3: shows OPEC meetings' official announcements dates as well as the meeting outcomes, starting from meeting 61 held on 05/08/1986 until meeting 157 held on 14/10/2010.

Event analysis is well known in financial literature endeavours to determine whether certain events (even if the event is anticipated) affect the asset/ commodity return. Almost all published studies about this issue followed the event analysis approach (Brown and Warner 1980; 1985) which aims to find if an event, in this case an OPEC meeting, affects the crude oil return at all. Hence, the abnormal return for a given commodity at time t is g which represents the difference between the observed return R and the forecasted return $\hat{R}$.

$$g_{i,t} = R_{i\,,t} - \hat{R}_{i,t} \tag{6.2}$$

The key issue here is how to compute $\hat{R}$.

Therefore, if an OPEC meeting announcement has no effect on the return then the cumulative average return (CAR) should not be significantly different from zero (Kothari & Warner, 2007).

$$CAR(t_1, t_2) = \sum_{t=t_1}^{t_2} AR_t \tag{6.3}$$

where $AR_t = \frac{1}{N}\sum_{i=1}^{N} g_{i,t}$.

The results will depend on two issues: (i) how to estimate the abnormal return, and (ii) the length of the window used. Since this approach has been used for the effect of OPEC meetings on crude oil returns and volatility, we find no point in repeating this approach. Alternatively, we endeavoured to create dummy variables in an attempt to account for OPEC meeting announcements and then use them alongside the returns to make a forecast.

Date	Price	Decision	Date	Price	Decision	Date	Price	Decision
5/08/1986*	14.35	No change	**26/06/1997**	18.84	No change	**4/12/2003**	31.24	No change
22/10/1986*	14.85	Cut	**1/12/1997**	18.76	Increase	**10/02/2004**	34.03	Cut
22/12/1986*	16.95	Increase	**30/03/1998**	16.32	Cut	**31/03/2004**	35.75	No change
29/06/1987	20.38	Increase	**24/06/1998**	14.54	Cut	**3/06/2004**	39.29	Increase
14/12/1987	17.47	Cut	**25/11/1998**	10.86	No change	**15/09/2004**	43.83	Increase
14/06/1988	16.85	No change	**23/03/1999**	15.36	Cut	**10/12/2004**	40.71	No change
28/11/1988	14.93	Increase	**22/09/1999**	24.26	No change	**31/01/2005**	48.25	No change
7/06/1989	19.7	Increase	**29/03/2000**	26.36	Cut	**16/03/2005**	56.5	Increase
28/09/1989	19.99	Increase	**21/06/2000**	33.64	Increase	**15/06/2005**	55.53	Increase
28/11/1989	19.33	Increase	**11/09/2000**	35.14	Increase	**10/10/2005**	60.71	No change
27/07/1990	20.07	Increase	**13/11/2000**	34.3	No change	**12/12/2005**	61.36	No change
13/12/1990	26.45	No change	**17/01/2001**	29.77	Cut	**31/01/2006**	67.86	No change
12/03/1991	20.06	Cut	**19/03/2001**	26.17	Cut	**8/03/2006**	60.06	No change
4/06/1991	20.9	No change	**5/06/2001**	27.84	No change	**1/06/2006**	70.11	No change
25/09/1991	22.11	Increase	**3/07/2001**	26.28	No change	**11/09/2006**	65.42	No change
27/11/1991	21.38	Cut	**25/07/2001**	26.71	Cut	**20/10/2006**	57.35	Cut
17/02/1992	19.42	Cut	**27/09/2001**	22.8	No change	**14/12/2006**	62.48	Cut
22/05/1992	20.79	No change	**14/11/2001**	19.63	Cut	**15/03/2007**	57.52	No change
17/09/1992	22.23	Increase	**28/12/2001**	20.42	Cut	**11/09/2007**	78.16	Increase
27/11/1992	20.29	Increase	**15/03/2002**	24.47	No change	**5/12/2007**	87.45	No change
16/02/1993	19.59	Cut	**26/06/2002**	26.67	No change	**1/02/2008**	89.03	No change
10/06/1993	19.27	No change	**19/09/2002**	29.49	No change	**5/03/2008**	104.45	No change
29/09/1993	18.73	Increase	**12/12/2002**	28.2	Increase	**10/09/2008**	102.66	Increase
24/11/1993	15.73	No change	**13/01/2003**	32.08	Increase	**24/10/2008**	63.34	Cut
28/03/1994	14.15	No change	**11/03/2003**	36.81	No change	**17/12/2008**	40.17	Cut
16/06/1994	19.83	No change	**24/04/2003**	27.52	Increase	**15/03/2009**	46.22	No change
22/11/1994	17.7	No change	**11/06/2003**	32.17	No change	**28/05/2009**	65.9	No change
20/06/1995	18.01	No change	**24/04/2003**	27.52	Increase	**10/09/2009**	71.95	No change
22/11/1995	17.93	No change	**11/06/2003**	32.17	No change	**22/12/2009**	73.42	No change
7/06/1996	20.28	Increase	**31/07/2003**	30.56	No change	**17/03/2010**	82.9	No change
29/11/1996	23.7	No change	**24/09/2003**	28.19	Cut	**14/10/2010***	82.7	No change

Table 6-3: OPEC meeting and production outcomes
* Meetings that are not included in our data sample.
Source: (Schmidbaue & Rosch, 2012, p. 17)

6.3.2 Switching model for crude oil returns

We start investigating whether there is more than one state, *regime*, that governs crude oil prices and returns. Markov Regime Switching was first introduced by Hamilton (1989, 1990) and is a suitable model to answer our question.

In the Markov Regime Switching, the series y is presumed to change its state function to the unobserved variable (state) of the process at time t denoted by S_i, where $i = 1, \ldots, m$ regimes.

According to the Markov model, the probability distribution of the state at any time t depends on the state at the previous time, which allows the model to capture the changes in variance of the state process and the change in mean (Hamilton, 1989):

$$Prob[S_t = 1|S_{t-1} = 1] = p_{11} \quad (6.4)$$

$$Prob[S_t = 0|S_{t-1} = 1] = 1 - p_{11} \quad (6.5)$$

$$Prob[S_t = 0|S_{t-1} = 0] = p_{22} \quad (6.6)$$

$$Prob[S_t = 1|S_{t-1} = 0] = 1 - p_{22} \quad (6.7)$$

$$y_t = \mu + \sum_{i=1}^{n} \phi_i \ (S_t) y_{t-i} + \epsilon_t, \quad (6.8)$$

where $y_t$is the time series, μ_t is the expected value of y_t, σ_i^2 is the variance and $\epsilon_t \sim N\left(0, \sigma_i^2{}_{S_t}\right)$.

For variables that deemed to follow the Markov switching process, two things are required to forecast whether the variable is going to be a given regime at a certain time: (i) the probability of the current period, and (ii) the transition probability matrix given in Equations (6.9 and 6.10) for models with two regimes (Brooks, 2008).

$$P = \begin{bmatrix} P_{11} & \cdots & P_{1m} \\ \vdots & \ddots & \vdots \\ P_{m1} & \cdots & P_{mm} \end{bmatrix} \quad (6.9)$$

where P_{ij} denotes the probability for the process to move from one regime to another, and must sanctify:

$$\sum_{j=1}^{m} P_{ij} = 1 \forall_i \quad (6.10)$$

The model parameter can be estimated using the maximum likelihood function. Considering the model in Equation (7.9) the log likelihood of this model is:

$$\ln L = \sum_{t=1}^{T} \ln\left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{y_t - \mu_{S_t}}{2\sigma^2}\right)\right) \quad (6.11)$$

However, since not all the world of states, (S_t) in Equation (6.8) are known beforehand, we cannot estimate Equation (6.11) outright. Therefore, we have to change the notation in Equation (6.11) to the following (Perlin, 2010):

	$\ln L = \sum_{t=1}^{T} \ln \sum_{j=1}^{2} (f(y_t|S_t = j, \omega) \Pr(S_t = j)$	(6.12)

where $f(y_t|S_t = j, \omega)$ is the likelihood function for the state j and ω are a set of conditional parameters for $f(.)$.

6.3.2.1 Empirical analysis

We start estimating the regime switching model using return (all) in-sample of the crude oil return. The model separated the data into two regimes. The results in

Table 6-4 shows that, for all three crude oil sub-series, there are two main dominating regimes, a tranquil regime and a volatile one. For state 1, the parameter μ_1 was positive for all sub-series corresponding with the positive return tranquil period. On the other hand the parameter μ_2 for state 2 was negative for all crude oil sub-series tested, indicating a negative return in the volatile regime. The transition probabilities are $p_{11,}$ and p_{22} for state 1 and state 2 respectively. Transition probability shows the probability the crude oil return will stay in a given regime at time t given that it was in the same regime at time $t-1$. The large values of $p_{11,}$ and p_{22} suggest that both regimes are generally stable and they are unlikely to change from one state to another. The third panel of Figure 6-2 shows a plot of the smoothed probability of each state for crude oil return (all). It is clear from this figure that from 31/1/1991 until 2/01/1998 the probability of being in state 1 was dominant. This has changed henceforth (from 1999 until 28/2/2003) during which period the return switched between the two states quite frequently. Moreover, from the same figure, although the probability frequently switches around, the volatile period remains either at zero or one. It seems that return I was more stable as the expected duration of each regime calculated as $[\frac{1}{(1-p_{ii})}]$, (Engel & Hamilton, 1990, p. 699) was relatively higher indicating the return I was less volatile compared to return II.

The next question is: can we find empirical evidence that OPEC announcements act as a trigger to regime switch? Yang (2004) applied the Markov switching model and showed that all OPEC countries[48] switch frequently between two production regimes. However, Yang's (2004) study stopped short of trying to find if this production change corresponded to a switch in crude oil price. We know that at particular points, historical events (e.g., the Gulf War of 1991 and the GFC of 2008) were the major cause of the changes in the crude oil market. Moreover, after the oil shock of 1990-1991 there was a stable period while from 1995-2003 the return seems to switch frequently between states 1 and 2. Liu and Zhang (2010) showed empirically that ANN can work side by side with a Markov switching model to predict financial data more accurately. In the next section we try to employ the findings using an ANN approach to answer the above question.

[48] The results of Yang (2004) showed stronger evidence of production regime switch for Iran, Saudi Arabia, UAE, Nigeria, Lybia Qatar and Venezuela than for Algeria and Indonesia.

Parameters	Return all (1)	Return I (2)	Return II (3)
$\boldsymbol{\mu_1}$	0.0008 (0.0003)	0.0003 (0.0004)	0.0014 (0.0004)
$\boldsymbol{\mu_2}$	-0.0023 (0.0017)	-0.0011 (0.0023)	-0.0078 (0.0031)
$\boldsymbol{\sigma_1^2}$	0.000304 (0.0000)	0.000219 (0.0000)	0.000411 (0.0000)
$\boldsymbol{\sigma_2^2}$	0.002255 (0.0001)	0.002273 (0.0001)	0.003085 (0.0002)
$\boldsymbol{p_{11}}$	0.98 (0.01)	0.98 (0.02)	0.98 (0.02)
$\boldsymbol{p_{22}}$	0.86 (0.01)	0.93 (0.00)	0.86 (0.01)
$\boldsymbol{ED_1}$	41.50	54.43	53.87
$\boldsymbol{ED_2}$	7.22	15.33	7.27
Log likelihood	12651.5826	7036.6387	6553.6841

Table 6-4: This table shows the parameters for each regime for each time series

The standard error of estimate is shown in parentheses under each parameter. ED_1 and ED_2 are the expected duration for regime 1 and regime 2 respectively; calculated as: $[\frac{1}{(1-p_{ii})}]$ (Engel & Hamilton, 1990).

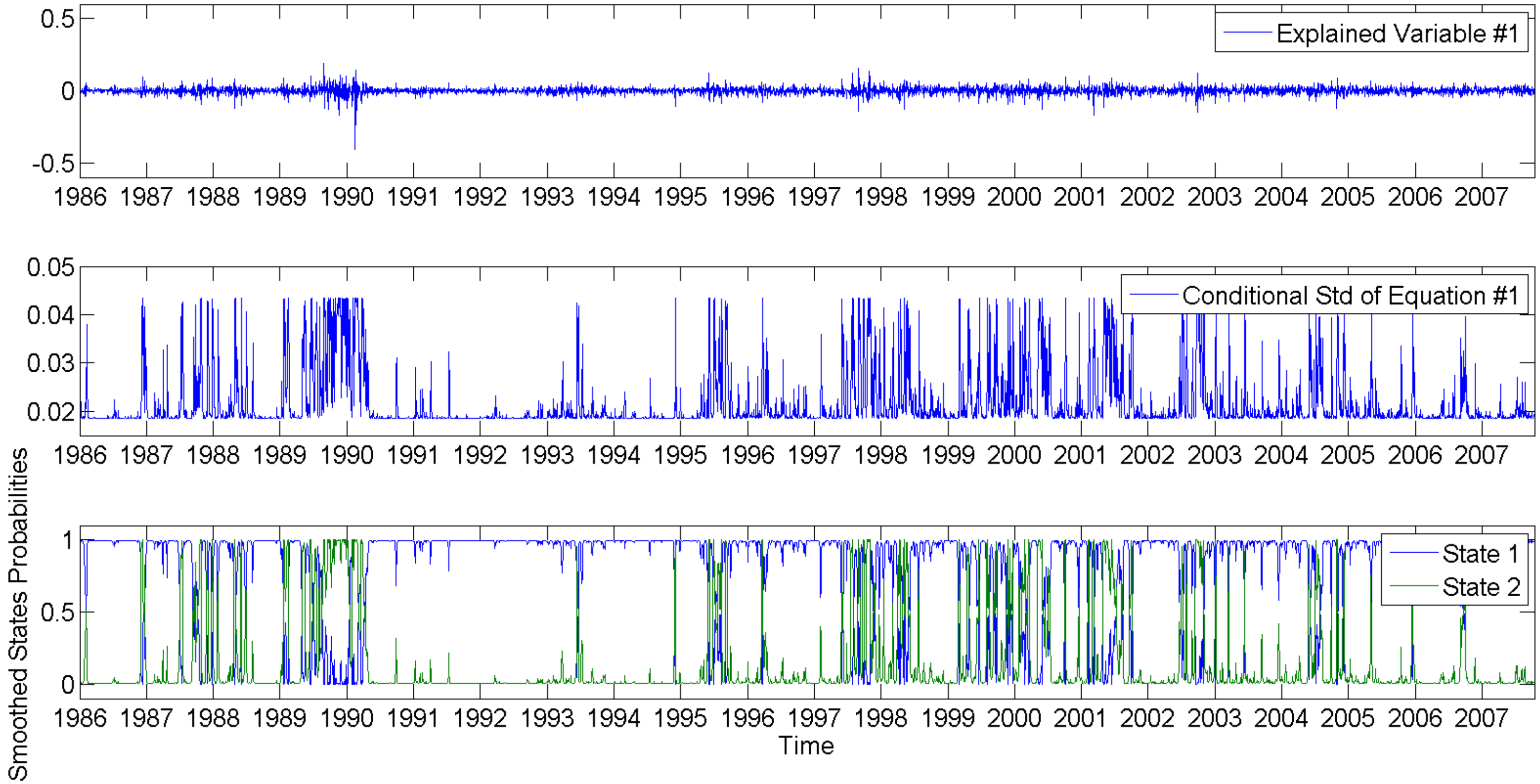

Figure 6-2: Crude oil return (1st panel), the conditional standard deviation (2nd panel) and the smoothed probability of each regime (3rd panel)

6.3.3 **Accounting for OPEC meetings' announcements**

The simplest way to account for OPEC meeting announcements and outcomes is by using dummy variables. We formed four dummy variables to account for the announcement dates, and the direction of the production, as follows:

- dummy variable 1: 1 if there was an announcement at that date and 0 otherwise
- dummy variable 2: 1 if there was an announcement of a production increase and 0 otherwise
- dummy variable 3: 1 if there was an announcement of a production decrease and 0 otherwise
- dummy variable 4: 1 if there was an announcement of production remaining steady and 0 otherwise.

These variables were used in addition to the crude oil lagged return to determine if using these variables will make any significant improvement to the return forecast. We used these variables as additional input to the network to forecast the return and the squared returns; the results are presented in the tables below.

As can be seen in Table 6-5, the hit rate on average was better than the benchmark but it was still poor. Moreover the DA statistic was not significant either, even for in-sample. On the other hand, using the squared returns as an input generated a better hit rate but not one better than the squared returns for the benchmark (no dummy variables). However, as can be seen from the results, adding these dummy variables helped the network to capture the oil shock of October 2008 very well. Normally, the network simulation is poor around this period because there are no patterns in the training sample to indicate that a shock is foreseen. This finding could be useful for forecasting crude oil returns in uncertain situations. Overall the simple dummy variable did not improve the forecast and performed poorly in term of forecast error. A possible explanation for this issue is that the simple dummy has a high value only at the actual date of the event. They do not account for the anticipation before the meeting and the market reaction to OPEC's decision, i.e., production cut, increase or standstill. In the next section we introduce a number of modified dummy variables to take these factors into account.

Metrics	Dummy		Benchmark	
	in-sample	**out-of sample**	**in-sample**	**out-of-sample**
Hit rate	50.1661	49.50952	49.69	48.821
RMSE	2.434727	3.980873	0.025556	0.035563
R^2	0.020718	0.004042	0.028949	0.018154
IC	0.699933	0.721674	0.687541	0.752927
MSE	5.931092	15.90783	0.000653	0.001265
MAE	1.685515	2.786954	0.018633	0.025279
SSE	31108.58	7683.48	1.812428	0.375617
DA	0.289866	-0.15485	-0.37257	-0.43037
P val	0.434943	0.545401	0.645264	0.666538

Table 6-5: Performance of one-step forecast of crude oil return using the dummy variables
The 95% confidence level in the out-of-sample hit rate was 45.7-53.4% over 1000 trials.

	Dummy		control	
Metrics	**in-sample**	**out-of-sample**	**in-sample**	**out-of-sample**
RMSE	0.003304	0.002244	0.003172	0.002153
RR^2	0.033025	0.050685	0.027486	0.046323
IC	0.777559	0.846068	0.746525	0.811873
MSE	1.11E-05	5.12E-06	1.01E-05	4.67E-06
MAE	0.000925	0.000986	0.000833	0.000907
SSE	0.044625	0.008845	0.040419	0.008058

Table 6-6: Performance of one-step forecast of crude oil squared return using the dummy variables compared to the control with no dummy variable
The results presented in this table represent the average performance metrics over 1000 trials. Since squared return was used as input and target, we rely on the error measures to assess performance.

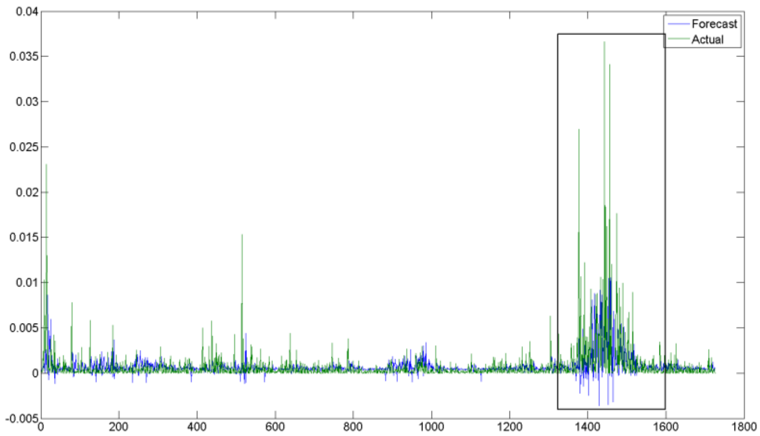

Figure 6-3: Out-of-sample one-step-ahead forecast compared to the actual squared return

6.3.3.1 Modified dummy variable

Dummy variables have several disadvantages as they do not reflect the anticipation that precedes the announcements (especially when OPEC meeting announcements are expected events and not surprising ones). Therefore, the goal here is to create a virtual series that reflects the effect of OPEC meeting announcements, and at the same time, tries to account for the anticipation of the meeting and the post-announcements effect. Also, we tried to find if there is any correlation between the announcements and the changes in the crude oil returns.

We started with the first dummy variable in the previous section (OPEC announcement date [0-1]). As this variable does not reflect the anticipation of the OPEC meeting, or the extended effect of the announcement, the dummy variable was modified. There is no consensus in the published literature around this topic on: (i) how long the effect of an OPEC announcement will last, and (ii) whether the effect differs based on the outcome of the announcement, i.e., production cut, increase, or no change. For example, Sharon and Lin (2009) used an event window of 20 days (divided into ten days before the announcement and ten days after) to account for the OPEC announcement. However, the authors stated that their choice of the length window seemed to be *ad hoc* and they tried to validate this choice empirically. Another recent study by Schmidbaure and Rösch (2012) used a sophisticated approach to decaying the OPEC announcement dummy based on the type of announcement. The authors generated different decay methods for: production cut, production increase or no change in production. Nevertheless, they could not provide clear theoretical justification for this approach. The only justification came through empirical experiments. In this section we chose a simple method and tried to validate empirically. We modified the dummy variable from [0-1] adding a diminishing value for three days before the announcement and three days after [0, 0.12, 0.25, 0.5, 1, 0.5, 0.25, 0.12, 0]. Then using wavelet multi-scale principal component analysis, based on a discrete approximation of a Meyer wavelet with eight levels of decomposition, the dummy variable was smoothed as can be seen in Figure 6-5. The smoothed dummy variable with memory (SDWM) represents our virtual variable, which is used as DK to aid the learning process for ANN, i.e., to account for the decision reached at the OPEC meetings. It seems that, scholars in the field only agree that there should be decaying effect of OPEC announcement. Therefore, we argue that our choice of a decaying OPEC announcement dummy effect in and by itself is a suitable one. Also, with the absent of theoretical assumption we selected a simple linear approach to do this because the goal of this experiment is to demonstrate a novel application of the existing soft computing method (wavelet analysis) to create the smoothed dummy variable.

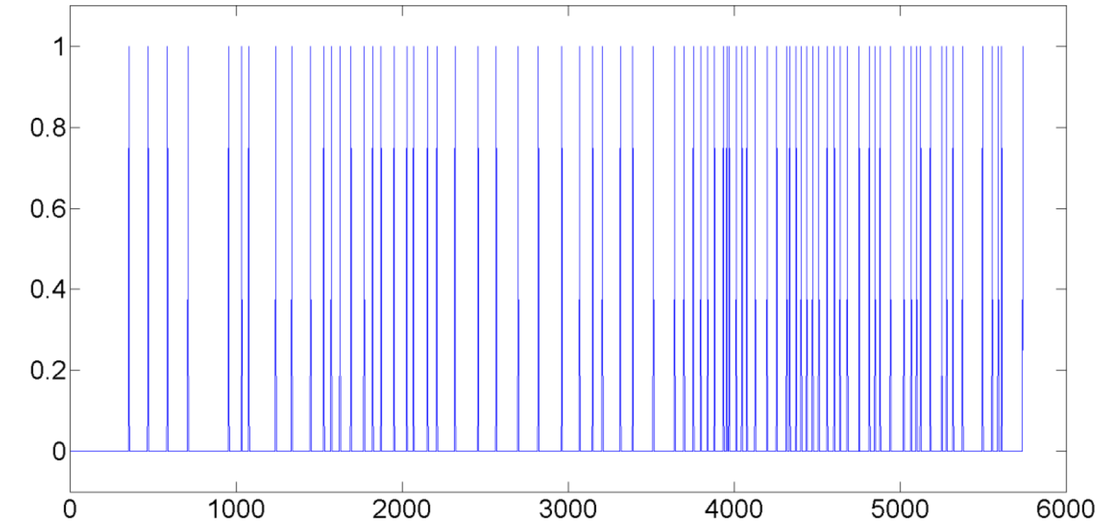

Figure 6-4: The modified dummy variable for OPEC announcement

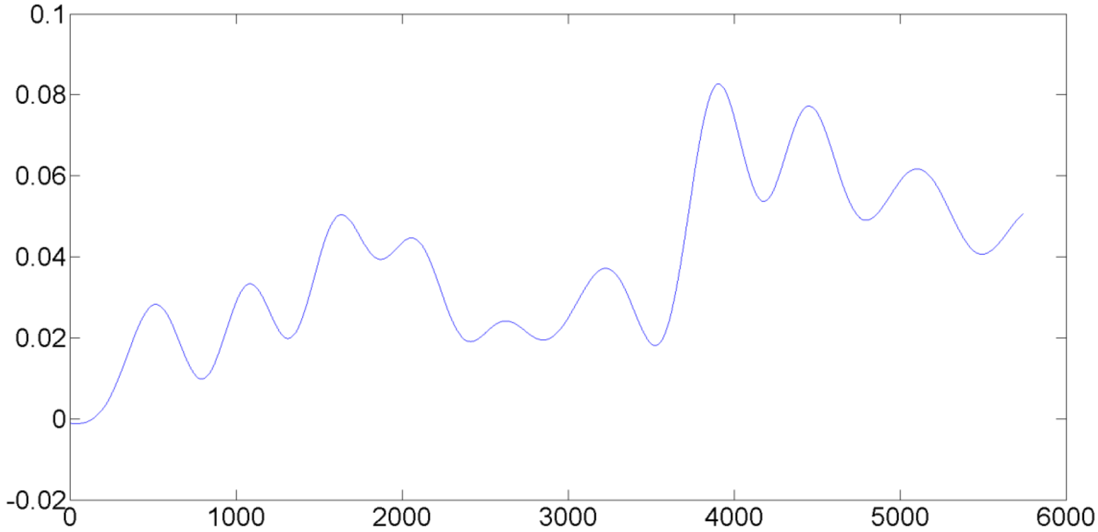

Figure 6-5: Modified OPEC dummy SDWM

Moreover, Figure 6-6 shows a plot of the OPEC SDWM variable and the crude oil returns after transforming the returns with the same wavelet simplifying approach (eight stages of discrete approximations of Meyer wavelet decomposition). Both series were normalised -1 to 1 for easy comparison. Visually, as can be seen from Figure 6-6, there is some correlation between OPEC announcement dates and the significant swings in crude oil returns.

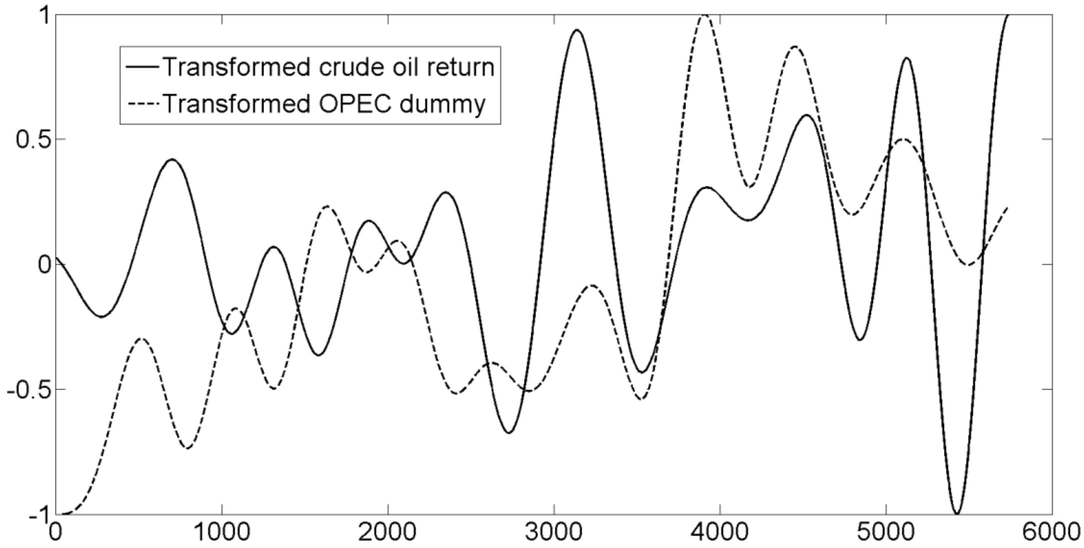

Figure 6-6: OPEC announcement modified dummy SDWM and the transformed return
The OPEC virtual variables (the dashed line) and the crude oil returns after transforming the returns with the same wavelet simplifying approach (eight stages of discrete approximations of Meyer wavelet decomposition).

The question here is, "Does this new information help in forecasting the actual return (not the one plotted in the figure above) or not?" Hence, lagged returns were used as input in addition to using the virtual series as auxiliary input.

The second dummy variable series also derived the same dummy variable which was then multiplied by the actual return to achieve symmetry and change the magnitude; we call it, symmetric dummy 1 (SD1). Since most of the dummy values are zero, the change took place only around the meeting dates.

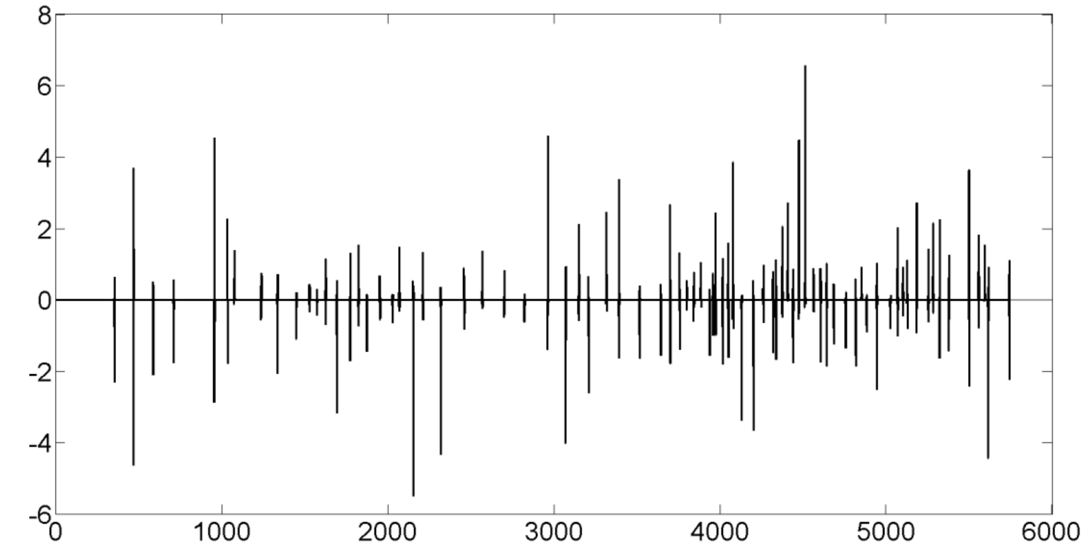

Figure 6-7: The symmetrical dummy variable for OPEC announcement

The third virtual variable represents the announcement date and outcome of the OPEC meeting. We started with the announcement date in Table 6-3**:** and gave:

- 1 if the meeting's outcome was a production increase
- -1 for a production cut
- 0 for no change until the next announcement.

This process will create another series which is not smoothed, and then using the same wavelet decomposition (six levels of decomposition were used) the variable was smoothed; we call it symmetric dummy 2 (SD2).

Table 6-7 shows that for SDWM the average hit rate was improved notably using this dummy variable. However, the 95% confidence level shows that the lowest out-of-sample hit rate obtained was just below the 50% mark. Moreover, the out-of-sample DA statistics on average were not significant which undermines the confidence in the hit rate. As for SD1 the out-of-sample hit rate is improved, but the RMSE is significantly higher than the benchmark; in other words, this dummy variable helped the model. For SD2, the average out-of-sample hit rate was 50%.

	SDWM		**SD 1**		**SD 2**	
	in-sample	**out-of-sample**	**in-sample**	**out-of-sample**	**in-sample**	**out-of-sample**
Hit rate	56.2857	54.34631	49.85756	50.64943	49.8576	50.6494
RMSE	0.02448	0.039305	2.67106	4.15626	2.67106	4.15626
R^2	0.03466	0.005853	0.024711	0.007493	0.024711	0.007493
IC	0.70304	0.735185	0.767064	0.777408	0.767064	0.777408
MSE	0.00062	0.001556	17.89811	27.75073	17.89811	27.75073
MAE	0.017	0.02727	1.911515	2.923439	1.911515	2.923439
SSE	3.22614	0.824465	92855.42	14707.89	92855.42	14707.89
DA	1.29699	0.005275	196.2836	197.1481	196.2836	197.1481
P value	0.24087	0.500547	0.507042	0.401027	0.507042	0.401027

Table 6-7: This table presents the average performance metrics for the three dummy variables tested, SDWM, symmetric dummy 1 and symmetric dummy 2
All experiments were repeated 1000 times.
The 95% confidence level for SDWM out-of-sample hit rate was 49-59% over 1000 trials.
The 95% confidence level for SD 1in the out-of-sample hit rate was 49-54% over 1000 trials.
The 95% confidence level for SD2 the out-of-sample hit rate was 47-59% over 1000 trials

In conclusion, only the first modified variable outperformed the benchmark significantly based on our performance metrics. However, we noticed that using dummy/modified dummy variables enabled the network to capture the big changes in the return better than the benchmark.

6.4 **Data transformation: Technical indicators**

In this section we assess the technical analysis method for crude oil forecasting as a first step in implementing the multi-agents model. In this section we follow Vanstone (2005) for transforming the data in a way that reflects the technical indicators. The idea of this section is to use data transformation, which falls under problem representation, to improve the forecast. In Chapter 5 we discussed some general transformation methods (Equations 5.5 and 5.6). In this section we test if transforming raw data using technical indicators would help train neural networks to find structure in the raw noisy data.

As such ten technical indicators were calculated; these indicators are widely used by technical investors (Vanstone 2005, pp.126-135):

1) three days moving average closing price (3MA)

2) 15 days moving average closing price (15MA)

3) short moving average (SMA):

$$SMA = \frac{3MA}{15MA}. \tag{6.13}$$

The use of a moving average is very common amongst financial market participants in general and amongst technical traders in particular. The rationale of using a short moving average (SMA) is that the technical traders might take a long position in a stock or commodity when the short moving average (three days in this case) and the longer moving average (in this case 15 days) crosses each other's paths (Vanstone, 2005). Also traders might do the same when the price moves from below to above the moving average, while technical traders might take a short position when the opposite happens (Vanstone, 2005). It is important to note that a moving average is a *lagging index* so the longer the moving average window, the bigger the lag from the price and the less accurate the strategy will be.

4) SMA_vol is calculated in the same way as SMA but for crude oil futures (near month) volume. This series was not available on Bloomberg so we purchased it from Norman's historical data (http://www.normashistoricaldata.com/). The time index was matched with the other series using our data processing code. Volume is regarded as a confidence indicator about the expected price movement, especially when the volume changes sharply contrasted to what is regarded as normal volume (Vanstone, 2005). Since we are dealing with spot prices, the best figure we could get is the near month rolling futures, which is the closest contract to spot price; also in the case of crude oil generally, the volume changes mainly in the near month futures only.

$$SMA_{vol} = \frac{3MA(vol)}{15MA(vol)} \tag{6.14}$$

5) LPR represents the lowest price of the past financial year (250 days). The period slides forward one day each time as soon as we accumulate one year. To avoid misrepresentation of the time index, after calculating this ratio, the first 250 days were removed from the series, and all other series, so our new sample starts from 1992.

$$LPR = \left(\frac{lowest(close, 250)}{close}\right), \tag{6.15}$$

where *close* is the close price and *lowest (close, 250)* is the lowest price in the last 250 days on a sliding basis (one day at a time).

According to Vanstone (2005) some stocks (or in this case crude oil commodity) are better traded when the price is lower than usual. Therefore, a trader could buy the stock or commodity when the price is low and resell it when it reverts to its long term mean. The issue here is to correctly identify the low price (bottom); hence, this ratio takes the lowest price in the last year as a representation of low.

6) HPR is the opposite of the previous ratio and is given by:

$$HPR = \left(\frac{close}{highest(close, 250)}\right), \tag{6.16}$$

where *close* is the close price and *highest (close, 250)* is the highest price in the last 250 days on a sliding basis (one day at a time).When the price of a stock or commodity is high some traders might want to take advantage of this if they believe there are better arbitrage opportunities since (a) the commodity price is high, thus it is not accessible for small traders or (b) the price did not reach the peak (Vanstone, 2005).

7) and 8) both represent Stochastic (3) or %K(3) and Stochastic (15) or %K(15) respectively, which are momentum oscillators and considered to be helpful for traders when the price is hovering around a stable range (Vanstone, 2005). Because this ratio is very sensitive it is usually smoothed over a period of time, in this case three days and 15 days respectively, and is calculated as follows (Vanstone, 2005):

$$\%K = \left(\frac{C - L_n}{H_n - L_n}\right) \times 100, \tag{6.17}$$

where C is the closing price, L_n is lowest price of n days, H_n is the highest price of n days, and n is the number of the days in this study: n=3 for % K(3) and n=15 for % K(15).

9) %K_ratio

$$\%K_{ratio} = \frac{\%K(3)}{\%K(15)}. \tag{6.18}$$

10) MACD ratio represents the difference between 26 days of exponential moving average and 12 days of exponential moving average (Vanstone, 2005).

6.4.1 **Networks structure**

Several combinations of these variables were tested as input for the network with different sets of lags. The number of lags and the number of neurons were reached as follows. We started with a maximum number of hidden neurons (n) and a maximum number of lags (m). These numbers are selected subjectively and they are relatively large. Then a matrix of neural networks was created (feedforward). The first network contained one lag and one hidden neuron and as we moved steadily to the last network in the matrix it consisted of (n x m) neurons and lags respectively. All networks were trained with in-sample data only for a fixed number of iterations and the network with the lowest RMSE was selected for the actual training. The number of lags reached in this manner was consistently between 10 and 12; the number of neurons selected was the maximum (10). However, so far the code does not accept a two-layer network; hence, we chose to use two layers with 25 and eight hidden neurons in each hidden layer respectively.

6.4.1.1 Experimental results

First a benchmark based on the crude oil spot price solely was created; in this benchmark 12 lags of spot closing price were used. Since the raw price is used, the hit rates are calculated as the percentage of predicting the directional change. There is no consensus in the literature on using the raw price with neural networks. For example, Azoff (1994) claimed that it is desirable to use the raw data when ANN is used, since pre-processing the data could affect the delicate structure of the original time series. This view was also shared by Yao, North and Tan (2001) and Venstone (2005) amongst others; however, other authors argue against it, for example, McNelis (2005) claim that like econometrics models, the input for ANN should be stationary. We believe that using the raw price as an input and/or output for a neural network is justifiable but not recommended. Vanstone (2005) argues that, instead of using the price and the volume directly, one could use technical (and fundamental) ratios derived from the price. The inputs and output for the network are summarized in Table 6-8. Also, since raw price is used the hit rate is defined as in Equation (6.19):

$$Hit = \frac{1}{n}\sum_{i}^{n} a_i \tag{6.19}$$

where

$$a_i = \begin{cases} 1 \rightarrow if\ (x_{t+1} - x_t)(\hat{x}_{t+1} - x_t) \\ 0 \rightarrow otherwise \end{cases}$$

x_t is the target at time t, $\hat{x}_{t+1}$ is the network forecast for time $t + 1$.

Network name	Input	Target
Benchmark	12 lag of spot closing price	Spot closing price 1 day ahead
Benchmark 2	12 lag of spot closing price	Spot closing price 3 days ahead
Tech net 1	12 lags of closing price, 3MA, 15 MA, SMA, LPR, HPR, %K(3), %K(13), %K ratio, OI, SMA_vol	Spot closing price 3 days ahead
Tech net 2	12 lags of closing price, 3MA, 15 MA, SMA, LPR, HPR, %K(3), %K(13), %K ratio, OI, SMA_vol, futures volume, MACD	Spot closing price 3 days ahead
Tech net 3	12 lags of closing price, 3MA, 15 MA, SMA, LPR, HPR, %K(3), %K(13), %K ratio, OI, SMA_vol	$\left(\frac{(\max(close_{i+3}, close_{i+2}, close_{i+1}) - close_i}{close_i} \times 100\right)$

Table 6-8: The input and output for each network

Table 6-9 presents the performance of two benchmarks, *benchmark1* for network-trained using crude oil price as sole input for one-step ahead and *benchmark2*, the same as before but for the three-days-ahead forecast. Table 6-10 compares the results for the three networks, tech net 1, tech net 2 and tech net 3. As can be seen, tech net 1 generated the best performance compared to the other two networks and it also outperformed the benchmarks in Table 6-9. The autocorrelation in crude oil price is very persistent indicating long memory, which could explain in part why the hit rate was high.

Metrics	Benchmark1		Bechmark2	
	in-sample	**out-of-sample**	**in-sample**	**out-of-sample**
Hit rate	73.89918	63.80759	65.22772	59.93191
RMSE	0.815542	16.98815	1.20072	16.39049
R^2	0.993266	0.64774	0.990389	0.656469
IC	1.295948	9.598046	1.907621	9.256195
MSE	1.20945	358.8439	1.728714	310.6179
MAE	0.546873	9.224591	0.834888	9.42989
SSE	4575.35	336236.8	6536.268	290427.7

Table 6-9: The average performance metrics for the benchmark
The hit rate represents the percentage of time ANN forecasted the direction change correctly

Metrics	Tech net 1		Tech net 2		Tech net 3	
	in-sample	out-of-sample	in-sample	out-of-sample	in-sample	out-of-sample
Hit	78.49652	66.3979	64.60615	55.13024	56.13108	51.25184
RMSE	0.626284	13.31718	1.151025	20.24991	1.575782	2.226108
R^2	0.998052	0.890757	0.993734	0.583398	0.029064	0.029064
IC	0.938551	6.705703	1.771004	11.22433	0.69984	0.734869
MSE	1.149769	251.4381	1.328145	449.5869	2.483088	4.955556
MAE	36.00372	8.038397	0.824288	14.08019	1.151746	1.528486
SSE	4355.912	170977.9	4756.087	397434.9	9398.489	3369.778

Table 6-10: This table compares the average performance metrics (out of 1000 trials) of the three network trained using technical indicators as an input

Table 6-11 shows the best results (in terms of out-of-sample hit rate) obtained for Tech net 1. As can be seen, the best performance significantly outperformed the benchmark.

Metrics	Best network from 1000 trials	
	in-sample	**out-of-sample**
Hit rate	79.67759	79.38144
RMSE	0.469324	2.094487
R^2	0.999142	0.999142
IC	0.697615	1.057099
MSE	0.220265	4.386875
MAE	0.331866	1.396209
SSE	833.7033	2983.075

Table 6-11: The best performance achieved for Tech net 1 out of 1000 trials

6.5 Discussion

The methods and results presented in this chapter provided some insight about crude oil prices and returns. However, like any empirical analysis they are affected with a number of limitations and caveats.

Beginning with the Google experiment, the idea is to use these non-financial variables to supplement the crude oil price prediction. The sub-series of the crude oil price is not an ideal one to use as it contains significant shocks and departs significantly from the long run mean of this commodity price. Furthermore, it could be argued that the selection method of these terms was an *ad hoc* one. This process could be systemized by using text-mining techniques to generate a 'dictionary' of terms related to the crude oil price, gathered from a corpus related to the crude oil industry, which will reduce the human impact (bias) on the conclusion. Even then, there is no guarantee that these phrases will return search queries from Google. The number of observations is relatively small; however, we are limited by the Google search query index we retrieved for each phrase. Industry-specific terms are not always available.

In the second study presented in this chapter, we investigated whether OPEC meeting announcements affect crude oil returns. Our goal was to use soft-computing to incorporate the effect of these announcements. Empirically, the Markov switching model clearly showed that there are two dominant regimes for crude oil returns, a tranquil regime and volatile one. This is very important as Fong and See (2002) found that crude oil futures price also switches between two regimes. They also found that the high volatility regime correlates perfectly with major markets events. However, Fong and See did not test if the OPEC decision, e.g., production cut, would constitute a regime switch. Furthermore, Yang (2004) showed that OPEC countries switch production regimes frequently. In this chapter we presented several modifications to the normal dummy variables using wavelet analysis to account for the effect of OPEC meetings and the outcome of each meeting—production cut, increase or standstill. Overall, all modified dummy variables outperformed the traditional dummy variable but preformed and the benchmark. However, one of the three smoothed dummy SDWM performed reasonably well as it helped increase the average out-of-sample hit rate to 54% from the benchmark rate of 48%. It is worth noting here that we have been conservative in reporting the results. We report the average over 1000 trials after removing the top-performing 2.5% of networks and the bottom 2.5%. Almost all published research we reviewed reported the best performance achieved. The methods proposed in this section are also affected by a number of limitations. One of these caveats is the linear decaying factor we introduced to account for the anticipation of OPEC meeting announcements. As we discussed before, there is no strong evidence to support how long in advance the anticipation for each meeting could have started or how it would have decayed after the meeting. Realistically, each meeting would have had its own political and economic circumstances. Therefore, our approach of linearly decaying the anticipation as well as the approaches presented in the literature is over-simplification in reality. It is worth noting that, so far, we have used the whole series to generate these forecasts.

It is also important that the results of the Markov switching model presented in this chapter provide some insight about the issue with long memory presented in the previous chapter. A likely explanation of the good out-sample hit rate we achieved for steps 19-25 might be due to the switching process as explained by Granger and Hyung (2004).

The best results obtained in this chapter, rather remarkably, used real data with technical indicator transformation. More precisely, one of the three technical analysis networks ‘Tech net 1’ generated a significant prediction to the price direction three-days-ahead. The average hit rate was 65%. On the other hand, Tech net 2 performed modestly with a hit rate of 55% for the three-days-ahead forecast compared to 59% for the benchmark. A possible explanation of the poor performance of Tech net 2 is the inclusion of the noisy futures volume in the training input. Although, Tech net 1 achieved a high hit rate, it is important to note that a high hit rate does not always translate into profitability in the market as Azoff (1994) has demonstrated. Further, adding transaction costs might offset the profitability of the model.

Finally, the results presented in this chapter suffer from the inherent limitations of using artificial neural networks, such as the black box criticism, the selection of the network topology and the network complexity, and the tendency of ANN to get stuck in the local minima. Chapter 3 of this thesis provided a detailed discussion of these limitations. Although, we argue that NEAT is less likely to suffer from some of the limitations above, genetic algorithms in general (the foundation of NEAT) also have a number of limitations that could affect the results in particular. As an example of these limitations, the size of the population needs to be large enough, in order to have adequate genetic diversity to find a reasonable solution. However, a trade-off needs to be made between the population size and the time converge, i.e., have a significantly larger population will improve the likelihood of avoiding local minima, but the model will be of no practical use as it will need a much longer time to converge.

6.6 Conclusion

This chapter employed three types of non-financial data and data transformation to help to improve the forecasting accuracy of crude oil returns. In the first study we investigated whether the search activities of Google users are helpful in predicting the crude oil price in the short-term. We also presented a critical analysis of the feasibility of using such a method, in the hope that this research will guide other scholars and energy professionals in their own research. Therefore, as a pilot study, a very abstract list of words was constructed which we believed could have a relation with the crude oil market. These words were inserted into Google Insight for Search and the search query of each phrase was retrieved. Non-linear auto-regressive with exogenous variable (NARX) networks were selected as a forecasting tool for price, because of their desirable capability as a non-linear and universal function approximation model. This is a very important issue, especially since our tests revealed that most of Google's series follow non-linear dynamics; hence, a non-linear model, such as NARX, should be better equipped than linear models to deal with the complex problem of forecasting the crude oil price. Our empirical results show: the term 'Iran' selected from Google Insight for Search helped in improving the forecast for crude oil weekly price (hit rate 58% compared to the benchmark rate of 54%). We believe that further research is needed to determine if Google Insight for Search data could provide much needed additional information to achieve better forecasting results for a complex series like the crude oil price.

In the second study we presented a number of modified dummy variables series to account for OPEC meeting announcements regarding oil production in each period, as an additional input for ANN. Unlike the event analysis methodology used in the literature, we used a set of dummy variables and virtual variables (an artificial series) to account for the announcement dates and their outcomes. The virtual variables were created using wavelet transformation of artificially created variables to generate a smoothed continuous series.

The goal of this study was to demonstrate how we can apply our knowledge about: (i) the crude oil market, (ii) soft-computing methods, and (iii) economics models, to uncover new understanding about the market and improve the forecasting accuracy of ANN. Hence, we have shown a new way to construct a supplementary time series from OPEC meeting announcements (knowledge about the crude oil market) and wavelet analysis (soft-computing). Overall, and except for SD1, it would appear from our analysis that these modified dummy variables do not improve the hit rate significantly. However, it was noticed that in the case of feedforward networks, adding these variables enabled the model to capture big changes in crude oil returns more accurately than small changes.

In the third and final study of this chapter we introduced a model that relies on technical analysis for the crude oil market. The idea here was to demonstrate how technical analysis as domain specific DK (financial domain) can aid the learning process. Although the evidence from the stock market (Vanstone, 2005) showed the efficiency of this technique in trading systems, we looked at how these indicators help in the forecasting of commodity prices with higher level of noise. We found that such techniques, while simple, were very effective in aiding the learning process of ANN.

The main contribution of this chapter is the presentation of a method for creating non-financial series. This acts as a supplementary source of information and compensates for the inadequate amount of data available for modelling crude oil short-term price movements.

6.7 Summary

This chapter showed how artificial examples can be used to improve the generalization of ANN. Three different methods were presented. Three cases were assessed, non-financial data, modified dummy variables and fundamental oil data and pre-processing of real data.

CHAPTER 7: Multi-agents model for crude oil market

7.1 Introduction

We propose as a final part of this thesis, a multi-agents model for crude oil market forecast. Close

This model takes advantage of Grothmann's (2002) multi-agents neural network model, and the multi-agents genetic algorithm model of Palmer, Arthur, Holland, LeBorne, and Tayler (1994) and combines them with the concept of neuro-evolution (neural networks optimized by genetic algorithm and reinforcement learning), namely NEAT (see Chapter 3 for details about NEAT), in order to reach a more realistic representation of the crude oil market. We argue that a multi-agents model is better equipped to capture the micro-dynamics of a complex commodity market like crude oil because it is able to explore a larger parameter space than a single model. Therefore, our goal is to find if the forecast of artificially intelligent agents would add useful information to train traditional neural networks, i.e., to act as a hint.

The multi-agents model is a relatively new method designed to tackle complex problems where an analytical solution is not available (Levy, 2011). Moreover, agents-based modelling in finance provides an additional dimension in modelling as it takes into consideration the behaviour of the individual agent and how the decision of each agent reflects on the market (Grothmann, 2002; Levy, 2011).

Our motivation for applying a multi-agents model to crude oil price forecasting is based on the facts that: (i) crude oil dynamics are very complex, (ii) there is hardly any study in the crude oil literature investigating the feasibility of a multi-agents model for forming an effective forecasting model, and (iii) a multi-agents model provides generic imitation financial/commodity market functionality[49], hence, it could help to create a realistic model of the market.

The evidence from the crude oil literature, along with our results highlighted in Chapter 4 of this thesis, support the premise that the crude oil market follows very complex dynamics.

Schweitzer (2002), stated (citing Herbert Simon):

[49] The limit of the proposed market imitation is the result of having intelligent agents buying and selling the commodity in question; this concept can be applied to any financial instrument or commodity and is not unique to the crude oil market.

> Economics and the social sciences are, in fact the 'hard' sciences, as Herbert Simon argued, because the complexity of the problems dealt with cannot simply be reduced to analytically solvable models or decomposed into separate sub-processes (Schweitzer, 2002, p. v).

Furthermore, in relation to crude oil markets, Professor James D. Hamilton, who is considered an authority on econometrics and of energy market analysis literature, has stated:

> It is sometimes argued that if economists really understand something, they should be able to predict what will happen next. But oil prices are an interesting example (stock prices are another) of an economic variable which, if our theory is correct, we should be completely unable to predict. (Hamilton, 2009, p. 184).

On the same note, the Energy Modelling Forum (EMF) has stated:

> It should be remembered that the oil market is a highly complex, uncertain network of centralized and decentralized decision-making processes. (Energy Modeling Forum, 1982, p.12).

There is an endless number of quotes in this direction in the energy literature: combining these citations with the poor forecasting recorded of crude oil, one could strongly argue that analytical models could be the least suitable modelling strategy. This reduces the choice for modelling the crude oil price to deciding between deductive and inductive models, and also anything in between e.g., hybrid models that combine both approaches such as a fuzzy neural system, or hinted neural networks and a multi-agents model.

If we examine the micro-structure of financial markets in general and crude oil in particular we find it consists of a large web of "agents" buying and selling (H. G. Zimmermann, Neuneier, & Grothmann, 2001). These agents could be individuals, investors, hedgers and refinery companies amongst others. In a multi-agents model the market is modelled from "the bottom up", in order to capture the effect of agents' behaviour on price movement (Grothmann, 2002). In other words, we examine how the behaviour of a group of agents in one marketplace affects the price of the commodity in the short-term. According to Grothmann (2002) agents assess and respond to the market environment and events to establish expectations of the price direction. Moreover, because agents belong to different schools of thought and have different objectives (hedging, profit maximizing), each agent reaches its own conclusion about how the market is going to move in the future: up, down or steady (Grothmann, 2002). Therefore, theoretically, a multi-agents model is a reasonable approach to deal with crude oil market forecasting.

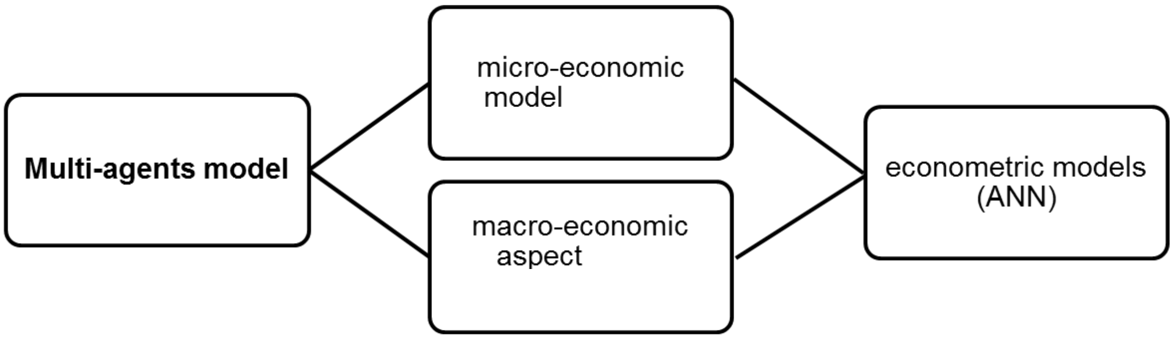

Figure 7-1: The modelling methods involved in our multi-agents model

To the best of our knowledge, only one study in the literature applied agent-based modelling for crude oil forecasting. Lean, Shouyang, and Kin Keung (2008) presented a multi-agents model for crude oil price forecasting based on a fuzzy system. The authors used a number of artificial intelligence agents (AI) to forecast the crude oil daily price. According to these researchers, all agents in their model have the same structure; they only differ in some parameters to achieve heterogeneity, i.e., to reach different forecast outcomes. The output of each agent was then fed to a fuzzy system for integration and then the output of all these agents was defuzzified to reach the final forecast. The findings of Lean, Shouyang, and Kin Keung (2008) suggested that multi-agents models outperformed all benchmarks (ANN, SVM and simple averaging) for out-of-sample testing.

However, in our opinion, there are a number of issues that raise concerns in Lean, Shouyang, and Kin Keung's (2008) model. Firstly, the authors did not explain what type of AI-based agents they were using, the quantity of these agents and the types of parameters they changed in each one of them. This is a very important point because it affects the outcome directly. Secondly, the authors used raw prices without any transformation, e.g., they did not use a logarithmic return; it is well known that the raw price of most commodity is non-stationary; hence, when used as an input this could lead to spurious regression (Refenes, 1995).

With all these points in mind, our aim is to present a novel multi-agents model to improve the forecasting accuracy for crude oil. Moreover, we want to test the feasibility of agents-based learning for this complex problem.

7.2 Design considerations

There are two main levels of design consideration for multi-agents models: (i) design specification of the agents, and (ii) the way agents interact (Zimmermann, et al., 2001). The interaction between both levels in a single modelling strategy is what distinguishes the multi-agents model from other modelling approaches.

7.2.1 Agent design consideration

Figure 7-2 shows the four steps involved in the designing of agents. Agents should be fairly heterogeneous in order to reach different decisions from each other and are able to learn in order to optimize their own objective function, e.g., profit maximization (Farmer & Joshi, 2002; Grothmann, 2002).

Heterogeneity is a very important design aspect of the agents because real world agents are different from each other in their aims, backgrounds and levels of complexity, i.e., some agents use simple regression to forecast the market while others use very complex models (Grothmann, 2002). Besides, if agents are homogeneous then they will reach almost the same conclusion about the market behaviour, which is unrealistic.

Figure 7-3 shows a taxonomy of agents' learning schemes for multi-agents models. Basically, there are two types of agents: (i) rule-based agents, in which the model attempts to evolve a complex set of rules to solve the problem at hand, e.g., evolving trading rules (Farmer & Joshi, 2002; Grothmann, 2002), and (ii) forecasting agents, where agents try to predict the market based on the information available to them at the current stage (Grothmann, 2002). Each of these learning strategies has its advantages and disadvantages; however, we believe that forecasting agents are the best type of agents for our problem as our goal is to forecast the market more accurately.

An objective function is at the heart of agents' learning and it is closely connected to the heterogeneity of agents. Real-world agents are in the market to optimise certain objective functions, e.g., making profits and hedging amongst others. Therefore, artificial agents also need to optimize an objective function (Farmer & Joshi, 2002; Grothmann, 2002).

Finally, the forecasts of the agents need to be mapped into a decision-making scheme. In other words, if the agents predicted the price is going up, down or steady, this prediction needs to be aggregated to affect the final price.

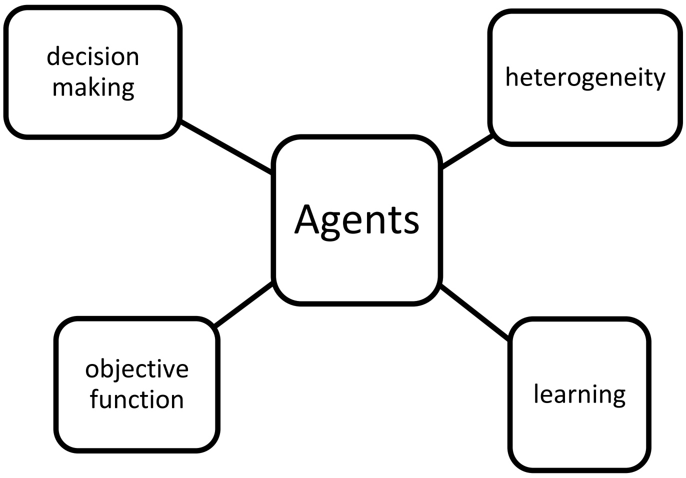

Figure 7-2: Agents design requirements for the multi-agents model
Source: (Grothmann, 2002, p. 148)

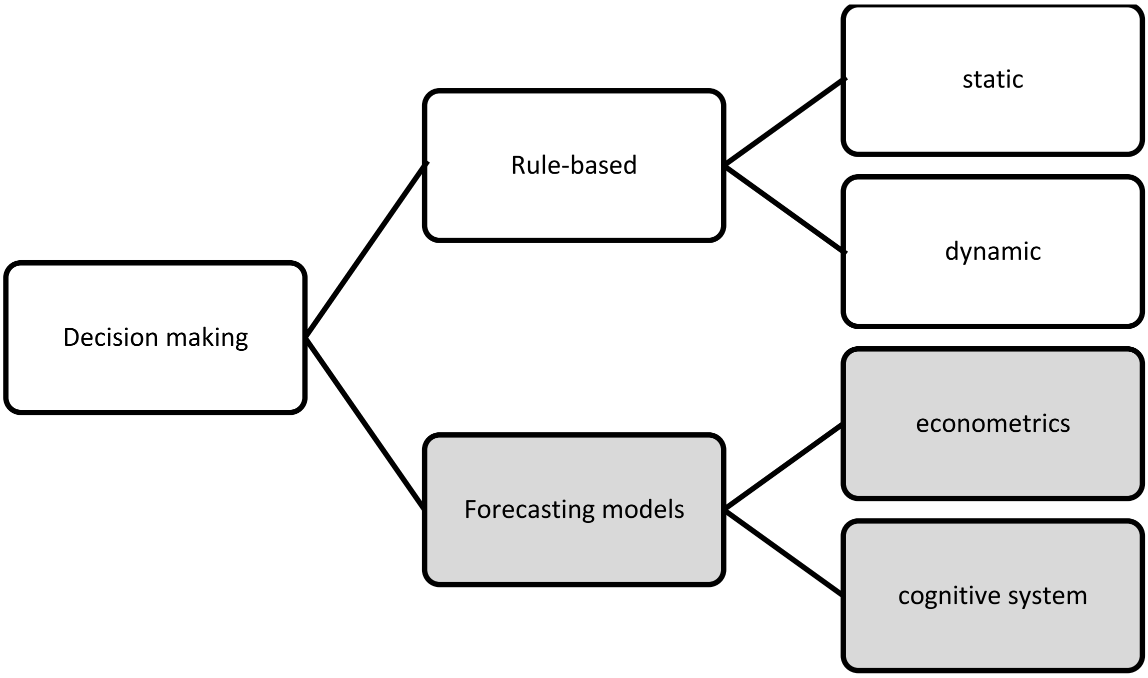

Figure 7-3: Decision-making schemes in a multi-agents model
In this figure the grey colour shows the type of agents used in our model and the white colour shows those not used in this research. We argue that our agents can be classified as econometric entities and as cognitive systems at the same time.
Source: (Grothmann, 2002, p. 164)

7.2.2 Agents interaction

The central part of this system is the way in which the agents interact to change the price, i.e., the *mechanism* of price formation system (micro-economics). According to Grothmann (2002) there are at least three different mechanisms to accomplish this (Grothmann, 2002):

1. price shifts as a response to the relationship between demand and supply of the commodity
2. price changes based on a predefined demand function
3. price changes in a way that close to the real market (bid and ask spread) functionality.

We rule out the second option because it is very difficult to define a demand function for the crude oil market. While the third option is indeed the most realistic option, in order to implement it one needs high frequency data (tick by tick) which are not available for crude oil prices.

We focus our attention on the first option. Here each agent forecasts the return for the next period; if the agent's outcome > 0, the agents request to buy, and vice versa. The market excess (supply) is then calculated and the price is changed based on Equation (7.1).

$$p_{t+1} - p_t = \varepsilon(d_t - s_t) \tag{7.1}$$

where p_{t+1} is the new price, p_t is the current price, ε is a positive constant representing the price shifting factor, and d_t and s_t are the demand and the supply at time t , respectively.

The market impact function in Equation 7.1 is calculated as follows: (i) We collect the order of all agents (buy if return > 0 and sell otherwise), (ii) The net of the market order is calculated, and (iii) The price level is then adjusted as a response to the market demand.

7.3 Multi-agents model for crude oil market

Perhaps the best models that used real world data were presented by Grothmann (2002). Grothmann's (2002) model aimed to forecast the weekly foreign exchange rate between the US dollar and the German mark. The author argued that a non-linear neuron of a neural network can simulate the behaviour of an agent. This agent (neuron) is able to perform three main tasks (Grothmann 2002, p.220):

- information prioritizing
- market evaluation
- action taking.

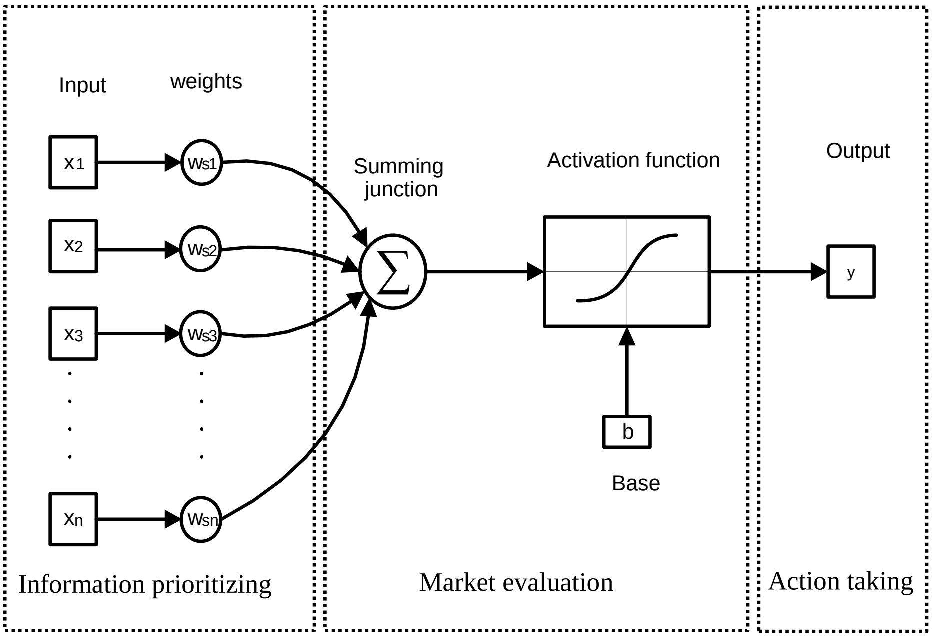

Figure 7-4: A single neuron as an agent

Here external information (inputs) are weighted and prioritized by the neuron's weights. Higher weights imply important information and vice versa. The summation function along with the non-linear activation function will perform the market evaluation task. The output of the neuron represents the action of the agents, e.g., a positive output is a signal to buy and a negative output is a signal to sell.

Source: (Grothmann, 2002, p. 221)

The external inputs $(x_1, \ldots, x_n)$ are weighted by the input weights w, (here, higher weights imply important information and vice versa). Based on this view of a neuron, a market place can be represented by a neural network with a large number of non-linear neurons that interact amongst themselves (Grothmann, 2002).

Grothmann (2002) presented several models accompanied with empirical results to support this view. However, in our opinion, while the argument of Grothmann (2002) is theoretically valid, we believe that a single neuron does not have enough processing power to represent an intelligent agent; rather, a neural network with a number of neurons is a much more powerful entity to represent a decision- making agent.

Therefore, we propose an agents-based model consisting of two stages, as illustrated in Figure 7-5.

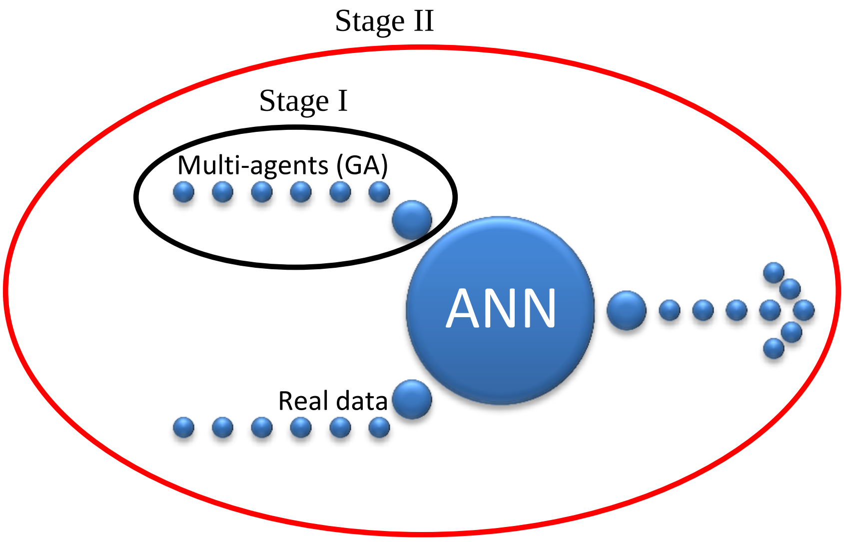

Figure 7-5: [coloured] A summary our model
Stage I is the multi-agent phase presented inside the black oval. Stage II is the econometric model presented inside the red oval

7.3.1 First stage

In this stage NEAT is allowed to run for a number of generations until one of the stopping criteria is met. Different fitness functions could be allocated to groups within the population. This point is inspired by the diversity in objective function in the real financial and commodity market. Each network within NEAT will generate forecasts and then the integration of all these networks, i.e., agents, will shift the price for the next step. Stage one includes agents' development and training, forecasting and price formation. Figure 7-6 illustrates Stage one.

- The input data include real financial data in addition to artificial data.
- As species are formed, each species will have a different number of networks (agents).
- Agents are heterogeneous in their objective function, input, complexity and structure.
- Agents' forecast of the future return, based on their objective function and the price, will be adjusted based on the trading decisions of all agents in the marketplace, i.e., price formation.

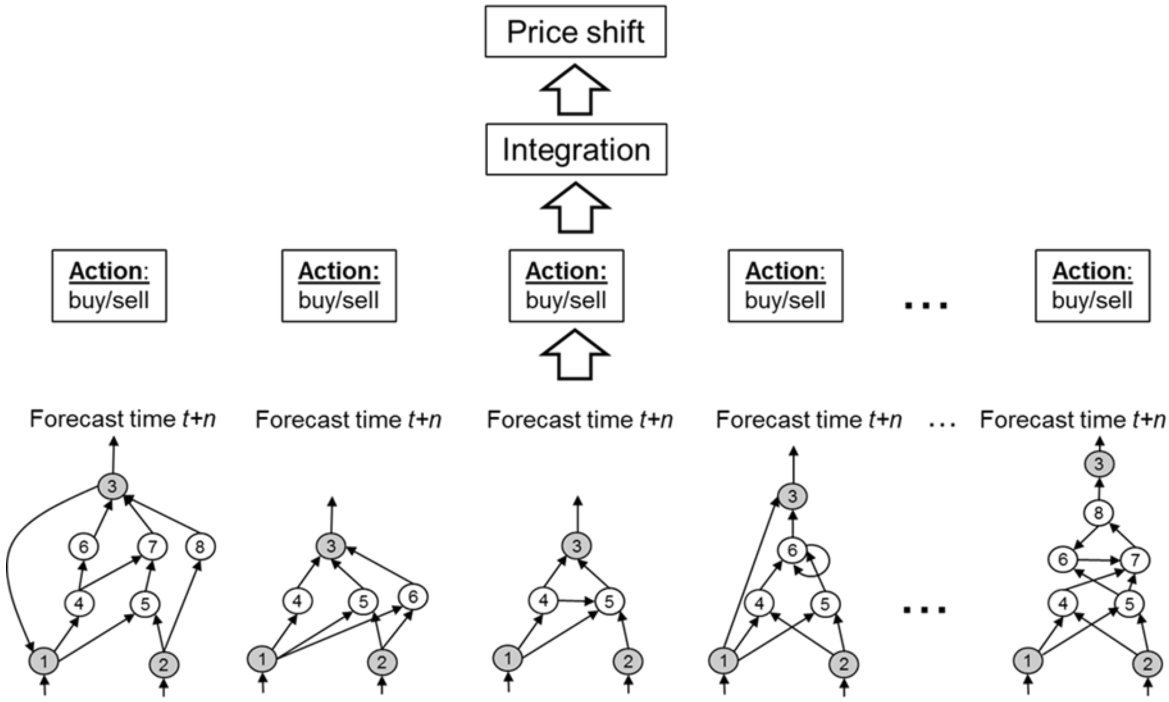

Figure 7-6: The first stage of our multi-agents model

In this stage NEAT is allowed to run for a number of generations until one of the stopping criteria is met. Different fitness functions could be allocated to groups within the population. This point is to simulate the diversity in objective function in the real market. Each network within NEAT will generate forecasts and then the integration of all these networks, i.e., agents, will shift the price for the next step.

7.3.2 **Second stage**

In this stage we examine whether agents-based forecasting is useful in improving the forecast accuracy of traditional ANN.

- The output of all the agents is fed into the ANN.
- The output of this network represents the market expectations of the price at time $t+n$.

In our model Stage II is justified as we cannot expect the artificial market to perform as accurately as the real market. To elaborate, Figure 7-7 compares the real market and the artificial market. First, the number of agents participating in the real market is huge; we cannot match this artificially due to computation time. Secondly, real agents in the market have access to a large amount of data and they are able to adapt to the new information that enters the market in real time (Abu-Mostafa, 1995a). In our artificial market we are restricted to the input space available to us (Abu-Mostafa, 1995a). Therefore, we are proposing Stage II to find if the output of the artificial market (stage I) contains new information that will help traditional ANN to reach more accurate forecast.

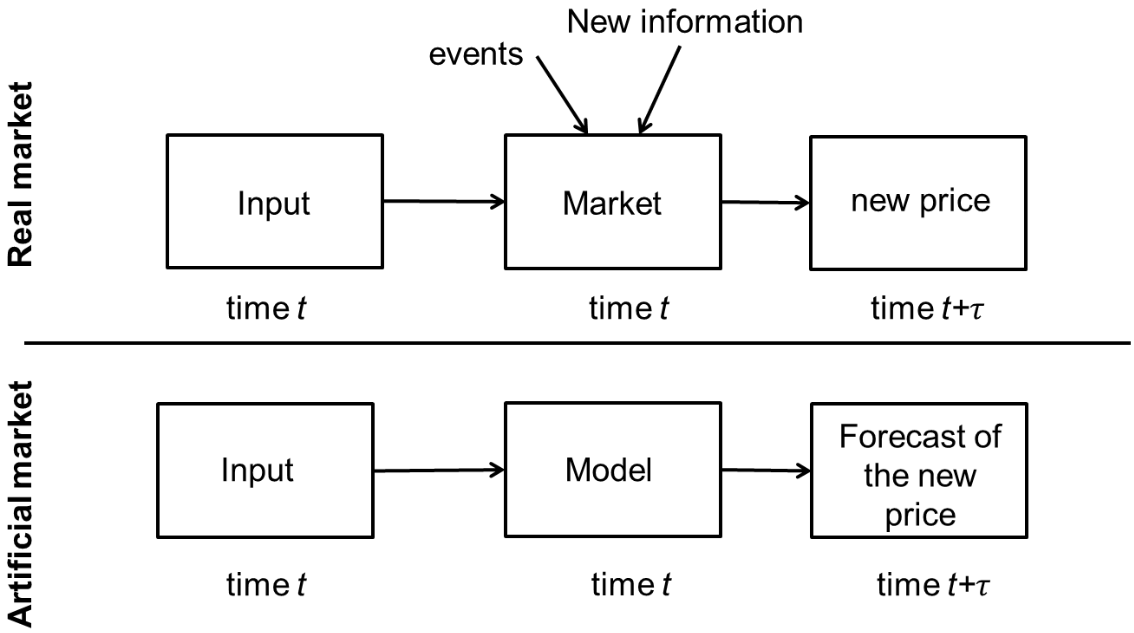

Figure 7-7: This figure compares the real market (upper panel) and the artificial one (bottom panel)
In the real market, a very large number of agents from very different backgrounds (such as technical traders and hedgers, etc.) interact in the market. These agents have the advantage of reacting to any new information during the day. In an artificial market, there are only a few agents who have access to relatively small amounts of information.
Source: Adopted from Abu-Mostafa (1995a, p.223)

7.4 Design assessment

In this section we show that our proposed model meets the design specification for a multi-agents model. We started with a simple multi-agents system to test the applicability of NEAT as a platform for our multi-agents model. The total number of agents was set to 30. All agents were trying to optimise the following fitness function:

$$E = \sum_{i}^{n} p_t \times (y_{t+1} - y_t) \tag{7.2}$$

where

$$p_t = \begin{cases} 1\ if\ (\hat{y}_{t+1} - y_t) > 0 \\ -1\ if\ (\hat{y}_{t+1} - y_t) < 0 \\ 0\ if\ (\hat{y}_{t+1} - y_t) = 0 \end{cases}$$

y_t is the actual crude oil return, $\hat{y_t}$ is the predicted return.

The goal of this experiment was to test the four steps involved in the micro-layer design specifications of the multi-agents model: heterogeneity, learning, objective function (here, this is to maximize the profits) and decision-making.

7.4.1 Heterogeneity

Heterogeneity of our agents was implemented in several ways:

1. **Input**: Different agents will connect to different inputs. This is similar to real markets as market participants rely on different types of information to make their forecast. Figure 7-9 shows the connection genes for each agent as an area. Each species was given the same colour code. What is clear from this figure is:

    a. There was a similarity in the pattern (input-output connectivity); this can be explained as we connected input 1 uniformly in the first generation.

    b. Some patterns were similar in the same species, which is expected and reasonable.

    c. The length and the pattern in connectivity were different across species (heterogeneity).

2. **Complexity and structure:** As agents (networks) have diverse structures, some agents will reach different decisions than others as they are able to process the information in several ways. Figure 7-8 shows the complexity of each agent, and the species are colour coded. Again we can see similarity within the species and heterogeneity across the species.

3. **Objective function**: Similar to the real market, agents in our model had diverse objective functions to optimize. So far all agents optimized the same fitness function. This was the case at the time of the experiment so now we can allocate several fitness functions for the population.

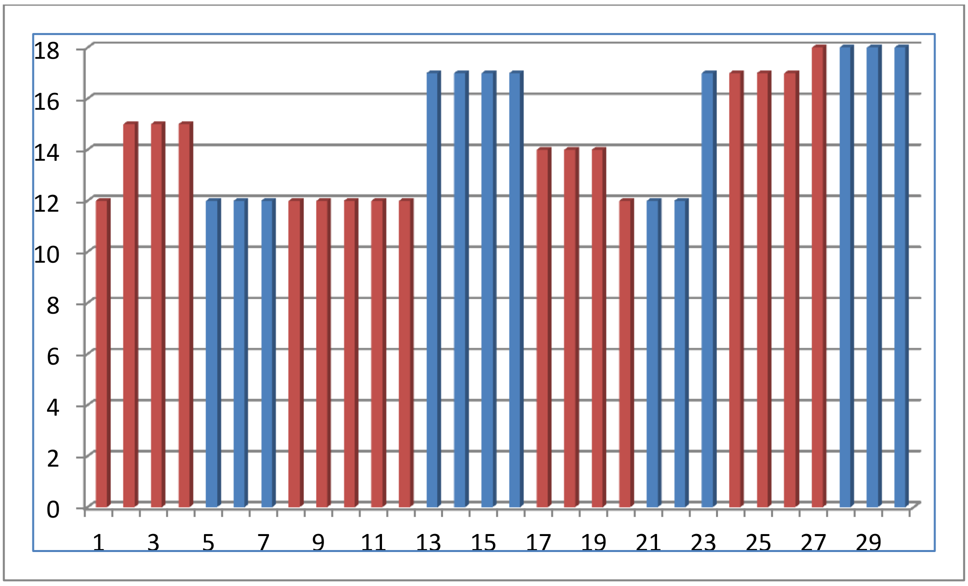

Figure 7-8: [coloured] The number of hidden neurons of each network (agents)
The colour of the column represents the species. The species in this figure are colour-coded; each column shows the number of hidden nodes for its corresponding agent.

Here each agent forecasted the return: if the agent outcome > 0 then the agent requests proposed to buy and vice versa. The market excess was then calculated and the price was changed based on

Equation (7.1). Agents were allowed to train for 200 generations; by the end of Generation 200, excess demand was calculated. The difference between supply and demand represents the excess demand. Figure 7-10 plots the excess demand for all agents. Once we have the excess demand, the price shift is calculated as in Equation (7.1). Figure 7-11 plots the actual crude oil price and the price from the multi-agents model.

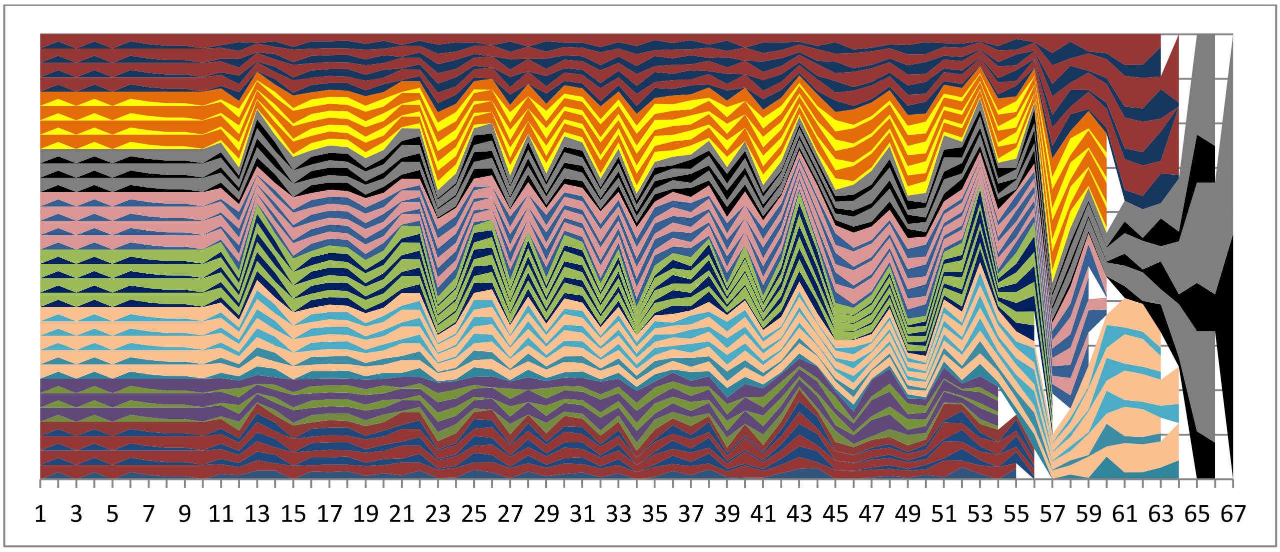

Figure 7-9: [coloured] The connectivity (input output) of each agents.
This figure illustrates the connectivity of each connection_from_node and connection_to_node for each agent. The representation here as an area is just for visual clarity. Agents from the same species are given the same colour code. In each species (from bottom to top) the connection_from_node is given a different colour to connection_to_node. We can see that the connection genes have different sizes and various patterns.

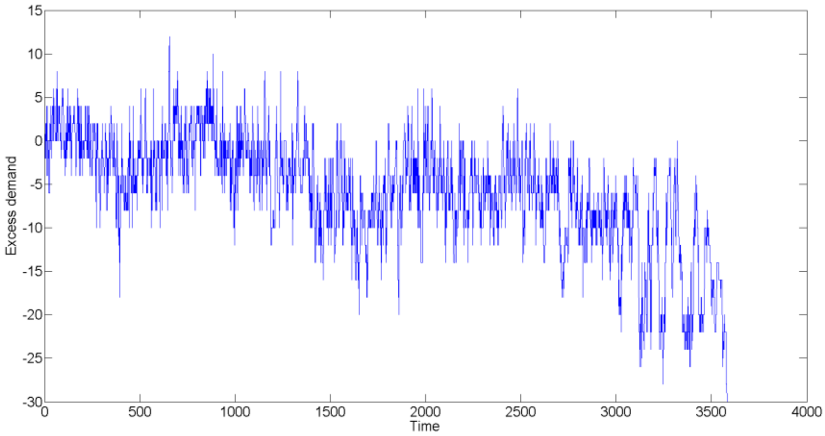

Figure 7-10: The excess demand (supply-demand) for all the agents in the final generation

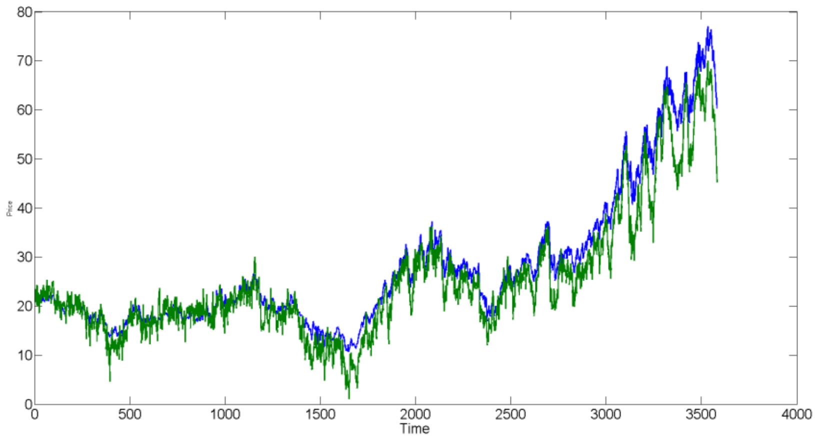

Figure 7-11: The actual crude oil price (blue) and the price derived from agent interaction (green) this plot is Stage 1 only.

7.4.2 **The effect of budget on the impact function**

In this experiment we allocated a budget to agents according to a beta distribution. For a random variable, X will have beta distribution if the pdf is as in Equation (7.3):

$$f(x) = \frac{x^{a-1}(1-x)^{b-1}}{B(a,b)} \tag{7.3}$$

for 0<x<1, a>0 and b>0,

where

$$B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}dt \tag{7.4}$$

For this experiment we used our new algorithm in Box 7.1 below with the budget allocated according to a beta distribution. The values of a and b were varied to change the wealth distribution between agents. We use the same output of agents, changing only the fund distribution for each agent based on the change in the value of a and b in Equation (7.4).

In this system the price change was based on the interaction between trading agents. The algorithm is summarized as follows:

- We allow the agents to train for a number of generations.
- We allocate different amounts of money according to beta distribution (and multiply the outcome by 1000000) for each agent.
- We assume this wealth is either cash or contracts (to facilitate selling).
- We calculate the hit rate for each agent.
- Only agents with a hit rate > 50% are allowed to trade.
- If we do not have more than 51% of the agents with a hit rate above 50%,

the agents are returned to train again (from where they left).

- The price will change as follow:
  - If an agent *i* hit rate >50% and forecast price t+1> 0

THEN place an order to buy with all budgets available

  - If an agent *i* hit rate >50% and forecast price t+1< 0

THEN look at OPEC meetings:

    - Place an order to sell with ½ of the budget available if OPEC production increased and
    - Do not place any order to sell if OPEC production decreased or remained steady.
  - Calculate the number of contracts for buy and sell (budget ÷ contract price).
  - Calculate ε as follows:

$$\varepsilon = \frac{The\ amount\ of\ money\ invested}{Total\ money\ available\ with\ investor} \qquad \text{(b.1)}$$

This concept is similar to the concept of capacity utilisation used in the crude oil market.

  - Calculate the market excess as in Equation (7.1)
  - Calculate the new price as in Equation (7.1) using the value of ε from Equation (b.1) as shifting factor

Box 7.1: The price formation algorithm

The results in Table 7-1 show that the distribution of the budget makes little difference to the final outcome. To eliminate the effect of the small population on the distribution, the same experiment was repeated with a population of 1000 agents. The results are shown in Table 7 2. In conclusion we cannot see any significant impact of changing the fund allocation distribution based on the results for this particular algorithm in Box 7.1.

(a, b)	(.5, .5)	(4, 4)	(2, 4)	(.5, 4)	(1, 1)
Hit rate	47.5771	46.2555	47.2834	47.4302	47.5771
RMSE	0.043385	0.05858	0.051713	0.041953	0.043371
R^2	0.008285	0.007969	0.00813	0.008294	0.008285
R	-0.09102	-0.08927	-0.09017	-0.09107	-0.09102
IC	1.601143	2.161919	1.908486	1.548293	1.600619
TUR	2.337717	3.156466	2.786446	2.260553	2.336952
MSE	0.001882	0.003432	0.002674	0.00176	0.001881
MAE	0.034098	0.046018	0.040627	0.032963	0.034087
SSE	1.281806	2.336902	1.821124	1.198582	1.280967
DA	-1.27006	-1.96176	-1.42348	-1.3455	-1.27006
P value	0.897969	0.975105	0.922702	0.910768	0.897969
AIC	0.00195	0.003555	0.00277	0.001823	0.001948
BIC	-0.72136	-0.65233	-0.68099	-0.72908	-0.72144
Net return	0.036245	0.036245	0.036245	0.036245	0.036245
Sharpe ratio	-0.00085	-0.00109	-0.00101	-0.00081	-0.00085
Rd ratio	28.98358	23.37639	27.09265	29.6634	28.98358

Table 7-1: The results of the multi-agents model using beta distribution to allocate the funds for each agent randomly with a different value of (a, b)

(a,b)	(.5, .5)	(4, 4)	(2, 4)	(.5, 4)	(1, 1)
Hit rate	50.0734	50.0734	50.0734	50.0734	49.9266
RMSE	1.025589	1.106699	1.136863	1.231633	1.229599
R^2	0.003944	0.004297	0.004698	0.006837	0.002463
R	0.062801	0.065555	0.068539	0.082686	0.049627
IC	37.85001	40.84343	41.95667	45.4542	45.37912
TUR	55.26214	59.63262	61.25798	66.36448	66.25486
MSE	1.051832	1.224783	1.292458	1.51692	1.511913
MAE	0.69405	0.759216	0.777967	0.81989	0.834468
SSE	716.2979	834.077	880.1642	1033.022	1029.613
DA	0.032781	0.032781	0.032781	0.035257	-0.04145
P value	0.486925	0.486925	0.486925	0.485938	0.516532
AIC	1.089566	1.268721	1.338825	1.571338	1.566152
BIC	0.005809	0.023308	0.029491	0.047899	0.047519
Net return	-0.03624	-0.03624	-0.03624	-0.03624	-0.03624
Sharpe ratio	0.001838	0.001849	0.001832	0.001738	0.001794
Rd ratio	1.369737	1.369737	1.369737	0.214628	0.478953

Table 7-2: The results of the multi-agents model (with 1000 agents) using beta distribution to allocate the funds for each agent randomly with a different value of (a, b)

7.5 Empirical results

In this section the price formation is approached as an averaging factor for all agents. Here, the output of all agents is fed into a single neural network trained in the supervised learning approach and MSE error function. This is initially planned as the second phase of our multi-agents system and is considered as part of DK incorporation.

7.5.1 Stage (II) no budget

We trained a multi-agents model with 1000 agents for only 13 generations (for time limitation) and the output of all agents was fed into a neural network. Although the in-sample is very good the out-of-sample is still poor. This is clear in the figure below.

	in-sample	out-of-sample
Hit rate	**88.6931**	**52.8689**
RMSE	0.008182	0.051908
R^2	0.814695	8.73E-05
R	0.902605	-0.00934
IC	0.301963	0.997818
TUR	0.440875	1.404393
MSE	6.69E-05	0.002694
MAE	0.005749	0.037676
SSE	0.04559	1.314904
DA	**20.20877**	**1.269194**
P value	0	0.102186
AIC	6.93E-05	0.00283
BIC	-1.10488	-0.90062
Net return	0.036245	0.024245
Sharpe ratio	-0.0111	-0.00865
Rd ratio	95.69552	39.64069

Table 7-3: The results for price formation based on averaging

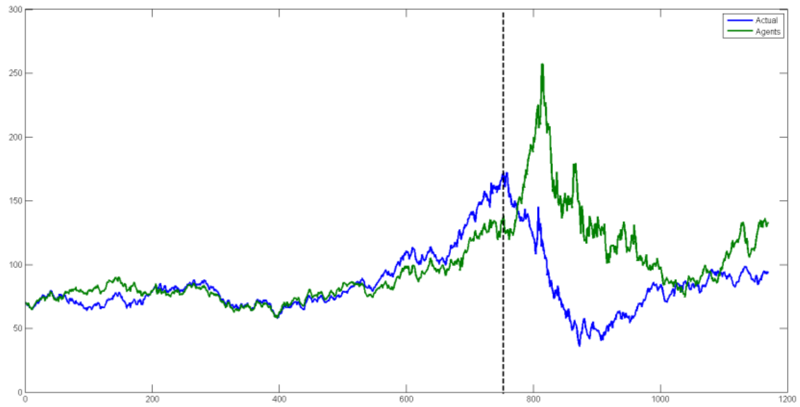

Figure 7-12: The actual price compared to the market price

7.5.2 Stage (II) with the role of budget

In this setting we train agents for a number of generations. After training we calculate the '*new price*' based on the interaction between all agents (as in Box 7.1) which completes stage I. Then, for stage II we train traditional ANN using the 'new price' as input, in addition we use each agent's output as part of the input for the neural network. We can clearly see the improvement in the performance (Table 7-5) and also we can see that the network did not overfit the output.

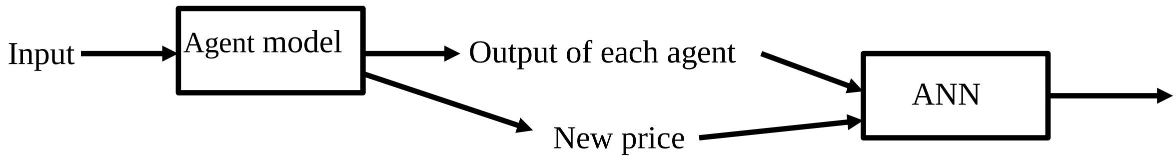

Figure 7-13: This diagram illustrates the way this model was trained.

	in-sample	out-of-sample
Hit rate	60.793	55.3279
RMSE	0.020654	0.052058
R^2	0.147877	0.095064
R	0.384547	0.308324
IC	0.762233	1.000696
TUR	1.112883	1.408443
MSE	0.000427	0.00271
MAE	0.016112	0.040572
SSE	0.290494	1.3225
DA	5.675893	2.635679
P value	6.90E-09	0.004198
AIC	0.000442	0.002847
BIC	-0.892	-0.89975
Net return	0.036245	0.024245
Sharpe	-0.16192	-0.59597
Rd ratio	70.24043	61.56852

Table 7-4: The results for price formation based on averaging with budget

7.5.3 Stage (II) with the role of budget and the initial input

This setting represents the incorporation of hints into a soft-computing model and it is the way we planned to use the multi-agents model in the first place. In this setting a feedforward network trained with the initial input and the output of the agents altogether. Hence, the input from the agents represents hints obtained from a panel of artificially intelligent agents to aid the learning process.

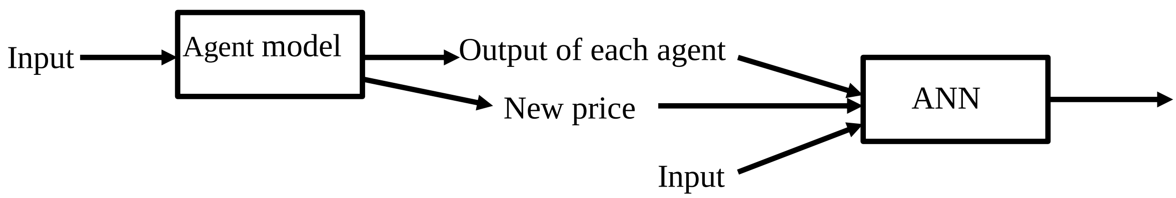

Figure 7-14: This diagram illustrates the way this model was trained.
After the multi-agents model has converged, the forecast of each agent is used as an additional input to train a feedforward neural network.

The results in Table 7-10 shows improvement compared to the results in Table 7-9. The Sharpe ratio for out-of-sample in both Tables 7-9 and 7-10 was negative; however, the ratio is also negative for the actual target as well (using the mean return as the risk-free rate) the Sharpe ratio for actual was -0.0136. This is not surprising as the out-of-sample data includes the unstable period of the global economic crises of 2008.

	in-sample	out-of-sample
Hit rate	62.26138	**58.60656**
RMSE	0.024384	0.044882
R^2	0.116114	0.15027
R	0.340755	0.387647
IC	0.89989	0.862759
TUR	1.313867	1.214303
MSE	0.000595	0.002014
MAE	0.018527	0.034923
SSE	0.404894	0.983039
DA	6.477928	**4.021026**
P value	4.65E-11	2.90E-05
AIC	0.000616	0.002116
BIC	-0.85383	-0.9449
Net return	0.036245	-0.02425
Sharpe	-0.19768	-0.53137
Rd ratio	61.32822	66.79486

Table 7-5: The results for price formation based on hints

7.6 Discussion

The multi-agents model presented in this chapter aimed to show a novel use of two machine learning algorithms, namely ANN and NEAT. The best results achieved in this chapter were by using the output return from multi-agent model as an additional variable (on top of the original input) in traditional ANN. The best hit rate in Table 7-10 was 58%.

The goal of the agents-based model is to imitate the functionality of a market through a group of adaptive agents. This is by definition a very difficult endeavour to achieve for a number of reasons. First, the design of an intelligent agent is still far from human. In this research we relied on NEAT as a base for our model. This gave our agent the ability to evolve over time, in terms of complexity and structure. However, under the current setting agents have to take action at each generation during their evolution in order to assess their fitness function. In the current model the price formation is only allowed after the agents have been evolved. Nevertheless, it might be more realistic if could be evolved using an objective function that does not involve trading, then they can trade based on the way they learnt.

Second, the objective of this model to imitate agents in the crude oil market is only through the information available to agents. In other words, agents are trained based on crude oil-related data and information about OPEC meetings. Beyond that the agents presented in this model can be applied to trade any financial stock or commodity. We argue that this might be an advantage although the model has to be validated using a set of financial data.

Third, the daily data frequency we used leads to an artificial trading strategy. Unfortunately we could not obtain high frequency data for this research. The availability of high frequency data (tick-by tick) could lead to a much more sophisticated and realistic trading strategy that is more close to the market functionality. However, the current model dealing with high frequency data is problematic due to the fact that the genetic algorithm is computationally intensive, i.e., it takes a long time to train agents. The effect of this issue is minimal when dealing with low frequency data, on a daily basis; however, the current model will not be able to generate the forecast for the next tick on time. Another issue is that, throughout this thesis, we did not consider transaction costs. This is very common in similar research as the objective is proof of concept.

Finally, the assumption that the data is available for all agents at the same time might not be realistic. We used fetcher selection NEAT to allow individual agents to connect only the input that improves their objective function.

7.7 Conclusion

In this Chapter we presented a new multi-agents model for the crude oil market based on NEAT. The multi-agents model is a relatively new concept in finance which distinguishes itself from other models by considering both the micro-economic aspect of the market and the macro-economic side (Gorthmann, 2002). Due to the complexity of real-world financial market, in this chapter we tested if the output of our multi-agents system contains new information useful in aiding the traditional ANN learning process. Therefore, we proposed a two-stage model. In the first stage NEAT ran for a number of generations until one of the stopping criteria was met. Each converged network in NEAT represents a trading agent. These agents/networks trade in the crude oil market based on their own forecast buy if the agent predicted the return to be > 0 in the future, and sell otherwise. As such, we were able to calculate the market excess then the price was shifted, based on a market impact function.

Our empirical results showed that when using the output of the multi-agents model to the original input spaces, the traditional ANN produced a superior forecast compared with that obtained by using just the output of the multi-agents model alone as an input, or ANN alone.

These results answer our third research question; a multi-agent model can be used to produce useful information.

7.8 Summary

We presented a novel multi-agents model for crude oil forecasting. The model was based on NEAT. Hence, each converged network from NEAT's population represents a trading agent. The output of the multi-agents model was fed to traditional ANN as a *hint* and the empirical results confirmed its efficacy.

CHAPTER 8: Conclusion

8.1 Introduction

This chapter presents concluding remarks. We reproduce the aim and objectives, a brief summary of the results, the contribution to knowledge and the connection to current literature. Finally, we discuss future research avenues and potential opportunities for improvement.

8.2 Summary of the objectives

The goal of this research is to create tools to improve the short-term direction forecasting of crude oil price and return series. Crude oil was selected as a target application in this thesis for the significant role it plays in the world economy. In Chapters 1 and 2 we discussed in detail the importance and complex dynamics of this commodity.

Our aim is to determine if we are able to use knowledge of the crude oil market, soft-computing methods and statistical inference to extract additional information, and whether the information content of the inputs (hints) is useful in improving forecasting performance. The term DK is used in the literature to describe many things. In this thesis there are four types of DKs organized from the simple to the complex. These are as follows: the problem representation, which includes data transformation, feature selection and noise control (amongst others), and the use of non-financial example and creating additional training examples based on our DK to supplement the lack of information and aid the learning process of ANN. In Chapter 6 we showed how the combination OPEC meeting announcements and wavelet analysis can be used to create an artificial series. In Chapter 3 we introduced a number of new fitness functions (for reinforcement learning) to incorporate DK. Finally, we used the output of an artificial market (multi-agents model) as supplementary input to ANN model. We selected soft-computing methods for this research owing to the complexity of the dynamics. Soft-computing methods have been chosen because they are able to tolerate imperfection.

From an econometric perspective, an ANN is: nonparametric (Grothmann, 2002); a flexible and multivariate group of models, which is able to incorporate a high degree of non-linearity; and able to handle high-dimensional problems. These characteristics make it appropriate for this task, given that we do not have a full description of the system dynamics. As a result, we rely upon a data-driven function generation process.

8.3 Summary of the results

We began by performing a comprehensive analysis on crude oil price dynamics. Our goal was to determine what type of dynamics governs the crude oil series. Specifically, we investigated if there was any non-linear deterministic dynamics (even chaos) which could be misspecified as a random walk. From a statistical perspective, have the dynamics of crude oil returns changed significantly during the past twenty years? To address these points, we applied a number of econometric tests and cross-matched the results. We find strong evidence of non-linear dynamics with a high level of noise governing crude oil prices and returns. Furthermore, we find evidence that these series follow low- dimensional dynamics, i.e., chaos. The implications of these results are: (i) theoretically, we can forecast crude oil returns in the short-term, (ii) long-term forecasting is not possible, and (iii) a non-linear non-parametric model such as ANN is suitable for this type of system. The details of this analysis and the results are presented in Chapter 4.

In Chapter 5 we test how a simple type of DK, the representation of the problem, affects the accuracy of crude oil forecasting. We argue that a better representation based on our DK will make it easier for the model in its generic form to improve the forecast. We tested a number of methods, including smoothing, transformation, changing the frequency (from daily to weekly or monthly) amongst other methods. The results in Chapter 5 showed that some of the methods applied, e.g., smoothing, are effective in improving the forecast accuracy and horizon, while other methods did not generate any significant improvement.

In Chapter 6 we tested the use of supplementary variables. Supplementary variables are modified dummy variables or non-financial time series we used to compensate for the lack of information, thereby aiding the model in selecting a more appropriate hypothesis. Three case studies were performed based on: (i) the number of virtual/dummy data obtained from OPEC meeting announcements, (ii) the search index from Google Insight for Search (non-financial), and (iii) technical analysis (domain specific data). The results show that one of the three OPEC dummy data improved the forecast. Also, some of the Google variables were able to marginally improve the forecasting results. Technical transformations seem to be one successful method of capturing the system dynamics and improving forecast accuracy.

In Chapter 7 we presented our novel multi-agents model as a source of DK. We tested if the output of these agents contains new information and its use would aid the learning process of the traditional black-box model. The results showed that output of our multi-agents model only useful as supplementary input. In other words the output of multi-agents model improved the forecasting accuracy (hit rate) when used in addition to the initial input to train ANN.

8.4 **Future research**

We have concentrated on four types of DKs. Our research in the future will expand to test other types of DK, such as news headline releases as an additional source of information.

In relation to our multi-agents model, currently, the agents evolve independently from each other. The only interaction between agents is through the fitness sharing between and the final trading through the market impact function. In the future we intend to alter the fitness sharing between agents in such a way that, during evolution, those agents compete on financial rather than statistical grounds. To elaborate, agents in each species are allowed to trade (buy and sell) based on their expectations and capital. The profitability of agents will determine the evolution process. Another issue for improvement is the price formation mechanism. In this thesis we used a simple market impact function for a number of reasons as outlined in Chapter 7. However, we anticipate a potential benefit in using a method that mimics the real market.

We have focused upon improving the forecasting performance using DKs. In the future we will deal with applying our forecasting methods into trading and hedging systems. This will provide more practical tools to crude oil market participants.

8.5 **Concluding remarks**

This thesis has narrowed the divide between three interrelated fields: (i) energy economics, (ii) time-series econometrics, and (iii) soft-computing. The primary contribution of this thesis is to the field of energy economics and soft-computing. From the energy economics point of view, we found strong evidence to support the hypothesis that the sign of returns can be successfully forecasted over short horizons. Prior to this research, there was no agreement as to what types of dynamics governed these series and if these dynamics were stable over time. From the soft-computing perspective, we have shown a new way of creating hints using well-established soft-computing methods such as fuzzy logics and wavelet analysis. Moreover, we introduced a novel multi-agents model using an innovative application of NEAT.

References

Abhyankar, A., Copeland, L. S., & Wong, W. (1997). Uncovering nonlinear structure in real-time stock-market indexes: The S&P 500, the DAX, the Nikkei 225, and the FTSE-100. *Journal of Business & Economic Statistics, 15*(1), 1-14.

Abramson, B. (1994). The design of belief network-based systems for price forecasting. *Computers & Electrical Engineering, 20*(2), 163-180.

Abramson, B., & Finizza, A. (1991). Using belief networks to forecast oil prices. *International Journal of Forecasting, 7*(3), 299-315.

Abu-Mostafa, Y. (1990). Learning from hints in neural networks. *Journal of Complexity, 6*, 192-198.

Abu-Mostafa, Y. (1993). Learning from hints in neural networks. In A.-P. Refenes (Ed.), *Neural networks in the capital markets* (pp. 221-232). New York: John Wiley & Sons.

Abu-Mostafa, Y. (1994). Learning from hints. *Journal of Complexity, 10*, 165-178.

Abu-Mostafa, Y. (1995a). Financial market applications of learning from hints. In A.-P. Refenes (Ed.), *Neural networks in the capital markets*. Chichester: John Wiley & Sons.

Abu-Mostafa, Y. (1995b). Hints. *Neural Computation, 7*, 639-671.

Abu-Mostafa, Y. (1995c). Machines that learn from hints. *Scientific American, 272*(4), 64.

Abu-Mostafa, Y. (2001). Financial model calibration using consistency hints. *Neural Networks, IEEE Transactions on, 12*(4), 791-808.

Adrangi, B., Chatrath, A., Dhanda, K. K., & Raffiee, K. (2001). Chaos in oil prices? Evidence from futures markets. *Energy Economics, 23*(4), 405-425.

Al-Qahtani, A., Balistreri, E., & Dahl, C. (2008). Literature review on oil market modeling and OPEC's behavior. Colorado School of Mines.

Anvik, C., & Gjelstad, K. (2010). *Just Google it! forecasting Norwegian unemploymnet figures with web queries.* Unpublished Master thesis, BI Norwegian School of Management.

Askitas, N., & Zimmermann, K. F. (2009). Google econometrics and unemployment forecasting. Unpublished Discussion Paper http://ftp.iza.org/dp4201.pdf. Bonn University.

Azoff, M. E. (1994). *Neural network time series forecasting of financial markets*. Chichester: John Wiley & Sons.

Bailey, T. L., & Elkan, C. (1995). *The value of prior knowledge in discovering motifs with MEME.* Paper presented at the the Third International Conference on Intelligent Systems for Molecular Biology, Cambridge, England.

Barnett, A., & Wolff, R. (2005). A time-domain test for some types of nonlinearity. *IEEE Transactions on Signal Processing, 53*(1), 26-33.

Bonissone, P. P., Subbu, R., Eklund, N., & Kiehl, T. R. (2006). Evolutionary algorithms + domain knowledge = real-world evolutionary computation. *Evolutionary Computation, IEEE Transactions on, 10*(3), 256-280.

Bowden, G. J. (2003). *Forecasting water resources variables using artificial neural networks.* Unpublished PhD thesis, University of Adelaide, Adelaide.

Brock, W. A., Scheinkman, J. A., Dechert, W. D., & LeBaron, B. (1996). A test for independence based on the correlation dimension. *Econometric Reviews, 15*(3), 197-235.

Brooks, C., & Henry, O. T. (2000). Can portmanteau nonlinearity tests serve as general mis-specification test? Evidence from symmetric and asymmetric GARCH model. *Economics Letters, 67*(2000), 245-251.

Buckland, M. (2002). Evolving neural network toplogy. In A. LaMothe (Ed.), *AI techqhniques for game programing*. Cincinnati: Stacy Hiquet.

Caruana, R. (1997). Multitask learning. *Machine Learning, 28*(1), 41-75.
Cheng, B., & Tong, H. (1992). On Consistent Nonparametric Order Determination and Chaos. *Journal of the Royal Statistical Society. Series B (Methodological), 54*(2), 427-449.
Cheong, C. W. (2009). Modeling and forecasting crude oil markets using ARCH-type models. *Energy Policy, 37*(6), 2346-2355.
Choi, H., & Varian, H. (2009). *Predicting the present with Google Trends*. Retrieved from www.google.com/googleblogs/.../google_predicting_the_present.pdf.
Choi, K., & Hammoudeh, S. (2009). Long memory in oil and refined products markets. *The Energy Journal, 30*(2), 21.
Crémer, J., & Salehi-Isfahani, D. (1991). *Models of the oil markets*. Chur, Switzerland: Harwood Academic Publishers.
De Santis, R. A. (2003). Crude oil price fluctuations and Saudi Arabia's behaviour. *Energy Economics, 25*(2), 155-173.
Energy Modeling Forum. (1982). *World oil* (EMF report 6). Stanford: Stanford University.
Engel, C., & Hamilton, J. D. (1990). Long swings in Dollar: Are they in the data and do markets know it? *American Economic Review, 85*(1990), 689-713.
Fan, Y., Liang, Q., & Wei, Y.-M. (2008). A generalized pattern matching approach for multi-step prediction of crude oil price. *Energy Economics, 30*(3), 889-904.
Farmer, J. D., & Joshi, S. (2002). The price dynamics of common trading strategies. *Journal of Economic Behavior & Organization, 49*(2002), 149-171.
Fattouh, B. (2005). The causes of crude oil price volatility. *Middle East Economic Survey, 48*(13), 1-22. Retrieved from http://www.mees.com/postedarticles/oped/v48n13-5OD01.htm
Flake, G. (1998). Square unit augmented radially extended multilayer perceptrons *Neural Networks: Tricks of the Trade* (pp. 552-552).
Frey, G., Manera, M., Markandya, A., & Scarpa, E. (2009). Econometric models for oil price forecasting: A critical survey. *CESifo Forum, 10*(1), 29.
Gately, D. (1984). A ten-year retrospective: OPEC and the world oil market. *Journal of Economic Literature, 22*(3), 1100-1114.
Ghaffari, A., & Zare, S. (2009). A novel algorithm for prediction of crude oil price variation based on soft computing. *Energy Economics, 31*(4), 531-536.
Granger, C., & Hyung, N. (2004). Occasional structural breaks and long memory with an application to the S&P 500 absolute stock returns. *Journal of Empirical Finance, 11*(2004), 399-421.
Grothmann, R. (2002). *multi-agent market modling based on neural networks*. Unpublished PhD Thesis, University of Bremen , Bremen.
Haidar, I. (2008). *Short-term forecasting model for crude oil price based on artificial neural networks*. University of Ballarat, Ballarat.
Hamilton, J. D. (1983). Oil and the macroeconomy since World War II. *The Journal of Political Economy, 91*(2), 228-248.
Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica, 57*(2), 357-384.
Hamilton, J. D. (1990). Analysis of time series subject to changes in regime. *Journal of Econometrics 45*(1990), 39-70.
Hamilton, J. D. (2003). What is an oil shock? *Journal of Econometrics, 113*(2), 363-398.
Hamilton, J. D. (2009). Understanding crude oil prices. *The Energy Journal, 30*(2), 29.
Haykin, S. (1998). *Neural networks: A comprehensive foundation* (Second ed.). Upper Saddle River, New Jersey: Prentice-Hall, Inc.
Hilario, M., & Rida, A. (1997). The use of prior knowledge in neural network configuration and training *Biological and Artificial Computation: From Neuroscience to Technology* (pp. 227-236).
Hinton, G. (1999). Supervised learning in multilayer neural networks. In R. A. Wilson & F. C. Keil (Eds.), *The MIT Encyclopedia of the Cognitive Sciences*. London, England: The MIT Press.

Hinton, G. (2007a). Learning multiple layers of representation. *Trends in Cognitive Sciences, 11*(10), 428-434.
Hinton, G. (2007b). To recognize shapes, first learn to generate images. In P. Cisek, T. Drew & J. Kalaska (Eds.), *Computational Neuroscience: Theoretical Insights into Brain Function* (Vol. 165). Amsterdam: Elsevier.
Hinton, G. (Producer). (2009, 25 September 2011) Deep Belief Networks. *videolectures*. Video Lecture retrieved from http://videolectures.net/mlss09uk_hinton_dbn/
Hinton, G., Osindero, S., & Tech, Y.-W. (2006). A fast learning algorithm for Deep Belief Nets. *Neural Computation*, *18*(2006), 1527-1554.
Ho, Y.-C., & Pepne, D. (2001). *Simple explanation of the no free lunch theorem of optimization*. Paper presented at the 40th IEEE conference on Decision and control.
Hooker, M. A. (1996a). This is what happened to the oil price-macroeconomy relationship: Reply. *Journal of Monetary Economics, 38*(2), 221-222.
Hooker, M. A. (1996b). What happened to the oil price-macroeconomy relationship? *Journal of Monetary Economics*, *38*(2), 195-213.
Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*(5), 359-366.
Huntington, H. G. (1994). Oil price forecasting in the 1980s: What went wrong? *Energy Journal, 15*(2), 1.
Kaastra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. *Neurocomputing, 10*, 215-236.
Kaboudan, M. A. (1995). Measuring the complexity of nonlinearity by a relative index with application to financial time series. In R. Trippi (Ed.), *Chaos and nonlinear dynamics in financial markets: Theory, evidence and applications* (pp. 417-438). Chicago: Irwin Professional Publishing.
Kaboudan, M. A. (1999). Diagnosing chaos by a fuzzy classifier. *Fuzzy Sets and Systems, 108*(1), 1-10.
Kaboudan, M. A. (2001). *Compumetric forecasting of crude oil prices*. Paper presented at the 2001 Congress on Evolutionary Computation.
Kaboudan, M. A. (2005). *Wavelets in multi-step-ahead forecasting*. Paper presented at the the16th IFAC World Congress.
Kang, S. H., Kang, S.-M., & Yoon, S.-M. (2009). Forecasting volatility of crude oil markets. *Energy Economics, 31*(1), 119-125.
Kaufman, S., & Rosset, S. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, *6*(4), 15:11-15:21.
Khazem, H. A. (2007). *Using artificial neural networks to forecast the futures price of crude oil.* Unpublished Doctor of Business Administration, Nova Southeastern University, Broward County.
Kothari, S. P., & Warner, J. B. (2007). Econometrics of event studies. In B. E. Eckbo (Ed.), *Handbook of corporate finance* (Vol. 1). Hanover: North-Holland.
Kuremoto, T., Obayshi, M., & Kobayashi, K. (2005). Nonlinear prediction by reinforcemnet learning. In D. S. Huang, X.-P. Zhang & G. B. Huang (Eds.), *ICIC 2005, Part I, LNCS* (pp. 1085-1094). Berlin Heidelberg: Springer-Verlag.
Labonte, M. (2004). *The effect of oil shocks on the economy: A review of the empirical evidence*. Retrieved from http://www.ncseonline.org/NLE/CRSreports/04Jun/RL31608.pdf.
Lackes, R., Börgermann, C., & Dirkmorfeld, M. (2009). Forecasting the price development of crude oil with artificial neural networks *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living* (pp. 248-255).
Lai, D., & Chen, G. (1998). Statistical analysis of Lyapunov exponents from time series: A Jacobian approach. *Mathematical and Computer Modelling, 27*(7), 1-9.

Lean, Y., Shouyang, W., & Kin Keung, L. (2008, 12-15 Oct. 2008). *A generalized Intelligent-agent-based fuzzy group forecasting model for oil price prediction.* Paper presented at the IEEE International Conference on Systems, Man and Cybernetics,(SMC 2008).

Lee, K., & Ni, S. (1995). Oil shocks and the macroeconomy: The role of price variability. *Energy Journal, 16*(4), 39.

Levenberg, K. (1944). A Method for the Solution of Certain Non-Linear Problems in Least Squares. *Quarterly of Applied Mathematics, 2,* 164-168.

Levy, M. (2011). Agent Based Computational Economics. In R. A. Meyers (Ed.), *Complex Systems in Finance and Econometrics* (pp. 1-21): Springer New York.

Lin, S., & Tamvakis, M. (2009). OPEC announcements and their effects on crude oil prices. *Enrgy Policy 38,* 1010-1016.

Liu, D., & Zhang, L. (2010). *China Stock Market regimes prediction with artifcial neural networks and Markov regime switching*. Paper presented at the The World Congress on Engineering

Liu, F., Quek, C., & Ng, G. S. (2005). *Neural networks model for time series prediction by reinforsmnet learning.* Paper presented at the The intrnational Joint Conferecnce on Neural Networks, Canada.

Liu, J., Bai, Y., & Li, B. (2007). *A new approach to forecast crude oil price based on fuzzy neural network*. Paper presented at the FSKD '07: Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery.

Mabro, R. (1998). OPEC behaviour 1960-1998: A review of the literature. *The journal of Energy Literature, 4*(1), 3-27.

Marquardt, D. (1963). An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *SIAM Journal on Applied Mathematics, 11*(2), 431.

Matilla-Garrcia, M. (2007). Nonlinear Dynamics in Energy Futures. *Energy Journal, 28*(3), 7-29.

Miikkulainen, R. (2010). Neuroevolution *Encylopdia of Machine Learning*. New York: Springer.

Miikkulainen, R., Bryant, B., Cornelius, R., Karpov, I., Stanley, K. O., & Young, C. H. (2006). Computaional inteligence in games. In D. B. Yen G. and Fogel (Ed.), *Computational intelligence: Principles and practice*: IEEE Computational Intellignce Society.

Misity, M., Misity, Y., Oppenheim, G., & Poggi, J.-M. (1997). Wavelet toolbox 4: The MathWorks.

Mohamed, A., Hinton, G., & Penn, G. (2012). *Understanding how Deep Belief Networks perform acoustic modelling.* Paper presented at the Acoustics, Speech and Signal Processing, Kyoto, Japan.

Moody, J., Wu, L., Liao, Y., & Saffell, M. (1998). Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting, 17*(5-6), 441-470.

Mork, K. A. (1989). Oil and the macroeconomy when prices go up and down: An extension of Hamilton's results. *The Journal of Political Economy, 97*(3), 740-744.

Moshiri, S., & Foroutan, F. (2006). Forecasting nonlinear crude oil futures prices. *Energy Journal, 27*(4), 81-95.

Neuneier, R., & Zimmermann, H. (1998). How to Train Neural Networks *Neural Networks: Tricks of the Trade* (pp. 550-550).

Palmer, R., Arthur, B., Holland, J., LeBorne, B., & Tayler, P. (1994). Artifical economic life: A simle model of stock market. *Physica A: Statistical Mechanics and its Applications, D*(17), 264-274.

Pan, H., Haidar, I., & Kulkarni, S. (2009). Daily prediction of short-term trends of crude oil price using neural networks exploiting multimarket dynamics. *Frontiers of Computer Science in China, 3*(3), 15.

Perron, P. (1989). The great crash, the oil price shock and the unit root hypothesis. *Econometrica, 57*(1989), 1361-1401.

Pesaran, H., & Timmerman, A. (1992). A Simple nonparametric test of predictive performance. *Journal of Business & Economic Statistics, 10*(4), 461-465.

Refenes, A. (1995). *Neural networks in the capital markets*. New York: John Wiley & Sons.

Refenes, A., Bentz, Y., Bunn, D. W., Burgess, A. N., & Zapranis, A. D. (1997). Financial time series modelling with discounted least squares backpropagation. *Neurocomputing, 14*(2), 123-138.

Ruggeri, G. C. (1983). Market conditions and future oil prices. *Energy Economics, 5*(3), 190-194.

Sadorsky, P. (2000). The empirical relationship between energy futures prices and exchange rates. *Energy Economics, 22*(2), 253-266.

Sanders, D. R., Manfredo, M. R., & Boris, K. (2009). Evaluating information in multiple horizon forecasts: The DOE's energy price forecast. *Energy Economics, 31*(2009), 8.

Sarikaya, R., Hinton, G., & Deoras, A. (2014). Application of Deep Belief Networks for natural language understanding, *IEEE Transactions on Audio, Speech and Language Processing, http://www.cs.toronto.edu/~hinton/papers.html*.

Satchell, S., & Timmermann, A. (1995). An Assessment of the economic value of non-linear foreign exchange rate forecast. *Journal of Forecasting, 14*(1995), 477-497.

Schmidbaue, H., & Rosch, A. (2012). OPEC news announcements: Effects on oil price expectation and volatility. *Energy Economics*.

Schmidbauer, H., & Rosch, A. (2012). OPEC news announcements: Effects on oil price expectation and volatility. *Emergy Economics, 34*(2012), 1656-1663.

Schweitzer, F. (2002). *Modeling complexity in economic and social systems*. New Jersey: World Scientific Publishing Co.

Sims, C. A. (1980). Macroeconomics and Reality. *Econometrica, 48*(1), 1-48.

Stanley, K. O. (2004). *Efficient Evolution of neural networks through complexification.* Unpublished PhD Thises, The University of Texas, Austin.

Stanley, K. O., & Miikkulainen, R. (2002). Evolving neural networks through augmenting topologies. *Evolutionary computation, 10*(2), 99-127.

Steurer, E. (1995). Nonlinear modelling of the DEM/USD exchange rate. In A.-P. Refenes (Ed.), *Neural networks in the capital markets*. Chichester: John Wiley & Sons.

Suddarth, S., & Holden, A. (1991). Symbolic-neural systems and the use of hints for developing complex systems. *Journal of Man-Machine Studies* (35), 291-311.

Suddarth, S., & Kergosien, Y. (1990). Rule-injection hints as a means of improving network performance and learning time *Neural Networks* (pp. 120-129).

Sutskever, I., & Hinton, G. (2010). Temporal-kernel recurrent neural networks. *Neural Networks, 23*(2), 239-243.

Tan, C. (2001). *Artificial neural networks: Applications in financial distress prediction and foreign exchange trading*. Gold Coast: Wilberto Press.

Tao, D., & Hongfei, X. (2007). *Chaotic time series prediction based on radial basis function network.* Paper presented at the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing.

Tilakaratne, C., Mammadov, M., & Morris, S. (2008). Predicting trading signals of stock market indices using neural networks *AI 2008: Advances in Artificial Intelligence* (pp. 522-531).

Tilakaratne, C. D., Mammadov, M. A., & Morris, S. A. (2009). Modified neural network algorithms for predicting trading signals of stock market indices. *Journal of Applied Mathematics and Decision Sciences, 2009*.

Timmermann, A. (2001). Structural breaks,incomplete information and stock prices. *Journal of Business & Economic Statistics, 19*(3), 299-314.

Towell, G. G., & Shavlik, J. W. (1994). Knowledge-based artificial neural networks. *Artificial Intelligence, 70*(1-2), 119-165.

Valentinyi-Endrész, M. (2004). Structural breaks and financial risk management *Magyar Nemzeti Bank Working Paper,* (11), 1-60

Vanstone, B. (2005). *Trading in the Australian stockmarket using artificial neural networks.* Unpublished PhD thesis, Bond University.

Vanstone, B., & Finnie, G. (2010). Enhancing stockmarket trading performance with ANNs. *Expert Systems with Applications 37*(2010), 6602-6610.

Vo, M. T. (2009). Regime-switching stochastic volatility: Evidence from the crude oil market. *Energy Economics, 31*(5), 779-788.
Wang, S., Yu, L., & Lai., K. K. (2005). Crude oil price forecasting with TEI@I methodology. *Journal of Systems Science and Complexity, 18*(2), 22.
Whiteson, S., Stone, P., Stanley, K., Miikkulainen, R., & Kohl, N. (2005). *Automatic Feature Selection in Neuroevoulution.* Paper presented at the Genetic and Evolutionary Computation Conference (GECCO 05), Washington D. C.
Wiewiora, E., Cottrell, G., & Elkan, C. (2003). *Principled methods for advising reinforcement learning agents.* Paper presented at the the Twentieth International Conference on Machine Learning, Washington DC.
Winebrake, J. J., & Sakva, D. (2006). An evaluation of errors in US energy forecasts: 1982-2003. *Energy Policy, 34*(18), 3475-3483.
Wolff, R. C. (1995). A Poisson Distribution for the BDS Test Statistic for Independence in a Time Series. In H. Tong (Ed.), *Chaos and forecasting* (pp. 109-127). London: World Scientific.
Wolff, R. C., & P. M. Robinson. (1994). Independence in time series: Another look at the BDS test [and Discussion]. *Philosophical Transactions: Physical Sciences and Engineering, 348*(1688), 383-395.
Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on, 1*(1), 67-82.
Wu, L., & Brynjolfsson, E. (2009). The future of prediction: How Google searches foreshadow housing prices and sales.
Xie, W., Yu, L., Xu, S., & Wang, S. (2006). A new method for crude oil price forecasting based on support vector machines. In V.N. Alexanderov & e. al. (Eds.), *Lecture notes in computer science* Heidelberg: Springer.
Yang, B. (2004). *OPEC behavior* . Unpublished PhD thesis. The Pennsylvania State University, Pennsylvania.
Yao, J., North, P., & Tan, C. L. (2001). *Guidelines for financial forecasting with neural networks*. Paper presented at the International Conference on Neural Information Processing.
Yao, J., & Tan , C. L. (2001). *Guidelines for financial forecasting with neural networks* Paper presented at the International Conference on Neural Information Processing Retrieved from http://www.cs.uregina.ca/~jtyao/Papers/guide_iconip01.pdf
Ye, M., Zyren, J., Blumberg, C., & Shore, J. (2009). A Short-Run Crude Oil Price Forecast Model with Ratchet Effect. *Atlantic Economic Journal, 37*(1), 37-50.
Ye, M., Zyren, J., & Shore, J. (2005). A monthly crude oil spot price forecasting model using relative inventories. *International Journal of Forecasting, 21*(3), 491-501.
Ye, M., Zyren, J., & Shore, J. (2006). Forecasting short-run crude oil price using high- and low-inventory variables. *Energy Policy, 34*(17), 2736-2743.
Yu, L., Lai, K., Wang, S., & He, K. (2007). Oil Price Forecasting with an EMD-Based Multiscale Neural Network Learning Paradigm *Computational Science – ICCS 2007* (pp. 925-932).
Yu, L., Wang, S., & Lai, K. K. (2008). Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. *Energy Economics, 30*(5), 2623-2635.
Zadeh, L. A. (1994). *Fuzzy logic and soft computing: Issues, contentions and perspectives*. Paper presented at the In Proc. IIZUKA'94: Third Int. Conf. Fuzzy Logic, Neural Nets Soft Computing.
Zimmermann, G. (2010). *Advances in forecasting with neural networks system identification, forecasting, control*. Paper presented at the Methodological and empirical advances in financial analysis University of Sydney, Sydney Australia. Retrieved from http://www.forecasters.org/isf/pdfs/Zimmerman-Workshop.pdf
Zimmermann, G., Grothmann, R., Schäfer, A. M., & Tietz, C. (2005). Modeling large dynamical systems with dynamical consistent neural networks. In P. Haykin, Sejnowski, and McWhirter (Ed.), *New directions in statistical signal processing: From systems to brain*. Cambridge: The MIT Press.

Zimmermann, H. G., Neuneier, R., & Grothmann, R. (2001). Multi-agent modeling of multiple FX-markets by neural networks. *Neural Networks, IEEE Transactions on*, *12*(4), 735-743.
Zivot, E., & Andrews, D. (1992). Further evidence on the great crash, the oil-Price shock, and the unit-root hypothesis. *Journal of Business & Economic Statistics, 10*(3), 251-270.

Appendices

10.1 **Appendix I: Kaboudan Signal to noise ratio**

Dependent variable:	Omega		
Estimation Method	Least Square		
R Bar**2	0.99		
Regression F Statistic	16528		
Durbin-Watson Statistic	1.76		
Q Statistic	249.52		
Variable	Coefficient	Std. Error	t-Stat
Constant	17.746	0.089	199.86
Theta	-13.545	0.132	-102.45
DV0	-6.200	0.089	-69.97
DV1	-4.524	0.070	-64.33
DV2	-2.120	0.067	-31.68
DV3	-1.077	0.063	-16.99
DV30	1.608	0.070	22.90
DV40	2.705	0.71	38.04
DV50	3.613	0.071	50.54

Table 10-1 Regression to estimation Omega
Source: Kaboudan (1995), p.427

IF	Then Value of dummy variable is						
Theta range	DV0	DV1	DV2	DV3	DV30	DV40	Dv50
0.99=or<θ	1	0	0	0	0	0	0
0.95=or< θ < 0.99	0	1	0	0	0	0	0
0.90=or< θ <0.95	0	0	1	0	0	0	0
0.85=or < θ <0.9	0	0	0	1	0	0	0
0.33=or < θ < 0.34	0	0	0	0	1	0	0
0.32=or < θ < 0.33	0	0	0	0	0	1	0
θ < 0.32	0	0	0	0	0	0	1
All other values	0	0	0	0	0	0	0

Table 10-2 Dummy variable for each range of θ
Source: Kaboudan (1995), p. 428

10.1.1 **Diagnostic tests**

		Price I	Price II	Return I	Return II
N	Statistic	3097	3096	3096	3095
Minimum	Statistic	10.25	10.82	-0.41	-0.17
Maximum	Statistic	41.07	145.31	0.19	0.16
Mean	Statistic	19.4244	45.9229	-0.0001	0.0005
Std. Deviation	Statistic	3.61216	26.82036	0.02538	0.02699
Variance	Statistic	13.048	719.332	0.001	0.001
Skewness	Statistic	1.562	1.086	-1.506	-0.198
	Std. Error	0.044	0.044	0.044	0.044
Kurtosis	Statistic	6.169	0.948	27.848	4.488
	Std. Error	0.088	0.088	0.088	0.088

Table 10-3 Descriptive statistics of crude oil prices I, II and returns I, II

	ADF test			PP		
Lag	**P value**	**T stat**	**C value**	**P value**	**T stat**	**C value**
1	0.2323	-2.7280	-3.9634	0.2323	-2.7280	-3.9634
2	0.2892	-2.6130	-3.9634	0.2641	-2.6638	-3.9634
3	0.3501	-2.4901	-3.9634	0.2940	-2.6034	-3.9634
4	0.3195	-2.5518	-3.9634	0.2986	-2.5941	-3.9634
5	0.2838	-2.6240	-3.9634	0.2938	-2.6038	-3.9634
6	0.3625	-2.4651	-3.9634	0.3059	-2.5793	-3.9634
7	0.3600	-2.4700	-3.9634	0.3124	-2.5663	-3.9634
8	0.3804	-2.4290	-3.9634	0.3187	-2.5535	-3.9634

Table 10-4 Unit root test for crude oil daily spot price from Jan 1986-Feb 2010

There is no significant statistical evidence to reject the null hypothesis of unit root in the oil spot price for 1% significance

	ADF test			PP		
Lag	**P value**	**T stat**	**C value**	**P value**	**T stat**	**C value**
1	0.0013	-4.5762	-3.9669	0.0013	-4.5762	-3.9669
2	0.0013	-4.5735	-3.9669	0.0015	-4.5589	-3.9669
3	0.0028	-4.4125	-3.9669	0.0018	-4.5199	-3.9669
4	0.0096	-3.9843	-3.9669	0.0031	-4.3859	-3.9669
5	0.0088	-4.0180	-3.9669	0.0037	-4.3143	-3.9669
6	0.0141	-3.8608	-3.9669	0.0043	-4.2482	-3.9669
7	0.0255	-3.6616	-3.9669	0.0049	-4.1818	-3.9669
8	0.0213	-3.7234	-3.9669	0.0057	-4.1465	-3.9669

Table 10-5 Unit root test for crude oil daily spot price from Jan 1986 - end of Jan 1998

The ADF test shows that there is significant statistical evidence to reject the null hypotheses of a unit root for lags from 1 to 5 but not from 6 to 8 at 1% significance, while the results of PP test reject the null hypotheses of a unit root for all lags at a significance rate of 1%. For a 5% significance rate (results not shown here) both tests reject the null hypothesis of a unit root for all lags i.e., the crude oil spot price for this subsection was id(0).

	ADF test			PP		
Lag	**P value**	**T stat**	**C value**	**P value**	**T stat**	**C value**
1	0.2903	-2.6106	-3.9669	0.2903	-2.6106	-3.9669
2	0.3627	-2.4644	-3.9669	0.3251	-2.5404	-3.9669
3	0.4216	-2.3454	-3.9669	0.3566	-2.4767	-3.9669
4	0.3674	-2.4549	-3.9669	0.3540	-2.4819	-3.9669
5	0.3243	-2.5419	-3.9669	0.3438	-2.5026	-3.9669
6	0.4052	-2.3786	-3.9669	0.3533	-2.4833	-3.9669
7	0.3918	-2.4057	-3.9669	0.3563	-2.4773	-3.9669
8	0.4189	-2.3509	-3.9669	0.3607	-2.4684	-3.9669

Table 10-6 Unit root test for crude oil daily spot price from 28 Jan 1998 to the end of Feb 2010

There is no significant statistical evidence to reject the null hypothesis of a unit root in the oil spot price for 1% significance

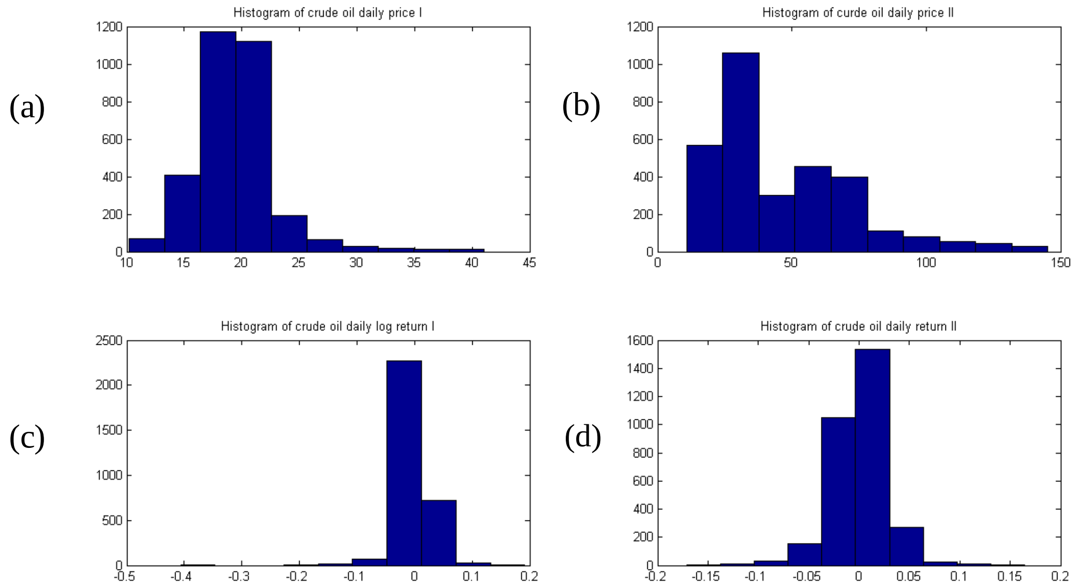

Figure 10-1 a) Histogram of crude oil price I, (b) histogram of crude oil price II, (c) histogram of crude oil return I, (d) histogram of crude oil return II

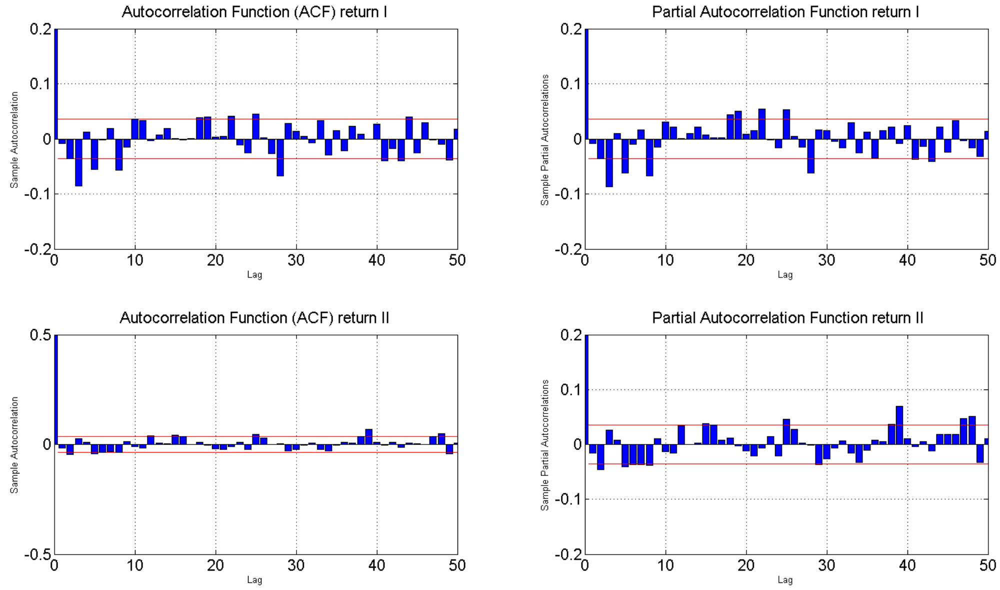

Figure 10-2 Autocorrelation function and partial correlation function for crude oil return I (upper), and crude oil return II (bottom)

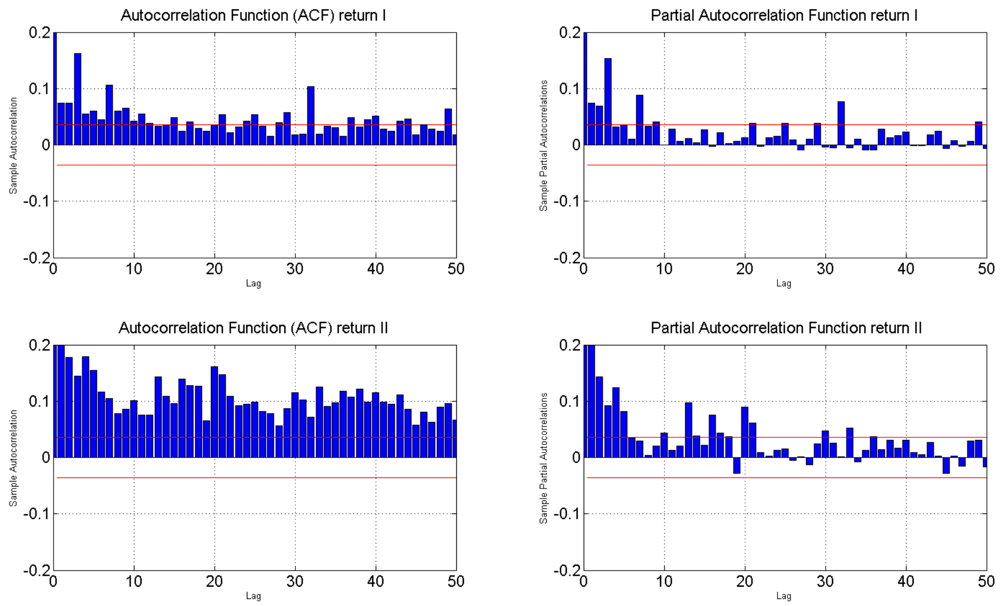

Figure 10-3 Autocorrelation function and partial correlation function for crude oil squared return I (upper), and crude oil squared return II (bottom)

10.2 Appendix II: Google experiment and financial data

10.3 The correlation coefficient

Phrase	Correlation I	Correlation coefficient II	Correlation coefficient III
War	-0.50606	-0.04963	-0.03828
OPEC	-0.20659	0.001819	-0.12136
Iran	-0.14825	-0.0371	0.039452
Iraq	-0.37678	0.024369	0.024102
Saudi	-0.28056	-0.05331	-0.00304
Speculation	0.188684	-0.00897	-0.01733
Cold Weather	-0.12186	-0.00898	0.001117
Supply	0.037813	-0.05657	0.025286
Petrol	0.514582	-	-
Petrol price	0.517232	-	-
Iran sanctions	0.475502	-	-
GFC	-0.14101	-	-
Middle East	-0.37709	-	-
Crude oil	0.33213	-	-
WTI	0.023601	-0.0125	0.030929
NYMEX crude oil price	-0.00465	-0.00465	-0.00465
UK petrol price	0.03227	-0.02042	0.022712
Growth China	0.033215	-0.01486	0.024559
Coal price	0.002957	0.002957	-0.04307

Table 10-7 Correlation coefficient for each of the phrases from Google
Correlation I is the correlation coefficient between each phrase and the crude oil weekly return; correlation II is the coefficient between each phrase and the crude oil weekly return

10.4 Unit root tests

ADF test	War	Dickey-Fuller unit root test based on AR model with drift				
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	1
P value	0.005665	0.001	0.002832	0.001	0.001	0.003053
T stat	-3.64492	-4.22058	-3.92361	-4.2312	-4.13699	-3.89872
Critical value	-3.45507	-3.4552	-3.45534	-3.45548	-3.45562	-3.45576
Significance 1%						
ADF test	**Supply**	**Dickey-Fuller unit root test based on AR model with drift**				
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	0	0	0	0
P value	0.001	0.001	0.025904	0.020509	0.049238	0.103731
T stat	-5.28336	-5.02308	-3.12467	-3.20965	-2.87721	-2.55628
Critical value	-3.45507	-3.4552	-3.45534	-3.45548	-3.45562	-3.45576
ADF test	**OPEC**	**Dickey-Fuller unit root test based on AR model with drift**				
lags	**2**	**4**	**6**	**8**	**10**	**12**

H	1	1	0	1	1	1
P value	0.001	0.001456	0.013001	0.00899	0.009015	0.008153
T stat	-5.43409	-4.08171	-3.37362	-3.49974	-3.49876	-3.53664
Critical value	-3.45507	-3.4552	-3.45534	-3.45548	-3.45562	-3.45576

ADF test	**Middle East**	**Dickey-Fuller unit root test based on AR model with drift**				
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	0	0	0	0
P value	0.001	0.002227	0.010906	0.026604	0.050923	0.053002
T stat	-4.52969	-3.99305	-3.43067	-3.11533	-2.86358	-2.84756
Critical value	-3.45507	-3.4552	-3.45534	-3.45548	-3.45562	-3.45576

ADF test	**speculation**	**Dickey-Fuller unit root test based on AR model with drift**				
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	1
P value	0.001	0.001	0.001	0.001	0.001	0.004993
T stat	-4.87048	-4.24722	-4.20793	-4.21105	-4.21067	-3.67556
Critical value	-3.45507	-3.4552	-3.45534	-3.45548	-3.45562	-3.45576

ADF test	**Iran**	**Dickey-Fuller unit root test based on AR model with drift**				
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	0
P value	0.001	0.001	0.001	0.001766	0.004329	0.025082
T stat	-6.02371	-5.14239	-4.58909	-4.04638	-3.75176	-3.13585
Critical value	-3.45507	-3.4552	-3.45534	-3.45548	-3.45562	-3.45576

ADF test	**Iraq**	**Dickey-Fuller unit root test based on AR model with drift**				
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	0	0	0
P value	0.001	0.003692	0.009518	0.018439	0.043192	0.089791
T stat	-4.93035	-3.82458	-3.47647	-3.24956	-2.93001	-2.6217
Critical value	-3.45507	-3.4552	-3.45534	-3.45548	-3.45562	-3.45576

ADF test	**Saudi**	**Dickey-Fuller unit root test based on AR model with drift**				
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	0	0	0	0
P value	0.001	0.003534	0.03544	0.057333	0.080252	0.092841
T stat	-4.77483	-3.8427	-3.00678	-2.81522	-2.67147	-2.60655
Critical value	-3.45507	-3.4552	-3.45534	-3.45548	-3.45562	-3.45576

ADF test	**Petrol**	**Dickey-Fuller unit root test based on AR model with drift**				

lags	2	4	6	8	10	12
H	1	1	1	1	1	1
P value	0.001	0.001	0.004495	0.004146	0.007747	0.005369
T stat	-4.75834	-4.16035	-3.73244	-3.77268	-3.55429	-3.65861
Critical value	-3.45507	-3.4552	-3.45534	-3.45548	-3.45562	-3.45576

ADF test	Petrol Price	Dickey-Fuller unit root test based on AR model with drift				
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	0	0
P value	0.001	0.001408	0.004514	0.005024	0.011116	0.010334
T stat	-4.76086	-4.08723	-3.73019	-3.67342	-3.42522	-3.44664
Critical value	-3.45507	-3.4552	-3.45534	-3.45548	-3.45562	-3.45576

ADF test	Cold weather	Dickey-Fuller unit root test based on AR model with drift				
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	1
P value	0.001	0.001	0.001835	0.003604	0.002743	0.001
T stat	-4.38911	-4.23006	-4.03825	-3.83501	-3.93416	-4.17528
Critical value	-3.45507	-3.4552	-3.45534	-3.45548	-3.45562	-3.45576

ADF test	Oil	Dickey-Fuller unit root test based on AR model with drift				
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	0	0	0	0	0	0
P value	0.369647	0.256134	0.199283	0.079992	0.046283	0.020568
T stat	-1.83495	-2.09213	-2.22142	-2.67285	-2.90243	-3.20886
Critical value	-3.45507	-3.4552	-3.45534	-3.45548	-3.45562	-3.45576

ADF test	Oil Price	Dickey-Fuller unit root test based on AR model with drift				
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	0	0	0	0	0	0
P value	0.044238	0.178691	0.229453	0.254488	0.310983	0.193222
T stat	-2.92025	-2.28203	-2.15258	-2.09585	-1.96783	-2.23926
Critical value	-3.45507	-3.4552	-3.45534	-3.45548	-3.45562	-3.45576

Table 10-8 The Phillips-Perron test

Phillips-Perron	significant 1%					
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	1
P value	0.003663	0.003132	0.002695	0.002944	0.003183	0.00365
T stat	-3.82764	-3.88863	-3.93884	-3.91027	-3.88275	-3.82907
Critical value	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493

Phillips-Perron

lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	1
P value	0.001	0.001	0.001	0.001	0.001	0.001
T stat	-10.1085	-10.6347	-11.2137	-11.7408	-12.2089	-12.6434
Critical value	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493

Phillips-Perron **OPEC**

lags	**2**	**4**	**6**	**8**	**10**	**12**
	1	1	1	1	1	1
H	0.001	0.001	0.001	0.001	0.001	0.001
P value	-7.66706	-7.74497	-7.98562	-8.34961	-8.63705	-8.82388
T stat	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493
Critical value						

Phillips-Perron **Middle East**

lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	1
P value	0.001	0.001	0.001	0.001	0.001	0.001
T stat	-5.40574	-5.23558	-5.17728	-5.13557	-5.18852	-5.27949
Critical value	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493

Phillips-Perron **Speculation**

lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	1
P value	0.001	0.001	0.001	0.001	0.001	0.001
T stat	-9.01128	-9.52989	-9.99032	-10.4103	-10.7373	-10.9707
Critical value	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493

Phillips-Perron **Iran**

lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	1
P value	0.001	0.001	0.001	0.001	0.001	0.001
T stat	-7.51522	-7.17691	-7.12678	-7.11119	-7.18564	-7.21583

Critical value	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493

Phillips-Perron	**Iraq**					
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	1
P value	0.001	0.001	0.001	0.001	0.001	0.001
T stat	-6.31331	-6.23421	-6.40853	-6.61294	-6.78835	-6.95105
Critical value	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493

Phillips-Perron	**Saudi**					
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	1
P value	0.001	0.001	0.001	0.001	0.001	0.001
T stat	-7.83201	-8.16066	-8.51904	-8.95352	-9.354	-9.72699
Critical value	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493

Phillips-Perron	**Petrol**					
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	1
P value	0.001	0.001	0.001	0.001	0.001	0.001
T stat	-5.7003	-5.56559	-5.65428	-5.77612	-5.89956	-6.00095
Critical value	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493

Phillips-Perron	**Petrol Price**					
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	1
P value	0.001	0.001	0.001	0.001	0.001	0.001
T stat	-5.20601	-5.066	-5.03631	-5.06842	-5.0967	-5.13551
Critical value	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493

Phillips-Perron	**Cold weather**					
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	1	1	1	1	1	1
P value	0.001	0.001	0.001	0.001	0.001	0.001

T stat	-6.30108	-6.47279	-6.71464	-6.92182	-7.1009	-7.24541
Critical value	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493
Phillips-Perron	**Oil**					
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	0	0	0	0	0	0
P value	0.408382	0.370393	0.337831	0.306245	0.271501	0.238828
T stat	-1.74719	-1.83326	-1.90704	-1.9786	-2.05732	-2.13135
Critical value	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493
Phillips-Perron	**Oil Price**					
lags	**2**	**4**	**6**	**8**	**10**	**12**
H	0	0	0	0	0	0
P value	0.013561	0.026783	0.028822	0.028208	0.022747	0.017698
T stat	-3.35801	-3.11271	-3.08538	-3.09361	-3.17309	-3.26419
Critical value	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493	-3.45493

Table 10-9 Augmented Dickey-Fuller test

10.5 Nonlinearity test (FCS test)

Data set	Fitted ARIMA	R^2	θ	Decision
Oil Price WTI	(3,1,8)	0.98	0.098	SL-WN
War	(0,1,2)	0.86	0.94	SL-NL-HN
OPEC	(1, 0, 2)	0.47	0.95	WL-NL-HN
Supply	Simple	0.37	0.88	WL-NL-HN
Iran	(3, 0, 0)	0.58	0.9	FL-NL-HN
Iraq	(0, 1, 1)	0.57	1.07	FL-WN
Saudi	(0,1,2)	0.48	0.95	WL-NL_HN
GFC*	(0, 1, 11)	0.94	0.08	SL-NL
Petrol-Price	(1,0,0)	0.71	0.72	FL-NL-MN
Petrol	(0,1,2)	0.64	0.89	FL-NL-MN
Cold Weather	(0,1,1)	0.55	0.98	FL-WN
Speculation	Simple	0.39	0.92	WL-NL-HN
Middle East	(0,1,2)	0.69	0.91	FL-NL-HN
Crude oil (phrase)	(0,1,12)	0.86	0.81	SL-NL-HN
Iran sanctions	Simple	0.46	0.28	WL-NL
WTI price	Simple	0.595	0.93	FL-NL-HN
NYMEX oil price	Simple	0.609	0.27	SL-NL
UK petrol price	Simple	0.596	0.79	FL-NL-MN
Growth GDP China	(0,1,3)	0.342	0.93	WL-NL-HN
Coal price	Simple	0.632	0.86	FL-NL-HN

Table 10-10 The FCS test on Google data

10.6 **NARX raw price forecast plots**

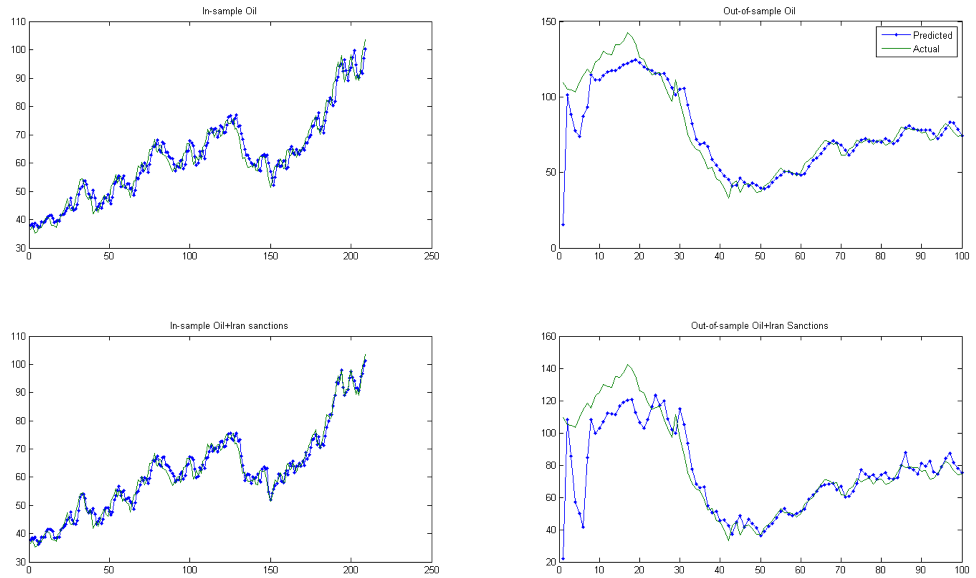

Figure 10-4 One-step-ahead forecast for crude oil price (benchmark) and Iran (bottom)

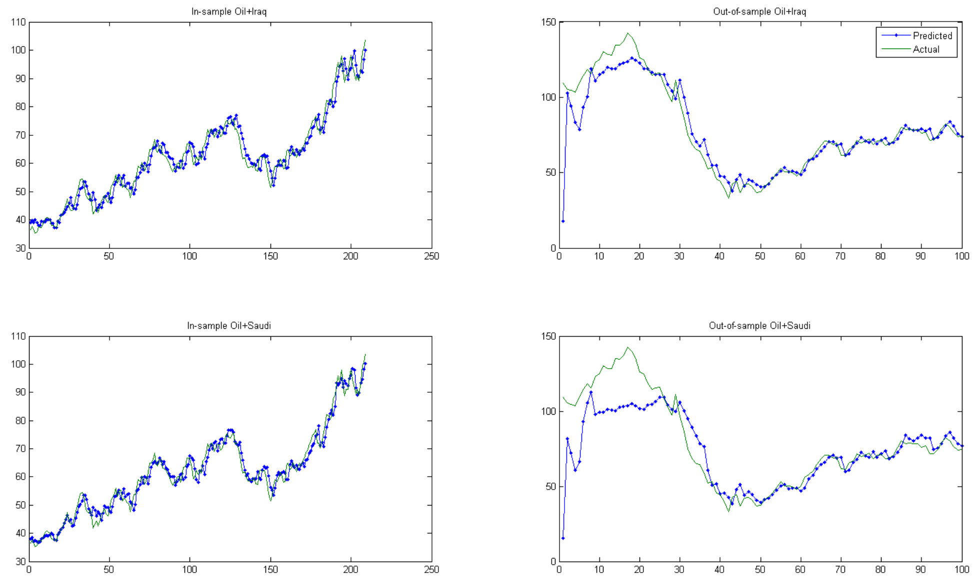

Figure 10-5 One-step-ahead forecast for oil price and Iraq (upper) and oil price and Saudi (bottom)

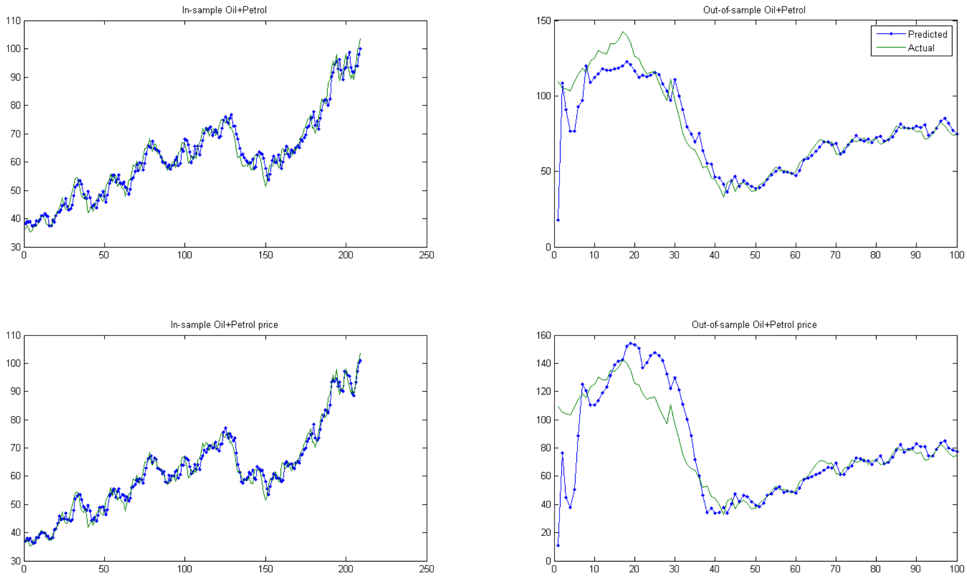

Figure 10-6 One-step-ahead forecast for crude oil price and petrol (upper) and crude oil price and petrol price (bottom)

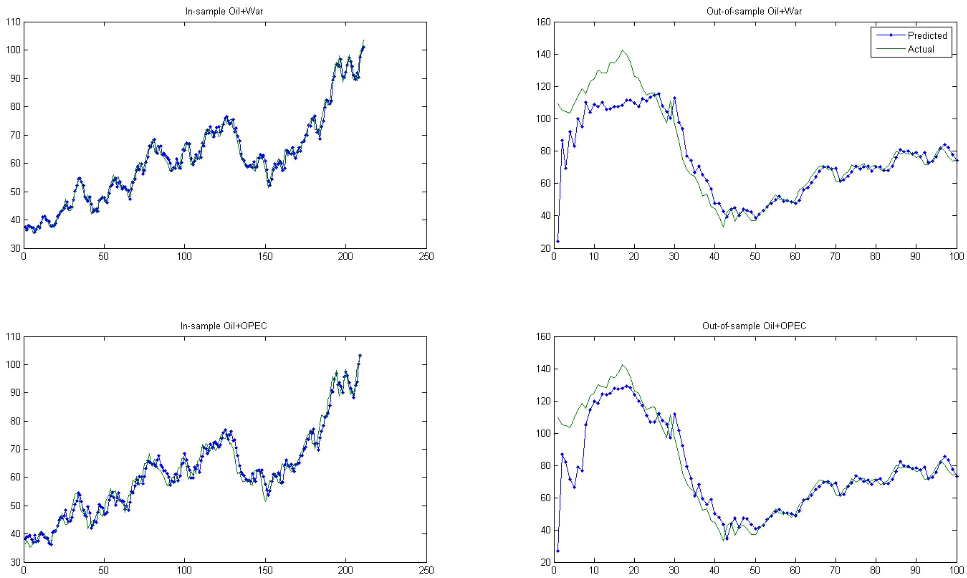

Figure 10-7 One-step-ahead forecast of oil price and war (upper) and oil price and OPEC (bottom)

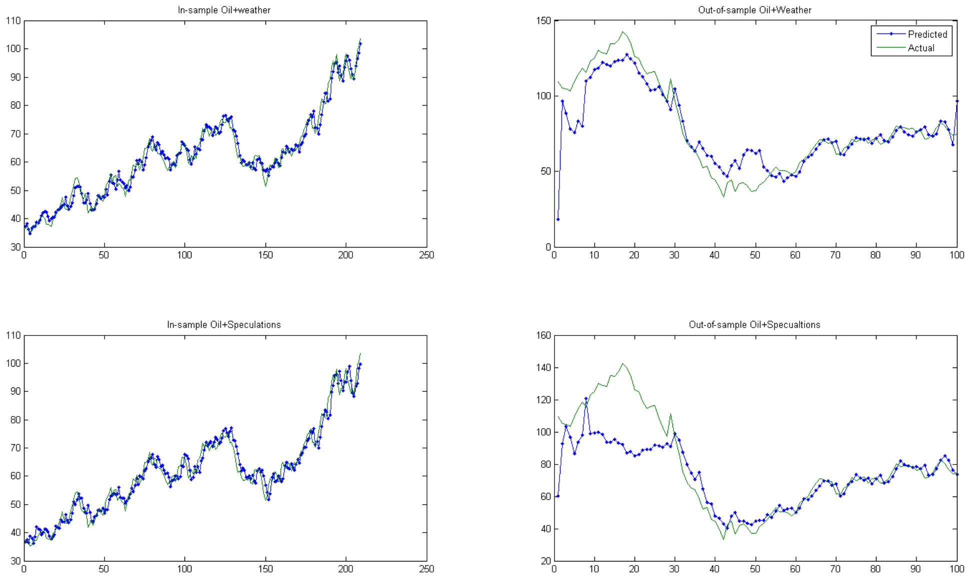

Figure 10-8 One-step-ahead forecast of oil price and weather (upper) and oil price and speculations (bottom)

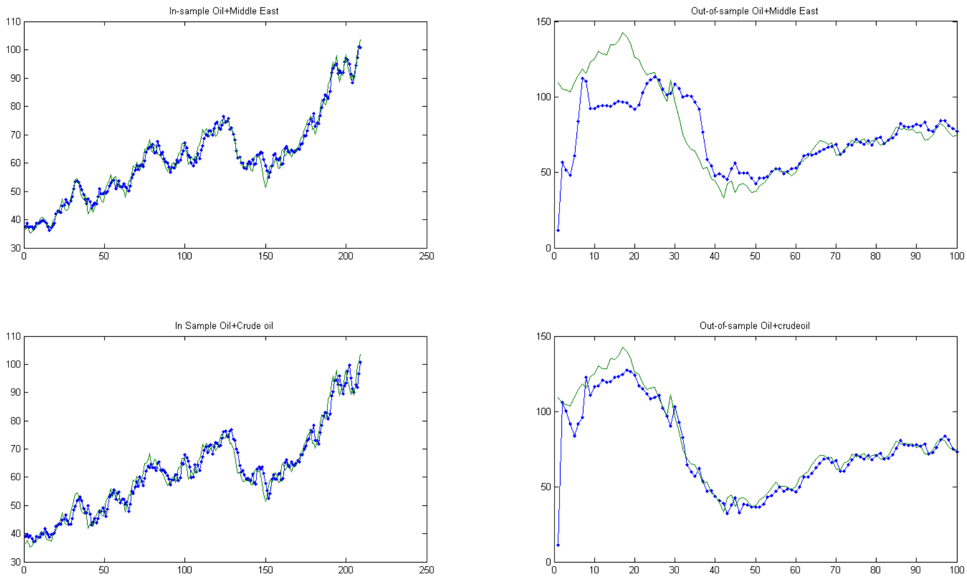

Figure 10-9 One-step-ahead forecast of oil price and Middle East (upper) and oil price and crude oil phase (bottom)

10.7 **Information gain (financial market)**

The Table 10-11 shows the results of the FCS test applied to the linearly filtered S&P GSCI price and return. According to the FCS the price for the entire series was mostly chaotic with some linear trend. In comparison the dynamics of the price and return sub-series are very similar to the crude oil price and return series.

S&P GSCI	Filter	R^2	θ	Decision
Price	Simple	0.99	0.39	SL-CHT
Price I	ARIMA(1,1,15)	0.99	0.97	SL-NL-HN
Price II	Simple	0.99	0.68	SL-NL-MN
Return	Simple	0.02	0.95	NL-HN
Return I	ARIMA(0,0,15)	0.013	0.97	NL-HN
Return II	ARIMA(0,0,14)	0.002	0.99	WN

Table 10-11 The FCS test for the S&P GSCI index

Twelve lags of each variable (crude oil Return II and S&P GSCI Return II) were used as an input for feedforward (results in Table 10-12) while in the next experiment, in addition to these two variables, a squared version of them was also added[50]. It is clear from Table 6-7 that the hit rate was slightly better than the benchmark. Nevertheless, the DA statistic was not significant enough to reject the null hypotheses that the hit rate was random; therefore, the improvement could be due to the noise factor, i.e., it behaves as if we added random noise to the input.

Metrics	Return II and S&P GSCI index		Return II and S&P GSCI index and squared inputs	
	in-sample	out-of-sample	in-sample	out-of-sample
Hit rate	55.26	50.1	53.45	50.19
RMSE	0.0248	0.0247	0.0273	0.0232
R2	0.2112	0.0051	0.0781	0.0007
R	0.3858	-0.0634	0.2565	0.0024
IC	0.6166	0.8272	0.679	0.7761
MSE	0.0006	0.0006	0.0007	0.0005
MAE	0.018	0.0195	0.0195	0.0184
SSE	1.7452	0.1269	2.0637	0.111
DA	5.824	0.0543	4.1434	-0.0843
P value	0.0881	0.4215	0.0152	0.5325

Table 10-12 Average results of using S&P GSCI as explanatory variable

[50] The reason for adding the squared return is explained later in this section.

10.8 **Appendix III: Empirical results for Chapter 3**

10.8.1 **Modified fitness function**

Each experiment took the algorithm around eight hours to complete 200 generations of search. The search is terminated by the 200th generation, if none of the stopping criteria has met the results obtained from the individual with the best fitness. This is by no mean ideal; however, it is acceptable practice when dealing with genetic algorithms. Besides, for a one-day-ahead forecast it is not logical to allow the algorithm to run longer; otherwise, there will not be enough time for any trader to act on this forecast.

The results showed that the new fitness function produced, as expected, a better Sharpe ratio than the benchmark and all other fitness functions (including the supervised ANN). However, the benchmark was better in terms of the realized potential in addition to the hit rate. We argue that a higher Sharpe ratio is more important than other metrics because it provides an indication of the risk involved.

10.8.1.1.1 Yao and Tan (2001) function

Metrics	In-sample	Out-of-sample
Hit rate	0.619677	0.680328
RMSE	0.588957	0.548635
R^2	0.00873	0.009187
R	0.093433	-0.09585
IC	1.253296	1.541884
TUR	1.82985	2.170146
MSE	0.34687	0.301001
MAE	0.486974	0.455375
SSE	236.2184	146.8884
DA	1.870203	0.082283
P value	0.030728	0.467211
AIC	0.360371	0.317477
BIC	-0.13186	-0.19799
Net return	0.628594	-0.16583
Sharpe	-0.48109	-0.73644
Realized potential	36.31652	23.39257

Table 10-13 One-day-ahead forecast of crude oil return using NEAT and the Yao and Tan (2001) fitness function

Here the fitness function is $= (\mathbf{10} - \boldsymbol{E_{DP}})^{\mathbf{2}}$, where $\boldsymbol{E_{DP}}$ is Yao and Tan (2001) function

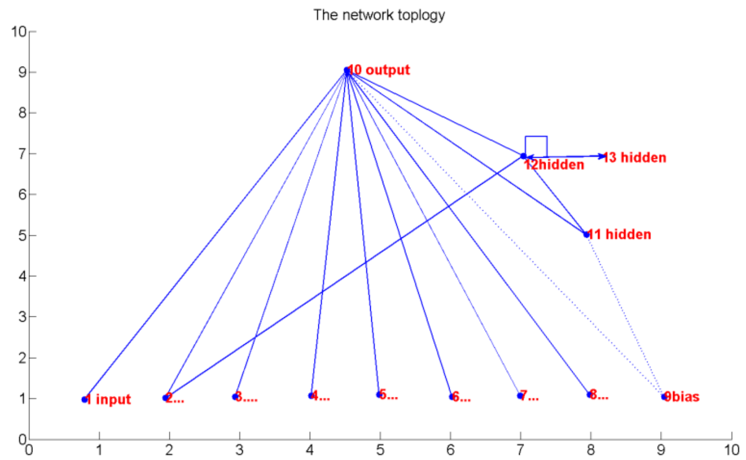

Figure 10-10 A plot of the network topology of NEAT

10.8.1.1.2 Refenes, et al. (1997) function

Metrics	In-sample	Out-of-sample
Hit rate	0.574156	0.686475
RMSE	0.638031	0.572141
R^2	3.01E-05	0.009188
R	0.005489	-0.09585
IC	1.357727	1.607943
TUR	1.982322	2.263122
MSE	0.407084	0.327345
MAE	0.529836	0.477044
SSE	277.2242	159.7444
DA	-0.8184	0.760504
P value	0.793434	0.223477
AIC	0.420451	0.342443
BIC	-0.0947	-0.15583
Net return	0.628594	-0.16583
Sharpe	-0.47631	-0.85807
Realized potential	27.66746	23.06812

Table 10-14 One-day-ahead forecast of crude oil return using NEAT and the Refenes, et al. (1997) fitness function. Here the fitness function is $= (\mathbf{10} - \boldsymbol{E_{DLS}})^2$, where $\boldsymbol{E_{DLS}}$ is Refenes, et al. (1997) function.

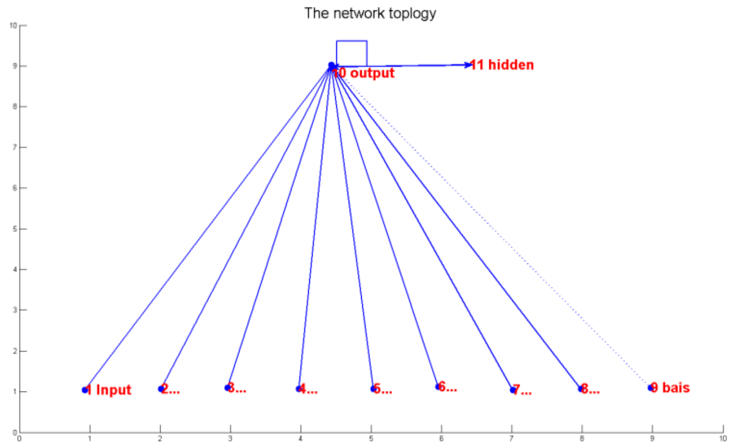

Figure 10-11 A plot of the network topology of NEAT

10.8.1.1.3 Yao and Tan (E2)

Metrics	In-sample	Out-of-sample
Hit rate	0.591777	0.688525
RMSE	0.717488	0.679448
R^2	0.002649	0.018616
R	0.051464	-0.13644
IC	1.526811	1.909519
TUR	2.22919	2.687581
MSE	0.514789	0.46165
MAE	0.620087	0.586376
SSE	350.5716	225.2851
DA	0.803291	1.485699
P value	0.210903	0.068679
AIC	0.533257	0.484926
BIC	-0.07633	-0.11766
Net return	0.628594	-0.16583
Sharpe	-0.46044	-0.78881
Realized potential	19.12153	17.24941

Table 10-15 One-day-ahead forecast of crude oil return using NEAT and the Yao and Tan (2001) fitness function.

Here the fitness function is $= (\mathbf{10} - \boldsymbol{E_{TDP}})^2$, where $\boldsymbol{E_{TDP}}$ is Yao and Tan (2001) function.

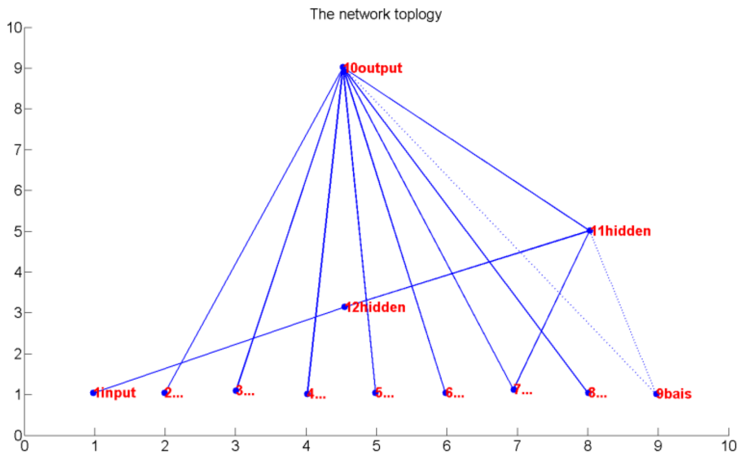

Figure 10-12 A plot of the network topology of NEAT

10.8.1.1.4 Fuzzy fitness function

	In-sample	Out-of-sample
Hit rate	60.7353	55.4415
RMSE	0.57553	0.506493
R^2	0.000778	0.002354
R	0.02789	-0.04851
IC	1.224724	1.423445
TUR	1.788134	2.003449
MSE	0.331235	0.256535
MAE	0.47091	0.41359
SSE	225.5708	125.189
DA	-0.55306	-2.87463
P value	0.709888	0.997977
AIC	0.341107	0.267268
BIC	-0.10585	-0.17258
Net return	0.628594	0.165835
Sharpe	1.101827	1.324217
Realized potential	36.62616	30.93998

Table 10-16 One-day-ahead forecast of crude oil return using NEAT and the new fitness function. Here the fitness function is same as.

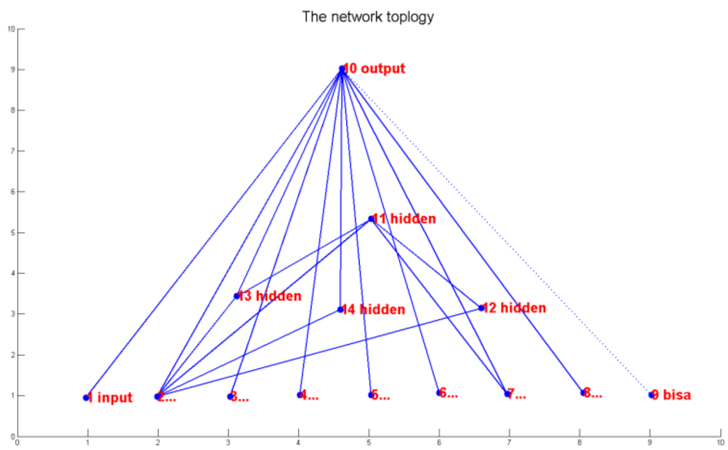

Figure 10-13 A plot of the network topology of NEAT

10.8.1.2 Internal pre-processing layer

In a previous experiment we tested the internal filter described in Neuneier and Zimmermann[51] (1998) to remove outliers from the input at each iteration using a fixed topology network. Here we try to interweave this model within NEAT. In the original Neuneier and Zimmermann (1998) model the weights were updated using the BP method. Clearly this approach of weight updating will not

---

51 A quick reminder of the Neuneier and Zimmermann (1998) approach: the data is transformed into momentum and force (see Neuneier and Zimmermann, 1998) with respect to the forecast horizon, then it also scaled to have a mean of 0 and a standard deviation of 1. The input is then passed through a special network layer with a diagonal weight matrix (initiated with +0.1) and squashed by a hyperbolic tangent. These small values of the input matrix will allow the input to pass the *tanh* function unchanged and as the weight increases the outliers will be removed from the input space by the *tanh* function.

work in NEAT as it updates its weight by mutation. To overcome this issue we propose that at each generation, if the input is classified so that it contains outliers, then it will be processed through the internal filter first, otherwise it will be processed normally.

The outliers are defined as:

$$out\ if\ [input - mean(input) > 2 \times std(input)]$$

The internal filter consists of a square diagonal matrix with small network weight values (+0.1) so when it passes through the *tanh* function it will not make much difference (Neuneier and Zimmermann, 1998). After the first generation, if there are still outliers, then the weight will be updated by mutation with the same probability of all other weights in NEAT; the only difference is that the weights are restricted to being positive. This condition was set by Neuneier and Zimmermann (1998).

10.9 **Error correction NEAT**

Error correction models are very useful in domains where full knowledge about the system dynamics, i.e., all factors and events affecting the system, is not available for the model (Grothmann, 2002). In this case, the error of the model itself during the past state can be used to aid finding good hypotheses (Grothmann, 2002). Grothmann (2002) argued that, for most financial time series, it is often rare to have a complete account of all external factors affecting the market. Therefore, the error of the model itself can be viewed as a measurement of the short-term influence of external forces when used as additional input to the recurrent network (Grothmann, 2002).

According to Grothmann (2002) this concept shares some similarity with two established models: ARIMA models and NARX networks. In an ARIMA model, the moving average is determined by the linear components of auto-regression and the stochastic segment of the moving average. On the other hand, NARX networks (non-linear auto-regressive with exogenous input) could be viewed as non-linear versions of an ARIMA model. However, Grothmann (2002) claimed that the recurrent error correction model differentiates itself from the ARIMA by its ability to model non-linear behaviour, and from NARX by its ability to model a long-term horizon (Grothmann, 2002). This way the state and transition equations (Equation 3.5 and 3.6) can be restated as:

$$\beta_t = f(\beta_{t-1}, u_t, e_t) \tag{10.1}$$

$$y_t = g(\beta_t) \tag{10.2}$$

$$y_t = g(\beta_t) \tag{10.3}$$

where $e_t = y_t - \hat{y}_t, \beta_{t-1}$ is the internal hidden state, u_t is an external input and y_t is the network output.

There are several ways to implement this approach in the context of NEAT. We chose a simple, yet robust method described in the following steps which present another extension to NEAT by using an error correction model.

The new algorithm is summarized as follows:

1. We start training as usual (either with a fully connected or a partly connected network).
2. After 100 generations (this number was found experimentally based on the best results achieved) we calculate the error from all individual networks in the population in our task.
3. We select the errors from the network that produce the smallest RMSE in our task, as long as they have at least one hidden neuron (the reason for this condition is that we want the error from non-linear model).
4. The error is placed as additional input to the training population.
5. At the first instance the error is left disconnected to avoid destabilizing the reproduction process.
6. The error is later connected during evolution.

10.9.1 **Hybrid supervised and reinforcement learning**

In an attempt to speed up the learning process we created a hybrid learning approach using supervised training and reinforcement learning. Three different settings were tested. First, NEAT started the learning process normally, then, if after 15 connective generations (to allow networks time to learn) the maximum fitness of the population did not meet the stopping criteria, feedforward networks were created with one hidden layer and a number of hidden nodes equal to the average number of nodes in the current generation (the average complexity of NEAT across the population). Then the output of each NEAT network was used as an input to each feedforward network and the actual target was used for training. All networks were trained with the Levenberg Marquardt algorithm proposed by Levenberg (1944) (and revived by Marquardt (1963)) until one of the stopping criteria for each feedforward was achieved. The outputs of this network were to replace the output on NEAT in the following generation. This process was repeated every 15 generations.

In the second approach, a feedforward network was used to train the input-output sets for a number of iterations, then its output (and in another experiment, the error) was used as an additional feature in the input set. Only the feedforward output was connected in the first generation.

In the last approach, NEAT was run for a number of generations; then its output was used as a sole (or additional) feature to train the feedforward network. The number of hidden nodes and layers was set to be the same as in NEAT (even so, NEAT and FF-ANN do not necessarily share the same complexity due to the unique structure of NEAT networks).

The results in Appendix III show that only the third approach produces superior results (for in-sample and out-of-sample testing) compared to the DK-NEAT, feedforward network and recurrent network, while the first approach appears to be harmful to the learning process. The NEAT network structure in the second approach shows that the input from ANN has much more value in the search process of GA than the actual inputs.

The main problem in this method is, unlike NEAT, we cannot justify the selection of the local network complexity and structure, especially in the first approach.

10.9.2 **The first hybrid approach**

A combination of training with NEAT and using ANN every 15 generations was trialled. The fitness of the algorithm with this method improved significantly with supervised training and then dropped down below its initial value. This could be explained as NEAT structures' complexity at this stage was not able to maintain the fitness (as it was not complex enough), so it plummeted in the next generation. Therefore, it appears to be that this approach is harmful for the learning process.

10.9.3 **The second hybrid approach**

Training with ANN and using the output /and error as an additional feature was undertaken. Other than the hit rate, the performance of this hybrid model appears to be worse than for the DK-NEAT.

10.9.3.1 Input and error

In these experiments both the output of feedforward and its error was used as additional features, but only the output was connected in the initial population. The network structure at the end of training was based only on these two inputs, i.e., the ANN output and its error with several feedback loops while none of the original inputs was connected. This suggests that the ANN error of these inputs is of much more value to the learning process than the original inputs. One issue here is, if the ANN did not converge to a good solution, then its error will be big (and usually close to the target). If that was the case we expect the network not to be generalized. There is some evidence of over-fitting although the R^2 for out-of-sample was better than in-sample.

10.9.4 **The third hybrid approach**

In this approach we tried training with NEAT then applying ANN to its output in two different ways explained below.

10.9.4.1 As sole input

NEAT was set to the maximum of 100 generations; by 100 generations, if none of the stopping criteria was met then the algorithm was set to select the network with the best performance and generate output for in-sample and out-of-sample. The output was then fed to feedforward and recurrent networks, which trained them with the actual target until one of the stopping criteria was met.

10.9.4.2 As additional feature

The procedure was the same as in the previous section, but in addition to the NEAT output the original input was also used in the training results. The results show that the hybrid system outperformed using NEAT alone and also was superior (for all performance criteria including the IC metric) to feedforward and recurrent networks trained with same number of nodes. Although, to confirm these results the standard error of estimate needs to be generated and tested on several datasets. This method is worth further investigation, as it could save a significant amount of time and produce a better fit.