An ensemble-ANFIS and a hybrid SOM-ANFIS based model for stock market variable forecasting

School of Agricultural, Computational and Environmental Sciences

A Dissertation submitted by

John Worrall

Supervisor:

Dr Ravinesh C Deo

For the award of

Master of Science in Mathematics and Statistics

November 2018

Abstract

Forecasting the daily stock index value is a challenging process due to the stochastic non-linear nature of the financial market. While there have been a number studies on machine learning techniques for stock market prediction, this paper improves the current literature by investigating the novel development of hybrid Self Organising Maps (SOM)-Adaptive network based fuzzy inference system (ANFIS) and ensemble ANFIS techniques for the All Ordinaries Index (AORD).

The forecasting capabilities of both the SOM-ANFIS and ensemble ANFIS models are studied and compared to ANFIS and traditional Autoregressive Integrated Moving Average (ARIMA) models. The comparison was conducted through efficient modelling and forecasting of the daily value of All Ordinaries of the Australian Securities Exchange (ASX) over 10 years. The techniques develop will consider integrating indexes Financial Times Stock Index (FTSE), Dow Jones Industrial Average (DJI), Hang Seng Index (HIS) and Nikkei Index (N225). The study devises an optimal approach for forecasting the All Ordinaries applying hybrid SOM-ANFIS, which divides data using unsupervised cluster algorithm, SOM. While the average methods were used for integration of the ensemble of ANFIS

Results of the study show the best performing ARIMA model obtained high error values, due to the forecast converging to the mean after three or four forecasted values. ANFIS architecture of combining fuzzy logic and Artificial Neural Network (ANN) overcomes several statistical and neural network model reported issues in achieving predicted values of non-linear and chaotic index time series. Further, results from 100 ensemble ANFIS notably improved on lowering error metrics and variance while enabling uncertainty (error bounds) on forecast. The hybrid SOM-ANFIS achieved the best performing model, showing the random sampling input selection process driven by SOM implements the optimal model for forecasting time series and financial pattern extraction, advocating for future develop models not to be reliant on arbitrarily divided chronologically ordered time series data.

The study signifies the important role both hybrid and ensemble ANFIS to potentially developing an automated prediction system for forecasting stocks prices, patterns and volatility, with large potential application in the economic and financial sector, such as risk management, asset pricing and allocation.

Thesis certification

This Thesis is entirely the work of **John Worrall** except where otherwise acknowledged. The work is original and has not previously been submitted for any other award, except where acknowledged.

Principal Supervisor: **Dr Ravinesh C Deo**
Insert name only (signature not required)

Student and supervisors signatures of endorsement are held at the University.

Acknowledgements

My research would not have been possible without the continuous support of my supervisor, Dr Ravinesh Deo. I would like to express my sincere gratitude for the valuable guidance throughout my Masters dissertation.

Contents

Nomenclature

ACF	Autocorrelation function
ANFIS	adaptive neuro-fuzzy inference system
AORD	All Ordinaries
ANN	artificial neural network
ARIMA	auto regressive integrated moving average
ASX	Australian Securitas Exchange
DJI	Dow Jones Industrial Average
EMH	efficient market hypothesis
FCM	fuzzy c-means clustering
FIS	fuzzy inference system
FTSE	Financial Times Stock Index
HIS	Hang Seng Index
LSE	least square estimate
MF	membership function
MSE	mean square error
N225	Nikkei Index
PACF	partial autocorrelation function
SSR	sum of squared residuals
SOM	self-organising map

List of Figures

List of Tables

Chapter 1

Introduction

This chapter sets the scene for the study, starting with the background of the stock market, explaining why research is important and highlighting my motivations for undertaking this work. It continues on to describe the aim of the study, specifically the research questions and important contributions that can be made. The chapter finishes with a summarised guide to each chapter in this dissertation. Chapter sections are outlined in Figure 1.1.

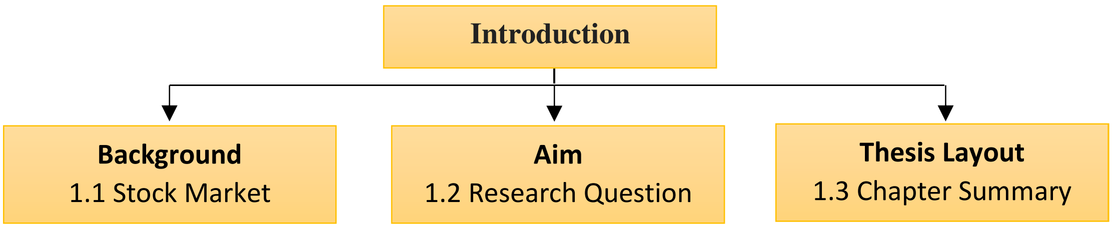


Fig. 1.1 Chapter 1 structure

1.1 Background

Stock market forecasting by means of times series analysis and modelling of related variables is of primary importance in the economic and financial sectors. The prediction of stock prices is useful for investors and traders, economists and policy makers in terms of allocating investment correctly leading to less wasted resources and a more prosperous economy. Developing and evaluating these predictive models is essential in macroeconomic strategies, construction of portfolio investments and risk management. Further analysis of time series may reveal the critical characteristics in behavioural and temporal patterns of the financial market.

Prediction and analysis of stock prices, however, is a challenging application due to the noisy, nonstationary chaotic characteristics, complex dimensionality and sheer quantity of data (Liao & Wang, 2010; Guresen, Kayakutlu & Daim, 2011; Bagheri, Peyhani & Akbari, 2014; Ruta, 2014; Rather, Agarwal & Sastry, 2015; Milosevic, 2016).
Previous studies have demonstrated the difficulties of using prediction models as they are nonlinear, reliant on polynomial functions consisting of variables in higher degrees. Additionally, there are numerous factors that affect the values of stock (e.g. economic, political and environment data), increasing the complexity of the system, which may appear random but relies on initial conditions for underlining patterns. Processing of the above mention complexities along with the sheer quantity (big data by nature), is suited towards machine learning.

Machine learning techniques are frequently used in stock market prediction. Milosevic (2016) demonstrated how machine learning enables analysts to predict the movement of stocks as well as the ratio of movement over a fixed amount of time. Choudhry and Garg (2008) and Tsai and Wang (2009) concluded better performance with multiple techniques such as ensembles and hybrid. Both algorithms, SOM and ANFIS have been frequently applied in stock modelling, enhancing decision-support systems in computing modelling and parameter estimation of stock indexes (Dablemont, Simon, Lendasse, Ruttiens, Blayo & Verleysen, 2003; Boyacioglu & Avci, 2010; Billah, Waheed & Hanifa, 2015). However, to date no known study has been done using a hybrid SOM/ANFIS and ensemble ANFIS applied for daily value prediction on major indexes.

The contribution of this research is to fill the gap by adding to the existing body of literature in stock index forecasting with machine learning techniques to risk management. The study analyses time series patterns and strategies to predictive movement in stock index that can be used to improve the overall risk and return required by investors.

1.2 Aim

The project aims are to investigate novel, hybrid and ensemble machine learning techniques for next day prediction of All Ordinaries (AORDS) indexes. In the study, four indexes are considered to be representative of developed nations' economies and are used in machine learning and prediction models (Figure 1.2): the Dow Jones Industrial Average (DJI), Hang Seng Index (HIS), Financial Times Stock Index (FTSE) and Nikkei Index (N225).

This research has two objectives with regard to machine learning next day predication of All Ordinaries:

1) Develop an Adaptive Neuro Fuzzy Inference System (ANFIS) machine learning model and evaluate performance accuracy against Auto-Regressive Integrated Moving Average (ARIMA) model using traditional machine learning techniques.

2) Develop and compare an ensemble ANFIS machine learning model against a hybrid Self-Organizing Map (SOM)/ANFIS and evaluate the model's performance and accuracy using traditional machine learning techniques.

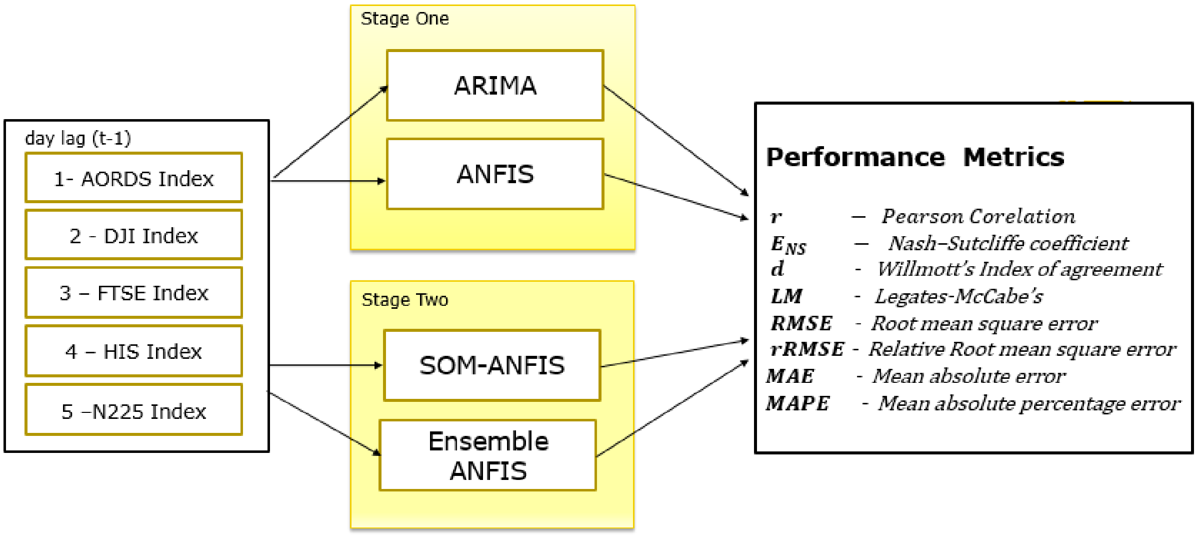


Fig. 1.2 Aim proposal diagram

The results of the prediction models can determine whether All Ordinaries prices can be effectively forecasted and whether the model can be applied in a broader financial context. The frameworks and overall findings can be useful to investors and the larger community, adding to current research.

1.3 Thesis Layout

This thesis is organised as follows. The literature review in Chapter 2 introduces the stock market, discussing its significance in predictions and various challenges in modelling. Then, the chapter presents the emergence of machine learning applications, specifically ARIMA, SOM, ANFIS, hybrid and ensemble techniques, in dealing with the inherent nature of the market. Chapter 3 describes data, All Ordinaries and other indexes used in the study. The chapter then provides theoretical overview of machine learning algorithms used for prediction and explains the procedure for model selection through performance metrics. Chapter 4 compares ARIMA and ANFIS based models for price prediction and discusses results. Chapter 5 extends on ANFIS best performing model through hybrid SOM-ANFIS and ensemble ANFIS, and performance is compared. Chapter 6 discusses the conclusion and future work to be done.

Chapter 2

Literature Review

This chapter offers a comprehensive survey of current literature, focusing on two specific areas (Figure 2.1). The first section reviews the stock market, defining the research problem. The second part focuses on the research design which is the proposed method for this thesis.

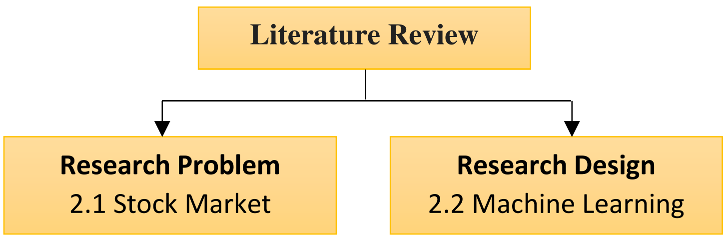


Fig. 2.1 Chapter 2 structure

2.1 Stock Market

There are two sections in reviewing the stock market: 1) the significance of forecasting stock prices and people who are affected by predictions; 2) modelling stock markets and difficulties due to their nature.

2.1.1 Significance

Analyses of the market are of primary importance to the financial and economic world. With analyses, a strong financial system can improve financing decisions, leading to better allocation of resources and thus economic growth.

As of 2017, the size of total stock market exchange with total market capitalisation has been estimated at $66.8 trillion, more than twice the value of all the physical money in the world (Figure 2.1).

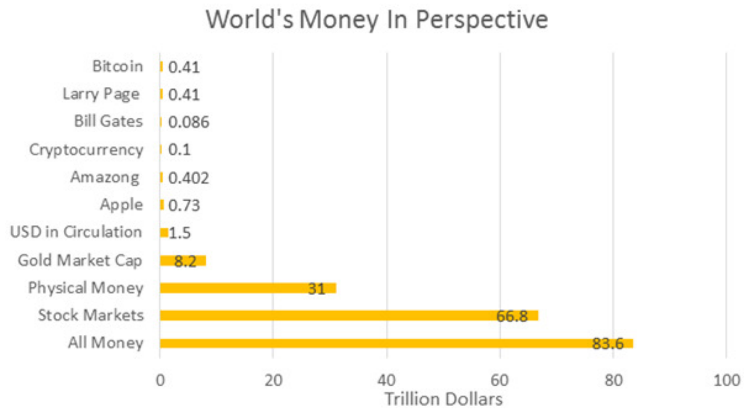


Fig. 2.1.1 Stock market estimates *(Market Watch 2018)*

The market plays a substantial role in the growth of industry and commerce of a country on a larger scale. There are multiple ways in which the stock market can directly affect a nation's economy. For example, if stocks fall, spending stops and consumers lose confidence. Conversely, if stocks increase, confidence spreads, and therefore, spending and investment grow.

Empirical investigations (Beck & Levine, 2004; Levine & Zervos, 1998; Rajan & Zingales, 1996) have quantitatively highlighted the influences the stock market has on banking and a nation's economic growth. These investigations have concluded that the market positively correlates with financial and economic development, suggesting that the market is an integral part of a nation's growth.

The wide influence of the stock market and close link to a nation's economy emphasises the necessity to focus on improving individuals' understanding of the stock market and alternative forecasting methods. This can ensure growth and stability is met for an ever changing environment.

Stake holders

The purpose of stock market forecasting is to reduce risk in financial decision making and is important for the following three groups: 1) traders/investors, 2) companies and 3) economists/policy makers.

Traders and investors can avoid blind decisions and can profit by correctly buying and selling shares. Fama & French (1993) state that it is necessary to predict future values of the stock market in order to correctly decide to buy or sell the shares.

Companies also use the stock market to raise capital and grow through acquisition. The raised funds from capital can then be used for research and/or to pay debt off. Accurate forecasting of the stock market assists companies in optimising their investments as well as implementing successful capital raising strategies.

Economists and policymakers are able to allocate resources effectively, avoiding wasting national resources and mitigation against financial crisis. Government capital gained can be optimised for development projects, while monetary policy benefits, based on the insights of the stock prediction, lead to more stable economies. Paper (Funke, 2004) examines the relationship between private consumption and the stock market, identifying this relationship as consistent with the fundamental theorem, which dictates that a booming stock market increases consumption pressures. This has implications for policymakers with regard to implementing monetary policy in reaction to emerging stocks.

Rigobon & Sack, (2003) found that more accurate modelling assists monetary policymakers in predicting stock market prices to mitigate against financial crises and protect a national economy. Accurate modelling reduces joblessness, homelessness, and foreclosures after an event like a market shock (Case et al., 2005; Sornette, 2017).

It is evident that further research on the prediction of stock market prices can have multiple benefits, not least ensuring the health and improvement of a nation's economy.

2.1.2 Modelling stock market

Numerous studies have tried to model the market and predict stock returns. However, these are challenging exercises due to the market's nature.

Modelling the market

The stock market is a market where several financial instruments are traded. These instruments can be modelled using various timescales. Timescales can be intraday, daily and monthly, while instruments traded on the stock market might include stocks, bonds and other securities.

Indexes are sample stocks representing a portion of the market and are used to track the performance of the overall stock market. Considered to be an important gauge of the overall market, they are known as barometers of stocks (Frino & Jarnecic, 2005). Market analysts typically believe there is a positive correlation between indexes and country performance.

There is merit in investigating modelling indexes in a daily interval to demonstrate the overall performance of the companies listed in the stock market. Past research describes the utilisation of indexes in forecasting and understanding stock market performance as a valuable dataset.

Efficient market hypothesis and random walk theory

Random walk theory (RWT) and efficient market hypothesis (EMH) have been the central propositions of the finance sector since the 1970s. Initial strong theoretical and empirical evidence supported EMH. However, more recent studies have refuted the EMH.

According to the EMH, market prices follow random walks that cannot be predicted with past information (Fama, 1965). All information that enters the market instantaneously affects the price. In other words, it would not be necessary to predict stock prices if the stock market were efficient because the price of the next day should be random. However, in recent years, technical analysts in finance industries have identified EMH as a null hypothesis (Rosenberg et al., 1998; Shleifer, 2000*).* A number of published studies (Kavajecz & Odders-White, 2004; Lo et al., 2000; Neftci, 1991) support technical analysis for predictions, displaying that the stock market is less efficient.

Nature of stock market data

Researchers have faced several difficulties when trying to model complex systems such as the stock market due to its nature. Described as non-linear, chaotic, highly dimensional and extensive (Schmidt, 2011) making forecasting a challenging process.

Models that approximate non-stationary financial time series may include noisy error prone features. The relationship between the model input and output is essentially non-linear (Xiao et al., 2012), where stock prices include variables of higher degrees, adding to the complexity of modelling and predicting the stock market.

Chaotic systems are deterministic non-linear. They have sensitive dependency to initial conditions, where subtle changes to initial conditions would lead to different results in the future. Given that the market is a complex system that appears random, studies have demonstrated (Ghosh et al., 2017) underlying patterns and repetition, relying on factors initial conditions. Two characteristics that generally require chaotic systems are fractional dimensions and the Lyapunov exponent. A previous published paper (Meng et al., 2013) explored stock market characteristic and found evidence of chaos in the stock market.

Another feature of financial time series is its multi-dimensionality. It is well known that stocks prices are multivariate in nature, with many factors influencing them, such as economics, political, temporal, environmental, etc. Papers have explored the connection of prices and behavioural, temporal and political patterns. Analysis of a high dimensional price prediction system suits numerical solutions, such as machine learning, distinguishing useful patterns. Pring (2002) has stated that a technical approach to investment is essential to reflect price movements determined by changing attitudes of investors who are part of a variety of economic, monetary, political and psychological forces.

Lastly, large quantities of scientific data are required in the generation, collection and analysis of data driven decision making of the stock market (Attigeri et al., 2015). Big data, in its nature, refer to data to be processed that are on the increase. The extraction of valuable information from raw data requires specific analytical approaches.

There are significant challenges in improving stock market forecasting for the formation of confident predictions. Future studies on the technical analysis of stocks must be performed based on accurately and reliably accounting for the nature of stock data in order to design potential models or forecast to aid economic development. Machine learning will now be presented as a method of mathematically modelling the above mentioned complexities.

2.2 Machine learning

This section on machine learning includes a number of sections, including a discussion of 1) the application of machine learning and its advantages when applied to finance data. Then, specific machine learning algorithms are described: 2) SOM, 3) ANFIS and techniques, 4) hybrid and 5) ensemble applied to financial time series.

2.2.1 Machine learning application

Given the modelling challenges of financial time series described previously, machine learning techniques offer suitable solutions (Chun & Kim, 2004; Liao & Wang, 2010). Studies have demonstrated numerical optimisation problems with chaotic, incomplete and non-stationary observed data in the domain of machine learning. Various current models (post 2000) have been researched for the forecast of market prices. A brief survey of literature is presented in table 2.2.1

Table 2.2.1 Recent research on machine learning applied to the stock exchange

Researchers	Date	Publication	Data	Method	Results
Afolabi, M.O. and Olude, O	2007	IEEE	Stocks	Hybrid SOM, SOM, NN	Hybrid SOM best predictor.
Ansari, T., Kumar, M., Shukla, A., Dhar, J. and Tiwari, R	2010	Expert Systems with Applications	Index -NASDAQ	ANFIS	ANFIS modelling issues .74% accuracy
Atsalakis, G.S. and Valavanis, K.P	2009	Expert Systems with Application	Stocks – ASE,NYSE	ANFIS	ANFIS concepts and technical, potential for prediction
Bagheri, A., H. Mohammadi Peyhani and M. Akbari	2014	Expert Systems with Applications	Foreign Exchange	Hybrid Wavelet +ANFIS QPSO for tuning parameters	Future trading advice. 69%
Billah, M., S. Waheed and A. Hanifa	2015	International Journal of Computer Applications	Stocks Bangladesh	ANFIS vs ANN	Next day predictions – ANFIS less error
Boyacioglu, M. A. and D.	2010	Expert Systems with Applications	Istanbul Index – ISE100	ANFIS	Economist – Prediction tool
Chen, Y., Yang, B. and Abraham, A.	2007	Neurocomputing	Index – NIFTY, NASDAQ	FNT+PSO vs FNT+GA	LWPR the best suited ensemble approach for FNT or NN.
Choudhry, R. and K. Garg	2008	World Academy of Science, Engineering and Technology	Stocks India	Hybrid GA+ SVM vs SVM	Direction of Stock. Hybrid more accurate
Dablemont, S., G. Simon, A. Lendasse, A. Ruttiens, F. Blayo and M. Verleysen	2003	WSOM'2003 proceedings	German Index - DAX30	Hybrid SOM + RBF vs RBF	Hybrid results exceed expectations
Ghosh, I., M. K. Sanyal and R. Jana	2017	Arabian Journal for Science and Engineering	International Indexes	ANFIS vs DENFIS vs JNN vs SVR vs RF	Critical characteristic. RF shows better results, author's opinions due to ensemble nature.
Guresen, E., G. Kayakutlu and T. U. Daim	2011	Expert Systems with Applications	Index- NASDAQ	MLP vs GARCH-MLP vs DAN2 vs GARCH-DAN2	MLP reported highest accuracy. Contrary to belief of hybrid showing better results.

(continue on next page)

Table 2.2.1 (Continued)

Researchers	Date	Publication	Data	Used method	Results
Huang, C.-L. and C.-Y. Tsai	2009	Expert Systems with Applications	Index - TAIEX	SOFM-SVR	Filter based feature selection of SOFM-SV produces good prediction results.
Liao, Z. and J. Wang	2010	Expert Systems with Applications	Indexes - SAI, SBI, HSI, DJI, IXIC and SP500,	Stochastic neural network	Effectiveness of including stochastic function with NN
Melin, P., Soto, J., Castillo, O. and Soria, J	2012	Expert Systems with Applications	Indexes - DJI, MSE	Ensemble ANFIS	Positive in its prediction of DJI and MSE.
Nguyen, D.H. and Le, M.T	2014	arXiv	Indexes - DJI, S&P500 and stocks	RBN, SOM-ANFIS, SOM+SVM, SOM+f-SVM	SOM improved execution time. SOM+f-SVM shows best results.
Pulido, M., Melin, P. and Castillo, O.	2014	Information Sciences	Mexican stock Exchange	Ensemble PSO NN	Best results from PSO optimising architecture NN
Rather, A. M., A. Agarwal and V. N. Sastry	2015	Expert Systems with Applications	Bombay- 25 stocks	Hybrid model (ARMR+ ES+RNN)	Proposed model achieved high accuracy
Shen, S., H. Jiang and T. Zhang	2012	Department of Electrical Engineering, Stanford	Stocks, Currency, Commodities	SVM vs MART vs GML vs LR	Models effectiveness
Svalina, I., Galzina, V., Lujić, R. and ŠImunović, G	2013	Expert Systems with Applications	Index - ZSEC	ANFIS	Forecasted 5 days ahead price, achieves satisfactory results.
Tan, L., Wang, S. and Wang, K	2017	Information Processing Letters	Stocks Shanghai A-share	FOA-ANFIS vs ANFIS	FOA-ANFIS far superior to ANFIS
Tay, F.E.H. and Cao, L.J	2001	Intelligent Data Analysis	Forex, Futures bonds, indexes	SVM vs SOM+SVM	Higher predication and performance of SVM with SOM.
Tsai, C. and S. Wang	2009	International Multi Conference of Engineers and Computer Scientists	Taiwan stocks	ANN vs DT vs ANN+DT vs DT+DT	Hybrid DT+ANN has highest accuracy.

.

Machine learning is a rapidly growing technology and has been applied in many disciplines, with great success in modelling highly non-linear predictive problems. In the finance industry, data analytics and forecasting have been used for credit scores, risk management and stock prices. Stock market price prediction with regard to time series prediction is a challenging task since the financial market is a complex, evolutionary, non-linear and dynamic system (Abu-Mostafa & Atiya, 1996; Addison, P.S, 2002).

Milosevic (2016) previously summarised that machine learning algorithms enable analysts to create models more easily. By applying their machine learning algorithms, they can predict the movement of stocks as well as the ratio of movement over a certain fixed amount of time. Shen et al. (2012) demonstrated the use of other global stock markets and temporal correlation to predict next day stocks. SOM models has also been studied for performance through benchmarking against other algorithms.

Machine learning has demonstrated success in solving complex predictive problems, particularly in regard to stock market forecasting. However, in spite of acceptable levels for model performance, there is need for better approximation.

2.2.2 Self Organising Maps

A popular machine- learning algorithm, self-organising map (SOM) (Teuvo Kohonen, 1982) is utilised to visualise high-dimensional data by configuring its neurons in order to quantise or cluster input space into a topological structure. These capabilities make SOM attractive in many applications involving clustering, particularly for stock price prediction.

With varying successes in the financial industries, Dablemont et al., (2003) have applied their models and found that results exceed expectations, particularly due to nonstationary dependencies. The use of SOM-based local models with Radial Basis Functions (RBF) networks for clustering of the German DAX30 over a period of ten years was suitable for this time series due to its past volatile nature. Afolabi & Olude (2007) attempted to solve this issue of non-linearity by comparing a hybrid Kohonen SOM with a Kohonen SOM and back-propagation, finding the hybrid is a better predictor as it is fast, efficient and has less errors in classification.

Tay & Cao (2001) benefited from improved final prediction results, by integrating SOM and SVM models to deal with large sizes and fuzzy rules from data. Huang & Tsai (2009) combined SOM and SVR to predict next day Taiwan index futures, resulting in improved accuracy based on several disjointed clusters. Another study (Nguyen & Le, 2014) used SOM to cluster input variables, reducing the size and also noise that is directly applied to fuzzy Support Vector Machine (SVM). This increased the number of rules applied to SVM and the complexity of system. However, results presented a better performing model.

When the SOM method is employed, there is no assumption that a single model is able to capture the dynamics of a whole series. Reviewed papers above had promising results when SOM was adopted for temporal pattern discovery, motivating us to employ this cluster approach to represent a different structure of data for the prediction algorithm. This paper aims to achieve a novel approach to forecasting by integrating a hybrid SOM with ANFIS, which has yet to be applied for forecasting of developed nations' index prices.

2.2.3 Adaptive Neural-Fuzzy Inference System

An ANFIS (Jang, 1991) combines both neural networks and fuzzy logic. Fuzzy logic theory is suitable for complex and nonlinear problems due to its ability in handling non probabilistic uncertainties. As a result, ANFIS has been applied widely in the sector of financial time series prediction for stocks (Atsalakis & Valavanis, 2009), foreign exchange (Boyacioglu & Avci, 2010) and indexes (Ghosh et al., 2017)

Billah et al. (2015) previously concluded that ANFIS has produced more accurate results than Artificial Neural Networks (ANN) in predicting the Dhaka Stock Exchange. An R squared value indicates strong correlation with observed outputs and predicts one day ahead with closing prices. Boyacioglu & Avci (2010) also reported significant enhancements when adopting the ANFIS method, demonstrating its potential for prediction and use in the financial industry.

Moreover, ANFIS methods (Ansari et al., 2010; Atsalakis & Valavanis, 2009) produced superior results when different membership functions and tuning parameters were compared based on error measures. The combination of the least squares method and back propagation to optimise membership and parameters resulted in an ANFIS function and a Sugeno-type fuzzy model, achieving accuracy up to 74%.

Svalina et al. (2013) presented interesting limits when using ANFIS to predict closing prices of Zagreb Stock Exchange Crobex index five days in advance. Prediction error increased each day, with the last Root Mean Square Error (RMSE) results on the fifth day being 0.5 greater than the first.

Comparing the use of ANFIS with other meta-heuristics models, Bagheri et al. (2014) proposed a model for market direction in the Foreign Exchange Market, using ANFIS tuned by Quantum-behaved particle swarm optimisation (PSO). The advantages included the ease of implementation flexibility, efficiency and rapid convergence speed. Tan et al. (2017) applied Fruit Fly Optimisation Algorithm (FOA) to adaptively adjust inference fuzzy rules, also producing results far superior to ANFIS.

The hybrid learning rule and logic theory is a strength of ANFIS, which has computational advantages over other methods in parameter identification. However, to the best of our knowledge hybrid SOM with ANFIS models have not been tested with our chosen indexes, despite the challenges and need for a robust model in forecasting developed nations’ stocks.

2.2.4 Hybrid

In literature, there are largely two types of machine learning models: standalone models, where a single algorithm is used for prediction, and hybrid models, where two or more algorithms are integrated to improve performance.

Although hybrid machine learning techniques are limited in the area of stock market forecasting, recent studies have demonstrated the increasing popularity of these techniques. Studies in traditional ANN combined with Decision Trees (DT) have reported successful results (> 77%) (Tsai & Wang, 2009). Other hybrid models, including GA (Genetic Algorithm) and SVM, have demonstrated that optimisation of GA improves performance significantly (Choudhry & Garg, 2008). While SOM and Support Vector Regression (SVR) have revealed increased accuracy and reduced training time for daily stock index prediction (Huang & Tsai, 2009).

Paper (Rather et al., 2015) credits the hybrid method with achieving a high degree of accuracy due to several factors. Time series input manipulation with unique regression on a sliding window and training separate series allowed recurrent neural network (RNN) to include a search for smaller weights, and the optimisation model minimised prediction error. The author's hybrid model's performance was outstanding in terms of prediction against RNN for the stocks in the national stock exchange in India.

However, Guresen et al., (2011) found that two types of hybrid methods, generalised autoregressive conditional heteroscedasticity (GARCH) and dynamic artificial neural network (DAN), did not perform better than the single ANN architecture. Applying multi-layer perceptron (MLP) and DAN with GARCH to NASDAQ index with one to four lags resulted in the classical ANN model MLP outperforming the hybrid forecasted results. Poor performance is likely contributed to by the architecture of the hybrid model, GARCH and DAN, which behaves like a statistical method.

Despite mixed hybrid method results, generally significant performance increase and accuracy are attributed to the hybrid approach. Since the hybrid approach does not ensure more accurate forecasting, time series and inner dynamics of the hybrid models should be researched further to improved architecture and lead to more powerful financial time series forecasting.

2.2.5 Ensemble

The ensemble method is a technique for combining the predictions of multiple classifiers to produce a single classifier (Breiman, 1996; Perrone, 1993; Wolpert, 1992), used to reduce the generalisation error. This can be described by training *n* different models separately and allowing each model to contribute to prediction. In general, the resulting classifier is more accurate than the individual classifiers making up the ensemble.

Chen et al. (2007) applied a flexible neural tree (FNT) method with its structure and parameters optimised by heuristic algorithm to NASDAQ100 and S&P500 indexes. Comparison of basic, generalised and local weighted polynomial regression (LWPR) ensemble approaches demonstrated that all ensemble learning paradigms performed well, with the LWPR resulting in higher correlation and the lowest error values.

Similarly, Melin et al. (2012) applied two different ANFIS ensemble models on indexes for next day price prediction. With five ensemble models using the weighted average and average approaches, predicted results ranged from 97% to 98% in accuracy. However, there was no comparison with other methods to verify that the approach provided better results.

Pulido et al. (2014) found that the best ensemble architecture of MLPs was optimised with a genetic algorithm, namely the number of individual MLPs and their parameters. Simulation of the ensemble model produced better results than other optimisation techniques.

The studies reviewed here inspired the aim of this project, which is to create an ensemble model comparing results against a hybrid model. Although an ensemble ANFIS model has been researched (Melin et al., 2012), no benchmark against a hybrid method has been applied, nor have international developed countries' indexes been studied in this context. Research providing a novel comparison of an ensemble ANFIS against a hybrid SOM with ANFIS can explore the efficiency and accuracy of the method in time series prediction and financial decision making. Furthermore, to this point, Melin et al. (2012) identified this as a future work to verify that their proposed approach produced useful results and to assess other researchers developed future models.

2.3 Summary

This chapter has presented an overview of financial time series and accomplishments in machine learning and artificial intelligence approaches to the stock market. A quantified search of literature was conducted to increase the author's knowledge on the relevant topics.

Initially, financial stock market significances with relevant stockholders were briefly investigated. The particular focus was on policy and economic benefits regarding the forecasting of stocks.

Existing academic understanding of efficient market hypothesis was briefly described. Efficient market hypothesis was described as being a useful heuristic concept with numerous recent papers demonstrating the ability of models to outperform the market. This discussion was followed by a presentation of the inability to more accurately forecast price due to factors relating to the nature of the data. Evidence presented has demonstrated that the nature of stock market is to a certain extent suitable for machine learning solutions.

A significant portion of the chapter was dedicated to research design concerning machine learning techniques, incorporating previous 'price of time' series. There was a specific focus on ANFIS, SOM techniques, hybrid and ensemble models within the last 15 years. Advantages and weaknesses along with similarities and differences have also been discussed.

2.3.1 Gaps

The literature review has revealed a wide range of publications on the topic of stock market forecasting using machine learning techniques. A number of gaps have been identified, and research aims have been formulated.

Given the scarcity of evidence regarding the use of ensemble ANFIS learning techniques in the financial prediction task, a large contribution will be made through this investigation into a novel approach. The study designs a well validated ensemble ANFIS model and comparing it with hybrid SOM for forecasting with five financial indexes.

Evidence on investigating times series stock data nature for insights through SOM has been identified through the research. The aim of the study is to reveal new patterns characteristic of developed economies in terms of five financial indexes, assisting in significant overall strategic management and economic development.

Enhanced financial and forecasting modelling as a result of improved machine learning techniques for stock markets can add to the current body of literature and the scientific community.

With regard to the gaps identified in the current research regarding the major five indexes, no study has focused on the utilisation of ensemble ANFIS model approach and compared it with a hybrid SOM and ANFIS for the major indexes, especially over ten years. This project aims to fulfil this gap.

We hope to reveal patterns and characteristics of developed economies in five financial indexes, assisting with significant overall strategic management and economic development.

Chapter 3

Methodology

This chapter investigates a novel methodology applied to financial indexes for future prediction of prices. The initial section will be a survey of the collected data and features extracted, inferring characteristics of prices for modelling. The second section investigates the theory of the adopted machine learning algorithm and techniques, then offers reasons for parameter selection. Finally, the performance measure section will describe how error metrics will capture advantages, limitations and overall performance of the model.

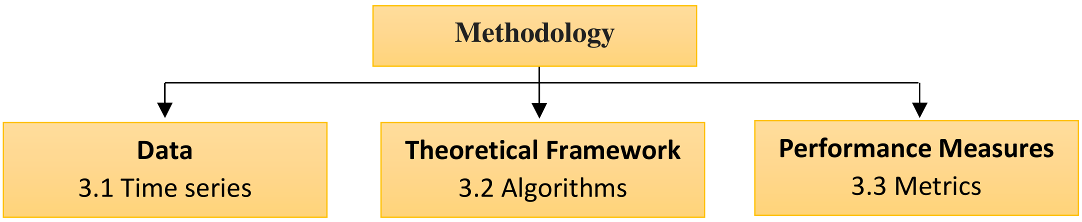


Fig. 3.1 Chapter 3 structure

3.1 Data

An initial assumption is that the stock market is not purely random. This paper investigates the stochastic nature and the EMH, suggesting that movements of the stock market follow random walk theory. It is important to understand the nature of the data and dynamics of the stock prior to the theory of predictive modelling.

The study uses daily *adjusted closing prices of All Ordinaries, Financial Times Stock Index, Dow Jones Industrial Average, Hang Seng Index and Nikkei Index from 1 January 2008 to 1 January 2018 that have been sourced from **Yahoo finance API.

Stock Market Index	Country	Ticker	Constituents
All Ordinaries	Australia	AORD	500
Financial Times Stock Index	England	FTSE	100
Dow Jones Industrial Average	USA	DJI	30
Hang Seng Index	Hong Kong	HSI	50
Nikkei Index	Japan	N225	225

Table. 3.1.1 Input stock indexes descriptions

The models in the study investigate and apply the actual returns value of indexes, represented as times series. Financial time series are a sequence of vectors $X_t = (x_{t-n}, x_{t-i}, \dots, x_{t-2}, x_{t-1})$ where x_{t-i} represents past values that vary with time. Figure 3.1.1 depicts the daily movement in value where the five indexes series represents the nature of stock prices as non-linear, dynamic, nonparametric and complex by nature.

Prior to modelling the data is to be normalised (Figure 3.1.2), described as follows.

$$X_{t-i} = \frac{X_{t-i} - X_{min}}{X_{max} - X_{t-i}}$$

**adjusted – "Adjusted close is the closing price after adjustments for all applicable splits and dividend distributions. Data is adjusted using appropriate split and dividend multipliers, adhering to [Center for Research in Security Prices (CRSP)] standards"*
https://sg.help.yahoo.com/kb/finance-for-desktop/adjusted-close-sln28256.html

***Yahoo finance API - http://download.finance.yahoo.com/d/quotes.csv*

Index	Min	Max	Median	1st Quartile	3rd Quartile	Std. Dev	Kurtosis	Skewness	Variance	NA
AORD	3112	6206	5047	4513	5486	637	-0.401	-0.483	405458	326
DJI	6547	24585	13562	11407	17575	3883	-0.706	0.287	15081143	326
FTSE	3512	7562	6053	5508	6709	859	-0.135	-0.478	737167	335
HSI	11016	30003	22286	20504	23636	3023	1.471	-0.572	9136835	387
N225	7055	22939	13372	9625	16886	4136	-1.199	0.349	17105591	404

Table. 3.1.2 Stock indexes statistics

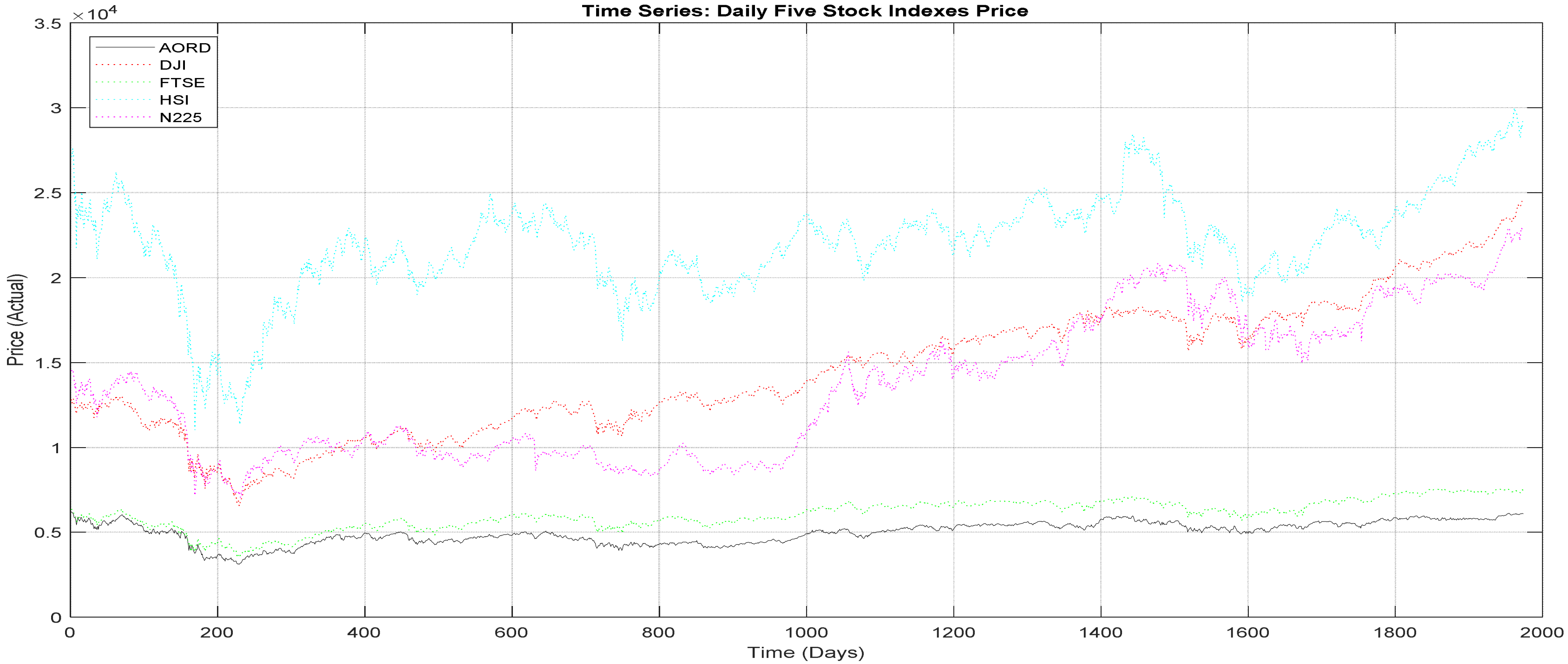


Fig. 3.1.1 Actual daily returns of five stock indices (1 January 2008 – 1 January 2018)

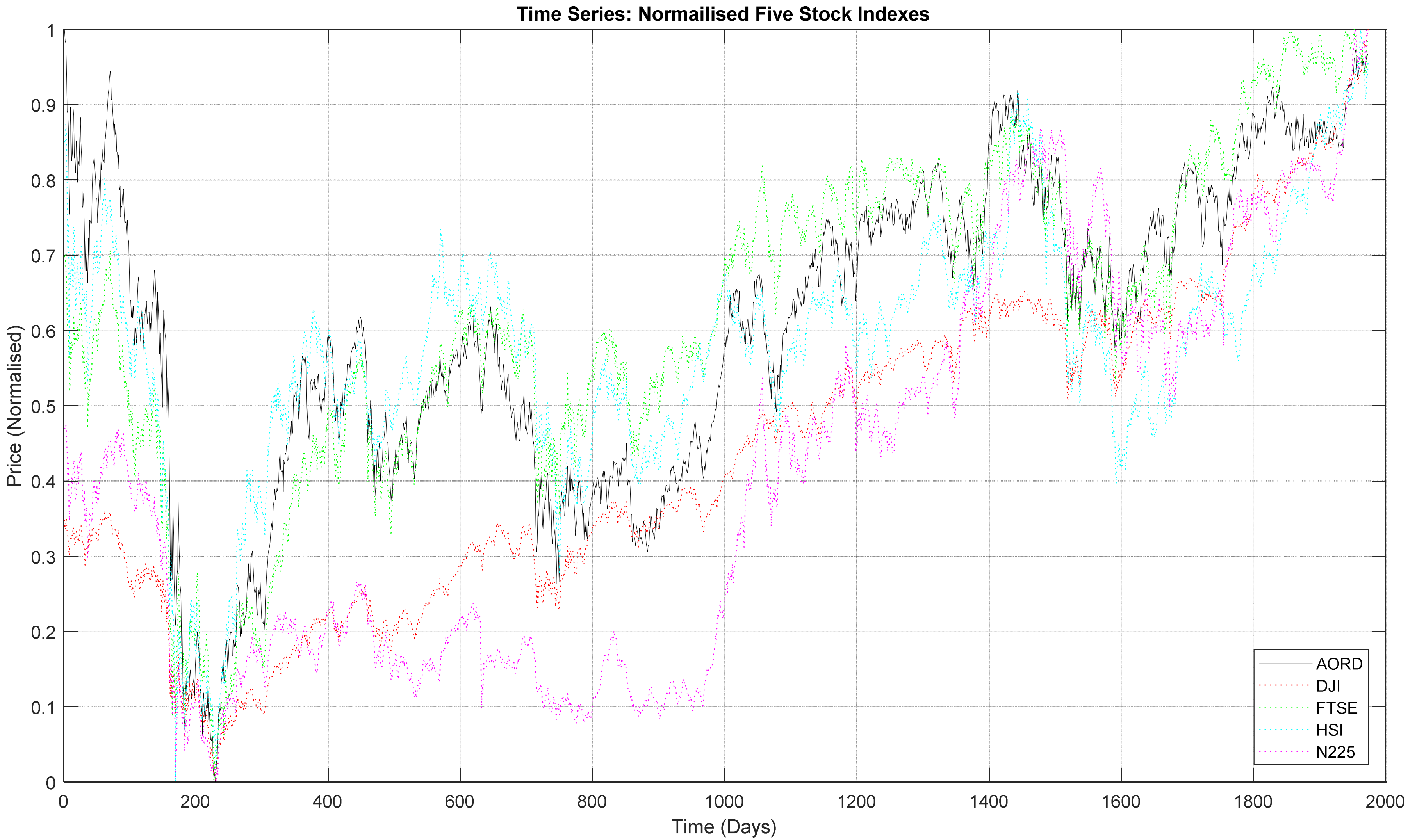


Fig.3.1.2 Normalised daily stock value of five stock indices (1 January 2008 – 1 January 2018)

A preliminary investigation was performed to display the relationship between the time series data. The correlation matrix plot (Figure 3.3) below demonstrates that the AORD, DJI, FTSE, HIS and N225 are highly correlated. A strong positive linear relationship exists between features DJI, FTSE, HIS, N225 and the AORD (greater than 0.8) with DJI having weakest (0.83) and FTSE the strongest (0.89).

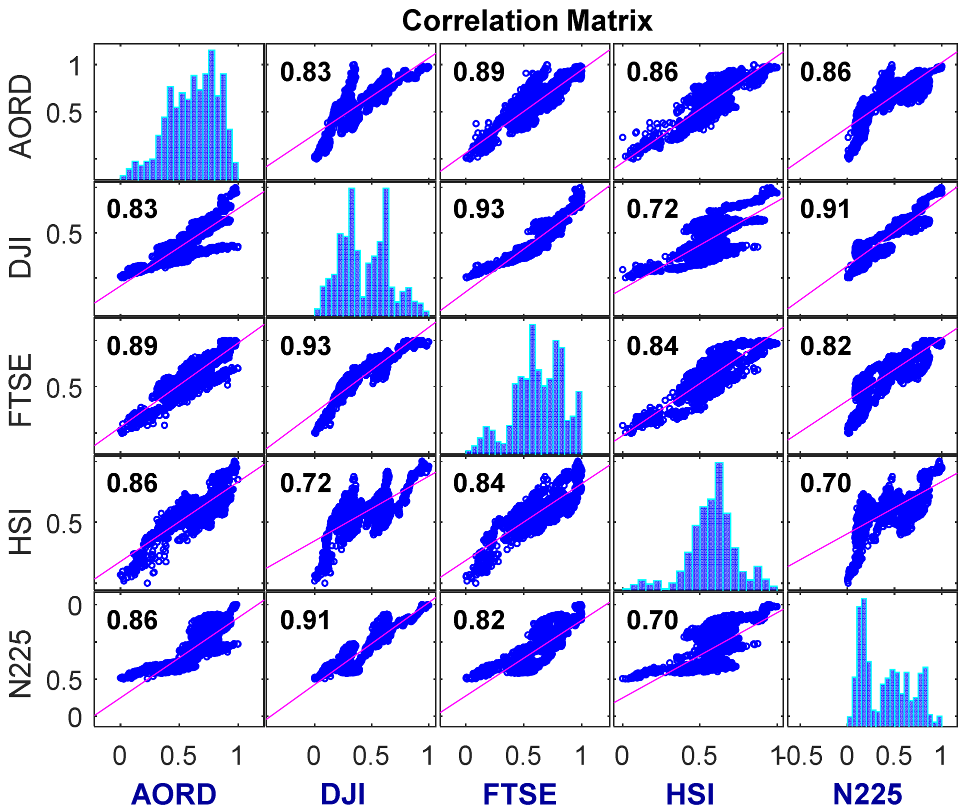


Fig.3.1.3 Indexes correlation matrix

To investigate the dependency of input variable with itself, the autocorrelation function (ACF) is the linear correlation with the signal at two different points in time, function of the lag *h* between two points displayed below.

$$acf(h) = corr(x_t, x_{x+h1})$$

Similar to ACF the partial autocorrelation function (PACF) determines linear dependency with signal at shorter lags removed, ACF between x_t and x_{x+h1} without the contribution of $x_{t+1}, x_{t+2} \dots, x_{t+h-1}$.

Figure 3.1.4 demonstrates PACF spike at lag, t-1 of greater than 90% for all indexes. This supports the significant correlation of one day (unit timescale) and the use of chosen input variable for ANFIS model development.
The ACF demonstrates pattern similarity across time, highlighting the non-stationarity nature of data.

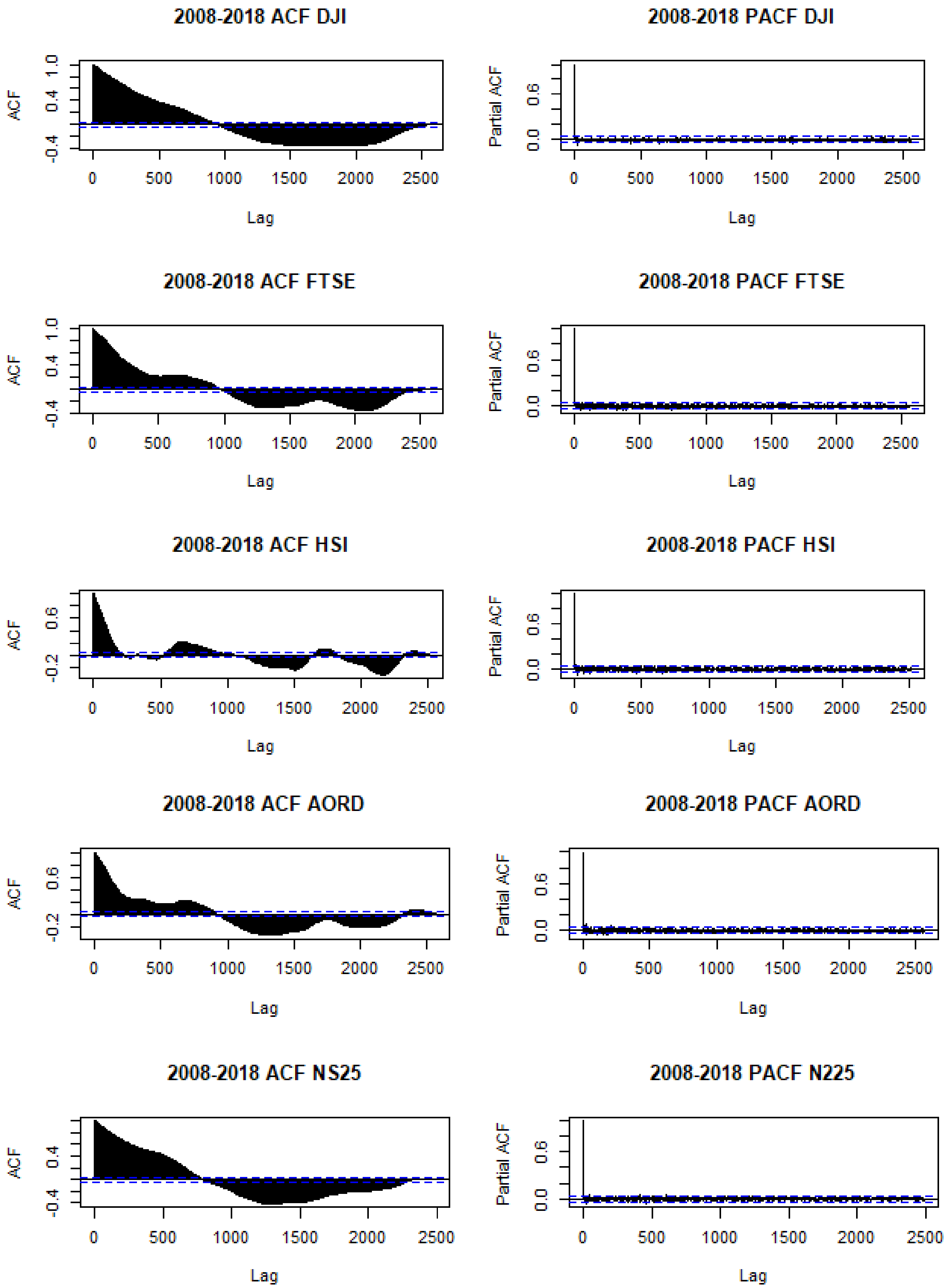


Fig.3.1.4 ACF (autocorrelation function) and PCAF (partial autocorrelation function) of daily stock value

Data investigation continued for the presence of trends (periodic and non-periodic), and statistical test (parametric and non-parametric) was performed to infer characteristics of prices and other nonlinear dynamic methods. However, given the above preliminary investigation on dataset particularly the PACF and literature review to date. Our proposal is to proceed with machine learning algorithms, ensemble based ANFIS and SOM as a viable solutions.

3.2 Theoretical Framework Methodology

For the project, we include a two-staged machine learning method. Initially implementing an ARIMA against ANFIS model then followed by SOM-ANFIS hybrid against ensemble ANIFIS model, benchmarking results against other machine learning techniques.
In the first stage, best performing conventional ARIMA Model will be developed to be compared against best performing ANFIS model.

The second stage, develops SOM clusters are developed to be selected as features for the ANFIS model. SOM clustering of historic time series data into disjointed clusters that may enhance recognition and classification. ANFIS model then applies fuzzy rules such as prior knowledge which are inserted into the network which may provide a high probability to converge to more accurate result compared to that of a traditional ANN.

Next, the study implements an ensemble ANFIS model. Features are not grouped, however, an integrated ensemble ANFIS approach is applied to chaotic time series where the goal is to minimise prediction error through an average or weighted average. The performance of the model is compared against the first stages for accuracy and computational complexity of time series prediction.
The following section outlines the SOM and ANFIS algorithm and criteria's in the parameters selection.

3.2.1 Auto-Regressive Moving Average model

A commonly used numerical forecasting model for market time series, the ARIMA model systematically characterises past, current and future trends in the data. Generally the data model consists of a autoregressive (AR), integrated (I) and a moving average (MA) component of varying values, which may be identified in seasonal and/or non-seasonal occurrences (Box et al., 2015).

The general form of ARIMA is as follows:

$ARIMA\ (p, d, q) - Non\ Seasonal$

Autoregressive, AR(p) relates to the current value of the time series to past values of order p. Moving average, MA (q) relates to the past forecast errors of order q and differencing (d), and adjusts for non-stationarity.

$ARIMA\ (p, d, q)(P, D, Q)S - Seasonal$

Seasonal AR (P) is the current value of time series to regular (period frequency, S) past values of order P. Moving average (Q) relates to regular (Period S past forecast errors of order Q. Differencing(D) is, adjusted for seasonal period nonstationary state.

Under the linear ARIMA models, it is required that the time series be free of any deterministic structures such as level shifts, local time trends and seasonal pulses (Harvey et al., 1999). Assumptions of ARIMA models are that the series have constant error variance and that the parameters of the proposed model remain constant over the course of time.

More appropriate ARIMA models may be determined through the three stages of model fitting: identification estimation and diagnostic check (Box et al., 2015). Stage one, identification, includes the selection of a set of more appropriate models through an examination of the ACF and PACF distribution of the time series. The recommendation or most optimally performing model is based on the combined testing for the minimum value of the Akaike Information Criterion (AIC) and Schwarz Bayesian criterion, also known as Bayesian information criterion(BIC) (Desai et al., 2005).

The Akaike Information Criterion is (Akaike, 1974) defined as follows:

$$AIC = -2\mathrm{k} - 2\ln(\hat{\mathrm{L}})$$

Eq.1 Akaike information criterion, where k is the number of parameters and $\hat{\mathrm{L}}$ is the maximum value of likelihood function.

The Schwarz Bayesian criterion (Schwarz,1978) is defined as follows:

$$AIC = \ln(n)\mathrm{k} - 2\ln(\hat{\mathrm{L}})$$

Eq.2 Bayesian information criterion, where k is the number of parameters, n the number of data points and $\hat{\mathrm{L}}$ the maximum value of likelihood function.

3.2.2 Self-Organizing Map (SOM)

Kohonen (1982) proposed the SOM to be an unsupervised and competitive learning algorithm. In the context of this project the data is manipulated in adequate way for feature assignment.
Topological maps are formed based on learning algorithm relating the input samples similarity, for example the conversion of higher dimensional space to a spatial relation among neurons representing inputs. Algorithm then updates weights of used neuron but those within spatial proximity based on activation zones of each neuron.
The steps in training, as shown below are repeated for weight adaptation and competiveness until stopping criteria is met.

1. A vector x input is introduced into the net.

2. Through the competitive process, distance between input vector and all neurons of output layer are computed to find the winning neuron (Eq.1).

$\|x - w_i\| \; i = 1,2, \dots, l$

Eq.1 Where l is the number of output neurons; w is the weight vector.

3. Through the weight adaptation process, the new weights are updated at time t + 1 of the wining neuron and the neighbouring neurons defined by activation zone.

$$w_i(t+1) = w_i(t) + \eta(t)\pi_{i,i^*}(t)(x(t) - w_i(t))$$

Eq.2 Where the topological neighbourhood is $\pi_{i,i^*}(t)$ and learning rate is $\eta(t)$ dependent on lateral distance between the winner neuron i^* and neuron i. Neighbourhood function is Gaussian-like, centred on winning neuron i^* defined by

$$\pi_{i,i^*}(t) = \exp\left(\frac{-\|p_i - p_{i^*}\|}{g(t)}\right)$$

Eq.3 Where the p_{i^*} is the wining neuron, the p_i is the position vector of the neighbourhood neuron and $g(t)$ is a parameter that gradually decreased.

The number of clusters used in the ANFIS model is determined by comparison prediction error testing. Mean absolute percentage error (MAPE), mean square error (MSE) and mean absolute error (MAE) are used to evaluate which number of clusters outperform the others.

3.2.3 Adaptive Neuro Fuzzy Inference System model

Adaptive Neuro Fuzzy Inference System was introduced by Jang in 1993 and derived from adaptive neuro-fuzzy inference system. This is an improved ANN technique that is a fuzzy inference system (FIS), employing the merits of both ANN and FIS on a common framework.

Artificial Neural Network
The ANN theory is elements operating in parallel to form a neural network, similar in principle to the biological nervous system. Connections between elements (neurons) are trained to a function by the adjustments of weights (connections) to elements, as demonstrates in Figure 3.2.3.1 below.

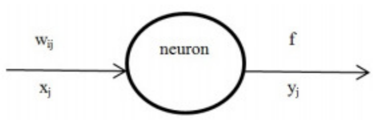


Fig. 3.2.3.1 Typical neuron in a neural network system

Typically ANN has an input, output layers and many (hidden) layers. x_1 is multiplied by its weight w_1 for $x_j w_{ij}$ product. The network is turned by comparing output and target until it converges to target.

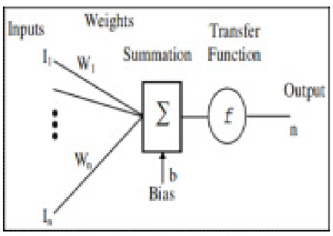


Fig. 3.2.3.2 Artificial Neural Network

Fuzzy Inference Systems

The FIS constructs membership function parameters that are tuned with either a back propagation algorithm or in combination with a least squares method. This allows the fuzzy system to learn from the data modelled.

For an FIS that has two inputs, x and y inputs and one output z. The first-order Sugeno fuzzy model has the following rules.

$$Rule\ 1\text{:}\ \ If\ x\ is\ A_1\ and\ y\ is\ B_1, then\ f_1 = p_1x + q_1x + r_1$$
$$Rule\ 2\text{:}\ \ If\ x\ is\ A_2\ and\ y\ is\ B_2, then\ f_2 = p_2x + q_2x + r_2$$

$$Therefore,\ \ output = \frac{w_1f_1 + w_2f_2}{w_1 + w_2}$$

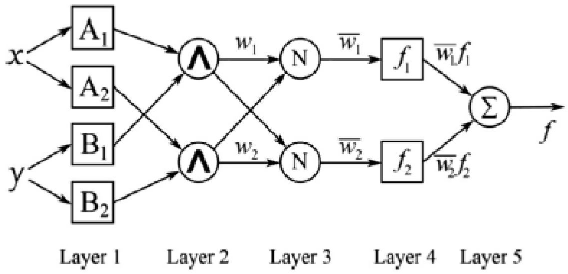
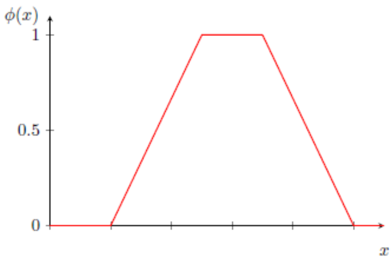


Fig. 3.2.3.3 ANFIS architecture

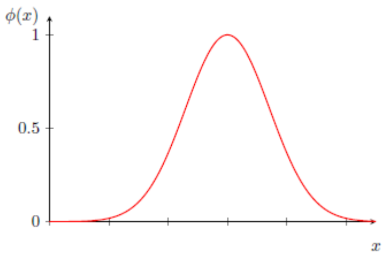
The membership function and their parameters brings the variation in ANFIS results. It is required for the model to select the correct membership function to achieve accurate results.



Trapezoid function,

$$trapezoid(x; a, b, c, d) =$$

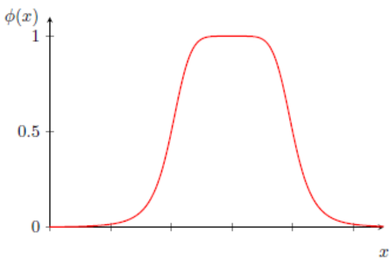
$$\max\left[min\left(\frac{x-a}{b-1}, 1, \frac{d-x}{d-c}\right), 0\right]$$



Gaussian function

$$gaussian(x; c, \sigma) =$$

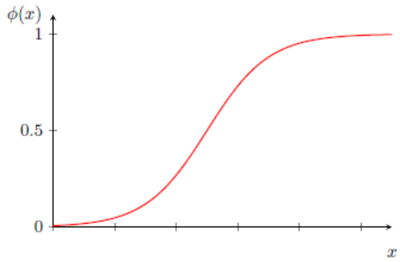
$$\exp\left[-\left(\frac{x-c}{\sigma}\right)^2\right]$$



generalised bell function

$$generalised\ bell\ (x; a, b, c) =$$

$$\frac{1}{1+abs\left(\frac{x-c}{a}\right)^2}$$



sigmoid function

$$sigmoid(x; a, c) =$$

$$x\left[\frac{1}{1+exp(-ax+ac)}\right]$$

Fig. 3.2.3.4 ANFIS membership functions

Learning

Learning can be described as the use of observations to solve a task optimally. The ANN represents a class of functions F that attempts to solve a certain task. The learning process tries to find, $f^* \in F$, to optimise the task. Learning can be further described as minimising a cost function $C: F \rightarrow \mathbb{R}$ such that $C(f^*) \leq C(f)$ for all $f \in F$.
This section introduces learning for an ANN, describing the chosen cost function, then introducing optimisation algorithms and, lastly, presenting a modification of the optimisation algorithm for the special case of ANN, the ANFIS.

Cost function

ANN allows for modelling linear as well as nonlinear relationships between the input and output space. To model the relationship of function $f^* \in F$ the parameter set of $f^*$is required. A measure of how well the ANN reflects the sought-after relationship is the cost function, C. The cost function C then evaluates the residuals ep with $p = \{1,2 \ldots, P\}$, which are defined as the difference between the observed output y_p and the output predicted by the model $\hat{y}_p$:

$$e_p = y_p - \hat{y}_p$$

There are various cost function that may be chosen. This study follows Jang (1993) defined cost function as the sum of square errors of p^{th}observations.

$$E_p = \sum_{k=1}^{N(L)} e^2{}_{p,k} = \sum_{k=1}^{N(L)} \left(y_{pk} - \hat{y}_{p,k}\right)^2$$

$N(L) > 1$ represents a case of ANN with multiple outputs, where L denotes layer. The term y_{pk} is a the actual observed output for p^{th} observation in the k^{th}variable, the $\hat{y}_{p,k}$ denotes the prediction equivalent. The overall cost function including cost functions of all P observation is defined as

$$E = \sum_{p=1}^{P} E_p$$

Backpropagation Method

Ultimately the parameter set determines the ANN output. The Figure below displays the relationship between the change of parameter and the overall cost function. That is the optimisation of E is an optimisation with respect to parameters of the ANN.

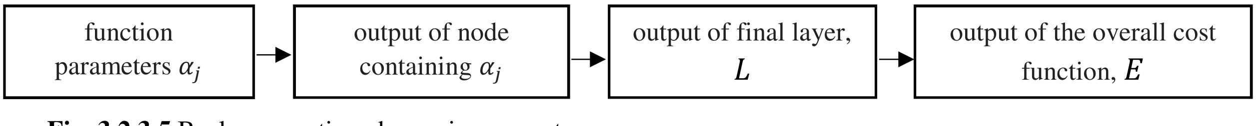


Fig. 3.2.3.5 Backpropagation change in parameters α_j.

There are a number of different methods for training an ANN. The backpropagation method is a commonly used method of computing the gradient of a cost function. The ANFIS utilises this gradient descent optimising technique to minimise the cost function. Where the gradient is a generalisation of a one-dimensional concept of the function derivative and the important property of the gradient is the direction of greatest rate of increase. Minimisation of the descent utilises the negative gradient that points in the opposite direction (local minima) of the greatest rate of increase. The concept is to create sequences that ‘wanders’ in each iteration a step by step moving further in the direction of greatest decrease until the minimum is reached. If the minimum is reached the sequence $(x_1, x_2 \ldots)$ converges.

For each parameter α the updated formula is as follows:

$$\Delta\alpha_i = -\eta \frac{\partial E_p}{\partial \alpha_i}$$

$$\text{Where } \eta = \frac{K}{\sum_{p=1}^{P}\left(\frac{\partial E}{\partial \alpha_i}\right)^2} \quad and \;\; K \; is \; step \; size$$

Hybrid learning Rule

Due to the gradient descent process the backpropagation method has a disadvantage in being computational intensive. Jang (1993) proposed the hybrid learning rule (HLR) to reduce the computation requirements in the learning process. This algorithm uses a combination of the least-squares and backpropagation gradient descent methods to model the training data set.

	Forward Pass	Backward Pass
Parameter 1	Fixed	Gradient Descent
Parameter 2	Least Square Estimation	Fixed
Signals	Node outputs	Error Signals

Table 3.2.3.1 Two pass hybrid learning rule.

However, hybrid learning has an assumption of linearity in some of its parameters, critical in the least squares estimation. The Sugeno inference system suits HLR as it contains two groups of parameters. Where parameters describing the fuzzy sets' MFs is contained in the first group and the consequent function of the fuzzy if-then rules are included in the second group.

Improving the adaptability of HLR classical ANFIS we will design x membership framework to attain to more accurate results. SOM hybrid, ensemble techniques is explored as a learner function as there are several lots of training performing on the same objective function. Comparison of x training and testing models will be determine through performance matrix

The next section will cover an evaluation of all simulations, with the absence of a standard framework Dawson et al (2007) suggest a variety of indices to be used.

3.3 Performances Measures

To discuss the performance of the ARIMA, ANFIS, SOM-ANFIS and Ensemble ANFIS based models the following mathematical assessment metrics are recommend for evaluating models simulated results against observation. (Dawson et al., 2007; Deo et al., 2016; Legates and McCabe, 1999; Willmott, 1981; Willmott, 1982; Willmott, 1984).

Regression Statistics

Coefficient of Determination (R^2)

$$R^2 = \left(\frac{\sum_{i=1}^{n} (Obs_i - \overline{Obs})\,(Sim_i - \overline{Sim})}{\sqrt{\sum_{i=1}^{n} (Obs_i - \overline{Obs})^2}\,\sqrt{\sum_{i=1}^{n} (Sim_i - \overline{Sim})^2}} \right)^2 \qquad (1)$$

$Where,$

$- Sim\ referes\ to\ the\ simulated\ values\ for\ predicted\ All\ Ords\ daily\ priceds, produced$
$from\ machnine\ learning\ models.$
$- Obs\ is\ the\ observed\ actual\ values\ of\ All\ Ords\ daily\ allocated\ from\ origninal\ dataset\ for\ comparision.$

The R^2 can be interpreted as the ratio of variation of the observed data that is explained by the predicted values. The range of values or $R^2$is 0 to 1, with 1 meaning the observed value can be used to predict the simulated values without error (while 0 is the opposite). Although used widely, this measurement is overly sensitive to high values and insensitive to new values of proportional differences. The $\boldsymbol{r}$ can be used initially to interpret the initial the optimum number of hidden neurones, the highest correlation coefficient is the optimal ANN model structure and is why it is the performance metric used for model development.

Model Evaluation Statistic - Dimensionless

Nash–Sutcliffe coefficient (E_{NS})

$$E_{NS} = 1 - \left(\frac{\sum_{i=1}^{n} (Obs_i - Sim_i)^2}{\sum_{i=1}^{n} (Obs_i - \overline{Obs})^2} \right), 0 \le E \le 1 \qquad (2)$$

A normalised statistic, E_{NS} determines the relative magnitude of residual variance compared to measured data variance.
E_{NS} ranges form $-\infty$ to 1, 1 being optimal and values < 0 indicating the mean observed value is a better predictor than simulated and therefore it is unacceptable .
Similar to R^2 , the disadvantage of E_{NS} arises from squared value differences from observed and measured, with larger values overestimated and smaller values negated.

Willmott's Index of agreement ($\boldsymbol{d}$)

$$d = 1 - \left(\frac{\sum_{i=1}^{n} (Obs_i - Sim_i)^2}{\sum_{i=1}^{n} \left(|Sim_i - \overline{Obs}| + |Obs_i - \overline{Sim}| \right)^2} \right), 0 \le d \le 1 \qquad (3)$$

The index of agreement represents the ratio between the mean square and potential error (Willmott, 1984). The potential error is then represented as the sum of the squared absolute values of the distances from the predicted values to the mean observed values and the distances from the observed values to the mean observed value. d is a standardised measure and varies between 0 and 1, with 1 indicating a perfect agreement between the measured and predicted values, and 0 indicating no agreement at all. Wilmott's index of agreement is able to consider nonlinear relationships and therefore does not standardise according to the means and variances of the observed and simulated values. Additionally the index has the advantage over the $\boldsymbol{R^2}$ and $\boldsymbol{E_{NS}}$ as values are not calculated by squaring the value, hence larger values are not overestimated and smaller values are neglected. Willmott (1981) introduced a proposed index of agreement (3) to overcome the insensitivity of Nash-Sutcliffe efficiency (ENS) and coefficient of determination (R2) differences in the observed and simulated means and variances. Although the index (Legates and McCabe, 1999) overcomes the challenge of high values associated with squaring, there have been a number of studies that have reported high values that are acceptable in fitting models (Krause et al., 2005).

Legates-McCabe's($\boldsymbol{LM}$)

$$LM = 1 - \left(\frac{\sum_{i=1}^{n}|Obs_i - Sim_i|}{\sum_{i=1}^{n}|Obs_i - \overline{Sim_i}|}\right), 0 \leq d \leq 1 \qquad (4)$$

Legates-Mcabe's (LM) takes absolute values into account and provides appropriate weight to errors and differences. This is a simple and easy metric to interpret a relative assessment of model performance.

Error Index

Root mean square error ($\boldsymbol{RMSE}$)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{n}(Sim_i - Obs_i)^2}$$

When one is measuring the goodness of fit, including for observed and simulated data, the $root\ mean\ square\ error$ ($RMSE$) values should be as small as possible to indicate small deviations between observed and predicted values. Additionally, RMSE is more sensitive than the mean absolute error (MAE), when extreme values are involved. One can define RMSE as measuring the goodness of fit relevant to high values. This can be a disadvantage to metrics like MAE that are not weighted towards high or low values, and that instead evaluate all deviations from observed data, equally and regardless of the sign.

Relative Root mean square error ($\boldsymbol{RRMSE}$)

$$RRMSE = 100 \times \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{n}(Sim_i - Obs_i)^2}}{\frac{1}{N}\sum_{i=1}^{n}(Obs_i)} \qquad (6)$$

Relative root mean square error (RRMSE) is expressed relative to the variability of measurements of the goodness of fit, including for observed and simulated values. The RRMSE values are expressed in percentages as small as possible to indicate small deviations between observed and predicted data.

Mean absolute error (MAE)

$$MAE = \frac{1}{N}\sum_{i=1}^{n}|Sim_i - Obs_i| \tag{7}$$

Not weighting the higher or lower values, MAE evaluates all deviations from observed data. Like $RMSE$, smaller values indicate less deviations

Mean absolute percentage error (MAE) (8)

$$MAPE = 100 \times \frac{1}{N}\sum_{i=1}^{n}\left|\frac{(Sim_i - Obs_i)}{Obs_i}\right| \tag{8}$$

The mean absolute percent error (MAPE) error refers to the forecast. Percentages are summarised without regard to the sign and error. In terms of percentage, the smaller the better.

For the purpose of modelling results, particularly where errors are concerned, both RMSE and MAE are ideal in that they calculate the aggregation of residuals of both observed and simulated data. The RMSE is included in the squared residual results to measure the goodness of fit to high values, while MAE which does not include square values, suits a more equally spread distribution. The weaknesses of both metrics is that they are expressed in their absolute units. When relative errors are required, RRMSE and MAPE can be utilised.

Chapter 4

Results 1

In this section an ARIMA and ANFIS model is developed for forecasting the future index value and trends of All Ordinaries. The two models are analysed and compared for performance and effectiveness based on an evaluation of simulated results. As per Figure 4.1, a section for each model outlines the predictive model's development, including required parameters, results and discussion section. Finally, the conclusions present findings, comparing ARIMA and ANFIS models in context.

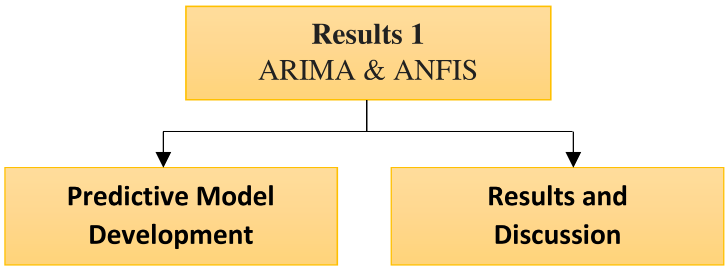


Fig. 4.1 Chapter 4 structure

4.1 Autoregressive Integrated Moving Average Model

The model of ARIMA development for All Ordinaries indexes forecast will be determined and discussed. First the configuration of the ARIMA model, which assumes a linear relationship between returns and lagged returns, is detailed.

4.1.1 ARIMA parameters

Descriptive statistics for the daily values of All Ordinaries (Table 3.2) are estimated for the purpose of illustration. The median value of the time series is estimated at 5,047, and the standard deviation at 637. The original time series is split for ARIMA model development, taking approximately 90% (1775/1973 days) input for model parameters generation. Therefore, all the analyses below on model development were conducted based on set time series of the All Ordinaries. Figure 4.2.1.1(top) represents the dataset that we analysed and is indicative of a non-stationary time series. This series varies randomly over ten years (January 1, 2008 to January 1, 2018), and there is no evident global trend or seasonal note. There was a notable sharp decline in the stock market at the end of the year 2008 (200 days mark), as this was during the period of the economic crisis which affected all global financial markets.

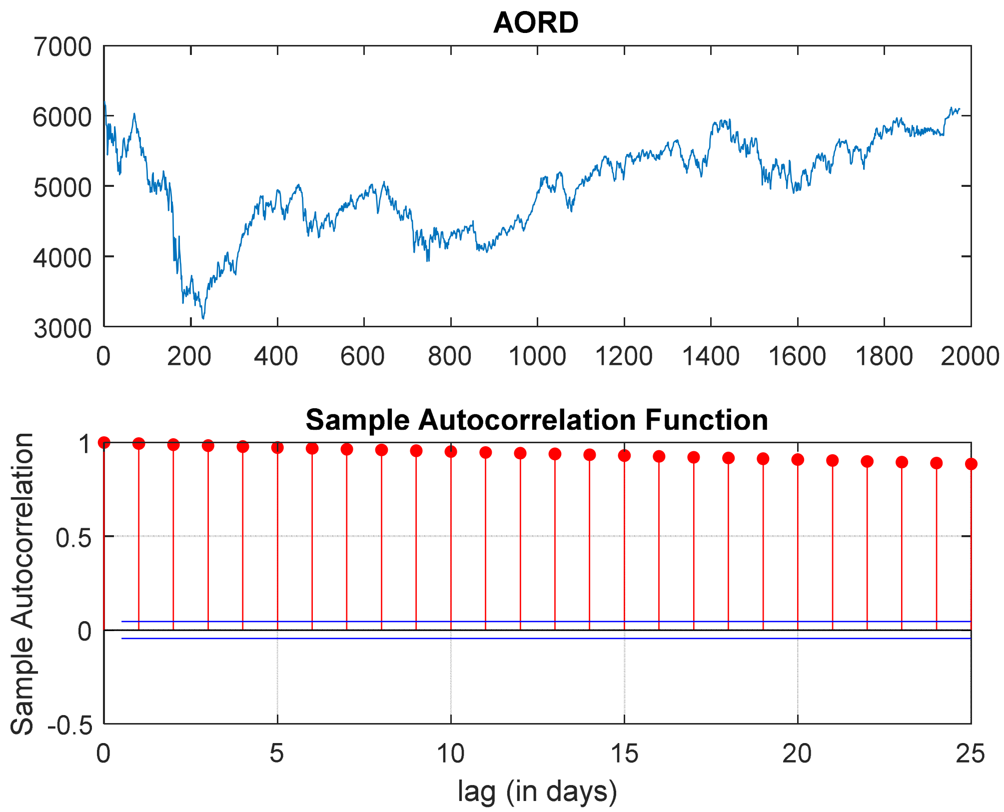


Fig. 4.1.1.1 Training time series (top) and ACF (bottom)

The autocorrelation function (Figure 4.1.1.1, bottom) of the All Ordinaries time series refers to all values that are 'significantly far from zero', where the only pattern is a linear decrease. In addition, the sample Partial Autocorrelation Function is also indeterminate with an initial increasing lag and cut-off after the first lag (bottom, 4.1.1.2). This typically implies that we are dealing with a correlogram of a non-stationary time series.

Using the original time series in order to achieve a stationary series, Figure 4.2.1.2(top) illustrates the first difference of the of the original All Ordinariess index time series. When examining the correlogram of the first difference, we can say that the All Ordinaries series has become stationary.

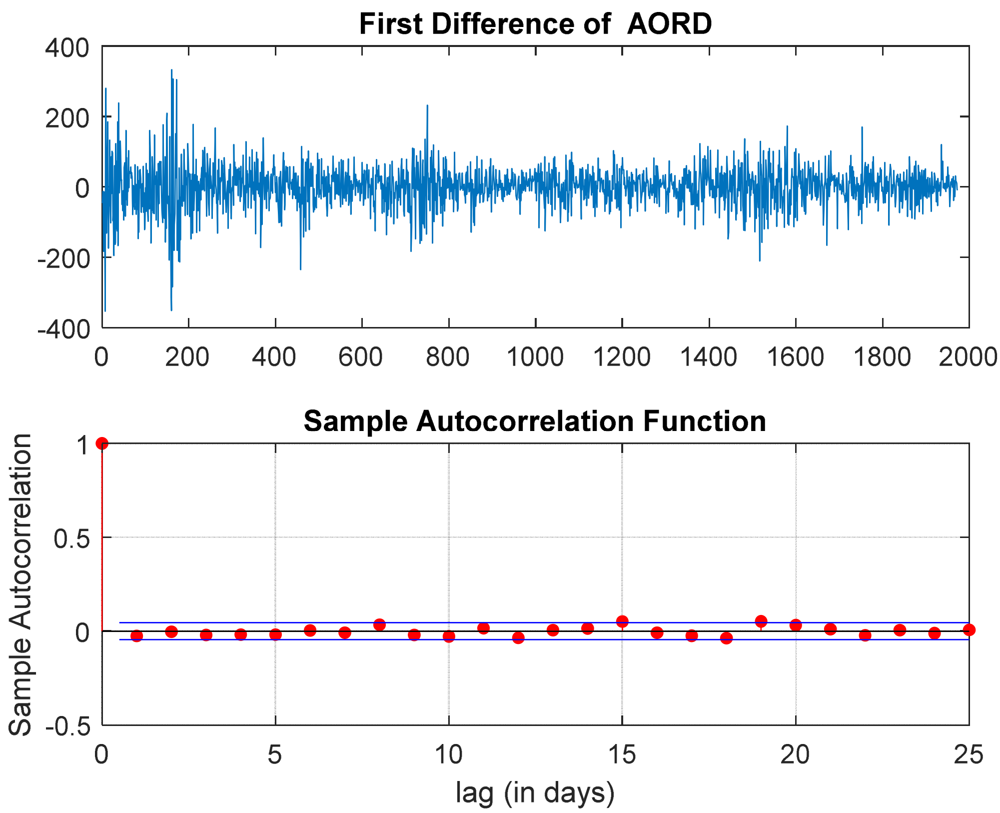


Fig. 4.1.1.2 First difference (top) and PACF (bottom) correlogram.

Fitting ARIMA(p,1,q) for the Stock price index

When we extend the model by allowing the AR polynomial to have one characteristic root, the model then becomes the ARIMA model. As it was seen, the time series are nonstationary and this fact implies the necessity to use the first-difference form data. Now, the time series is integrated of order 1, which means that we will have d=1 in the ARIMA(p,d,q) and seasonal ARIMA(P,D,Q)(freq) model. Next, we need to determine the order of autoregressive (p), moving average (q) and seasonal (P),(Q) parameters that are necessary to give an effective model fit.

The AIC are a widely used measure of ARIMA model goodness of fit and simplicity of the model statistics. In comparing various ARIMA models, the lower AIC is generally considered the 'better 'and therefore used to determine the model parameters.
In order to generate the ARIMA model's associated statistic (e.g AIC,BIC, loglikilihood) the tseries and forecast libraries under R, version 3.3.3, was used. Table 4.2.1.1 and Table 4.2.1.2 rank models on their performance from lowest AIC to highest and according to seasonal pattern. The results for this analysis demonstrate that the best model for daily prediction is the ARIMA (0, 1, 0), and for season, the optimal model is ARIMA(0,1,0)(0,0,1)[12].

	Type of ARIMA	Statistics			Parameters					
Rank	Structure (p,d,q)	AIC	Log likelihood	Sigma^2	AR1	AR2	AR3	MA1	MA2	MA3
1	ARIMA(0,1,0)	**19564.96**	-9781.48	3581						
2	ARIMA(1,1,0)	19565.66	-9780.83	3579	-0.0271					
3	ARIMA(0,1,1)	19565.67	-9780.83	3579				-0.0269		
4	ARIMA(2,1,0)	19567.62	-9780.8	3579	-0.027	0.0054				
5	ARIMA(0,1,2)	19567.64	-9780.81	3579				-0.0267	0.0049	
6	ARIMA(1,1,1)	19567.67	-9780.83	3579	-0.0136			-0.0134		
7	ARIMA(0,1,3)	19568.49	-9780.24	3576				-0.028	0.0051	-0.0255
8	ARIMA(3,1,0)	19568.58	-9780.24	3577	-0.0269	0.0047	-0.0242			
9	ARIMA(2,1,1)	19569.63	-9780.8	3579	-0.0132	0.0054		-0.0135		
10	ARIMA(1,1,2)	19569.64	-9780.81	3579	-0.0137			-0.0132	0.005	

Table 4.1.1.1 Ten best performing ARMIA daily models with characteristics

	Type of ARIMA Seasonal	Statistics			Parameters					
Rank	Structure (p,d,q)(P,D,Q)(freq)	AIC	Log likelihood	Sigma^2	AR1	AR2	AR3	MA1	MA2	MA3
1	ARIMA(0,1,0)(0,0,1)[12]	**1426.026**	-710.96	9052				-0.0558		
2	ARIMA(0,1,0)(0,0,2)[12]	1428.131	-710.96	9052				-0.0559	-0.0005	
3	ARIMA(0,1,0)(0,0,3)[12]	1429.332	-710.49	8948				-0.0642	-0.0104	-0.1069
4	ARIMA(0,1,0)(1,0,0)[12]	1426.024	-710.96	9052	-0.0557					
5	ARIMA(0,1,0)(1,0,1)[12]	1428.006	-710.9	9038	0.4926			-0.5565		

6	ARIMA(0,1,0)(1,0,2)[12]	1430.147	-710.9	9038	0.4969		-0.5597	-0.0021	
7	ARIMA(0,1,0)(1,0,3)[12]	1430.73	-710.1	8823	-0.6532		0.6017	-0.0471	-0.149
8	ARIMA(0,1,0)(2,0,0)[12]	1428.123	-710.96	9051	-0.0548	0.0088			
9	ARIMA(0,1,0)(2,0,1)[12]	1429.642	-710.65	8979	-0.7801	0.0234	0.734		
10	ARIMA(0,1,0)(2,0,2)[12]	1431.767	-710.62	8978	-0.2365	0.5095	0.1555	-0.5026	

Table 4.1.1.2 Ten best performing ARMIA Seasonal models with characteristic

4.1.2 Autoregressive Integrated Moving Average results

The results in the Table 4.2.1.2 and Figure 4.1.2.1 illustrate the prediction of the top four ARIMA models ranked for daily and seasonal prediction compared with the testing set (10%). As the model does not contain a great deal of autocorrelation, the forecasts quickly settle down to the mean of the series, and the forecast limits contain all of the actual values.

Testing Performance Matrix	Models				
	ARIMA (0,1,0)	**ARIMA** (0,1,1)	**ARIMA** (2,1,0)	**ARIMA** (1,1,0)	**ARIMA Seasonal** (0,1,0) (0,0,1)
r	0	0	0.105	**0.106**	0.096
Nash Ens	**-1.852**	-1.864	-1.871	-64.013	-1.872
Willmott Index	-3.405	-3.380	-3.389	-3.381	**-0.249**
Legates	**-0.686**	-0.695	-0.692	-0.695	-8.878
RMSE	**186.541**	187.201	186.957	187.176	890.686
rRMSE	**3.193**	3.204	3.200	3.204	15.246
MAE	**150.874**	151.667	151.373	151.638	883.826
MAPE	**2.549**	2.562	2.557	2.562	15.099

Table 4.1.2.1 Testing of ARIMA models performance

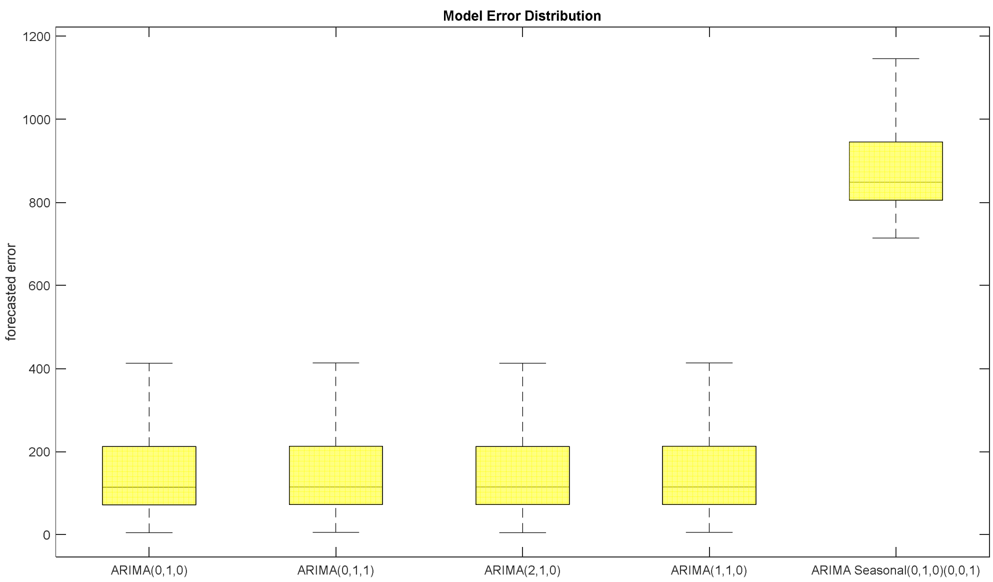


Fig 4.1.2.1 Boxpot distribution of ARIMA models absolute errors

The results of above table and figure indicate the poor performance of the non-seasonal model. This model has comparatively lower error values across the metrics, and the distribution of errors is well above the other models. The remaining top four ARIMA models have small discernible differences, though a case for the best model, ARMIA (0,1,0) can be made due its having the lowest RMSE at 186.5 and MAE at 150.8.

4.2 Adaptive Neuro Fuzzy Inference System model

The ARIMA model was not able to perform well for future prediction with large error metrics and poor correlations compared to the test set. Therefore, the next model, ANFIS, a more complex and highly parametric model was investigated and compared to the test set.

4.2.1 Adaptive Neuro Fuzzy Inference System parameters

As ANFIS is a high parametric model, the challenge is to control parameters in order to avoid overfitting. In this study ANFIS is based on first order Sugeno and Mamdani inference system with a number of indexes (Table 4.2.1.1). The amount of parameters depends on several factors:

1. Number of input variables: u
2. Number of MFs per input variable: v
3. Number of parameters of the chosen membership functions (MFs): x

The total amount of parameters can be calculated as $x.(v^u) + (u+1).(v^u)$ where v^u is the amount of rules for the ANFIS. The first summation contains the set of parameters defining all MFs, the second summation contains the set of parameters defining the first order polynomials.

As the amount of rules grows exponentially to the base of v and the power of u, controlling v and u has a significant impact on the total amount of parameters in the ANFIS model. Due to thedescribed impact of variable u's on the total amount of parameters and in order to address the Curse of Dimensionality the ANFIS model we will contain the number of inputs to five or less variables. Additionally the amount of MFs v per input variable will also be set to only two or lower. The three-parametric generalized bell-shaped MF has been chosen as MF more representative of nature of dataset. Therefore the total amount of parameters for the first ANFIS model, will have the following chosen configuration is $3.2^2 + (2+1).2^2 = 24$.

The training model process is now to be described, with the number of features (inputs, u) ranging from one to five: one day lag for All Ordinaries, Financial Times Stock Index, Dow Jones Industrial Average, Hang Seng Index and Nikkei Index. Splitting dataset into 80% training and 10% validation set.

In order to generate the FIS structure to implement ANFIS algorithm the fuzzy logic toolbox of MATLAB, version R2017b was used. ANFIS models GENFIS3 was preselected by hypothesizing best model structure relating to input/output data. A number of settings in GENSFIS3 were adjusted to the ANFIS networks to yield the most effective model, and the best most model was selected with the minimum errors RMSE/MAPE and high correlation (observed vs forecast).

Settings (Table 4.2.1.1) included the logic techniques, types of membership functions, the number of membership functions and the learning methods. Trials on two different structure models are shown diametrically in Matlab dialog box in figure 4.2.1.1 and 4.2.1.2. Generating and training a number of fuzzy inference systems based on above ANFIS structures were then optimised by tunning each parameter. FIS structure training output was plotted against the actual training data and errors (top and bottom figure 4.2.1.3). Interesting larger errors around the 200 days, where the financial crisis of 2008 occurred. However, outside of this event the overall ANFIS data matches well with the training data. To further improve the match, parameters including the number of membership function and epochs were tuned.

Figure 4.2.1.4 highlights at epoch 10 where the minimum validation error occurs. The increase in validation error (red) after this point suggests overfitting of the model parameters to the training data. Epoch 10 is therefore chosen when tunning FIS to exhibit the best generalisation performance.

Step size rate is monitored (Figure 4.2.1.5) in the training stages, by increasing the step size can improve the convergence, however too much may lead to poor convergence. The optimal step profile as shown, is initially increasing reaching the maximum then decreasing for the remaining of the training.

Membership rules are investigated in both the training and validation stage, since fuzzy c means clustering generates all possible rules, it is then evaluated and updated to produce better performing model. Figure 4.2.1.6, Matlab's Nuro-Fuzzy designer shows the effect of the number of MF's and their parameters has on model prediction, while Figure 4.2.1.7 illustrates the number and types or shapes of membership functions, in this case Gaussian.

Output variable is then examined against two other input variables as a surface, figure 4.2.1.8 and 4.2.1.9. As there are a number of input FIS variables in most model system, the input variables are changed in relation to the output variable.

Model Number	Technique	Type of mf			Number of mf		Learning method		
		Type	Clustering	Output	Input	Epoch	Models	Type	Features Combination
1	genfis 3 Sugeno	Gauss	FCM	Linear	3	10	Single	Hybrid	1 (t-1 All Ords)
2	genfis 3 Sugeno	Gauss	FCM	Linear	6	10	Single	Hybrid	5 (all Indexes t-1)
3	genfis 3 Sugeno	Gauss	FCM	Linear	4	10	Single	Hybrid	5 (all Indexes without All Ords t-1)
4	genfis 3 Mamdani	Gauss	FCM	Linear	2	15	Single	Hybrid	1 (t-1 All Ords)

Table 4.2.1.1 Characteristics of ANFIS model

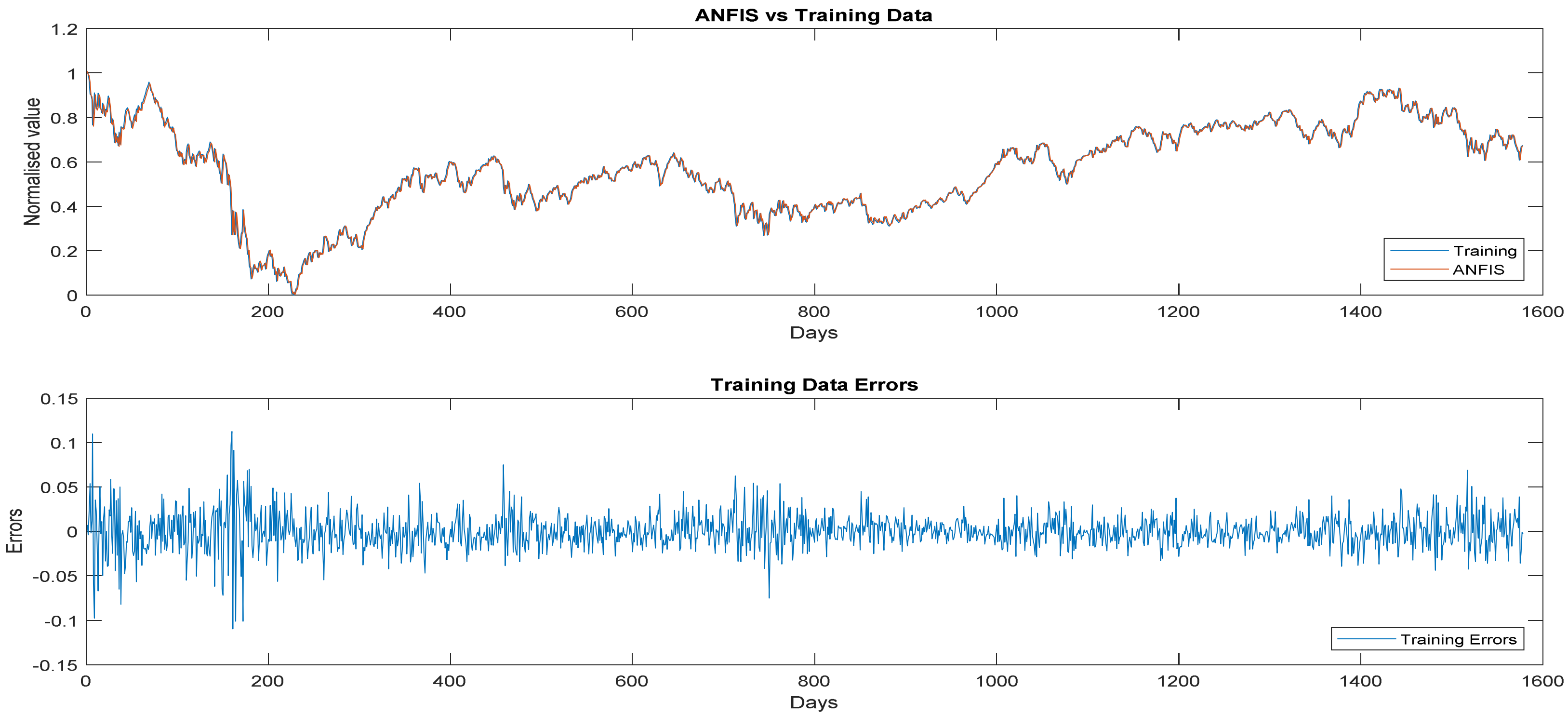


Fig 4.2.1.3 ANFIS model 1 training error and time series

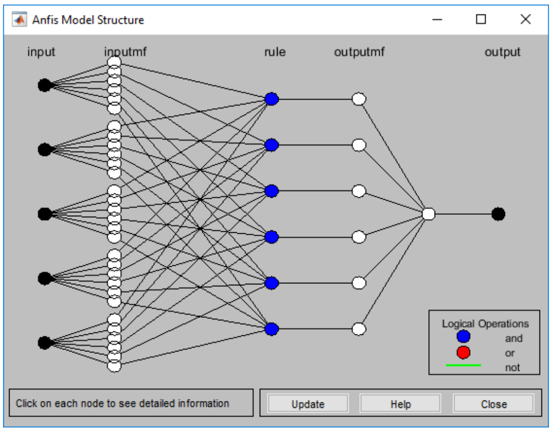


Fig. 4.2.1.1 Model 1 structure

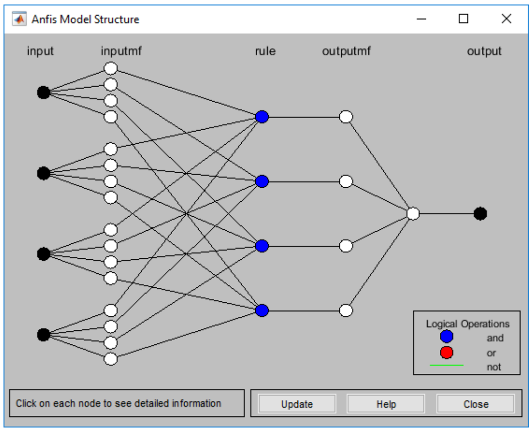


Fig 4.2.1.2 Model 2 structure

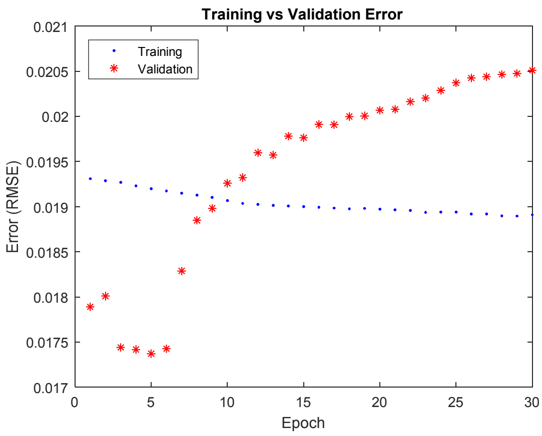


Fig. 4.2.1.4 Training and validation error epoch sample

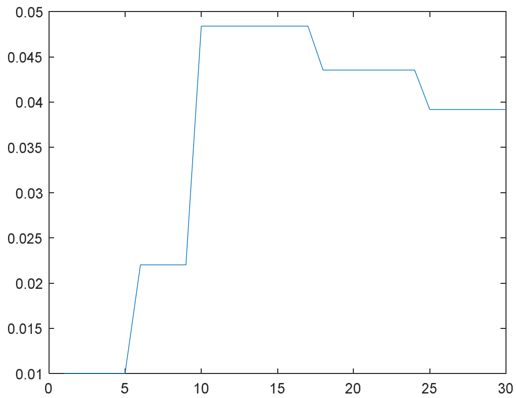


Fig. 4.2.1.5 Step size profile sample

Fig. 4.2.1.6 Model 2 mf rules

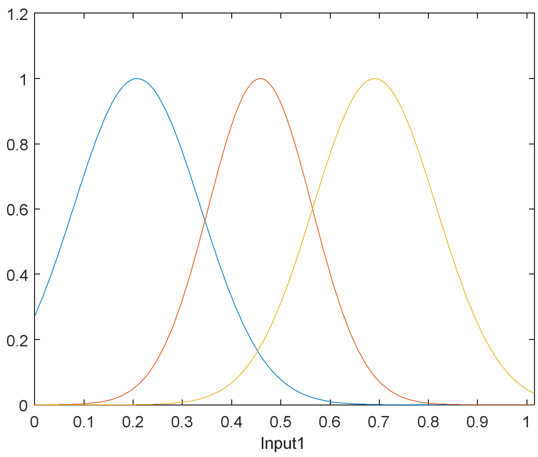


Fig 4.2.1.7 Model 1 mf distribution plot

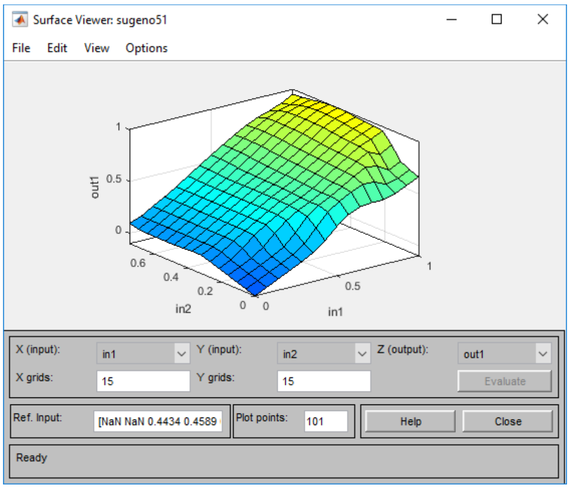


Fig. 4.2.1.8 Model 2 FIS surface

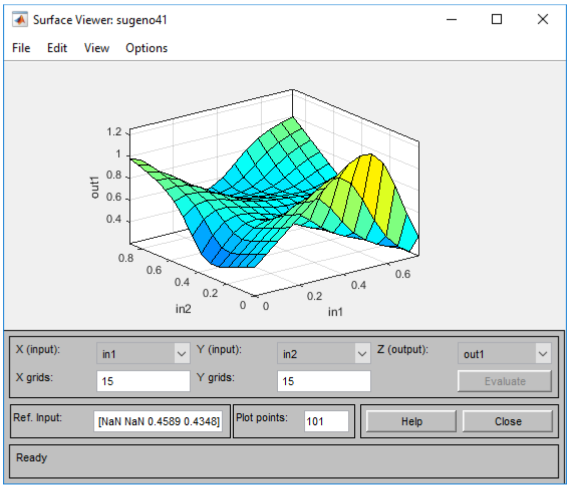


Fig. 4.2.1.9 Model 1 FIS surface

Model	Model		r	RMSE	MAPE
ANFIS	Type	Sugerno	0.9849	76.7927	1.0673
	Epoch	10			
	Membership function	gaussmf			
	Optimisation method	hybrid			
	orMethod	probor			
	Clusters	auto			
	dufuzzMethod	wtaver			
	impMethod	prod			
	aggMethod	sum			
	Input	1x5 - [0,1] [0,0.7444] [0,0.8868] [0,0.9178] [0,0.8696]			
	Ouput	1x1			
	rule	1x4			

Table 4.2.1.2 Model 2 parameters

Training Performance Matrix	Models			
	Sugeno (t-1 All Ords)	Sugeno (all Indexes t-1)	Sugeno (all Indexes without All Ords t-1)	Mamdani (t-1 All Ords)
r	0.9894	**0.9896**	0.5375	0.9894
Nash Ens	0.9788	**0.9792**	-26.472	0.9788
Willmott Index	0.9845	**0.9845**	0.2691	0.9845
Legates	0.858	**0.8599**	-5.3357	0.858
RMSE	109.145	**107.977**	578.986	109.145
rRMSE	2.1604	**2.1373**	9.9106	2.1604
MAE	82.8895	**81.8117**	566.854	82.8895
MAPE	1.720	**1.695**	9.719	1.72

Table 4.2.1.3 Training ANFIS models performance

4.2.2 Adaptive Neuro Fuzzy Inference System results

A comparison of ANFIS model results are presented in Table 4.2.2. The results of the experiments show that ANFIS Sugeno type with fuzzy c-mean clustering with Gaussian membership function performed best among other types of ANFIS function types. Further the two best models (*) based on r and RMS are the Surgeno with single feature and combination of all indexes features (t-1 day lag).

Training Performance Matrix	Models			
	*Sugeno (t-1 All Ords)	*Sugeno (all Indexes t-1)	Sugeno (all Indexes without All Ords t-1)	Mamdani (t-1 All Ords)
r	**0.946**	0.935	0.538	-0.782
Nash Ens	**0.861**	0.855	-26.472	-33.273
Willmott Index	0.937	**0.943**	0.269	-0.385
Legates	0.622	**0.636**	-5.336	-6.109
RMSE	**41.15**	42.113	578.986	646.695
rRMSE	**0.704**	0.721	9.911	11.07
MAE	33.841	**32.594**	566.854	636.006
MAPE	0.578	**0.56**	9.719	10.853

Table 4.2.2 Testing of ANFIS models performance

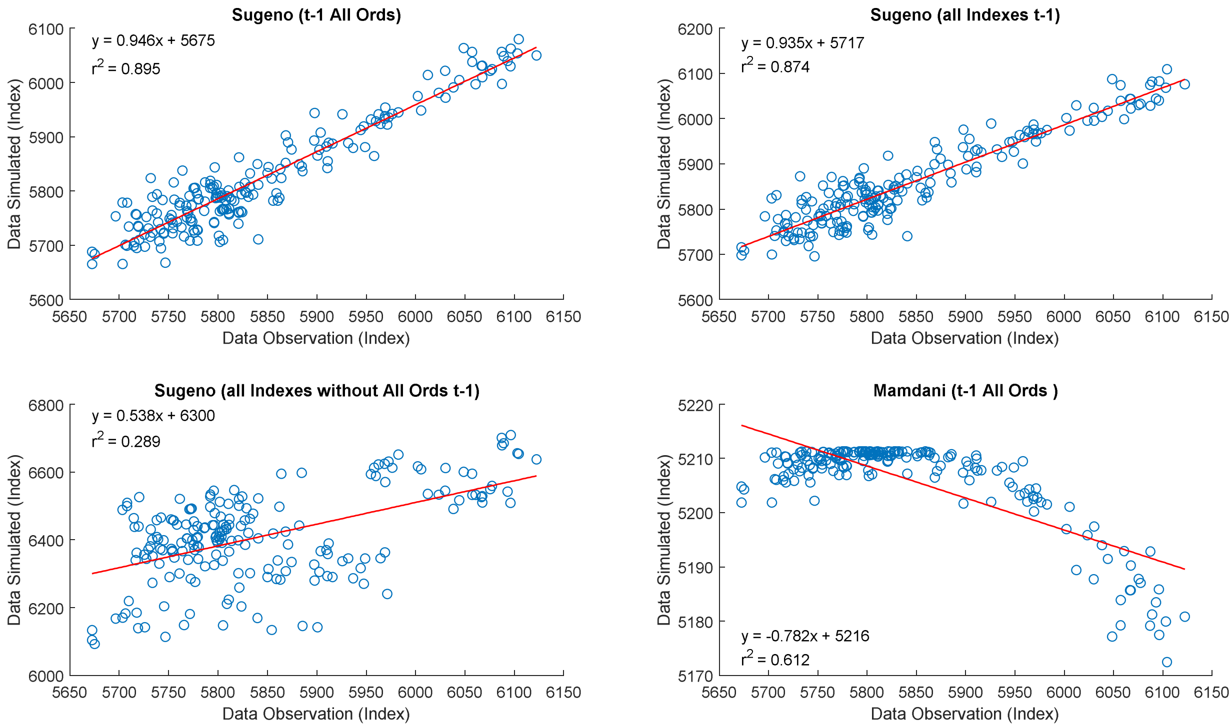


Fig. 4.2.2 Testing error observed vs simulated scatter plot

4.3 Conclusion

This study presents the analyses of ANFIS against the traditional ARIMA models performed over a number of different parameters. The results of the top two rank ANFIS and best rank ARMIA, are shown in table 4.3.1 and figure 4.3.1 where clearly the ANFIS models outperforms ARMIA model for forecasting accuracy. Figure 4.3.1 show ARIMA model (yellow) forecast values converging to a mean after just a few forecast values, while ANFIS forecasts follow closely the tested non-linear value (black).

The top two rank ANFIS model (Surgeon single and multiple feature) scored high r values of greater than 0.93 and RMSE of below 42. While the top performing ARIMA score a 0 r value indicating no relationship between observed and predictive values and a RMSE of 186, that is equivalent to 4 times the ANFIS model.
Further the error distribution in boxplots (figure 4.3.2), calculated absolute difference of observed to simulated values depicts ARIMA's significantly larger inter quantile range and the ANFIS models similarities in comparison.

We can conclude from forecasting financial time series, such as the All Ords indexes that the process is nonlinear in nature with a range of related factors such as interest rates changes, announcement of macroeconomic news and political events that affects the forecasting accuracy. Conducting an ARIMA (0, 1, 0) model best fit All Ordinaires among other ARIMA models, supported by statistics AIC, log likelihood and RMSE. Best fitted model showed forecasted results that were not accurate more than a few days in the future, however gave a benchmark to compare ANFIS model against.

ANFIS model using a hybrid optimisation algorithm with Sugeno system with 5 inputs features (t-1), six membership function and an epoch of 10 is the best fitting for All Ords forecasting. Forecasting accurately stock price index for several future points and perform very well when trained in economic and financial data, and thus it makes a great contribution as an efficient tool for forecasting in financial markets.

Training Performance Matrix	Models		
	Multiple Input (5 Indexes) **ANFIS Sugeno** model	Single Input (t-1) **ANFIS Sugeno** model	**ARIMA** (0,1,0))
r	0.935	**0.946**	0
Nash Ens	0.8547	**0.8612**	-1.852
Willmott Index	**0.9432**	0.9372	-3.405
Legates	**0.6357**	0.6218	-0.686
RMSE	42.1132	**41.15**	186.541
rRMSE	0.7209	**0.7044**	3.193
MAE	**32.5942**	33.8409	150.874
MAPE	**0.5599**	0.5779	2.549

Table 4.3.1 Performance ARIMA versus ANFIS.

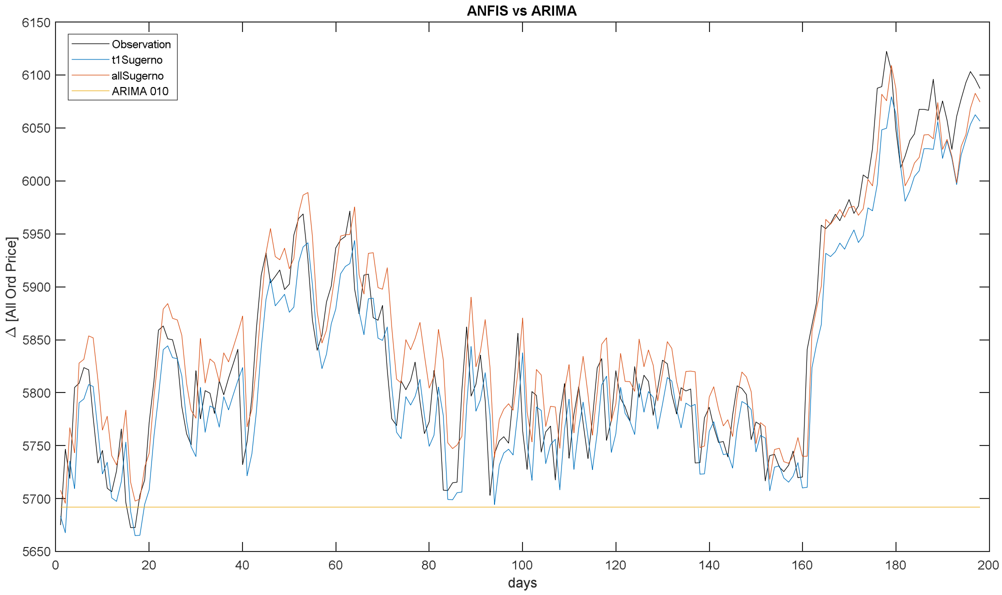


Fig. 4.3.1 ARIMA and ANFIS prediction against observation time series

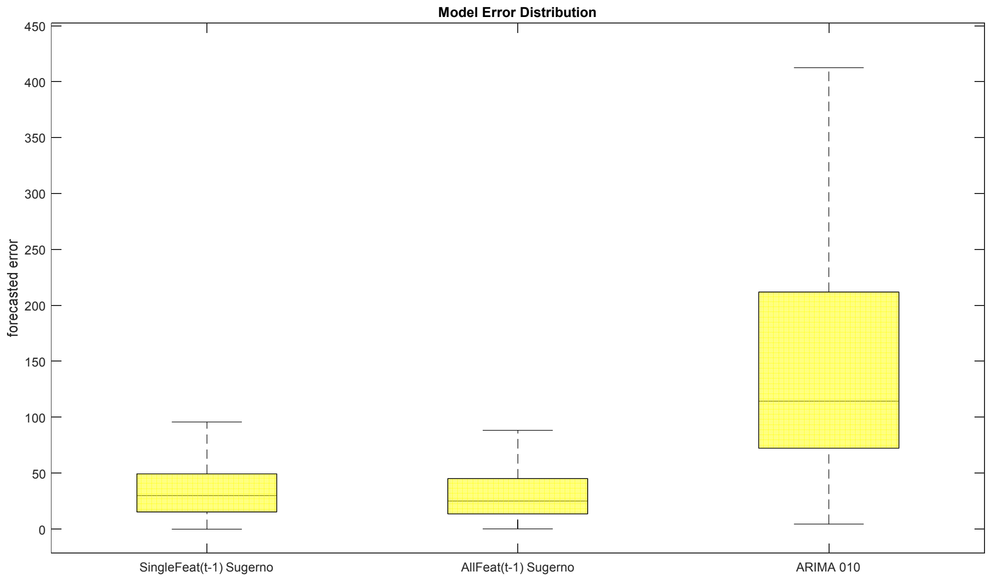


Fig. 4.3.2 Absolute error distribution of ARIMA and ANFIS prediction boxplot

Chapter 5

Results 2

In light of the findings from the previous chapter, we can conclude that ANFIS outperforms traditional ARIMA. The study's subsequent aim is to improve classical well trained ANFIS and contribute to literature with regard to popular approaches of ensemble and hybrid methods, specifically SOM-ANFIS and ensemble ANFIS.

This chapter provides the results of an extensive evaluation of the proposed hybrid, SOM-ANFIS, against the ensemble ANFIS model. As outlined in Figure 5.1, the initial section discusses the parameters and then presents the comparison of the results of the models' performance in context.

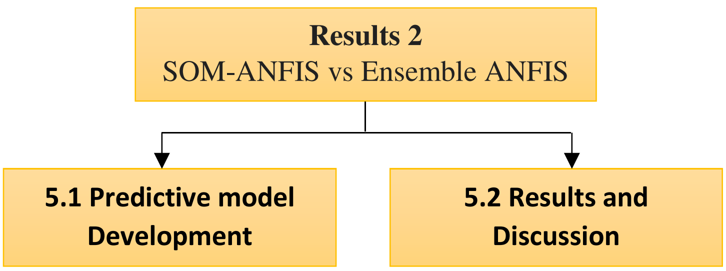


Fig. 5.1 Chapter 5 structure

5.1 Hybrid Self Organising Maps (SOM) and ANFIS Model

The complex, highly parametric model ANFIS performed well for future prediction with small error metrics and high correlations against the test set, as outlined in chapter 4. To further improve forecasting accuracy, the next model SOM-ANFIS adopts a pre-process unsupervised clustering technique stage. This clustering of stock indexes training and testing samples aims to improve on the practice of feeding input datasets chronically, advocating for model development that is not reliant on arbitrarily dividing sets in a chronologically ordered time series.

5.1.1 SOM-ANFIS parameters

The study aims to combine SOM with ANFIS, a hybrid approach, which to date and to the author's knowledge, has not been studied in the context of financial markets. Historic index data (AORD, FTSE, DJI, HIS, N225) was used to developed the proposed SOM-ANFIS hybrid. The original data with the historical antecedent day at t-1 as input predictors was employed to forecast price. The development of parameters for the proposed SOM-ANFIS model unfolded in the following two steps:

Step 1: First the historical antecedent day order at (t−1) for assignment to a training and validation set was calculated based on the SOM. In order to generate SOM-ANFIS models, the required algorithm selforgmap of the Neural Net clustering and genfis3/ANFIS of the fuzzy logic toolbox of MATLAB, version R2017b, were used.

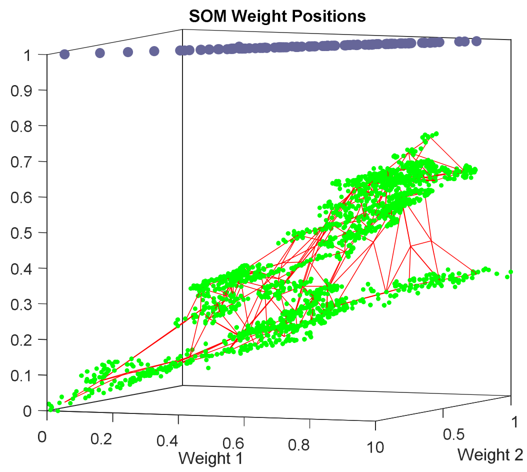
As there is no adopted standard for determining the optimum size of the Kohonen layer for unsupervised training, the dimension used in the paper by Bowden et al. (2002) was used. This dimension is a 100 clusters topology, represented by 10 rows by 10 columns. Initial neuron placement was selected as a grid 2d structure. Figures 5.2.1 and 5.2.2 display the final updated neuron position (displayed as blue points) adapted to the topological shape of the dataset. These visualisations demonstrate the SOM learning algorithm during the final stage of iteration when it had converged. The evolution of the position of neurons is closer to the data points (in green), reflecting underlying clusters. In the 2d Figure, we clearly see a high correlation of weight 1 versus weight 2, producing a grid at 45 degrees and spacing compact in the centre (approx. 0.5), while the 3d Figure depicts the dataset's higher dimensionality nature

The visualisation of the weights applied to the inputted six vectors (Figure 5.1.1.1 and 5.1.1.2) connected to each neuron (darker colours represent higher weights) demonstrates similar inputs. Weights from input 1 and 2 display almost identical patterns (high values on the top right and low values to the left), representative of the similarity between All Ordinaries and All Ordinaries day lag price. It is important to note that weighting input 8 or Japan's Nikkei Index (N225) was the most dissimilar, with high weighting to right side.

The U-Matrix, in Figures 5.1.1.3 to 5.1.1.4 displays distances to neighbouring groups (darker colours represent larger values) with blue hexagons representing neurons. A dark segmentation crossing the top central region to the lower left region indicates distinct grouping of indexes.

Finally, figure 5.1.1.5 provides a visualisation of the distribution of the train and validation samples across the SOM grid topology. The largest data points to cluster, 41 located on the lower left and no point's clusters down the centre indicating a clear separation.

The dataset split in training and validation is compared pre and post SOM model in Tables 5.1.1.1 and 5.1.1.2, respectfully. Post SOM algorithm statistics (Table 5.2.2) demonstrate that training has larger minimum values, and the variance is evenly spread, with the exception of N225 in the validation set.



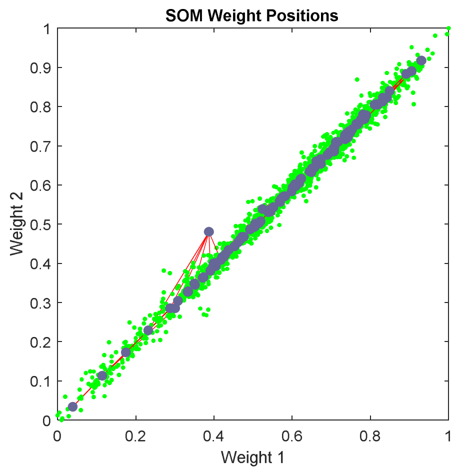
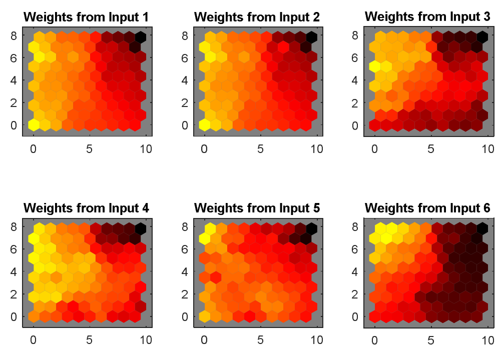


Fig. 5.1.1.1 SOM neuron 3d position

Fig.5.1.1.2 SOM neuron 2d position.



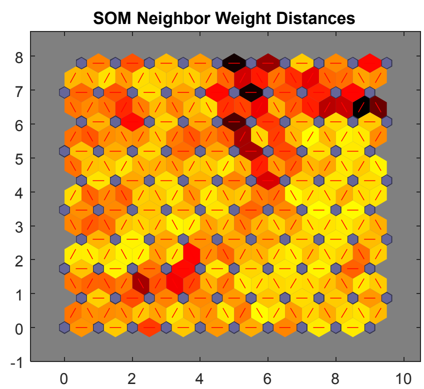


Fig. 5.1.1.3 SOM input weights

Fig. 5.1.1.4 SOM weighted distances, u-matrix

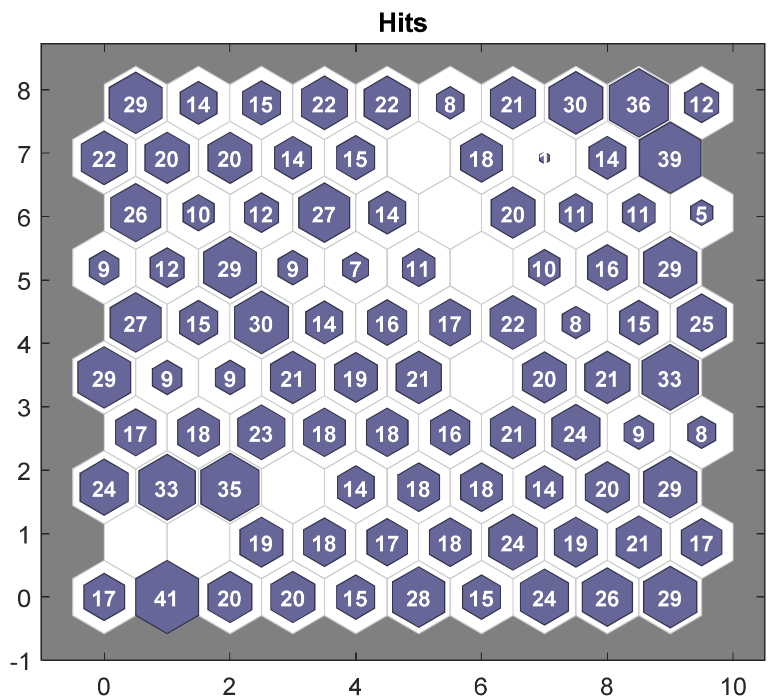


Fig. 5.1.1.5 Distribution of training and testing sets in SOM clusters topology

Training										
Index	**Min**	**Max**	**Median**	**1st Quartile**	**3rd Quartile**	**Std. Dev**	**Kurtosis**	**Skewness**	**Variance**	**Sample Size**
AORD	**0.000**	**1.000**	0.567	0.417	0.718	0.196	-0.385	-0.345	0.038	1578
DJI	**0.000**	0.652	0.337	0.244	0.523	0.165	-1.027	0.100	0.027	1578
FTSE	**0.000**	0.887	0.581	0.457	0.739	0.192	-0.117	-0.631	0.037	1578
HSI	**0.000**	0.918	0.579	0.490	0.646	0.152	1.687	-0.958	0.023	1578
N225	**0.000**	0.870	0.225	0.149	0.476	0.225	-0.475	0.786	0.051	1578
Validating										
Index	**Min**	**Max**	**Median**	**1st Quartile**	**3rd Quartile**	**Std. Dev**	**Kurtosis**	**Skewness**	**Variance**	**Sample Size**
AORD	0.577	0.827	0.738	0.679	0.779	0.066	-0.812	-0.479	0.004	197
DJI	0.511	0.744	0.635	0.610	0.660	0.046	0.460	-0.309	0.002	197
FTSE	0.534	0.880	0.729	0.650	0.819	0.093	-1.460	-0.086	0.009	197
HSI	0.396	0.689	0.547	0.487	0.622	0.075	-1.189	-0.045	0.006	197
N225	0.497	0.783	0.610	0.589	0.637	0.054	1.110	0.886	0.003	197

Table. 5.1.1.1 Normalised input indexes statistics lagged input for training, validation and testing dataset chronological (80:10:10)

Training										
Index	**Min**	**Max**	**Median**	**1st Quartile**	**3rd Quartile**	**Std. Dev**	**Kurtosis**	**Skewness**	**Variance**	**Sample Size**
AORD	0.061	**1.000**	0.604	0.453	0.742	0.177	-0.323	-0.376	0.031	1578
DJI	0.037	0.744	0.358	0.271	0.567	0.166	-1.171	0.101	0.027	1578
FTSE	0.062	0.887	0.604	0.495	0.758	0.169	0.166	-0.605	0.028	1578
HSI	0.000	0.918	0.587	0.506	0.645	0.127	3.407	-1.087	0.016	1578
N225	0.007	0.868	0.347	0.158	0.524	0.219	-0.981	0.450	0.048	1578
Validating										
Index	**Min**	**Max**	**Median**	**1st Quartile**	**3rd Quartile**	**Std. Dev**	**Kurtosis**	**Skewness**	**Variance**	**Sample Size**
AORD	**0.000**	0.828	0.491	0.209	0.653	0.250	-1.568	-0.002	0.063	197
DJI	**0.000**	0.643	0.209	0.104	0.581	0.241	-1.879	0.135	0.058	197
FTSE	**0.000**	0.849	0.323	0.191	0.642	0.258	-1.587	0.174	0.067	197
HSI	0.017	0.876	0.404	0.305	0.555	0.206	-0.376	0.382	0.043	197
N225	**0.000**	0.870	0.244	0.133	0.724	0.311	-1.744	0.216	0.097	197

Table. 5.1.1.2 Normalised input indexes statistics lagged input training, validation and testing dataset (SOM 80:10:10)

Testing										
Index	**Min**	**Max**	**Median**	**1st Quartile**	**3rd Quartile**	**Std. Dev**	**Kurtosis**	**Skewness**	**Variance**	**Sample Size**
AORD	0.828	0.973	0.871	0.857	0.902	0.035	-0.066	0.923	0.001	198
DJI	0.731	1.000	0.824	0.790	0.869	0.064	-0.276	0.626	0.004	198
FTSE	0.871	1.000	0.955	0.936	0.974	0.028	-0.045	-0.555	0.001	198
HSI	0.559	1.000	0.783	0.698	0.887	0.108	-1.067	-0.098	0.012	198
N225	0.715	1.000	0.807	0.776	0.829	0.071	0.559	1.188	0.005	198

Table. 5.1.1.3 Normalised indexes statistics output training, validation and testing dataset

Step 2: In the second stage, after assignment of historic price at one day lag (t−1) by the SOM algorithm, the ANFIS model was applied to forecast the All Ordinaries price.

The ANFIS model has been configured according to the best performing structure defined in results 1 (Section 4.3). For this application, the total amount of inputs into the model is five, and consisted of the day lag of all five indexes from SOM normalised with the min-max method (ranged from 0 -1) . The inputs were fuzzified by means of three generalised bell-shaped membership functions and trained by means of the classical hybrid learning algorithm.
After 10 epochs, the prediction with the training set can be seen in Figure 5.1.1.8 for the validation set. The resulting performance of the trained model was validated through achieved r, RMSE and MAPE statistical parameters, which are displayed in Table 5.1.1.4.

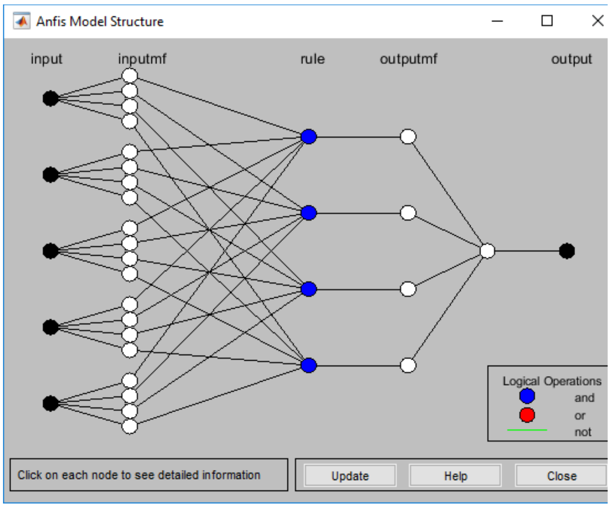


Fig. 5.1.1.6 ANFIS structure, SOM-ANFIS

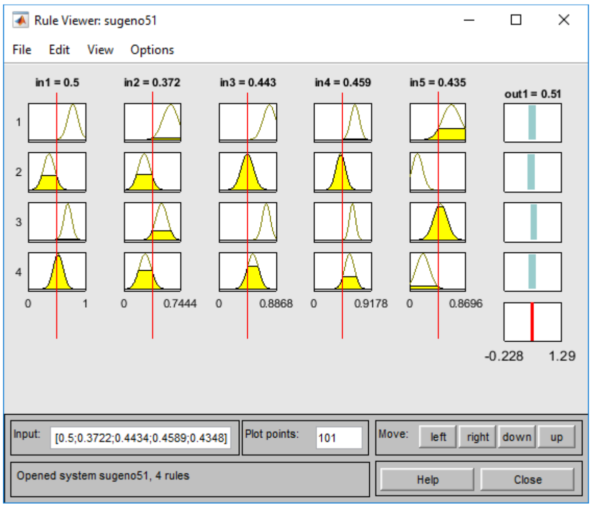


Fig. 5.1.1.7 Membership rules

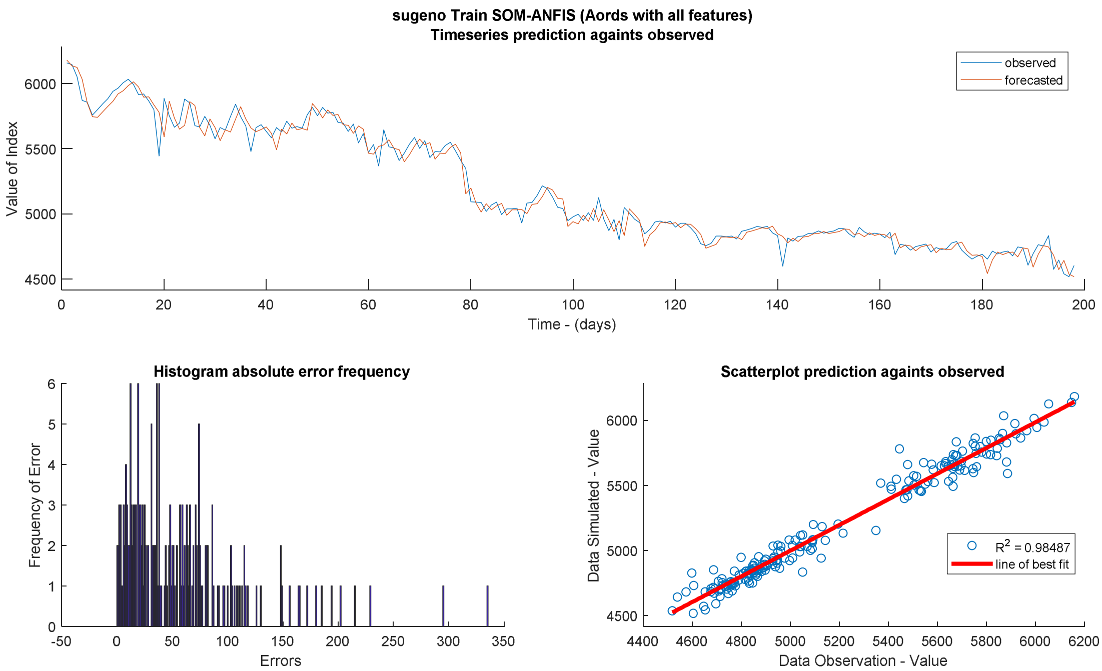


Fig. 5.1.1.8 Training performance metrics, SOM-ANFIS

Model	Model		r	RMSE	MAPE
SOM -ANFIS	Type	Sugerno	0.9849	76.7927	1.0673
	Epoch	10			
	Membership function	gaussmf			
	Optimisation method	hybrid			
	orMethod	probor			
	Clusters	auto			
	dufuzzMethod	wtaver			
	impMethod	prod			
	aggMethod	sum			
	Input	1x5			
	Ouput	1x1			
	rule	1x4			

Table 5.1.1.4 SOM-ANFIS parameters and performance training metrics

5.1.2 SOM-ANFIS results

Figure 5.1.2 below, presents the performance of the SOM-ANFIS model against the observed data given a number of metrics. The bottom right, scatter plot displays high positive correlation ($r^2 = 0.947$) of observation to simulation, indicate by the cluster of points around the red line.

Furthermore, the visualisation of the absolute difference of frequency between observed and simulation values is depicted on the bottom left histogram. The multi peak errors are below the frequency of 100 and there is a right skewed distribution with an extreme value at approximately 130. The time series (top Figure) demonstrates the change in the observed variable given the change in days for both simulated (red) and test (blue).The SOM-ANFIS simulated values follow the test closely with no discernible difference in pattern, except a higher overall value.

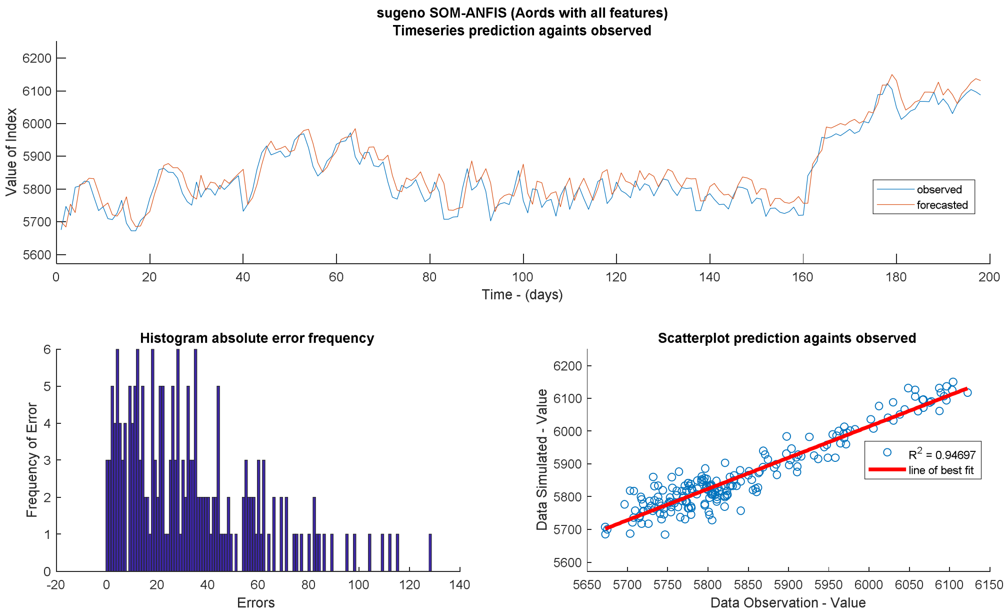


Fig. 5.1.2 Testing performance metrics, SOM-ANFIS

On inspection of the model's performance (below, Table 5.1.2), the SOM-ANFIS hybrid performs strongly across the metrics. With high correlation between actual and predicted values and small error values compared to previous models. The Pearson correlation, $r = 0.945$ demonstrates a close relationship between the observed and simulated values performed. Further error investigation of the goodness of fit ($RMSE$) and the deviations from observed values regardless of the sign (MAE) both indicate a well-fitting model (below 2). Both ENS and Pdv agree with low error metrics showing inversely high values. The Willmot Index of agreement, d , which is able to consider non-linear relationships, has the largest value because it does not overestimate the larger values and negate small values, suggesting an acceptable fitting model.

Testing Performance Metrics	Model
	SOM-ANFIS Multiple Input Sugeno model
r	0.947
Nash Ens	0.855
Willmott Index	0.953
Legates	0.632
RMSE	41.967
rRMSE	0.718
MAE	32.902
MAPE	0.565

Table 5.1.2 Testing performance metrics SOM-ANFIS

5.2 Ensemble ANFIS Model

To further improve model accuracy of the highly complex parametric model a hybrid SOM-ANFIS in previous section was develop with successful results. The ensemble model approach is now explored for comparison with the SOM-ANFIS model. Ensemble methods have been known to improve accuracy and measure variance. However, they have not yet been investigated in application with ANFIS to the financial index, such as All Ordinaries.

5.2.1 Ensemble ANFIS parameters

Similar to the SOM-ANFIS model, the ANFIS structure has been configured according to the best performing model defined in results 1 (Section 4.3). Again, the total amount of inputs into the model is five, and consisted of the day lag of all five indexes from SOM, normalised with the min-max method (ranged from 0 -1). The inputs were 'fuzzified' by means of three generalised bell-shaped membership functions and trained through the classical hybrid learning algorithm.
To obtain the average of the ANFIS ensemble, we added the results of the models and divided them by the number of ANFIS models. For the study, there were four ensemble ANFIS models that were developed: 5, 10, 50 and 100 models.

A combination of trained models was used to assess the performance through achieved r, RMSE and MAPE statistical parameters, which are displayed below in Table 5.2.1**.** The ensemble -ANFIS hybrid performed strongly across the metrics. With high Pearson correlation between all four models and four significant figures used to distinguish, $r = 0.9951$ ranked the highest for models 10 and 5. The error metric ($RMSE$) displays low values compared to previously developed models, with little discernible differences (less than 1) between all ensemble models.

Training Performance Matrix	Models			
	Ensemble ANFIS (100 models)	**Ensemble ANFIS** (50 models)	**Ensemble ANFIS** (10 models)	**Ensemble ANFIS** (5 models)
r	0.9950	0.9950	**0.9951**	**0.9951**
Nash Ens	0.9901	0.9901	0.9902	0.9903
Willmott Index	0.9945	0.9945	0.9945	0.9946
Legates	0.913	0.913	0.9134	0.9133
RMSE	60.303	60.442	**60.294**	**59.421**
rRMSE	1.2477	1.2511	1.2467	1.2265
MAE	43.448	43.557	43.469	43.145
MAPE	0.9242	0.927	0.9251	0.9151

Table 5.2.1 Training performance of n-ensemble models

5.2.2 Ensemble ANFIS results

In the below Figure 5.2.2.1 (5, 10, 50,100 ensemble model), we illustrate a boxplot of the ensemble-ANFIS models forecasting error for All Ordinaries value. The outliers specified by the end of whisker in each boxplot represent the extreme magnitudes of the forecasting error within the testing phase along with other statistics; upper quartile, median and lower quartile. The distributed forecasting errors are justified by these boxplots displaying a relative small spread around tested values.
Figure 5.2.2.2 depicts best performing model (100 ensemble ANFIS) minimum and maximum forecast error against tested values. While Figure 5.2.2.3 adds statistical properties (inter quantile range) illustrating ensemble ability to add confidence interval to a predicted value.

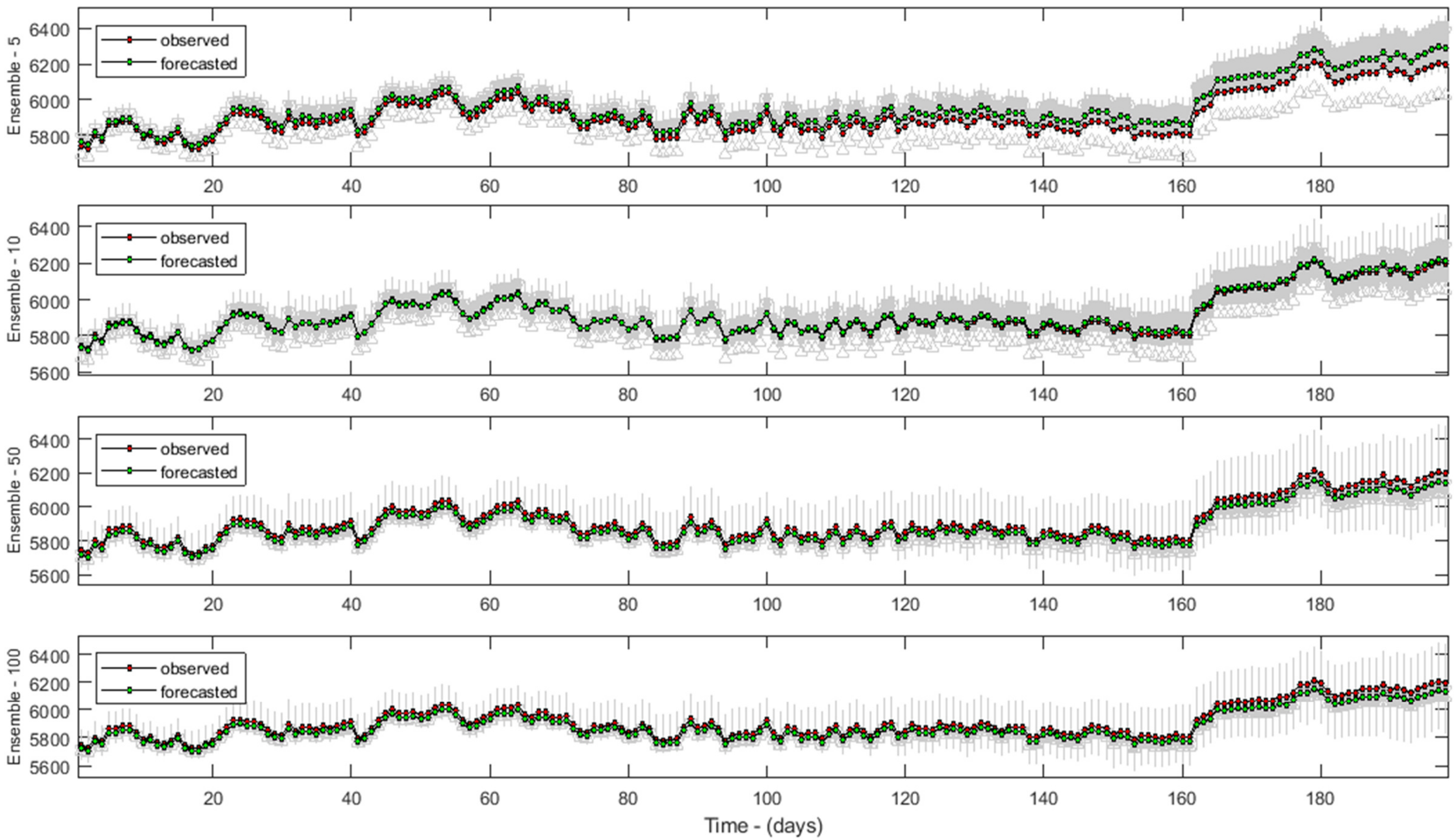


Fig. 5.2.2.1 n-ensemble ANFIS statistical forecast results (ensemble in green, actual value in red)

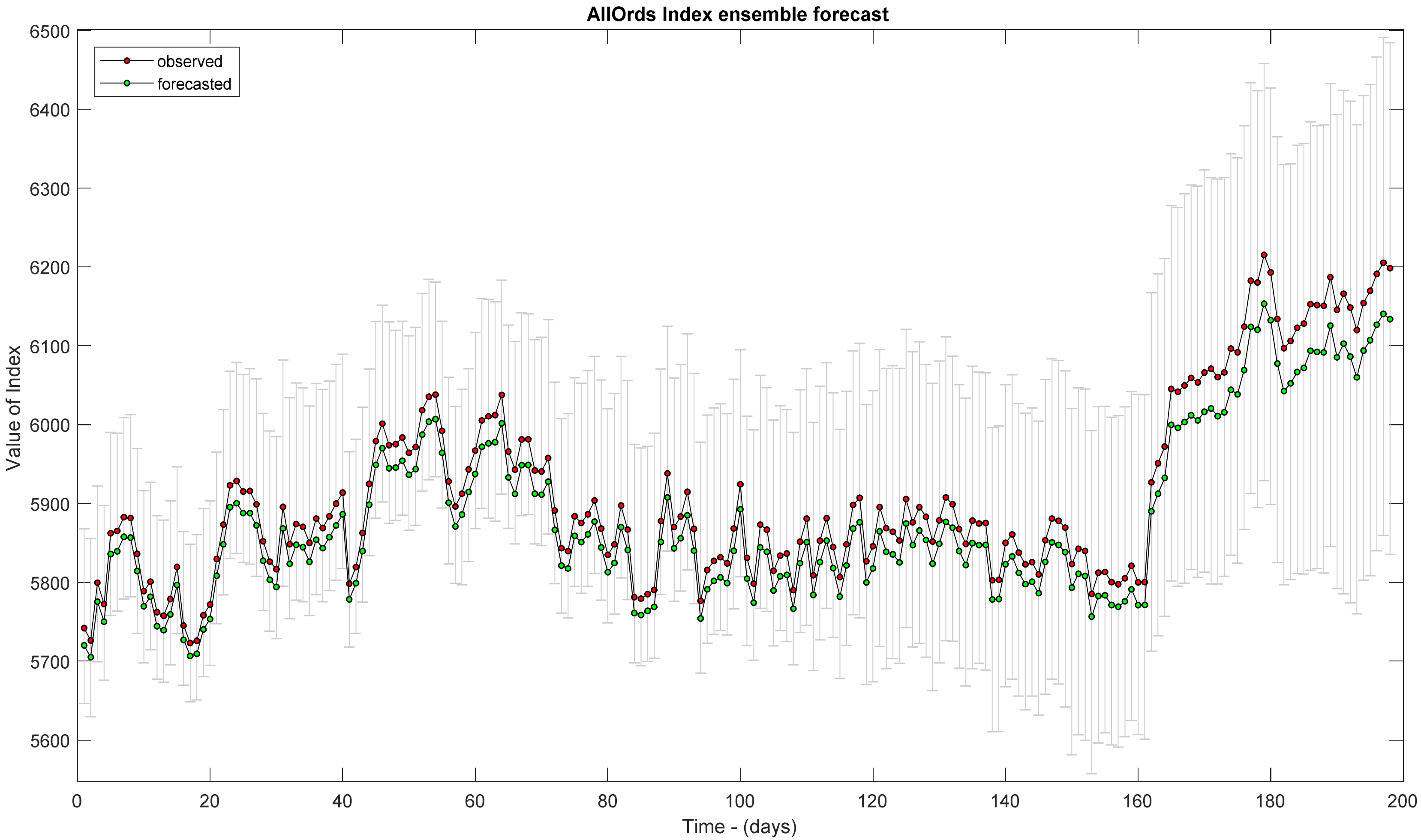


Fig. 5.2.2.2 100 ensemble ANFIS in terms of minimum and maximum forecast error (ensemble in green, actual value in red)

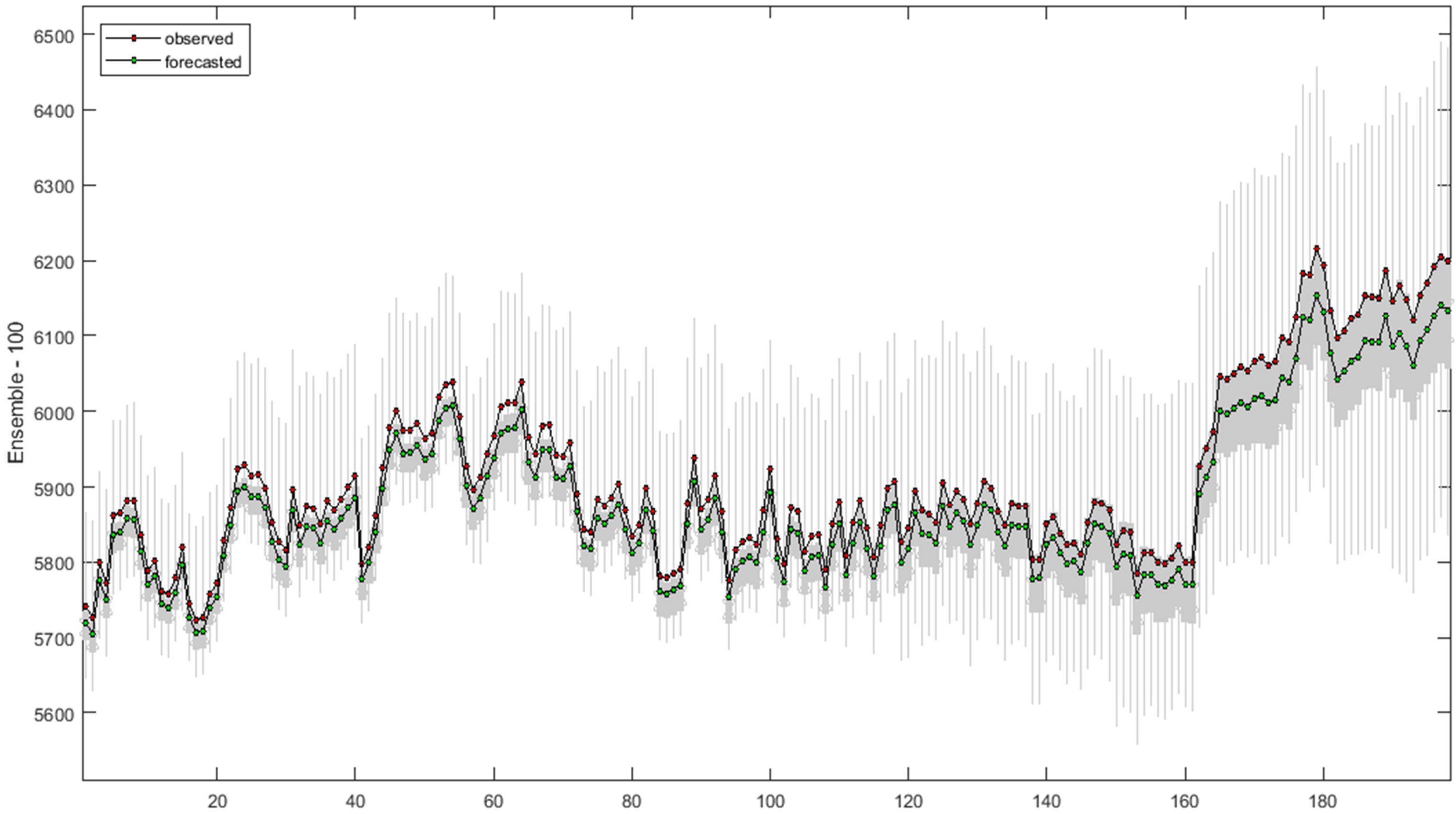


Fig. 5.2.2.3 100 ensemble-ANFIS results in terms statistical whisker box (IQR in dark grey) forecast (ensemble in green, actual value in red)

Below, Table 5.2.2 presents the testing errors and correlation metrics of ensemble ANFIS, displaying higher performing models as n (number of models) increases. The Pearson correlation of the highest model (100), $r = 0.945$, versus the lowest model (5) ,$r = 0.941$ demonstrate a slight improvement in the relationship between observed and simulated values. However, it is in the error investigation of goodness of fit ($RMSE$) and the deviations of absolute error (MAE) that the difference in model becomes evident. Ensemble model (100) improves close to $2/5$ the RMSE of ensemble model (5) (52/124).

Testing Performance Matrix	Models			
	Ensemble ANFIS (100 models)	**Ensemble ANFIS** (50 models)	**Ensemble ANFIS** (10 models)	**Ensemble ANFIS** (5 models)
r	**0.9456**	0.9464	0.9449	**0.9409**
Nash Ens	0.77	0.7568	0.3962	-0.2707
Willmott Index	0.9298	0.9279	0.8669	0.7898
Legates	0.5174	0.4969	0.1347	-0.2984
RMSE	**52.9812**	54.4722	85.8324	**124.5222**
rRMSE	0.9069	0.9324	1.4692	2.1315
MAE	43.1779	45.0129	77.4228	116.1694
MAPE	0.7426	0.7735	1.3253	1.9857

Table 5.2.2 Testing performance of n-ensemble models

Furthermore, exploring the difference in errors between ensemble models, Figure 5.2.2.4 histograms visualisation depicted ensembles models (5, 10, 50, and 100) of absolute difference between observed and simulation All Ords values.

As the models increase, the peak shifts to the right (towards 40 errors) and frequency increases from 20 to 30 range. The bimodal histogram of ensemble 5 signals a shift in average error possibly due to a large variation in the small sample of 5. The histogram depicting ensemble 10 reflects the common bell shaped normal distribution typically expected. However, as we increase the sample size to 50, there is a clear tendency in the errors, falling within the 30 to 40 range, reflective of a positive rightly skewed distribution. From sample size 50 to 100, there is a slight improvement with less frequency in the tail. However given the increase in size, this is a relatively small improvement.

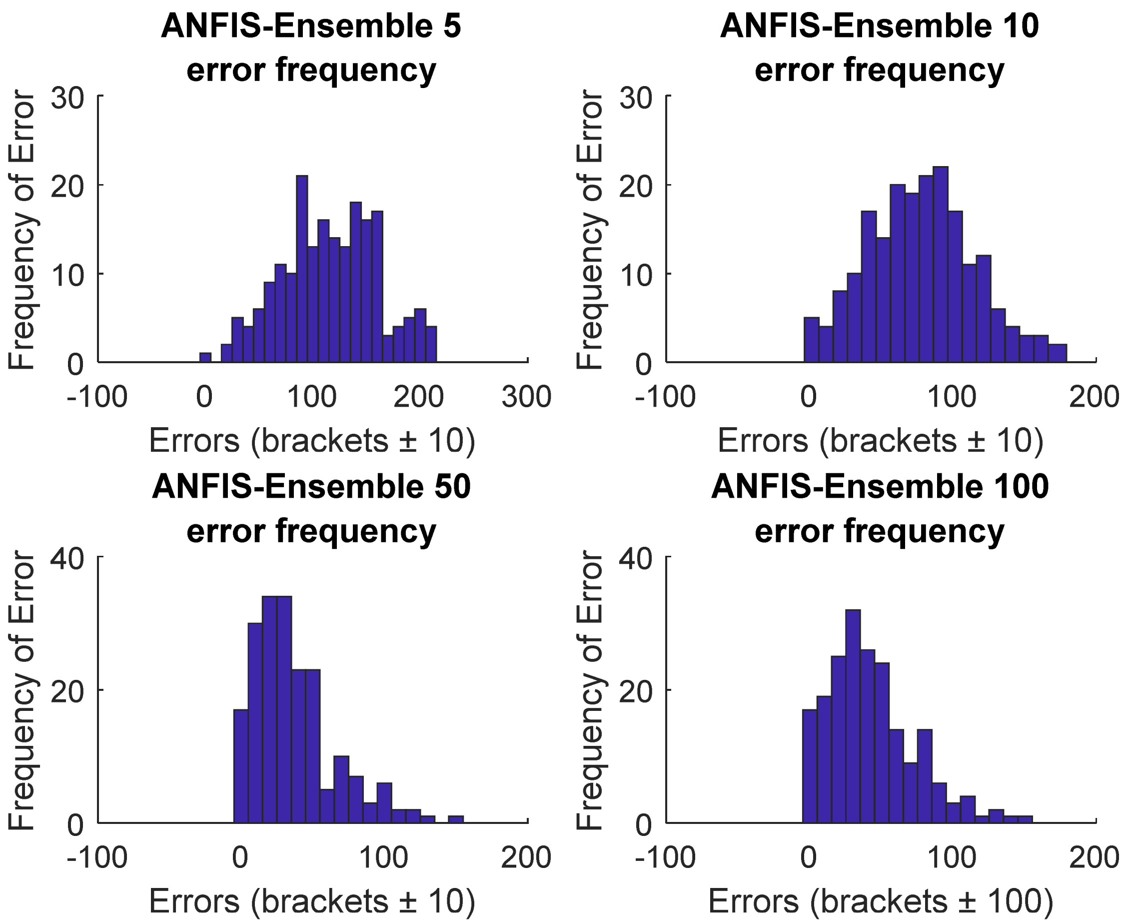


Fig. 5.2.2.4 n-ensemble ANFIS testing absolute error results histograms

5.3 Conclusion

This section presents the analyses of the hybrid and ensemble methods applied to develop ANFIS. The hybrid SOM-ANFIS and ensemble ANFIS models both demonstrate competitive predictive performance for daily stock indexes values.

The SOM methodology represented higher dimensionality data into two dimension. With the SOM algorithm well suited to finding relationships between the stock prices at given times over a ten year period. On the other hand, the ensemble approach displayed a decrease in error variance with improved overall goodness of fit and an added statistical confidence around a value.

Table 5.3.1 represents the predicted values of performance for developed models. The top two rank ensemble ANFIS model (n= 50 and n =100) scored high r values of greater than 0.94 and RMSE's of below 54. The top performing SOM-ANFIS scored a 0.97 r value, a slightly stronger relationship then the ensemble models. The SOM-ANFIS predictive error value, RMSE, produced 41.967, the lowest of the developed models. Figure 5.3.1 boxplot of the three models errors supported the error metrics findings, with SOM-ANFIS clearly the best performing model, while the ensemble models showed a similar distribution. Time series Figure 5.3.2 showed the performance of All Ordinaries over the tested days.

Performance Matrix	Models		
	SOM-ANFIS Multiple Input Sugeno model	**Ensemble ANFIS** (100 models)	**Ensemble ANFIS** (50 models)
r	0.947	0.9456	0.9464
Nash Ens	0.8557	0.77	0.7568
Willmott Index	0.9532	0.9298	0.9279
Legates	0.6322	0.5174	0.4969
RMSE	41.9678	52.9812	54.4722
rRMSE	0.7184	0.9069	0.9324
MAE	32.9028	43.1779	45.0129
MAPE	0.5653	0.7426	0.7735

Table 5.3.1 Testing performance of SOM-ANFIS vs n-ensemble ANFIS (n =50 and n=100)

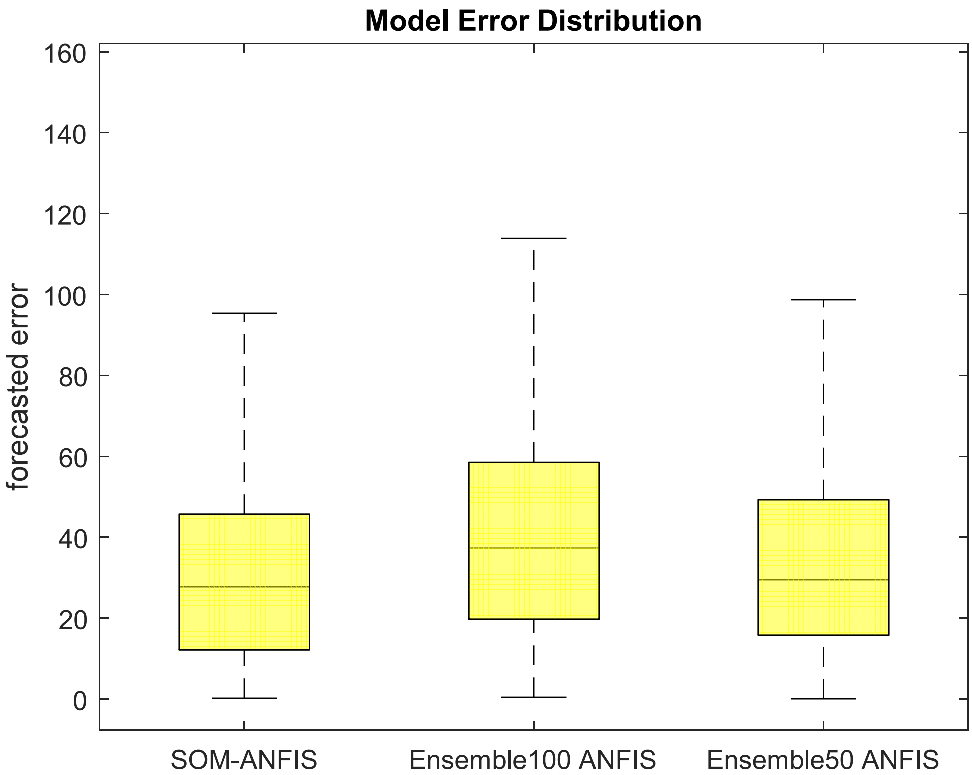


Fig. 5.3.1 Absolute error distribution of SOM-AFINS and n-ensemble ANFIS (n =50 and n=100) prediction boxplot

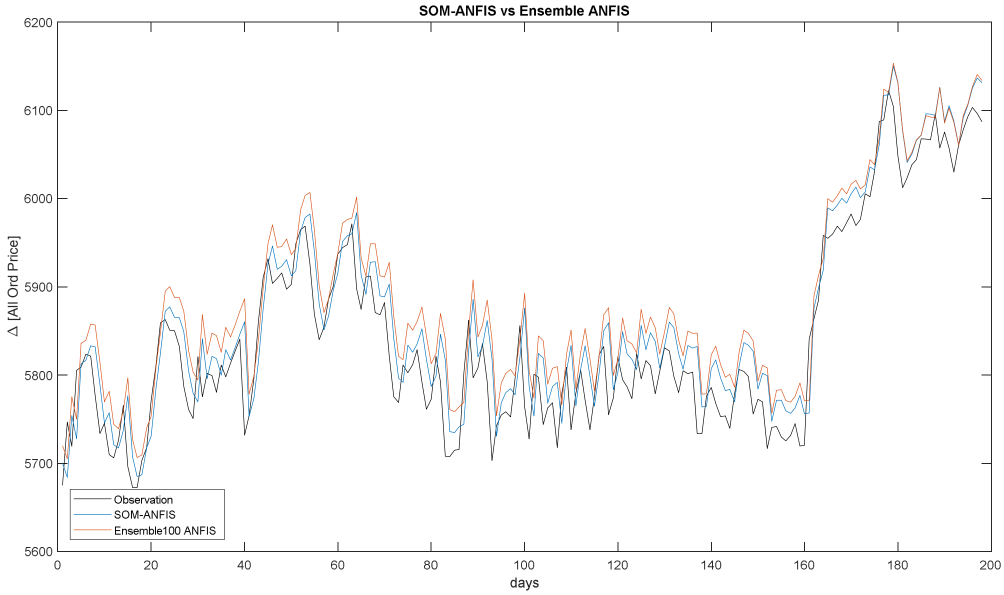


Fig 5.3.2 Time series of SOM-ANFIS and 100 ensemble ANFIS prediction with observation

Chapter 6

Conclusions

This chapter presents the findings and conclusions to the study, discussing the limitations and presenting possible future works.

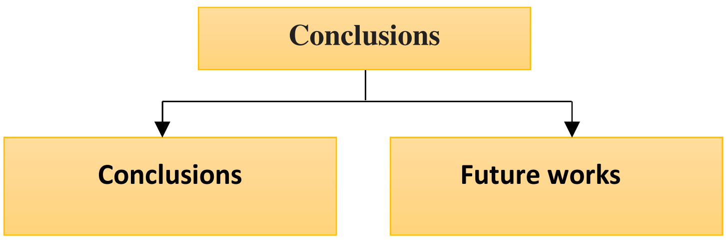


Fig. 6.1 Chapter 6 structure

6.1 Conclusion

Using a set of limited inputs defined by a day lag of five stocks, the present paper established the function of the ANFIS for modelling short term (daily) positions of All Ordinaries stock prices.

The previous days of All Ordinaries stock prices along with data from the FTSE, DJI, HSI and N225 serve to develop an ANFIS model along with other data driven models (hybrid SOM-ANFIS, Ensemble ANFIS and ARMIA) to serve as benchmarks. Models were evaluated based on plots of All Ordinaries observations versus predicted values, error metrics RMSE, MAE, rRMSE, MAPE and evaluation statistics r, E_{NS} and d.

The study resulted in the following findings:

For short-term forecasting, the performance of the ANFIS Sugerno FCM using All Ordinaries featuring day lag and the remaining indexes feature model remained similar in correlation coefficients for the linear relationship between All Ordinariess observation and simulated values over the test period which corresponded with the relatively large values of the ENS, WI, and ELM. In the training and testing stages, the ANFIS models achieved overall high evaluation statistics and a promising error index, while the ARIMA models performed poorly in comparison. The ARIMA model (010) displayed little to no correlation and four and a halve times the RMSE of the ARIMA model (186/42). The seasonal ARIMA (010)(0,001) , obtained the largest RMSE (890) with a significantly low E_{NS} value performing the worst out of all the data driven models developed.

When the ANFIS-Sugerno FCM model was evaluated for daily forecasting, its accuracy was superior to all other data-driven models tested, prompting investigation into the hybrid and ensemble methods to further improve performance. This was confirmed with the hybrid approach of SOM with ANFIS, which yielded the highest r value and lowest RMSE, outperforming all models. The ensemble ANFIS model also improved performance with 100 model high r value (0.95) and similar large values for E_{NS} (0.77) and d (0.93).

Overall, the study highlights the appropriateness of the ANFIS and hybrid/ensemble based approaches to modelling for daily forecasting of All Ordinaries. These approaches present advantages to similar indexes incorporating ANFIS machine learning techniques for improved performance with regard to different stocks in the financial market. Furthermore, this research paper provides a baseline relevant to SOM-ANFIS and ensemble ANFIS index forecasting, which have not yet studied. The study illustrates models' ability to forecast All Ordinaries stocks, providing insightful information on the phenomena of market events for investors, policy makers and other stakeholders to make more informed and profitable decisions.

6.2 Limitations and recommendations for future works

Despite this research paper's provision of the baseline relevance of ANFIS models for All Ordinaries forecasting, the models should be tested in a wide range of international and other national indexes where market prices are targeted confirm the models' practicality.

While the results of our study provide interesting insights given the incorporated features, recommended future research could include the implementation of additional features such as inter-daily scales (i.e. hourly prices), sentiment analysis and other market data such as the individual stocks of the indexes.

References

Abu-Mostafa, Y.S. and Atiya, A.F., 1996. Introduction to financial forecasting. Applied Intelligence, 6(3), pp.205-213.

Afolabi, M.O. and Olude, O., 2007, January. Predicting stock prices using a hybrid Kohonen self organizing map (SOM). In System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on (pp. 48-48). IEEE.

Akaike, Hirotugu. "A new look at the statistical model identification." IEEE transactions on automatic control 19.6 (1974): 716-723.

Ansari, T., Kumar, M., Shukla, A., Dhar, J. and Tiwari, R., 2010. Sequential combination of statistics, econometrics and Adaptive Neural-Fuzzy Interface for stock market prediction. Expert Systems with Applications, 37(7), pp.5116-5125.

Atsalakis, G.S. and Valavanis, K.P., 2009. Forecasting stock market short-term trends using a neuro-fuzzy based methodology. Expert systems with Applications, 36(7), pp.10696-10707.

Attigeri, G.V., MM, M.P., Pai, R.M. and Nayak, A., 2015, November. Stock market prediction: A big data approach. In TENCON 2015-2015 IEEE Region 10 Conference (pp. 1-5). IEEE.

Bagheri, A., Peyhani, H.M. and Akbari, M., 2014. Financial forecasting using ANFIS networks with quantum-behaved particle swarm optimization. Expert Systems with Applications, 41(14), pp.6235-6250. Beck, T. and Levine, R., 2004. Stock markets, banks, and growth: Panel evidence. Journal of Banking & Finance, 28(3), pp.423-442.

Billah, M., Waheed, S. and Hanifa, A., 2015. Predicting Closing Stock Price using Artificial Neural Network and Adaptive Neuro Fuzzy Inference System (ANFIS): The Case of the Dhaka Stock Exchange. International Journal of Computer Applications, 129(11), pp.1-5.

Box, G.E., Jenkins, G.M., Reinsel, G.C. and Ljung, G.M., 2015. Time series analysis: forecasting and control. John Wiley & Sons.

Boyacioglu, M.A. and Avci, D., 2010. An adaptive network-based fuzzy inference system (ANFIS) for the prediction of stock market return: the case of the Istanbul stock exchange. Expert Systems with Applications, 37(12), pp.7908-7912.

Breiman, L., 1996. Bagging predictors. Machine learning, 24(2), pp.123-140.

Case, K.E., Quigley, J.M. and Shiller, R.J., 2005. Comparing wealth effects: the stock market versus the housing market. Advances in macroeconomics, 5(1).

Chen, Y., Yang, B. and Abraham, A., 2007. Flexible neural trees ensemble for stock index modeling. Neurocomputing, 70(4-6), pp.697-703.

Choudhry, R. and Garg, K., 2008. A hybrid machine learning system for stock market forecasting. World Academy` of Science, Engineering and Technology, 39(3), pp.315-318.

Chun, S.H. and Kim, S.H., 2004. Automated generation of new knowledge to support managerial decision□making: case study in forecasting a stock market. Expert Systems, 21(4), pp.192-207.

Dablemont, S., G. Simon, A. Lendasse, A. Ruttiens, F. Blayo and M. Verleysen (2003). Time series forecasting with SOM and local non-linear models-Application to the DAX30 index prediction. In: Proceedings of the workshop on self-organizing maps, Kitakyushu, Japan, Citeseer.

Desai, M., Misra, A. and Doyle, W.D., 2005. Effect of interface roughness on exchange coupling in synthetic antiferromagnetic multilayers. IEEE transactions on magnetics, 41(10), pp.3151-3153.

Fama, E.F., 1965. The behavior of stock-market prices. The journal of Business, 38(1), pp.34-105.

Fama, E.F., 1970. Efficient capital markets: A review of theory and empirical work. The journal of Finance, 25(2), pp.383-417.

Fama, E.F. and French, K.R., 1993. Common risk factors in the returns on stocks and bonds. Journal of financial economics, 33(1), pp.3-56.

Frino, A. and Jarnecic, E., 2005. Introduction to futures and options markets in Australia.

Funke, N., 2004. Is there a stock market wealth effect in emerging markets?. Economics Letters, 83(3), pp.417-421.

Ghosh, I., Sanyal, M.K. and Jana, R.K., 2017. Fractal Inspection and Machine Learning-Based Predictive Modelling Framework for Financial Markets. Arabian Journal for Science and Engineering, pp.1-15.

Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y., 2016. Deep learning (Vol. 1). Cambridge: MIT press.

Guresen, E., Kayakutlu, G. and Daim, T.U., 2011. Using artificial neural network models in stock market index prediction. Expert Systems with Applications, 38(8), pp.10389-10397.

Harvey, A., Jan Koopman, S. and Penzer, J., 1999. Messy time series. In Messy Data (pp. 103-143). Emerald Group Publishing Limited.

Huang, W., Nakamori, Y. and Wang, S.Y., 2005. Forecasting stock market movement direction with support vector machine. Computers & Operations Research, 32(10), pp.2513-2522.

Huang, C.L. and Tsai, C.Y., 2009. A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting. Expert Systems with applications, 36(2), pp.1529-1539.

Jang JS. ANFIS: adaptive-network-based fuzzy inference system. IEEE transactions on systems, man, and cybernetics. 1993 May;23(3):665-85.

Kavajecz, K.A. and Odders-White, E.R., 2004. Technical analysis and liquidity provision. Review of Financial Studies, 17(4), pp.1043-1071

Kohonen, T., 1982. Self-organized formation of topologically correct feature maps. Biological cybernetics, 43(1), pp.59-69.

Krause, P., Boyle, D.P. and Bäse, F., 2005. Comparison of different efficiency criteria for hydrological model assessment. Advances in geosciences, 5, pp.89-97.

Legates, D.R. and McCabe Jr, G.J., 1999. Evaluating the use of “goodness-of-fit” measures in hydrologic and hydroclimatic model validation. Water resources research, 35(1), pp.233-241.

Levine, R. and Zervos, S., 1998. Stock markets, banks, and economic growth. American economic review, pp.537-558.

Liao, Z. and Wang, J., 2010. Forecasting model of global stock index by stochastic time effective neural network. Expert Systems with Applications, 37(1), pp.834-841.

Lo, A.W., Mamaysky, H. and Wang, J., 2000. Foundations of technical analysis: Computational algorithms, statistical inference, and empirical implementation. The journal of finance, 55(4), pp.1705-1765.

Market Watch. 2018. *Statistics on world's money perspective*. [ONLINE] Available at: https://www.marketwatch.com/story/how-big-is-bitcoin-really-this-chart-puts-it-all-in-perspective-2017-06-21. [Accessed 19 April 2018].Melin, P., Soto, J., Castillo, O. and Soria, J., 2012. A new approach for time series prediction using ensembles of ANFIS models. Expert Systems with Applications, 39(3), pp.3494-3506.

Meng, J., Zhu, T., Chen, X. and Yin, X., 2013, March. The nonlinear dynamics characteristics of stock market and its variation. In Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013). Published by Atlantis Press, Paris, France (pp. 0450-0455).

Milosevic, N., 2016. Equity forecast: Predicting long term stock price movement using machine learning. arXiv preprint arXiv:1603.00751.

Neftci, S.N., 1991. Naive trading rules in financial markets and wiener-kolmogorov prediction theory: a study of" technical analysis". Journal of Business, pp.549-571.

Nguyen, D.H. and Le, M.T., 2014. A two-stage architecture for stock price forecasting by combining SOM and fuzzy-SVM. arXiv preprint arXiv:1408.5241.

Perrone, M.P., 1993. Improving regression estimation: Averaging methods for variance reduction with extensions to general convex measure optimization (Doctoral dissertation, Brown University).

Pring, M.J., 2002. How to select stocks using technical analysis. McGraw Hill Professional.

Pulido, M., Melin, P. and Castillo, O., 2014. Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange. Information Sciences, 280, pp.188-204.

Rajan, R.G. and Zingales, L., 1996. Financial dependence and growth (No. w5758). National bureau of economic research.

Rather, A.M., Agarwal, A. and Sastry, V.N., 2015. Recurrent neural network and a hybrid model for prediction of stock returns. Expert Systems with Applications, 42(6), pp.3234-3241.

Rigobon, R. and Sack, B., 2003. Measuring the reaction of monetary policy to the stock market. The quarterly journal of Economics, 118(2), pp.639-669.

Rosenberg, B., Reid, K. and Lanstein, R., 1998. market inefficiency. Streetwise: the best of the Journal of portfolio management, p.48

Ruta, D., 2014, June. Automated trading with machine learning on big data. In Big Data (BigData Congress), 2014 IEEE International Congress on (pp. 824-830). IEEE.

Schmidt, A.B., 2011. Financial markets and trading: an introduction to market microstructure and trading strategies(Vol. 637). John Wiley & Sons.

Schwarz, G., 1978. Estimating the dimension of a model. The annals of statistics, 6(2), pp.461-464.

Shen, S., Jiang, H. and Zhang, T., 2012. Stock market forecasting using machine learning algorithms. Department of Electrical Engineering, Stanford University, Stanford, CA, pp.1-5.

Shleifer, A., 2000. Inefficient markets: An introduction to behavioural finance. OUP Oxford. Nature of stock market data

Sornette, D., 2017. Why stock markets crash: critical events in complex financial systems. Princeton University Press.

Svalina, I., Galzina, V., Lujić, R. and ŠImunović, G., 2013. An adaptive network-based fuzzy inference system (ANFIS) for the forecasting: The case of close price indices. Expert systems with applications, 40(15), pp.6055-6063.

Tan, L., Wang, S. and Wang, K., 2017. A new adaptive network-based fuzzy inference system with adaptive adjustment rules for stock market volatility forecasting. Information Processing Letters, 127, pp.32-36.

Tay, F.E.H. and Cao, L.J., 2001. Improved financial time series forecasting by combining support vector machines with self-organizing feature map. Intelligent Data Analysis, 5(4), pp.339-354.

Tsai, C. and S. Wang (2009). Stock price forecasting by hybrid machine learning techniques. Proceedings of the International MultiConference of Engineers and Computer Scientists.

Willmott, C.J., 1982. Some comments on the evaluation of model performance. Bulletin of the American Meteorological Society, 63(11), pp.1309-1313.

Wolpert, D.H., 1992. Stacked generalization. Neural networks, 5(2), pp.241-259.

Xiao, Y., Xiao, J. and Wang, S., 2012. A hybrid forecasting model for non-stationary time series: An application to container throughput prediction. International Journal of Knowledge and Systems Science (IJKSS), 3(2), pp.67-82.

Program Code

Matlab code

Non-linear model development

```
% ANFIS Multiple Input, trial on membership functions
% MSC8001/MSC8002
% Author: John Worrall
% Description: Given input data, design ANFIS, Ensemble ANFIS, SOM/ANFIS
% and compare results
% Requirements: Indexes.xlsx (Data Input) ResultsRun.xlsx (Assessment)
%----------------------------

close all force
clear
clc

% Model Run users Choice options to run
% 1 = single run (all inputs)
% 2 = Ensemble
% 3 = SOM+ANFIS,
% 4 = single run (all features except t-1)
% 5 = single run (just t-1)
% 6 = Mamdani model
% 7 = ARIMA
% 8 = Comparison of models
% 9 = plot timeseries (initial)
% 10 = Ensemble plot

ModelRun = 10;

MemFun = 'sugeno';% Only Sugerno argument for ANFIS not Madami
%randomise parameters, so historic runs have no influence
rand('state',0);

%Raw values --------------------------------
closeData=xlsread('Indexs.xlsx','AlIndex');
Day = closeData(:, 1);
AORD = closeData(:, 2);
DJI = closeData(:, 3);
FTSE = closeData(:, 4);
HSI = closeData(:, 5);
N225 = closeData(:, 6);
A = [AORD DJI FTSE HSI N225 Day];
A(any(isnan(A), 2), :) = []; %Remove Rows with NAN values
clearvars -except ModelRun MemFun A Day; %Remove all variable except A

%Prep features --------------------------------
% ouput delete first row
AORDOut = A(:, 1);
AORDOut(1,:)=[];
DayA = A(:, 6);
DayA(1,:)=[];
% t-1 delete last row
```

```
AORDIn = A(:, 1);
AORDIn(end,:)=[];
DJIIn = A(:, 2);
DJIIn(end,:)=[];
FTSEIn = A(:, 3);
FTSEIn(end,:)=[];
HSIIn = A(:, 4);
HSIIn(end,:)=[];
N225In = A(:, 5);
N225In(end,:)=[];

%Normalise values --------------------------------
%Min Max values
MaxAordOt = max(AORDOut);
MinAordOt = min(AORDOut);
MaxAordIn = max(AORDIn);
MinAordIn = min(AORDIn);
MaxDji = max(DJIIn);
MinDji = min(DJIIn);
MaxFtse = max(FTSEIn);
MinFtse = min(FTSEIn);
MaxHsi = max(HSIIn);
MinHsi = min(HSIIn);
MaxN225 = max(N225In);
MinN225 = min(N225In);

for x = 1:numel(AORDOut)
   AORD_out(x) = (AORDOut(x) - MinAordOt)/(MaxAordOt - MinAordOt);
   AORD_t1(x) = (AORDIn(x) - MinAordIn)/(MaxAordIn - MinAordIn);
   DJI_t1(x) = (DJIIn(x) - MinDji)/(MaxDji - MinDji);
   FTSE_t1(x) = (FTSEIn(x) - MinFtse)/(MaxFtse - MinFtse);
   HSI_t1(x) = (HSIIn(x) - MinHsi)/(MaxHsi - MinHsi);
   N225_t1(x) = (N225In(x) - MinN225)/(MaxN225 - MinN225);
end
AORD_out = AORD_out';
AORD_t1 = AORD_t1';
DJI_t1 = DJI_t1';
FTSE_t1 = FTSE_t1';
HSI_t1 = HSI_t1';
N225_t1 = N225_t1';

A_Final = [AORD_out AORD_t1 DJI_t1 FTSE_t1 HSI_t1 N225_t1 DayA];

%Correlation plot
% Index_norm = [AORD_out DJI_t1 FTSE_t1 HSI_t1 N225_t1];
% corrplot(Index_norm,'varNames',{'AORD','DJI','FTSE','HSI','N225'});

% Split and train dataset --------------------------------
nData = size(A_Final,1);
NInput = [AORD_t1 DJI_t1 FTSE_t1 HSI_t1 N225_t1];
NOutput = AORD_out;

PERM = 1:nData;
pTrain=0.8;  %80 train
pValid=0.1;  %10 validate
pTest= 0.1;  %10 test
%Apply in order
TrainInd=PERM(1:round(pTrain*nData));
ValidInd=PERM(round(pTrain*nData)+1
:round(pTrain*nData)+round(pValid*nData));
```

```
TestInd=PERM(round(pTrain*nData)+round(pValid*nData)+1 :end);

TrainInputs=NInput(TrainInd,:);
TrainTargets=NOutput(TrainInd,:);
ValidInputs=NInput(ValidInd,:);
ValidTargets=NOutput(ValidInd,:);
TestInputs=NInput(TestInd,:);
TestTargets=NOutput(TestInd,:);

clearvars -except ModelRun MemFun A A_Final pTrain pTest pValid TrainInd
ValidInd TestInd MaxAordOt MinAordOt TrainInputs TrainTargets ValidInputs
ValidTargets TestInputs TestTargets

% Machine learning --------------------------------
if ModelRun == 1 % Single Run ANFIS

% training stage --------------------------------
    %Set up structure for anfis
    fis = genfis3(TrainInputs, TrainTargets, MemFun, 'auto', [])

%     %Plot Membership rules
%     for input_index=1:1
%         subplot(1,1,input_index)
%         [x,y]=plotmf(fis,'input',input_index);
%         plot(x,y)S
%         axis([-inf inf 0 1.2]);
%         xlabel(['Input' int2str(input_index)]);
%     end

%     %Plot training output
%     subplot(2,1,1)
%     x = 1:length(TrainInputs);
%     yp = evalfis(TrainInputs, fis);
%     plot(x,TrainTargets,x,yp);
%     title('ANFIS vs Training Data')
%     xlabel('Days');
%     ylabel('Normalised value');
%     legend('Training','ANFIS','Location','SouthEast')
%     %Training Errors
%     ypEr = yp - TrainTargets;
%     subplot(2,1,2)
%     plot(x,ypEr);
%     title('Training Data Errors')
%     xlabel('Days');
%     ylabel('Errors');
%     legend('Training Errors','Location','SouthEast')

%     %GUI training output
%      neuroFuzzyDesigner(fis)
%      fuzzyLogicDesigner(fis)
%      surfview(fis)


 % validation stage --------------------------------
    %Configure setting for validation
    EpochNum = 30;
    opt = anfisOptions('InitialFIS',fis);
    opt.DisplayANFISInformation = 0;
    opt.DisplayErrorValues = 0;
```

```
    opt.DisplayStepSize = 0;
    opt.DisplayFinalResults = 0;
    opt.ValidationData = [ValidInputs ValidTargets];
    opt.EpochNumber = EpochNum;
    opt.StepSizeIncreaseRate = 2*opt.StepSizeIncreaseRate;

    %Run validation and predicion
    %out_fis = anfis([TrainInputs TrainTargets],opt);
    [out_fis,trn_error,step_size,chk_out_fismat,chk_error] =
anfis([TrainInputs TrainTargets],opt);

    %Plot training output
    x = 1:length(TestTargets);
    yp = evalfis(TestInputs, out_fis);
    plot(x,TestTargets,x,yp);

    %Plot step size
    plot(step_size);

    %plot training and validation errors
    figure(2)
    x = [1:EpochNum];
    plot(x,trn_error,'.b',x,chk_error,'*r');
    title('Training vs Validation Error')
    xlabel('Epoch');
    ylabel('Error (RMSE)');
    legend('Training','Validation','Location','NorthWest')

%     %GUI validation output
%     neuroFuzzyDesigner(out_fis)
%     fuzzyLogicDesigner(out_fis)
%     surfview(out_fis)

    %Predict Train
    predictTrain = evalfis(TrainInputs, fis);
    for x = 1:numel(TestTargets)
       dataSimTrain(x) = predictTrain(x) * (MaxAordOt - MinAordOt)+
MinAordOt;
       dataObsTrain(x) = TrainTargets(x) * (MaxAordOt - MinAordOt)+
MinAordOt;
    end
    dataSimTrain = dataSimTrain';
    dataObsTrain = dataObsTrain';

    [nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnrMAE,nnPI]
=asseMetric(dataObsTrain,dataSimTrain);
    sugenoErrorsTrain = nnPI;
    asseMetricVis(dataObsTrain,dataSimTrain,nnR,2, strcat(MemFun, ' Train
(Aords with all features)'));

 % test stage --------------------------------
    %predict Test
    predict = evalfis(TestInputs, out_fis);

    for x = 1:numel(TestTargets)
       dataObs(x) = TestTargets(x) * (MaxAordOt - MinAordOt)+ MinAordOt;
       dataSim(x) = predict(x) * (MaxAordOt - MinAordOt)+ MinAordOt;
    end
    dataObs = dataObs';
```

```
    dataSim = dataSim';

    [nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnrMAE,nnPI]
=asseMetric(dataObs,dataSim);
    A_SugenoAllFeatErrors = nnPI;
    asseMetricVis(dataObs,dataSim,nnR,2, strcat(MemFun, ' (Aords with all
features)'));


%----------------------------------------------------------------
elseif ModelRun == 2    % Enesemble ANFIS
%----------------------------------------------------------------
    nModels = 10;

    TrainSet = [TrainTargets TrainInputs];
    ValidSet = [ValidTargets ValidInputs];
    for x = 1:numel(TestTargets)
       dataObs(x) = TestTargets(x) * (MaxAordOt - MinAordOt)+ MinAordOt;
    end
    dataObs = dataObs';

    for x = 1:nModels
        clearvars dataSim predict TrainTargets TrainInputs ValidTargets
ValidInputs

        %Trainset and Validation set for random sample with replacement.
        TrainSamp = datasample(TrainSet,size(TrainSet,1));% returns k
observations sampled uniformly at random, with replacement, from the data
in data.
        TrainTargets =  TrainSamp(:, 1);
        TrainInputs = TrainSamp(:,[2 3 4 5 6]);
        ValidSamp = datasample(ValidSet,size(ValidSet,1));% returns k
observations sampled uniformly at random, with replacement, from the data
in data.
        ValidTargets =  ValidSamp(:, 1);
        ValidInputs = ValidSamp(:,[2 3 4 5 6]);

        %ANFIS
        fis = genfis3(TrainInputs, TrainTargets,'sugeno', 'auto', []);

        %View Membership rules
%          for input_index=1:1
%              subplot(1,1,input_index)
%              [x1,y]=plotmf(fis,'input',input_index);
%              plot(x1,y)
%              axis([-inf inf 0 1.2]);
%              xlabel(['Input' int2str(input_index)]);
%          end

        %Configure validation
        opt = anfisOptions('InitialFIS',fis);
        opt.DisplayANFISInformation = 0;
        opt.DisplayErrorValues = 0;
        opt.DisplayStepSize = 0;
        opt.DisplayFinalResults = 0;
        opt.ValidationData = [ValidInputs ValidTargets];

        [out_fis,trn_error,step_size,chk_out_fismat,chk_error] =
anfis([TrainInputs TrainTargets],opt);
```

```
        mfEpoch(:,x)= opt.EpochNumber; % Stores number of epochs
        [m,n2]= size(out_fis.rule);  % Stores number of mf
        mfNumbers(:,x)= n2;

        %Predict Train
        predictTrain = evalfis(TrainInputs, fis);
        for z = 1:numel(TrainTargets)
           dataSimTrain(z) = predictTrain(z) * (MaxAordOt - MinAordOt)+
MinAordOt;
           dataObsTrain(z) = TrainTargets(z) * (MaxAordOt - MinAordOt)+
MinAordOt;
        end
        dataSimTrain = dataSimTrain';
        dataObsTrain = dataObsTrain';
        runsTrain(:,x) = dataSimTrain;

        [nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnrMAE,nnPI]
=asseMetric(dataObsTrain,dataSimTrain);
        runsTrainError(:,x) = nnPI;
        %asseMetricVis(dataObsTrain,dataSimTrain,nnR,2, strcat(MemFun, '
Train (Aords with all features)'));

        %Predict Train
        predict = evalfis(TestInputs, out_fis);

        % denormalise simulation values --------------------------------
        for y = 1:numel(predict)
            dataSim(y) = predict(y) * (MaxAordOt - MinAordOt)+ MinAordOt;
        end
        dataSim = dataSim';
        runs(:,x) = dataSim;

        RunNumber = num2str(x);
        %Error metrics --------------------------------
        [nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnrMAE,nnPI]
=asseMetric(dataObs,dataSim);
        runsError(:,x) = nnPI;
        %asseMetricVis(dataObs,dataSim,nnR,2,strcat('Sugeno, Aords with all
features - Run ',RunNumber));

    end


    %Mean model Train
    avModelTrain = mean(runsTrain,2);
    avModelTrainErr = mean(runsTrainError,2)';
%
[nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnrMAE,nnPI]=asseMetric(dataObsTr
ain,avModelTrain);
%     avModelTrainErr = nnPI';
%     dataObsTrain = dataObsTrain';
%     asseMetricVis(dataObsTrain,avModelTrainErr,nnR,2,strcat('Sugeno,
ENSEMBLE Train Aords with all features - Run ',RunNumber));

    %Mean model TEST
    avModel = mean(runs,2);
    [nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnrMAE,nnPI]
=asseMetric(dataObs,avModel);
    A_EnsembleANFIS_TRAIN_Errors = nnPI';
```

```
    asseMetricVis(dataObs,avModel,nnR,2,strcat('Sugeno, ENSEMBLE Aords with
all features - Run ',RunNumber));

    min(mfEpoch)
    max(mfEpoch)
    min(mfNumbers)
    max(mfNumbers)

    asseMetricVis(dataObs,runs,nnR,3, strcat(MemFun, ' (Aords with all
features)'));

    %box plot
    asseMetricVis(dataObs,runs,nnR,4, strcat(MemFun, ' (Aords with all
features)'));
    asseMetricVis(dataObs,avModel,nnR,5, strcat(MemFun, ' (Aords with all
features)'));


%----------------------------------------------------------------
elseif ModelRun == 3     % SOM-ANFIS
%----------------------------------------------------------------
%   Redo TrainSet to include dates and validation set
%   Use SOM to seperated training and validation set

    pTrainSOM = pTrain+pValid;  %e.g 0.8+0.1
    nData = size(A_Final,1);
    PERM = 1:nData;
    TrainIndSOM=PERM(1:round(pTrainSOM*nData));

    %TrainSetSom = A_Final(TrainIndSOM,:);
     TrainSetSom= A_Final;

    %SOM values --------------------------------
    AORD=TrainSetSom(:, 1:6)'; %all features as input as dimensions

    %Create Topologies
    nrow = 10;
    ncol = 10;
    pos = gridtop([nrow ncol]); %grid pattern
    plotsom(pos);
    %Distance between points
    D2 = dist(pos);

    % %Split into two
    % half = length(AORD)/2;
    % s1 = AORD(1:half);
    % s2 = AORD(half + 1 : end);
    % AORD2 = [s1' s2'];
    %AORD2=AORD2';
    AORD2 = AORD;

    net.trainParam.epochs = 1000;
    net.property.iteration.number = 200;

    net = selforgmap([nrow ncol]);
    net = train(net,AORD2); %Train
    disp(net)
    y = net(AORD2);%Test
    classes = vec2ind(y);
```

```
    disp(classes)

    %Visual  --------------------------------
    figure(1)
    view(net);
    plotsomnc(net);
    figure(2)
    plotsomnd(net);
    plotsomplanes(net);

    figure(3)
    plotsomhits(net,AORD2)
    figure(4);
    plotsompos(net,AORD2);

    %Extract clusters for input  --------------------------------
    hits=sum(sim(net,AORD2)');
    hits=hits';
    classes = vec2ind(net(AORD2))';
    AORD_cluster = [TrainSetSom classes]; %Add to main input

    %rearrane into cluster ascending --------------------------------
    %B_Final = sortrows(AORD_cluster, 8); %Reset vector (includes cluster
number and orginal ordered day)
    B_Final = AORD_cluster;
    %--------------------------------
    %Hybrid SOM-ANFIS
    %--------------------------------
    %TrainInd=PERM(1:round(pTrain*nData));
    %ValidInd=PERM(round(pTrain*nData)+1
:round(pTrain*nData)+round(pValid*nData));

    NOutputSOM =  B_Final(:,1); % SOM
    NInputSOM = B_Final(:,[2 3 4 5 6 8]); %SOM
    %NInputSOM = B_Final(:,[2 3 4 5 6 8]); %SOM with clusters

    %Split train and validate based on clusters from SOM
    TrainInputs=NInputSOM(TrainInd,:);
    TrainTargets=NOutputSOM(TrainInd,:);
    ValidInputs=NInputSOM(ValidInd,:);
    ValidTargets=NOutputSOM(ValidInd,:);
    TestInputs=NInputSOM(TestInd,:);
    TestTargets=NOutputSOM(TestInd,:);

    %Set up structure for anfis
    fis = genfis3(TrainInputs, TrainTargets, MemFun, 'auto', []);

    %View Membership rules
    for input_index=1:1
        subplot(1,1,input_index)
        [x,y]=plotmf(fis,'input',input_index);
        plot(x,y)
        axis([-inf inf 0 1.2]);
        xlabel(['Input' int2str(input_index)]);
    end

    %Configure validation
    opt = anfisOptions('InitialFIS',fis);
    opt.DisplayANFISInformation = 0;
```

```
    opt.DisplayErrorValues = 0;
    opt.DisplayStepSize = 0;
    opt.DisplayFinalResults = 0;
    opt.ValidationData = [ValidInputs ValidTargets];
    %opt.EpochNumber = 5;

    %Run validation and predicion
    %out_fis = anfis([TrainInputs TrainTargets],opt);
    [out_fis,trn_error,step_size,chk_out_fismat,chk_error] =
anfis([TrainInputs TrainTargets],opt);
%       neuroFuzzyDesigner(out_fis)
%       fuzzyLogicDesigner(out_fis)
%       surfview(out_fis)

    %Predict Train
    predictTrain = evalfis(TrainInputs, fis);
    for x = 1:numel(TestTargets)
       dataSimTrain(x) = predictTrain(x) * (MaxAordOt - MinAordOt)+
MinAordOt;
       dataObsTrain(x) = TrainTargets(x) * (MaxAordOt - MinAordOt)+
MinAordOt;
    end
    dataSimTrain = dataSimTrain';
    dataObsTrain = dataObsTrain';

    [nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnMAPE,nnPI]
=asseMetric(dataObsTrain,dataSimTrain);
    SomANFIS_TRAIN_Errors = nnPI;
    asseMetricVis(dataObsTrain,dataSimTrain,nnR,2, strcat(MemFun, ' Train
SOM-ANFIS (Aords with all features)'));

    %Predict Test
    predict = evalfis(TestInputs, out_fis);

   for x = 1:numel(TestTargets)
       dataObs(x) = TestTargets(x) * (MaxAordOt - MinAordOt)+ MinAordOt;
       dataSim(x) = predict(x) * (MaxAordOt - MinAordOt)+ MinAordOt;
    end
    dataObs = dataObs';
    dataSim = dataSim';

    [nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnrMAE,nnPI]
=asseMetric(dataObs,dataSim);
    A_SomANFIS_TEST_Errors = nnPI;
    asseMetricVis(dataObs,dataSim,nnR,2, strcat(MemFun, ' SOM-ANFIS (Aords
with all features)'));

%---------------------------------------------------------------
elseif ModelRun == 4    % single run surgeno (all features except t-1)
%--------------------------------------------------------------
    TrainInputs = TrainInputs(:,[2 3 4 5]);
    ValidInputs = ValidInputs(:,[2 3 4 5]);
    TestInputs = TestInputs(:,[2 3 4 5]);


    %Set up structure for anfis
    fis = genfis3(TrainInputs, TrainTargets, MemFun, 'auto', []);
    neuroFuzzyDesigner(fis)
    fuzzyLogicDesigner(fis)
```

```
    surfview(fis)
    %View Membership rules
    % for input_index=1:1
    %     subplot(1,1,input_index)
    %     [x,y]=plotmf(fis,'input',input_index);
    %     plot(x,y)
    %     axis([-inf inf 0 1.2]);
    %     xlabel(['Input' int2str(input_index)]);
    % end

    %Configure validation
    opt = anfisOptions('InitialFIS',fis);
    opt.DisplayANFISInformation = 0;
    opt.DisplayErrorValues = 0;
    opt.DisplayStepSize = 0;
    opt.DisplayFinalResults = 0;
    opt.ValidationData = [ValidInputs ValidTargets];
    %opt.EpochNumber = 5;



    %Run validation and predicion
    %out_fis = anfis([TrainInputs TrainTargets],opt);
    [out_fis,trn_error,step_size,chk_out_fismat,chk_error] =
anfis([TrainInputs TrainTargets],opt);
     neuroFuzzyDesigner(out_fis)
     fuzzyLogicDesigner(out_fis)
     surfview(out_fis)

    predictTrain = evalfis(TrainInputs, fis);
    for x = 1:numel(TestTargets) %SHOULD THIS BE TrainTargest INSTEAD of
TestTargets
       dataSimTrain(x) = predictTrain(x) * (MaxAordOt - MinAordOt)+
MinAordOt;
       dataObsTrain(x) = TrainTargets(x) * (MaxAordOt - MinAordOt)+
MinAordOt;
    end
    dataSimTrain = dataSimTrain';
    dataObsTrain = dataObsTrain';

    [nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnrMAE,nnPI]
=asseMetric(dataObsTrain,dataSimTrain);
    sugenoTrain_exceptTminus1_Errors = nnPI;
    asseMetricVis(dataObsTrain,dataSimTrain,nnR,2, strcat(MemFun, ' Train
(Aords with all features)'));

    predict = evalfis(TestInputs, out_fis);
    %plot
    % [a,b] = min(chk_error);
    % plot(1:5,trn_error,'g-',1:5,chk_error,'r-',b,a,'ko')
    % title('Training (green) and checking (red) error
curve','fontsize',10)
    % xlabel('Epoch numbers','fontsize',10)
    % ylabel('RMS errors','fontsize',10)


    for x = 1:numel(TestTargets)
       dataObs(x) = TestTargets(x) * (MaxAordOt - MinAordOt)+ MinAordOt;
       dataSim(x) = predict(x) * (MaxAordOt - MinAordOt)+ MinAordOt;
```

```
    end
    dataObs = dataObs';
    dataSim = dataSim';

    [nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnrMAE,nnPI]
=asseMetric(dataObs,dataSim);
    A_SugenoWithoutminus1Errors = nnPI;
    asseMetricVis(dataObs,dataSim,nnR,2, strcat(MemFun, ' (Aords without t-
1)'));

%----------------------------------------------------------------
elseif ModelRun == 5   % single run surgeno (just t-1)
%---------------------------------------------------------------
    TrainInputs = TrainInputs(:,[1]);
    ValidInputs = ValidInputs(:,[1]);
    TestInputs = TestInputs(:,[1]);

    %Set up structure for anfis
    fis = genfis3(TrainInputs, TrainTargets, MemFun, 'auto', []);
    gensurf(fis)

    %View Membership rules
    for input_index=1:1
        subplot(1,1,input_index)
        [x,y]=plotmf(fis,'input',input_index);
        plot(x,y)
        axis([-inf inf 0 1.2]);
        xlabel(['Input' int2str(input_index)]);
    end

    %Configure validation
    opt = anfisOptions('InitialFIS',fis);
    opt.DisplayANFISInformation = 0;
    opt.DisplayErrorValues = 0;
    opt.DisplayStepSize = 0;
    opt.DisplayFinalResults = 0;
    opt.ValidationData = [ValidInputs ValidTargets];
    %opt.EpochNumber = 5;


    %Run validation and predicion
    %out_fis = anfis([TrainInputs TrainTargets],opt);
    [out_fis,trn_error,step_size,chk_out_fismat,chk_error] =
anfis([TrainInputs TrainTargets],opt)

    %Predict Train
    predictTrain = evalfis(TrainInputs, fis);
    for x = 1:numel(TestTargets)
       dataSimTrain(x) = predictTrain(x) * (MaxAordOt - MinAordOt)+
MinAordOt;
       dataObsTrain(x) = TrainTargets(x) * (MaxAordOt - MinAordOt)+
MinAordOt;
    end
    dataSimTrain = dataSimTrain';
    dataObsTrain = dataObsTrain';

    [nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnrMAE,nnPI]
=asseMetric(dataObsTrain,dataSimTrain);
    sugenoTrain_withTminus1_Errors = nnPI;
```

```
    asseMetricVis(dataObsTrain,dataSimTrain,nnR,2, strcat(MemFun, ' Train
(Aords with all features)'));


    %Predict Test
    predict = evalfis(TestInputs, out_fis);
    for x = 1:numel(TestTargets)
       dataObs(x) = TestTargets(x) * (MaxAordOt - MinAordOt)+ MinAordOt;
       dataSim(x) = predict(x) * (MaxAordOt - MinAordOt)+ MinAordOt;
    end
    dataObs = dataObs';
    dataSim = dataSim';

    [nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnrMAE,nnPI]
=asseMetric(dataObs,dataSim);
    A_sugenoTEST_withTminus1_Errors = nnPI;
    asseMetricVis(dataObs,dataSim,nnR,2, strcat(MemFun, ' (Aords with just
t-1)'));

%-----------------------------------------------------------------
elseif ModelRun == 6   % single run madami (just t-1)
%----------------------------------------------------------------
    TrainInputs = TrainInputs(:,[1]);
    ValidInputs = ValidInputs(:,[1]);
    TestInputs = TestInputs(:,[1]);


    %As madami is not accepted in ANFIS -> apply test set (Test +
Validation)
    TrainInputs = [TrainInputs;ValidInputs];
    TrainTargets = [TrainTargets;ValidTargets];

    fis = genfis3(TrainInputs, TrainTargets, 'mamdani', 'auto', []);
    predict = evalfis(TestInputs, fis);

    for x = 1:numel(TestTargets)
       dataObs(x) = TestTargets(x) * (MaxAordOt - MinAordOt)+ MinAordOt;
       dataSim(x) = predict(x) * (MaxAordOt - MinAordOt)+ MinAordOt;
    end
    dataObs = dataObs';
    dataSim = dataSim';

    [nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnrMAE,nnPI]
=asseMetric(dataObs,dataSim);
    A_Mamdani_Errors = nnPI;
    asseMetricVis(dataObs,dataSim,nnR,2, strcat(MemFun, ' (Aords without t-
1)'));

%-----------------------------------------------------------------
elseif ModelRun == 7   % ARIMA
%----------------------------------------------------------------

    %Done in R

%-----------------------------------------------------------------
elseif ModelRun == 8   % comparison of modesl
%----------------------------------------------------------------
    clear
    clc
```

```
    close all

    reData=xlsread('ResultsRun.xlsx','Results_FINAL');
    test = reData(:, 1);
    arima1 = reData(:, 2);
    arima2 = reData(:, 3);
    t1mamdami = reData(:, 4);
    t1Sugerno = reData(:, 5);
    allSugerno = reData(:, 6);
    nt1Sugerno = reData(:, 7);
    en50ANFIS = reData(:, 8);
    en100ANFIS = reData(:, 9);
    SOMANFIS = reData(:, 10);
    arima3 = reData(:, 11);
    arima4 = reData(:, 12);
    arima5 = reData(:, 13);
    arima6 = reData(:, 14);
    arima7 = reData(:, 15);
    arima8 = reData(:, 16);
    arima9 = reData(:, 17);
    en5ANFIS = reData(:, 18);
    en10ANFIS = reData(:, 19);

    models = reData(:,[1 2 3 4 5 6 7 8 9 10]);

%--------------------------------------------------------------------------
-------
%   %Boxplot for ARMIA results 1
%     figure
%     models = reData(:,[11 13 14 12 15]);
%     arima3E = abs(test - arima3);
%     arima5E = abs(test - arima5);
%     arima6E = abs(test - arima6);
%     arima4E = abs(test - arima4);
%     arima7E = abs(test - arima7);
%
%
%     models = [arima3E,arima5E,arima6E,arima4E,arima7E];
%     %boxplot(models)
%
boxplot(models,'symbol','','PlotStyle','traditional','BoxStyle','outline','
Colors','k','Notch','off','MedianStyle','line','labels',{'ARIMA(0,1,0)','AR
IMA(0,1,1)','ARIMA(2,1,0)','ARIMA(1,1,0)','ARIMA Seasonal(0,1,0)(0,0,1)'});
%     ylabel('forecasted error')
%     %xlabel('t1Sugerno','allSugerno','ARIMA 010')
%     title('Model Error Distribution');
%     h = findobj(gca,'Tag','Box');
%     for j=1:length(h)
%         patch(get(h(j),'XData'),get(h(j),'YData'),'y','FaceAlpha',.5);
%     end
%
%--------------------------------------------------------------------------
-------
% %   scatter plot for ANFIS Results 1
%     subplot(2,2,1)
%     hold on;
%     scatter(test,t1Sugerno)
%     title('Sugeno (t-1 All Ords)');
%     xlabel('Data Observation (Index)');
%     ylabel('Data Simulated (Index)');
%     %add line
```

```
%     nnR = corrcoef(test,t1Sugerno);
%     nnR = nnR(1,2);
%     coeffs = polyfit(test, t1Sugerno, 1);
%     % Get fitted values
%     fittedX = linspace(min(test), max(test), 198);
%     fittedY = polyval(coeffs, fittedX);
%     C = round(polyval(coeffs, fittedX(1,1),0));
%     plot(fittedX, fittedY, 'r-', 'LineWidth', 1);
%     txt1 = ['y = ' num2str(round(nnR,3)) 'x + ' num2str(C)];
%     text(min(test),max(t1Sugerno), txt1);
%     text(min(test),max(t1Sugerno)-50, ['r^2 = '
num2str(round(nnR*nnR,3))]);
%     hold off;
%     subplot(2,2,2)
%     hold on;
%     scatter(test,allSugerno)
%     title('Sugeno (all Indexes t-1)');
%     xlabel('Data Observation (Index)');
%     ylabel('Data Simulated (Index)');
%     %add line
%     nnR = corrcoef(test,allSugerno)
%     nnR = nnR(1,2);
%     coeffs = polyfit(test, allSugerno, 1);
%     % Get fitted values
%     fittedX = linspace(min(test), max(test), 198);
%     fittedY = polyval(coeffs, fittedX);
%     C = round(polyval(coeffs, fittedX(1,1),0));
%     plot(fittedX, fittedY, 'r-', 'LineWidth', 1);
%     txt1 = ['y = ' num2str(round(nnR,3)) 'x + ' num2str(C)];
%     ylim =get(gca,'ylim')-30
%     text(min(test),max(ylim), txt1);
%     text(min(test),max(ylim)-50, ['r^2 = ' num2str(round(nnR*nnR,3))]);
%     hold off;
%     subplot(2,2,3)
%     hold on;
%     scatter(test,nt1Sugerno)
%     title('Sugeno (all Indexes without All Ords t-1)');
%     xlabel('Data Observation (Index)');
%     ylabel('Data Simulated (Index)');
%     %add line
%     nnR = corrcoef(test,nt1Sugerno)
%     nnR = nnR(1,2);
%     coeffs = polyfit(test, nt1Sugerno, 1);
%     % Get fitted values
%     fittedX = linspace(min(test), max(test), 198);
%     fittedY = polyval(coeffs, fittedX);
%     C = round(polyval(coeffs, fittedX(1,1),0));
%     plot(fittedX, fittedY, 'r-', 'LineWidth', 1);
%     txt1 = ['y = ' num2str(round(nnR,3)) 'x + ' num2str(C)];
%     ylim =get(gca,'ylim')-30
%     text(min(test),max(ylim), txt1);
%     text(min(test),max(ylim)-70, ['r^2 = ' num2str(round(nnR*nnR,3))]);
%     hold off;
%     subplot(2,2,4)
%     hold on;
%     scatter(test,t1mamdami)
%     title('Mamdani (t-1 All Ords )');
%     xlabel('Data Observation (Index)');
%     ylabel('Data Simulated (Index)');
%     %add line
%     nnR = corrcoef(test,t1mamdami)
```

```
%     nnR = nnR(1,2);
%     coeffs = polyfit(test, t1mamdami, 1);
%     % Get fitted values
%     fittedX = linspace(min(test), max(test), 198);
%     fittedY = polyval(coeffs, fittedX);
%     C = round(polyval(coeffs, fittedX(1,1),0));
%     plot(fittedX, fittedY, 'r-', 'LineWidth', 1);
%     txt1 = ['y = ' num2str(round(nnR,3)) 'x + ' num2str(C)];
%     ylim =get(gca,'ylim')+80
%     text(min(test),min(t1mamdami)+5, txt1);
%     text(min(test),min(t1mamdami)+1, ['r^2 = '
num2str(round(nnR*nnR,3))]);
%     hold off;

%  %Results 1
% Timeseries
%     figure
%     plot(test,'k');
%     hold on;
%     plot(t1Sugerno);
%     plot(allSugerno);
%     plot(arima3);
%     title('ANFIS vs ARIMA')
%     xlabel('days')
%     ylabel('\Delta [All Ord Price]')
%     var = {{'Observation','t1Sugerno','allSugerno','ARIMA
010'},'Location','northwest'}
%     legend(var{:})
%
%legend({'Test','arima1','arima2','t1mamdami','t1Sugerno','allSugerno','nt1
Sugerno','en50ANFIS','en100ANFIS','SOMANFIS'},'Location','northwest','NumCo
lumns',2)
%     hold off

%Boxplot
%     figure
%     models = reData(:,[1 5 6 11]);
%     t1SugernoE = abs(test - t1Sugerno);
%     allSugernoE = abs(test - allSugerno);
%     arima3E = abs(test - arima3);
%     models = [t1SugernoE,allSugernoE,arima3E];
%     %boxplot(models)
%
boxplot(models,'symbol','','PlotStyle','traditional','BoxStyle','outline','
Colors','k','Notch','off','MedianStyle','line','labels',{'SingleFeat(t-1)
Sugerno','AllFeat(t-1) Sugerno','ARIMA 010'});
%     ylabel('forecasted error')
%     %xlabel('t1Sugerno','allSugerno','ARIMA 010')
%     title('Model Error Distribution');
%     h = findobj(gca,'Tag','Box');
%     for j=1:length(h)
%         patch(get(h(j),'XData'),get(h(j),'YData'),'y','FaceAlpha',.5);
%     end

 %Results 2
% time series
%     plot(test,'k');
%     hold on;
%     plot(SOMANFIS);
%     plot(en100ANFIS);
%     title('SOM-ANFIS vs Ensemble ANFIS')
```

```
%     xlabel('days')
%     ylabel('\Delta [All Ord Price]')
%     var = {{'Observation','SOM-ANFIS','Ensemble100
ANFIS'},'Location','southwest'}
%     legend(var{:})
%
%legend({'Test','arima1','arima2','t1mamdami','t1Sugerno','allSugerno','nt1
Sugerno','en50ANFIS','en100ANFIS','SOMANFIS'},'Location','northwest','NumCo
lumns',2)
%     hold off

%
%   % boxplot
%     %models = reData(:,[1 8 9 10]);
%     en50ANFISE = abs(test - en50ANFIS);
%     en100ANFISE = abs(test - en100ANFIS);
%     SOMANFISE = abs(test - SOMANFIS);
%     models = [SOMANFISE,en100ANFISE,en50ANFISE];
%     %boxplot(models)
%
boxplot(models,'symbol','','PlotStyle','traditional','BoxStyle','outline','
Colors','k','Notch','off','MedianStyle','line','labels',{'SOM-
ANFIS','Ensemble100 ANFIS','Ensemble50 ANFIS'});
%     ylabel('forecasted error')
%     title('Model Error Distribution');
%     h = findobj(gca,'Tag','Box');
%     for j=1:length(h)
%         patch(get(h(j),'XData'),get(h(j),'YData'),'y','FaceAlpha',.5);
%     end




%
%     dataObs = test;
%     dataSim = arima2;
%     [nnR,nnENS,nnD,nnLeg,nnRMSE,nnrRMSE,nnMAE,nnrMAE,nnPI]
=asseMetric(dataObs,dataSim);
%     arima4_Errors = nnPI;



%plot of arima models
%     Arimalmodels = reData(:,[1 2 3 11 12 13 14 15 16 17]);
%     figure
%     plot(Arimalmodels);
%     title('Test vs Arima Model Forecast')
%     xlabel('days')
%     ylabel('\Delta [All Ord Price]')
%     var =
{{'Test','arima1','arima2','arima3','arima4','arima5','arima6','arima7','ar
ima8','arima9','arima10'},'Location','southeast'}
%     legend(var{:})
%
%     Arimalmodels = reData(:,[1 2 3 11]);
%     figure
%     plot(Arimalmodels);
%     title('Test vs Arima Model Forecast')
```

```
%     xlabel('days')
%     ylabel('\Delta [All Ord Price]')
%     var =
{{'Test','arima1','arima2','arima3','arima4'},'Location','southeast'}
%     legend(var{:})
%
%     Arimalmodels = reData(:,[1 12 13 14]);
%     figure
%     plot(Arimalmodels);
%     title('Test vs Arima Model Forecast')
%     xlabel('days')
%     ylabel('\Delta [All Ord Price]')
%     var = {{'Test','arima5','arima6','arima7'},'Location','southeast'}
%     legend(var{:})
%
%     Arimalmodels = reData(:,[1 15 16 17]);
%     figure
%     plot(Arimalmodels);
%     title('Test vs Arima Model Forecast')
%     xlabel('days')
%     ylabel('\Delta [All Ord Price]')
%     var = {{'Test','arima8','arima9','arima10'},'Location','southwest'}
%     legend(var{:})

    subplot(2,2,1)
    dataErr = abs(test - en5ANFIS);
    hmin = min(dataErr);
    hmax = max(dataErr);
    hold on;
    hist(dataErr,[hmin:10:hmax]);
    title({'ANFIS-Ensemble 5','error frequency'})
    ylabel('Frequency of Error');
    xlabel('Errors (brackets ± 10)');
    hold off;
    subplot(2,2,2)
    dataErr = abs(test - en10ANFIS);
    hmin = min(dataErr);
    hmax = max(dataErr);
    hold on;
    hist(dataErr,[hmin:10:hmax]);
    title({'ANFIS-Ensemble 10','error frequency'})
    ylabel('Frequency of Error');
    xlabel('Errors (brackets ± 10)');
    hold off;
    subplot(2,2,3)
    dataErr = abs(test - en50ANFIS);
    hmin = min(dataErr);
    hmax = max(dataErr);
    hold on;
    hist(dataErr,[hmin:10:hmax]);
    title({'ANFIS-Ensemble 50','error frequency'})
    ylabel('Frequency of Error');
    xlabel('Errors (brackets ± 10)');
    hold off;
    subplot(2,2,4)
    dataErr = abs(test - en100ANFIS);
    hmin = min(dataErr);
    hmax = max(dataErr);
    hold on;
    hist(dataErr,[hmin:10:hmax]);
    title({'ANFIS-Ensemble 100','error frequency'})
```

```
    ylabel('Frequency of Error');
    xlabel('Errors (brackets ± 100)');
    hold off;


%     %subplot(1,1,1)
%     violin([test arima1 arima2 t1mamdami t1Sugerno allSugerno nt1Sugerno
en50ANFIS en100ANFIS SOMANFIS],...
%         'facecolor',[[1 0 0];[0 0 1];[0 0 1];[0 0 1];[0 0 1];[0 0 1];[0 0
1];[0 0 1];[0 0 1];[0 0 1]],...
%         'medc','','mc','k')

%         %Kernel Desity function
%         'edgecolor','b',...
%         'bw',0.3,...
%         'mc','k',...
%         'medc','r--')
%         ylabel('\Delta [All Ord Price]','FontSize',14)



%'xlabel',{'Test','arima1','arima2','t1mamdami','t1Sugerno','allSugerno'},.
..
    %'xlabel',{'a','b','c','d','a','b','c','d','c','d'},...
    % xlabel:    xlabel. Set either [] or in the form
{'txt1','txt2','txt3',...}
    figure
    plot(models);
    title('Test vs Model Forecast')
    xlabel('days')
    ylabel('\Delta [All Ord Price]')
    var =
{{'Test','arima1','arima2','t1mamdami','t1Sugerno','allSugerno','nt1Sugerno
','en50ANFIS','en100ANFIS','SOMANFIS'},'Location','southwest'}
    legend(var{:})

%legend({'Test','arima1','arima2','t1mamdami','t1Sugerno','allSugerno','nt1
Sugerno','en50ANFIS','en100ANFIS','SOMANFIS'},'Location','northwest','NumCo
lumns',2)


%----------------------------------------------------------------
elseif ModelRun == 9   % plot inital time series
%---------------------------------------------------------------

% %Ploting actual stock Indexes
plot(A);
count1 = timeseries(A(:,1),1:size(A,1),'name', 'AORD');
count1.TimeInfo.Units = 'Daily Count';
% count1.TimeInfo.Units = 'days';
% count1.TimeInfo.StartDate = '01-Jan-2008';     % Set start date.
% count1.TimeInfo.Format = 'dd mmm, yy';       % Set format for display on
x-axis.
% count1.Time = count1.Time - count1.Time(1);        % Express time
relative to the start date.

plot(count1,'k','LineWidth',0.01)
grid on
hold on
```

```
count2 = timeseries(A(:,2),1:size(A,1),'name', 'DJI');
plot(count2,':r')
count3 = timeseries(A(:,3),1:size(A,1),'name', 'FTSE');
plot(count3,':g')
count4 = timeseries(A(:,4),1:size(A,1),'name', 'HSI');
plot(count4,':c')
count5 = timeseries(A(:,5),1:size(A,1),'name', 'N225');
plot(count5,':m')
title('Time Series: Daily Five Stock Indexes Price')
xlabel('Time (Days)')
ylabel('Price (Actual)')
legend('AORD','DJI','FTSE','HSI','N225','Location','northwest')


% % Ploting normalise stock Indexes
% plot(A_Final);
% count1 = timeseries(A_Final(:,2),1:size(A_Final,1),'name', 'AORD');
% count1.TimeInfo.Units = 'Daily Count';
% % count1.TimeInfo.Units = 'days';
% % count1.TimeInfo.StartDate = '01-Jan-2008';     % Set start date.
% % count1.TimeInfo.Format = 'dd mmm, yy';        % Set format for display
on x-axis.
% % count1.Time = count1.Time - count1.Time(1);        % Express time
relative to the start date.
%
% plot(count1,'k','LineWidth',0.01)
% grid on
% hold on
% count2 = timeseries(A_Final(:,3),1:size(A_Final,1),'name', 'DJI');
% plot(count2,':r')
% count3 = timeseries(A_Final(:,4),1:size(A_Final,1),'name', 'FTSE');
% plot(count3,':g')
% count4 = timeseries(A_Final(:,5),1:size(A_Final,1),'name', 'HSI');
% plot(count4,':c')
% count5 = timeseries(A_Final(:,6),1:size(A_Final,1),'name', 'N225');
% plot(count5,':m')
% title('Time Series: Normailised Five Stock Indexes')
% xlabel('Time (Days)')
% ylabel('Price (Normalised)')
% legend('AORD','DJI','FTSE','HSI','N225','Location','southeast')
%

%---------------------------------------------------------------
elseif ModelRun == 10   % ensemble plot
%--------------------------------------------------------------

    clear
    clc
    close all

    Sim=xlsread('ResultsRun.xlsx','Ensemble100');
    test = Sim(:, 2);
    Sim(:,1)=[];
    Sim5 = Sim(:,1:5)
    Sim10 = Sim(:,1:10)
    Sim50 = Sim(:,1:50)
    AvSim = mean(Sim,2)';
    AvSim5 = mean(Sim5,2)';
    AvSim10 = mean(Sim10,2)';
    AvSim50 = mean(Sim50,2)';
```

```
    %Sim = Sim.'
    ylimMax1 = max(Sim);
    ylimMax1 = max(ylimMax1);
    ylimMax2 = max(test);
    ylimMax = max(ylimMax1,ylimMax2);
    ylimMin1 = min(Sim);
    ylimMin1 = min(ylimMin1);
    ylimMin2 = min(test);
    ylimMin = min(ylimMin1,ylimMin2);


    Sim = Sim'
    pos = max(Sim);
    %pos = quantile(Sim,0.25)
    pos1= abs(AvSim-pos)

    neg = min(Sim);
    %neg= quantile(Sim,0.75)
    neg1= abs(AvSim-neg)


    titleN = "Ensemble 100 forecast";
     figure(1)
     %boxplot(Sim.','symbol','','plotstyle',
'compact','whisker',1.5,'BoxStyle','outline','Colors',[0.8 0.8
0.8],'Notch','off','MedianStyle','line');

%boxplot(Sim.','symbol','','PlotStyle','traditional','BoxStyle','outline','
Colors',[0 0 0],'Notch','off','MedianStyle','line');

boxplot(Sim,'symbol','','PlotStyle','traditional','BoxStyle','outline','Col
ors',[0 0 0],'Notch','on','Whisker',20,'MedianStyle','line');
     %errorbar(1:length(AvSim),AvSim,neg1,pos1,'Color',[0.8 0.8
0.8])%,neg,pos)
     title('AllOrds Index ensemble forecast');
     ylim([ylimMin-10 ylimMax+10])
     set(findobj(gcf,'LineStyle','--'),'LineStyle','-') % solid lines
     hold on;
     %axis auto
    h1=plot(test,...
    '-o','MarkerSize',3,...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerFaceColor','red','Color',[0.8 0.8 0.8]);
    h2=plot(AvSim,...
    '-o','MarkerSize',3,...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerFaceColor','green','Color',[0.8 0.8 0.8]);
    set(gca, 'XTickMode', 'auto', 'XTickLabelMode', 'auto')
    xlabel('Time - (days)');
    ylabel('Value of Index ');
    legend([h1 h2],'observed','forecasted','Location', 'northwest');
    hold off;

    figure(2)
    subplot(4,1,1)

boxplot(Sim5.','symbol','','PlotStyle','traditional','BoxStyle','filled','C
olors',[0.8 0.8 0.8],'Notch','on','Whisker',20,'MedianStyle','line');
    hold on;
```

```
    plot(test,'-o','MarkerSize',3,'MarkerEdgeColor',[0 0
0],'MarkerFaceColor','red','Color',[0 0 0]);
    plot(AvSim5,'-o','MarkerSize',3,'MarkerEdgeColor',[0 0
0],'MarkerFaceColor','green','Color',[0 0 0]);
    set(gca, 'XTickMode', 'auto', 'XTickLabelMode', 'auto')
    legend('observed','forecasted','Location', 'northwest');
    ylabel('Ensemble - 5','FontSize',9);
    subplot(4,1,2)

boxplot(Sim10.','symbol','','PlotStyle','traditional','BoxStyle','filled','
Colors',[0.8 0.8 0.8],'Notch','on','Whisker',20,'MedianStyle','line');
    hold on;
    plot(test,'-o','MarkerSize',3,'MarkerEdgeColor',[0 0
0],'MarkerFaceColor','red','Color',[0 0 0]);
    plot(AvSim10,'-o','MarkerSize',3,'MarkerEdgeColor',[0 0
0],'MarkerFaceColor','green','Color',[0 0 0])
    set(gca, 'XTickMode', 'auto', 'XTickLabelMode', 'auto')
    legend('observed','forecasted','Location', 'northwest');
    ylabel('Ensemble - 10','FontSize',9);
    subplot(4,1,3)

boxplot(Sim50.','symbol','','PlotStyle','traditional','BoxStyle','filled','
Colors',[0.8 0.8 0.8],'Notch','on','Whisker',20,'MedianStyle','line');
    hold on;
    plot(test,'-o','MarkerSize',3,'MarkerEdgeColor',[0 0
0],'MarkerFaceColor','red','Color',[0 0 0]);
    plot(AvSim50,'-o','MarkerSize',3,'MarkerEdgeColor',[0 0
0],'MarkerFaceColor','green','Color',[0 0 0])
    set(gca, 'XTickMode', 'auto', 'XTickLabelMode', 'auto')
    legend('observed','forecasted','Location', 'northwest');
    ylabel('Ensemble - 50','FontSize',9);
    subplot(4,1,4)

%boxplot(Sim.','symbol','','PlotStyle','traditional','BoxStyle','outline','
Colors',[0 0 0],'Notch','on','Whisker',20,'MedianStyle','line');

boxplot(Sim,'symbol','','PlotStyle','traditional','BoxStyle','filled','Colo
rs',[0.8 0.8 0.8],'Notch','on','Whisker',20,'MedianStyle','line');
    hold on;
    plot(test,'-o','MarkerSize',3,'MarkerEdgeColor',[0 0
0],'MarkerFaceColor','red','Color',[0 0 0]);
    plot(AvSim,'-o','MarkerSize',3,'MarkerEdgeColor',[0 0
0],'MarkerFaceColor','green','Color',[0 0 0])
    set(gca, 'XTickMode', 'auto', 'XTickLabelMode', 'auto');
    legend('observed','forecasted','Location', 'northwest');
    ylabel('Ensemble - 100','FontSize',9);
    hold off;
    xlabel('Time - (days)');

end
```

Performance function

```
function [r,ENS,d,Leg,RMSE,rRMSE,MAE,MAPE,PI]=asseMetric(dataObs,dataSim)
%Function requiring forcasted, observed values to then
%calculate acceptable assesment metrics.

% correlation coefficient, r:
R21=sum((dataObs -mean(dataObs)).*( dataSim - mean(dataSim)));
R22=sqrt(sumsqr(dataObs -mean(dataObs)).*sumsqr(dataSim - mean(dataSim)));
```

```
r = R21/R22;

% Nash-Sutcliffe coefficient, ENS
ENS = (1-(sumsqr(dataObs - dataSim)/sumsqr(dataObs - mean(dataObs))));

% Willmottr's Index, d
d =(1-(sumsqr(dataObs - dataSim)/sumsqr(abs(dataSim - mean(dataObs) +
abs(dataObs -mean(dataObs))))));

% % **** NOT USED ****
% % Peak Percentage Deviation, Pdv
% Pdv = (1-max(dataSim)/max(dataObs))*100;

% Root Mean Square Error, RMSE
RMSE = sqrt(sumsqr(dataObs - dataSim)/length(dataObs));

%Relative RMSE
rRMSE = (sqrt(sumsqr(dataObs -
dataSim)/length(dataObs)))/(sum(dataObs)/length(dataObs)) * 100;

% Mean Absolute Error, MAE
MAE = (sum(abs(dataSim - dataObs)))/length(dataObs);

%MAPE
MAPE =(sum(abs((dataSim - dataObs)./dataObs)))/length(dataObs)*100;

%Legates & McCabes Index, Leg
L = sum(abs(dataSim - dataObs));
M = sum(abs(dataObs - mean(dataObs)));
Leg = (1-L/M);

% All error values within the same matrix, PI
%PI = [r ENS d Leg RMSE rRMSE MAE rMAE] ;
PI = [r ENS d Leg RMSE rRMSE MAE MAPE] ;

end
```

Visual metrics function

```
function asseMetricVis(dataObs,dataSim,nnR,style,titleN)
%Function requiring forcasted, observed values and optional values
%nnR - R^2 value
%style - options for display of figures (1 or 2)
%titleN - title to be included (string)

clear title xlabel ylabel

n = get(gcf,'Number');%set figure number

[m1,n1] = size(dataSim);
[m2,n2] = size(dataObs);

%check dimension of matrix to set limits for x axis
if n1 == 1 && n2 == 1
    ylimMax1 = max(dataSim);
    ylimMax2 = max(dataObs);
```

```
    ylimMax = max(ylimMax1,ylimMax2);
    ylimMin1 = min(dataSim);
    ylimMin2 = min(dataObs);
    ylimMin = min(ylimMin1,ylimMin2);

else
    ylimMax1 = max(dataSim)
    ylimMax1 = max(ylimMax1)
    ylimMax2 = max(dataObs)
    ylimMax = max(ylimMax1,ylimMax2);
    ylimMin1 = min(dataSim);
    ylimMin1 = min(ylimMin1);
    ylimMin2 = min(dataObs);
    ylimMin = min(ylimMin1,ylimMin2);
end
     ylimMin = ylimMin-100
     ylimMax = ylimMax+100

if style == 1  %Tradition visualisation with each figure has separate
window

    %histogram
    figure(n+1)
    n = n+1;
    dataErr = abs(dataObs - dataSim);
    hmin = min(dataErr);
    hmax = max(dataErr);
    hold on;
    hist(dataErr,[hmin:1:hmax]);
    title({titleN,'Histogram absolute error frequency'});
    ylabel('Frequency of Error');
    xlabel('Errors ');
    hold off;

    % Scatterplot
    figure(n+1)
    n = n+1;
    hold on;
    scatter(dataObs,dataSim)
    ylim([ylimMin ylimMax])
    %text(2,2, ['R^2 = ' num2str(nnR)]);
    title({titleN,'Scatterplot prediction againts observed'});
    xlabel('Data Observation - Value');
    ylabel('Data Simulated - Value');
    %add line
    coeffs = polyfit(dataObs, dataSim, 1);
    % Get fitted values
    fittedX = linspace(min(dataObs), max(dataObs), 200);
    fittedY = polyval(coeffs, fittedX);
    plot(fittedX, fittedY, 'r-', 'LineWidth', 3);
    legend(['R^2 = ' num2str(nnR)],'line of best fit' ,'Location', 'Best');
    hold off;

    %time series
    figure(n+1)
    n = n+1;
    hold on;
    plot(dataObs);
    plot(dataSim);
    ylim([ylimMin ylimMax]);
```

```
    title({titleN,'Timeseries prediction againts observed'});
    ylabel('Value of Index');
    xlabel('Time - (days)');
    legend('observed','forecasted','Location', 'Best');
    hold off;

elseif style == 2 %visualisation in one window for easy comparison

    figure(n+1)
    %time series
    subplot(2,1,1)
    hold on;
    plot(dataObs);
    plot(dataSim);
    ylim([ylimMin ylimMax]);
    title({titleN,'Timeseries prediction againts observed'});
    ylabel('Value of Index');
    xlabel('Time - (days)');
    legend('observed','forecasted','Location', 'Best');
    hold off;

    %Histogram
    subplot(2,2,3)
    hold on;
    dataErr = abs(dataObs - dataSim);
    hmin = min(dataErr);
    hmax = max(dataErr);
    hist(dataErr,[hmin:1:hmax]);
    title('Histogram absolute error frequency');
    ylabel('Frequency of Error');
    xlabel('Errors ');
    hold off;

    % Scatterplot
    subplot(2,2,4)
    hold on;
    scatter(dataObs,dataSim)
    ylim([ylimMin ylimMax])
    title('Scatterplot prediction againts observed');
    xlabel('Data Observation - Value');
    ylabel('Data Simulated - Value');
    %add line
    coeffs = polyfit(dataObs, dataSim, 1);
    % Get fitted values
    fittedX = linspace(min(dataObs), max(dataObs), 200);
    fittedY = polyval(coeffs, fittedX);
    plot(fittedX, fittedY, 'r-', 'LineWidth', 3);
    legend(['R^2 = ' num2str(nnR)],'line of best fit' ,'Location', 'Best');
    hold off;

elseif style == 3 %visualisation with box plots

    figure(n+1)
    hold on;
    boxplot(dataSim);
    title('All Ords Index Forecasted Distribution')
    xlabel(titleN);
    ylabel('Price of Stock ($)');
    hold off;
```

```
elseif style == 4 %line graph for use with box plots - Styl5

%     figure(n+1)
%     hold on;
%    % boxplot(dataSim.', 'orientation', 'horizontal');
%     boxplot(dataSim.');
%     title('All Ords Index Forecasted Distribution')
%     ylabel('Price of Stock ($)');
     figure(n+1)

boxplot(dataSim.','symbol','','PlotStyle','traditional','BoxStyle','outline
','Colors',[0.7 0.7 0.7],'Notch','off','MedianStyle','line');
%
boxplot(dataSim.','symbol','','PlotStyle','traditional','BoxStyle','outline
','Colors',[0 0 0],'Notch','off','MedianStyle','line');
     title('All Ords Index ensemble Forecasted');

     ylim([ylimMin ylimMax])
     hold on;
     %axis auto
     plot(dataObs,...
    '-o','MarkerSize',3,...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerFaceColor','red','Color',[0 0 0]);

elseif style == 5 %linestyle with box plots

    %time series
    %plot(dataObs);
    %plot(dataSim);
    plot(dataObs,...
    '-o','MarkerSize',3,...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerFaceColor','red','Color',[0 0 0]);
    plot(dataSim,...
    '-o','MarkerSize',3,...
    'MarkerEdgeColor',[0 0 0],...
    'MarkerFaceColor','green','Color',[0 0 0]);

    ylim([ylimMin ylimMax]);
    ylabel('Value of Index');
    xlabel('Time - (days)');
    legend('observed','forecasted','Location', 'Best');
    hold off


end

end
```

R code

ARIMA model development

```r
# ARIMA model Development
# MSC8001/MSC8002
#Author: John Worrall
# Description: Given input data, develop both seansonal/nonseasonal ARIMA
# parameter development and performance
# Requirements: Train/Test.csv (Data Input)
#----------------------------

train = read.table("train.csv", header = FALSE)
test = read.table("test.csv", header = FALSE)
allData <- rbind(train,test)

library(forecast)

#Auto model development
fit <- auto.arima(train)
summary(fit)
forc <- forecast(fit,h=198)
plot(forc)
lines(allData, col='red', add = TRUE)
lines(forc$x, col='black', add = TRUE)
write.table(forc$mean, "AutoArima.txt", sep="\t")
#view autoarima vs observed test
predDF2 <- as.numeric(forc$mean)
predDF <- summary(forc)
Sim <- predDF[1]
Sim1 <- as.vector(Sim)
obs <- test
accuracy(Sim1,obs)

#Seasonal model development parameter assignment
fit_Seasonal <- auto.arima(train, seasonal = T)
summary(fit_Seasonal)
forS <- forecast(fit_Seasonal,h=198)
plot(forS)


#Analysis parameters - output AIC BIC
library(tseries)
adf = adf.test(train[,1])
Stl = Stl(train[,1],s.window=”periodic”)
(fit1 <- arima(train, c(1, 0, 0)))
(fit1 <- arima(train, c(1, 0, 0)))

qxts <- xts(train)
mod.arima <- get.best.arima(train[,1], maxord = c(2, 2, 2, 2, 2, 2))[[2]]

M=matrix(NA,36,13)
k=0 # current row being filled in
for(i in 0:5){
  for(j in 0:5){
    k=k+1
    fit <- arima(train[,1], order=c(i,0,j), include.mean=TRUE)
    if(i>0) M[k,c(1:   i) ]=fit$coef[c(   1 :   i )] # AR coefficients in
the 2nd-6th columns
    if(j>0) M[k,c(8:(7+j))]=fit$coef[c((i+1):(i+j))] # MA coefficients in
the 8th-12th columns
```

```
    M[k,      13  ]=tail(fit$coef,1)         # "intercept" (actually, mean)
in the 13th column
  }
}

#Model development manual parameter assignment
fit2 = arima(train, order = c(2, 0, 2),include.mean=FALSE)
summary(fit2)
forc2 <- forecast(fit2,h=198)
plot(forc2)
fit3 = arima(train, order = c(2, 1, 2),include.mean=FALSE)
summary(fit3)
forc3 <- forecast(fit3,h=198)
plot(forc3)
fit4 = arima(train, order = c(1, 1, 5),include.mean=FALSE)
summary(fit4)
forc4 <- forecast(fit4,h=198)
plot(forc4)
fit5 = arima(train, order = c(3, 1, 3),include.mean=FALSE)
summary(fit5)
forc5 <- forecast(fit5,h=198)
plot(forc5)

#Fit without drift
fitARIMA2 <- auto.arima(train, trace=TRUE, allowdrift=FALSE)
lines(fore3$pred, col='green', add = TRUE)
aa <- fore3[1]
lines(aa)
futurVal <- forecast.Arima(fitARIMA,h=10, level=c(99.5))
plot.forecast(futurVal)


#non-seasonal models
allData <- rbind(train,test)
modArima<-auto.arima(train,D=7,max.P = 5, max.Q =
5,trace=TRUE,approximation=FALSE, stepwise = F)
arima_010 = arima(train, order = c(0, 1, 0))
arima_110 = arima(train, order = c(1, 1, 0))
arima_011 = arima(train, order = c(0, 1, 1))
arima_210 = arima(train, order = c(2, 1, 0))
arima_012 = arima(train, order = c(0, 1, 2))
arima_111 = arima(train, order = c(1, 1, 1))
arima_013 = arima(train, order = c(0, 1, 3))
arima_310 = arima(train, order = c(3, 1, 0))
arima_211 = arima(train, order = c(2, 1, 1))
arima_112 = arima(train, order = c(1, 1, 2))
forc <- forecast(arima_210,h=198)
Sim <- as.numeric(forc$mean)
write.table(Sim, "arima_210.txt", sep=",")
#BIC values
AIC(arima_010)
bic=AIC(arima_010,k = log(length(train)))


library(lmtest)
#Seasonal models
TrainTs <- ts(train, start=c(2008,1), end=c(2017, 12), frequency=12)
sesArima <-auto.arima(myts,max.P = 5, max.Q =
5,trace=TRUE,approximation=FALSE, stepwise = F, seasonal = T)
(arimaSea_010_001 = arima(TrainTs, order = c(0, 1, 0),seasonal = list(order
= c(0,0,1))))
```

```
(arimaSea_010_002 = arima(TrainTs, order = c(0, 1, 0),seasonal = list(order
= c(0,0,2))))
(arimaSea_010_003 = arima(TrainTs, order = c(0, 1, 0),seasonal = list(order
= c(0,0,3))))
(arimaSea_010_100 = arima(TrainTs, order = c(0, 1, 0),seasonal = list(order
= c(1,0,0))))
(arimaSea_010_101 = arima(TrainTs, order = c(0, 1, 0),seasonal = list(order
= c(1,0,1))))
(arimaSea_010_102 = arima(TrainTs, order = c(0, 1, 0),seasonal = list(order
= c(1,0,2))))
(arimaSea_010_103 = arima(TrainTs, order = c(0, 1, 0),seasonal = list(order
= c(1,0,3))))
(arimaSea_010_200 = arima(TrainTs, order = c(0, 1, 0),seasonal = list(order
= c(2,0,0))))
(arimaSea_010_201 = arima(TrainTs, order = c(0, 1, 0),seasonal = list(order
= c(2,0,1))))
(arimaSea_010_202 = arima(TrainTs, order = c(0, 1, 0),seasonal = list(order
= c(2,0,2))))
forc2 <- forecast(arimaSea_010_011,h=198)
Sim <- as.numeric(forc$mean)
write.table(Sim, "arimaSea_010_003.txt", sep=",")
```