HYBRID APPROACH FOR FINANCIAL FORECASTING WITH SUPPORT VECTOR MACHINES

Withanage Don Sameera Roshan

(118006X)

Degree of Master of Science

Department of Mechanical Engineering

University of Moratuwa
Sri Lanka

June 2014

W.D.S. Roshan M.Sc 2014

HYBRID APPROACH FOR FINANCIAL FORECASTING WITH SUPPORT VECTOR MACHINES

Withanage Don Sameera Roshan

(118006X)

Thesis submitted in partial fulfillment of the requirements for
the degree Master of Science

Department of Mechanical Engineering

University of Moratuwa
Sri Lanka

June 2014

Declaration

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Mr. W.D.S. Roshan

Signature:

Date:

The above candidate has carried out research for the Master's thesis under our supervision.

Dr. R.A.R.C Gopura
Department of Mechanical Engineering,
University of Moratuwa

Signature of the supervisor:

Date:

Dr. A.G.B.P Jayasekara
Department of Electrical Engineering,
University of Moratuwa

Signature of the supervisor:

Date:

Dedicated
To my parents

Acknowledgements

I would like to thank my supervisors Dr. A.G.B.P Jayasekara and Dr. R.A.R.C. Gopura for their continuous support given to me throughout the master's program. Thank you for your immense support for thesis writing guidelines, reference materials, motivation and vision provided by Dr. A.G.B.P Jayasekara. It has been a great pleasure to explore the new knowledge with the friendly environment created by Dr. R.A.R.C. Gopura. I really appreciate the trust and confidence you gave me throughout the whole period of time. Thank you both for the great support given.

I would like to thank the members of the progress review committee, Dr. K.T.M.U. Hemapala and Dr.V.P.C. Dassanayake for giving their ideas, directions, positives and cons by overseeing the progress of the work.

Withanage Don Sameera Roshan
sameeraroshanuom@gmail.com

Abstract

Financial markets are the biggest business platforms in the world. Therefore, financial forecasting is getting a lot of attention in today's economic context. Accurate forecast is beneficial to broker firms, governments, individuals etc.

Vast range of forecasting methods, models have introduced by the research community. However, the risk involved with trading on those markets are very high. Such complexity makes a difficulty of making consistent profit. Building an accurate forecasting model is still an active and interesting research area for the academic community.

Recently, nonlinear statistical models such as neural network, support vector machine have shown greater capability to forecast financial markets over conventional methods. This dissertation proposed a hybrid support vector machine model which consists of wavelet transform and k-means clustering for foreign exchange market forecasting. The proposed model analyzes the trends and makes a forecast by entirely depending on the past exchange data. Wavelet transform is used to remove the noise of the time series. K-means clustering cluster the input space according to the similarities of the input vectors and finally support vector models make a forecast for the relevant cluster.

The proposed hybrid forecasting system was tested on real market environment to check the forecasting capability. Auto trading algorithm developed on 'metatrader4' platform used the forecast of the model to trade on the real conditions. Results confirmed that the proposed model can forecast price movements with greater accuracy that leads to profitable trades on foreign exchange market.

Table of content

List of figures

List of tables

List of abbreviations

ABC - Artificial Bee Colony Algorithm

APE - Absolute Percentage Error

ARIMA - Auto Regression Integrated Moving Average

BP - Back Propagation

CD - Correct Down Trend

DS - Directional Symmetry

DWT - Discrete Wavelet Transform

EA - Expert Advisor

EKF - Extended Kalman Filter

EMA - Exponential Moving Average

EUR - Euro

FFT - Fast Fourier Transform

FLNN - Functional Link Neural Network

FOREX – Foreign Exchange

GA - Genetic Algorithm

GARCH - Generalized Autoregressive Conditional Heteroskedasticity

GLAR - Generalized Auto Regression

IBCO - Improved Bacterial Chemotaxis Optimization

ICA - Independent Component Analysis

ICA - Independent Component Analysis

JPY – Japan Yen

MAD – Mean Absolute Deviation

MAE - Mean Absolute Error

MLP - Multilayer Perceptron

MSE - Mean Squared Error

NMSE - Normalized Mean Squared Error

PCA - Principal Component Analysis

PCA - Principal Component Analysis

PSNN - Pi-Sigma Neural Network

PSO - Particle-Swarm Optimization

RBF - Radial Basis Functions

RMSE - Root Mean Square Error

RNN - Recurrent Neural Network

RPNN - Ridge Polynomial Neural Network

RW - Random Walk

SOM - Self-Organizing Maps

SVM - Support Vector Machine

SVR - Support Vector Regression

SWT - Stationary Wavelet Transforms

TAIEX - Taiwan Capitalization Weighted Stock Index

TEMA - Triple Exponential Moving Average

USD - US Dollar

VC - Vapnik-Chervonenkis

WT - Wavelet Transform

1. INTRODUCTION

1.1 Financial Markets

Financial market is a place where we can trade financial securities, commodities and other fungible items. Various kinds of financial markets exist in the today's economic context. Large multinational firms, government firms, local firms, agencies, individual buyers are interacting each other using financial markets for the purpose of fund raising, transfer risks or international trading. Capital markets, Commodity markets, Money markets, Derivative markets, Futures markets, Insurance markets and Foreign exchange market (FOREX) are different forms of financial markets. Since those markets are highly volatile, the risk involved in the markets are very high. One can lose all the capital invested on the market within a week or even within a day without a good money management system. Therefore, financial market prediction is used to reduce the risk involved in trading. However, financial market prediction is a vast area. Hence the research is narrowed down to foreign exchange market which is known to be the most unpredictable market of them all.

Foreign exchange market is the world's largest financial market with average daily turnover of 5.3 trillion dollars in year 2013 [1]. FOREX market allows to trade major currency pairs in worldwide. Normally currency pairs consist with two currencies, e.g., EUR/USD pair consists of Euro currency value over US Dollar. Other major currency pairs are USD/JPY, GBP/CAD, GBP/USD, AUD/USD, USD/CHF, USD/CAD and GBP/EUR. Besides that there are some minor currency pairs which are GBP/CHF, GBP/JPY, GBP/NZD, GBP/ZAR, CAD/CHF, CAD/JPY, CHF/JPY and USD/ZAR. Most importantly, large firms and also individuals can trade on the FOREX market. Therefore FOREX market became the largest financial market in the world. Since the money transfer within the FOREX market is very high, profit that can be generated within the FOREX market is also a higher value. However high risk can cause all the money to lose if not traded correctly .Numbers of new forecasting models, approaches have been developed for decades to reduce the risk involved in the FOREX trading so that the significant profit can be generated. [2]

1.2 Prediction Approaches

The foreign exchange forecasting has drawn much interest of academic and economic communities for past two decades because of high profit making opportunities. The interest of financial predictions originated within the economic communities to better anticipate future and thus reduce the risks in markets or to make profits. For instance, accurate FOREX market forecasting can benefit with a huge profit if it correctly forecasts the future price movements. Stock markets, large multinational firms, medium and small firms and individual investors are highly depend on the forecasting methods.

Many fundamental and technical analysis applications have been used for financial forecasting. Fundamental analysis is mainly conducted with microeconomic and macroeconomics factors. Rate of economic growth, employment interest rate, inflation rate, consumer spending, and other economic factors have significant impact on exchange rates. However those analyses cannot be used to gain large profits because forecast can only be done for long periods of time with low risk involved.

Forecasts depend on past data can be grouped into technical analysis. Therefore statistical models and computer intelligence approaches can be categorized under technical analysis. Those methods include Logistic models, Factor analysis, Regression analysis, Discriminate analysis, Decision trees, Artificial neural networks , Fuzzy logic , Support vector machines and combination of those models. Furthermore those computational models have shown to be promising approaches to rely on. Results have shown that most linear models were unable to forecast with higher accuracy [3] [4] [5] [6] [7]. Recent researches on nonlinear models have shown positive sign of predicting foreign exchange market with greater accuracy.

1.3 Motivation

Even though recent researches on nonlinear methods have shown that FOREX market can be forecasted up to some extent, high volatility and uncertainty makes it very difficult to make a profit from foreign exchange trading. There are highly correlated economical social and even psychological factors which are affected for movements of the exchange rates. According to efficient market hypothesis [8], current asset market price fully reflects its all available information. That means price changes are occurring due to new information and past currency prices do not have an impact on next price. If the market goes according to the efficient market hypothesis, then every effort of predicting future market prices from past data will fail. Efficient market hypothesis usually test with random work model. Even though according to efficient market hypothesis it is impossible to predict market prices, many researchers have developed linear and nonlinear time series models for foreign exchange forecasting [2].

Therefore, this research aims to develop a new prediction system that can give more accurate prediction compared to existing models. Suitable forecasting model was expected to be selected by using empirical and mathematical evidence. EUR/USD and USD/JPY currency pairs are chosen to forecast with newly developed prediction model. Recently automated trading on market has gained a higher attention because of the trader emotions on trading often lead to loses in FOREX market. Finally it is expected to use predictions of the model to make profitable trades with an automated trading system. Among the variety of available forecasting methods, nonlinear data analysis mainly based on neural network is reviewed.

1.4 Overview

Chapter 1: Introduction

Chapter 1 is an introduction to financial markets. Market overview, prediction and research problem is presented in the chapter 1.

Chapter 2: Current Approaches for Financial Forecasting

Chapter 2 describes the current literature for nonlinear forecasting methods and data preparing and model designing. Data preparing includes previous research on input selection and pre-processing. Under model design subsection, learning algorithms, existing architectures and performance measuring are presented.

Chapter 3: Development of Forecasting Model

This chapter discusses the detailed description of development of the model. Noise removal, selected learning algorithms and input space clustering are discussed in this chapter.

Chapter 4: Real Time Implementation and Issues

Real-time implementation and issues found in the existing models have been discussed in chapter 4.

Chapter 5: Proposed Hybrid Model

Chapter 5 presents the proposed forecasting model.

Chapter 6: Model Testing

Testing of the model with Metatrader back tester is discussed in the chapter 6.

Chapter 7: simulations and results

Chapter 7 includes the simulation and results.

Chapter 8: conclusion

Chapter 8 presents the conclusions and future direction of the research.

2. CURRENT APPROACHES FOR FINANCIAL FORECASTING

The literature of current approaches for financial forecasting has been discussed in this chapter. Especially FOREX market based researches are considered. FOREX market allows buying or selling of major currency pairs such as EUR/USD, USD/JPY, EUR/AUD, EUR/JPY etc. EUR/USD currency pair is the mostly traded pair [9] and therefore it is the most volatile and most profitable pair if traded correctly. FOREX market opens within week days and closed on Saturday and Sunday [10]. All currency pairs can be traded on FOREX market continuously within working days. There is no central location for market. It is a combination of banks, organizations, traders all around the world. Since the market is very large and trading amount is very high, single trader or even a large company cannot perform a significant effect on the price movement and manipulation of the market. Therefore there is a high probability to forecast future movements depending on hidden patterns in past exchange prices with a good prediction model.

Even though lot of prediction models are available, FOREX market predictability is still a challenging task for researchers. It is found that opposing views exist within the researchers regarding the market predictability [11] [8] [12] [13] [14]. Some of them have shown that the market prices depend on the new information and therefore every attempt to forecast the market will fail. Random walk hypothesis that was first proposed by Louis Bachelier in 1900 [15] stated that past stock values has no real value in predicting future values because market response to the information that are random in nature.

In [12], the empirical tests on random work hypothesis have been carried out and concluded that "data seem to present consistent and strong support for the random walk hypothesis. This implies, that chart reading, though perhaps an interesting pastime, is of no real value to the stock market investor." However [16] has shown that the results of [12] suggests that there is no linear dependencies of past data. In references [17] [18] [19] have shown that nonlinear dependencies do exist on the past data and therefore market can be predicted up to some extent.

Recently many researchers have shown that FOREX and other financial markets can be predicted with nonlinear models that are entirely depending on past data [20]. Therefore most of the literature are based on promising nonlinear analyses. Despite with theoretical explanations, automated trading championship conducted by the MetaQuotes Software Corp [21] is a strong practical evidence for FOREX market predictability. Each year various algorithms compete with each other to make a better profitable model. The winner is chosen by competing auto trading models. Those models anticipate the future market movements and trade based on those movements. The results of the competitions can be summarized as in Table 2.1.

Table 2.1-Winners of the automated trading championship (2007-2011)

Year	Winner	Application	Inputs	Time period	Total profit after 3 months
2007	OlexandrTopchylo, Ukraine	Neural networks	Moving averages	N/A	$120 475.45
2008	KirilKartuniv ,Bulgaria	Decision trees	volatility adjusted oscillator	daily	$159 584.64
2010	Russian developer Boris Odintsov	Statistical market analysis, modeling and optimization.	short-term trend based method on moving average	(EUR/USD, 5 minutes)	$ 67 102.79
2011	Xupypr	N/A	MACD indicator - zero line crossover and signal line reversal	1hour	$113135.18

Net profit after three months have shown that models with proper algorithms can trade profitably on FOREX market. However the competition only continues for three months, robustness of the models for long time period trading cannot be proved. Almost all of the winners of the previous years have tried to win again in next championship by improving previous wining model. Even though they have won the championship once, none of them have claimed the first place twice. Therefore those algorithms are not yet capable of making consistent profit through years. Market is always evolving into more complex patterns and therefore the existing models are unable to make predictions leading to consistent profit.

This study focused on technical analysis based forecast that depends entirely on the historical data which can be used to make consistent profit through years. Building a forecasting model can be divided into two main stages as 1) Data preparing and 2) Model design. Literature on each of those sections are discussed in following sections.

2.1 Data Preparation

Data preparation includes input selection and input data preprocessing. Data preparation is very important for an accurate forecast. Past exchange data is inheritably noisy and therefore it is hard to distinguish the pattern hidden among the data without proper data preparation procedure. Section 2.1.1 compares the available input selection methods and procedure for proper input selection. Section 2.1.2 compares the literature of available data preprocessing methods.

2.1.1 Input selection

The selection of inputs for forecasting model directly affect to its performance. In the field of financial market forecasting, most of the researchers have used time delay inputs and various technical indicators (moving average, stochastic oscillator etc.) as inputs [2].Time delay inputs can be Close prices, Open prices, High prices, Low prices, Median prices, Typical prices and Weighted close price. There are large numbers of technical indicators and they got their own strengths over others. Vastly used on the literature are Moving averages (simple, exponential, weighted), Stochastic oscillator, Bollinger bands, historical volatility, Moving average convergence divergence, Relative strength index etc [14] [4] [22] [23] [24] [25] [26]. Those indicators can point out some hidden patterns and turning points of trend which cannot be seen at once. Individual trader or trading model can use those signals to trade in financial markets. However, none of those analyses can operate continuously. If the market is moving according to a pattern for a while, then those methods can be profitable. Otherwise it is difficult to gain a profit. Most reliable automated trading algorithm ever came to publicity was made by Olexandr Topchylo a Ukrainian scientist as shown in Table 2.1. The system was based on moving averages combined with a neural network. It was submitted to worldwide automated trading championship held on 2007 and claimed the first place with constant profit gain. Most importantly, the algorithm only used past price values to make a forecast and trade [27].

Researchers have used more than one technical indicator for their forecast claiming that combining indicators remove weaknesses of individual indicators [3] [28] [5] . However, use of technical indicators as much as possible does not necessarily lead to better forecasting results. Most often, it weakens the forecast due to curse of dimensionality [29]. Hidden relations among inputs can lead to the bias effect of the network [29]. Therefore, inputs should be selected carefully. Selected inputs should not be correlated with other inputs and should have maximum correlation with itself to achieve better forecast. There is no general method to select input parameters for the model. In [13], four inputs have been used to forecast Nikkei 225 opening cash index and seven technical indicators have been used to forecast TAIEX closing cash index. In [14], five technical indicators have been used while [3] uses 53 financial indices. Furthermore, [30] used 75 technical indicators. However, most researchers focused on finding optimal input variables among the available inputs. Rescaled range analysis [31] [4] [32] and Lyapunov exponent [33] are used as a measure of time series predictability. Forecastable time span is selected using those analyses. Trial and error, stepwise-regression [34] auto-regression testing [35] [36] [37] genetic algorithm based selection techniques [38] [5] are used to find most influential input variables by removing redundant ones. About 45% articles that are reviewed in the literature have used technical indicators while 35% used time lagged inputs. Others used fundamental indicators. Details of the selected inputs are presented in Table 2.2.

Table 2.2 - Types of inputs used for the forecasting model

Article	Input	Indica-tor
[13]	Four inputs(TAIEX index futures prices and technical indicators)	•
[3]	53 financial indices	•
[30]	75 technical indicators	•
[31]	Time delay inputs, indicators	✓
[33]	Reconstructed phase space variables, normalized daily closing price,	⊙
[5]	23 technical indicators	•
[39]	5 financial indices	
[40]	Technical indicators	•
[41]	Daily opening, closing, highest and lowest price and daily trade volume	✓
[42]	Current fundamental asset price, strike price and time-to-maturity	✸
[28]	27 technical indicators	•
[14] [4] [22] [23] [24] [25] [26]	4 to10 technical indicators	•
[38] [43] [44] [34] [45] [46] [47]	10 to 20 technical indicators	•
[35] [36] [37] [48] [49] [50] [6] [51] [52] [53] [54] [7] [32]	Time lagged price values	✓

- • Technical Indicators
- ✓ Time lagged inputs
- ✸ Fundamental indicators
- ⊙ Other

2.1.2 Input data preprocessing

Data preprocessing is essential for better performance, especially for financial time series forecasting. Financial time series has highly noisy, nonlinear and nonstationary characteristics which reduce the overall network performance. Almost all recently published articles have used advanced data preprocessing techniques.

K-means basic clustering algorithm was used in [30] to analyze NYSE and NASDAQ market indexes while [34] used dynamic clustering algorithm along with particle swarm optimization and recursive least-squares. Self-organizing maps (SOM) is an unsupervised leaning method which is used to map higher dimensional input space in to two dimensional spaces. Among SOM architectures, Kohonen network model was the most widely used model based on Hebbian learning principle. It has the capability of presenting unorganized data into an organized pattern. Then individual forecasting models were developed for each SOM groups [24] [25] [46] [48] [53]. Researchers have concluded that all above mentioned clustering based methods have improved the forecasting accuracy.

Apart from clustering, principal component analysis (PCA) [7] independent component analysis (ICA) [13] and wavelet transform (WT) [47] [53] methods were used to reduce the effect of noise and input correlation, and nonstationary effects. Principal component analysis is a data projection method according to variance [29]. It projects input data set into number of principal components which are linearly uncorrelated with each other and can describe the characteristics of original data set. In [7], PCA is used to remove redundant information when combining forecasts of multiple models.

ICA is quite different from PCA. Unlike PCA, ICA does not find principle components. Rather it tries to find independent sources which contained within input parameters. Original time series was broken down in to four components in [13]. Then component with highest noise ratio was removed as noise and rest is used to make a forecast and claimed that their method increased the forecasting capability. Wavelet transform (WT) is a superior alternative to Fourier transform because WT localize signal in both time and frequency domains. Therefore it is capable of removing noise and reduce the nonstationary effects of time series. WT has implemented in different ways on number of researches.

In [53] inputs are decomposed by wavelet basis into five mutually orthogonal series. Daubechies least asymmetric filters with length 8 have used for the decomposition. Finally forecast of each series summed up to get a final forecast. [47] used Haar wavelet to remove the noise of the input variables. It is done by removing small wavelet coefficients of the wavelet decomposition.

2.2 Model Design

Model design includes learning algorithm selection, architecture selection and performance measure selection. Model design is the next procedure after the data preparation is carried out. Development of a forecasting algorithm is extremely difficult as the complexity involved in the data are very high. Vast amount of models have been proposed [2] [17] [28] [14] [47] and literature of the models have presented under this section.

2.2.1 Learning algorithm

Learning algorithm is another key factor for forecasting performance. When selecting learning algorithm, [55] has shown that scaled conjugate gradient and Bayesian regularization based models show competitive results. These models forecast more accurately than that of back propagation (BP), according to [20], multilayer feed forward network with BP is most widely used network design and also it leads to satisfactory results. However, BP algorithm has some undesirable disadvantages such as slow convergence speed, higher likeliness to get trapped in local minima, initial weight determination, and selecting transfer functions. Recurrent neural network (RNN), a variant of feed forward ANN, has effectively used in some models [33] [47]. Recently, many researchers have shown that SVM is more suitable and has performed better than other algorithms [5] [39] [26] [43] [54].

2.2.2 Architectures

Generally, neural networks are fed with past data and get the future values as outputs. The network should be able to positively identify relationships and hidden patterns within inputs. If the network architecture becomes more complex it will be difficult to recognize patterns and often lead to over fit. Too many hidden neurons and higher number of training epochs can also be caused neural network to over fit. On the other hand, if the network is designed with a more simple architecture it will not be able to separate correct patterns from input data and often lead to under fitting. Recently more and many researchers are realizing that using a single neural network may lead to bias effect and combining multiple neural architectures can consistently outperform the single models [20]. There is an improving attention going on to apply hybrid models and to ensemble models for better performance [30] [32] [39] [52] [7].

2.2.3 Performance measure

The network performance is assessed by the performance measures. Mean squared error (MSE) [37] [41] normalized mean squared error (NMSE) [31] [24] root mean square error (RMSE) [28] [47] mean absolute error (MAE), directional symmetry (DS) [13] [34] and mean absolute percentage error(MAPE) [34] [42] are widely used indicators as performance measures. In addition, [47] and [48] used profit as the performance measure. Rather than using just one performance measure, combination of performance measures like error, directional statistics and profit [40] gives better understanding of the performance of the model. (See Appendix A for summery of revived articles)

3. DEVELOPMENT OF FORECASTING MODEL

Developing a suitable forecasting model for financial markets is an extremely difficult task. Even though some researchers have shown that some positive directions for modelling [26] [24] [47] , none of them were able to make consistent profits in highly unpredictable financial markets. Existing literature have shown that rather than taking one model for forecasting, hybrid and ensemble models have always outperformed single models [3] [30] [34] [48] [6].

3.1 Removal of Noise and Non-Stationary Characteristics of Input Data

Noise is the most disturbing factor for exchange values prediction [46] [47] [56] [45] [50]. Removing noise from original time series is essential for a better forecast. Otherwise, forecasting model will tries to model the noise instead of the original trend. That leads to poor forecasting results as noise is random in nature. Numerous techniques have been used for noise removal within past decades. In [57], it is suggested that the transforming the input into non correlated inputs can increase the forecasting performance. E. Bodt *et al* [11] suggested that the successive variations of the price are more suitable inputs rather than the direct price values. Input normalizing is done by many researchers in the data preparing stage. Normalizing is used to reduce the effect of noise further and improve the learning ability of the model. Among the techniques found on the literature, principal component analysis (PCA), independent component analysis (ICA), wavelet transform (WT) and combinations of those techniques have given the promising results. When selecting a specific method from above promising approaches, mathematical and empirical evidences should be considered.

PCA is a non-parametric data analysis technique which can be used to reduce the input dimension of the data set and to rearrange input into more simplified manner [58]. When applying for financial forecasting, PCA is a good tool to remove redundant information by mapping input orthogonally along with maximum variance directions. Components that contribute to higher variance can be used to describe the system. PCA is built on few assumptions that are disturbing when applying to financial data analysis. PCA assumes input data is linearly correlated and thus linear algebra has the capability to find a better representation for input data. Such assumption cannot be assumed in the financial data analysis because the data is inherently nonlinear. Kernel PCA has introduced to deal with nonlinearities and again the kernel building is a difficult problem for financial data because the data is non-stationary. Assumption on keeping high variance components and removing low variance components often removes the important data which can be used to identify the turning points of original trend. In a practical situation like financial data analysis; it is hard to define a margin for noise. Therefore assumptions of removing low variance components badly affect to the trend finding process of forecasting model [58].

ICA is often referred as blind source separation problem. Unlike PCA, ICA does not try to map the input into orthogonal vectors. ICA tries to find the underlying data structures by separating it to independent components. There are different algorithms which can be used to develop ICA. Independence of the inputs can increase performance of the learning model. However the fundamental assumption made in the ICA algorithm is independent components should be non-Gaussian for ICA to be possible. Random variables such as exchange rates do follow the Gaussian distribution in classical statistical theory. Central limit theorem has shown that sum of independent random variables shows the Gaussian distribution under certain conditions [59]. Therefore, use of ICA on exchange rates is still a doubt. Nonlinear ICA can be a good area to focus on because its capability to analyze sub-Gaussian and super-Gaussian mixtures. Unfortunately nonlinear ICA is a less developed area [60].

Apart from those techniques, Wavelet transform has emerged into a solid solution for noise removal in the area of financial forecasting. It was recently used with many forecasting methods as noise reduction tool [47] [53]. WT is a superior alternative for Fourier transform. Unlike Fast Fourier transform (FFT), wavelet analysis can be used to analyze the complex patterns in stationary and non-stationary time series which is a great advantage over other algorithms. Time signals that are made of constant frequencies can be analyzed with Fourier transform. Those kinds of signals are often called as stationary signals. However, real world signals such as exchange rates are made of frequencies that are changing with time. Those signals are non-stationary and cannot be analyzed with Fourier transform. Wavelets can analyze such signal by cutting it up to different frequency components and analyze the component by matching the resolution to its scale. Therefore it gives the information of the frequencies and its time of the occurrence. This gives the better understanding of the signal than any of the previously discuss methods. Therefore unlike, FFT, discrete wavelet transform (DWT) allows to remove the unwanted noise components at specific time frames. In DWT function $f[n]$can be approximated as follows.

$$f[n] = \frac{1}{\sqrt{M}} \sum_k W_\emptyset[j_0, k] \emptyset_{0,k}[n] + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_k W_\psi[j, k] \psi_{j,k}[n] \quad \text{......................} (3.1)$$

Where

$$\text{Scaling function } \emptyset_{j,k}(t) = 2^{j/2} \emptyset(2^j t - k) \quad \text{......................................} (3.2)$$

$$\text{Wavelet function } \psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad \text{......................................} (3.3)$$

j is the scaling parameter and k is the shift parameter. Wavelet coefficients can be calculated by taking inner product with scaling and wavelet functions as follows.

$$W_\emptyset[j_0, k] = \frac{1}{\sqrt{M}} \sum_n f[n] \emptyset_{j_0,k}[n] \quad \text{...} (3.4)$$

$$W_\psi[j, k] = \frac{1}{\sqrt{M}} \sum_n f[n]\, \psi_{j,k}[n] \ , \quad j \geq j_0 \quad \text{...} (3.5)$$

Here,

$\frac{1}{\sqrt{M}}$ is the normalization factor.

More importantly, it is not necessary to find a perfect wavelet to get better understanding of the signal. Only need is to avoid redundant information which can be done with discrete wavelet transform. Furthermore, stationary wavelet transform (SWT) makes sure no information loss and allows perfect reconstruction. Many researchers have shown that

wavelet decomposition and wavelet denoising can be effectively used in financial time series forecasting for non-stationary effect removal and noise removal [47] [53].
Wavelet multi resolution analysis allows decomposing the signal into scales with different time and frequency resolution. It gives a better time resolution and a poor frequency resolution at high frequencies and a better frequency resolution and a poor time resolution at low frequencies. This method can be used to remove noise of the signal at each sub level of decomposition by removing low value wavelet coefficients in their finer scales. However, the field is still at an exploratory stage and quantitative results are not available. Therefore, analyses are based on heuristic approaches. Finding proper wavelet to denoise the financial time series is done by trial and error approach.

There are few wavelet families and each of them has their strengths and weaknesses. Haar wavelet, Debauchee's wavelets and Coiflet wavelet families have taken for the comparison of denoising. Furthermore SWT and DWT are compared for the better denoising capability.

Threshold selection is the most important part in the wavelet denoising. Used thresholding method is a universal thresholding that can remove Gaussian random noise effectively [56]. It uses a fixed threshold values that are separately calculated for each decomposition level. Threshold value λ for each level has calculated as given in (6).

$$\lambda_j = \sigma_j \sqrt{2 * \log N_j} \quad \text{...} \; (3.6)$$

Here λ_j is the length of the coefficient vector at level j and σ_j is the standard deviation of the noise. That can be calculated as follows.

$$\sigma = \frac{MAD}{0.6745}$$

Here MAD is mean absolute deviation.
Calculated threshold value is used with soft and hard thresholding for both SWT and DWT. Soft thresholding changes the low level wavelet coefficients to the threshold value

and do the reconstruction. Hard thresholding sets all the wavelet coefficients that are lower than the threshold to zero and do the reconstruction.

Debauchees and Coiflet wavelets have been used for solving broad range of problems including noise removal and edge detection [61] [62] [63]. Debauchee's wavelet and Coiflet wavelets have been used for comparison of noise removal. Noise removal using Debauchee's 1, 2, 6 and 10 wavelets and Coiflet 1, 2 and 5 have been considered. Selected Debauchee's and Coiflet wavelets have shown in Figure 3.1 to Figure 3.7.

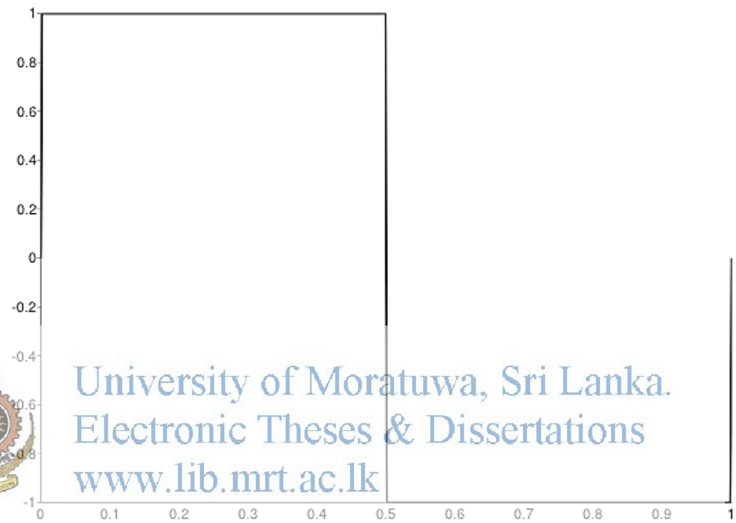

Figure 3.1- Debauchees 1 wavelet

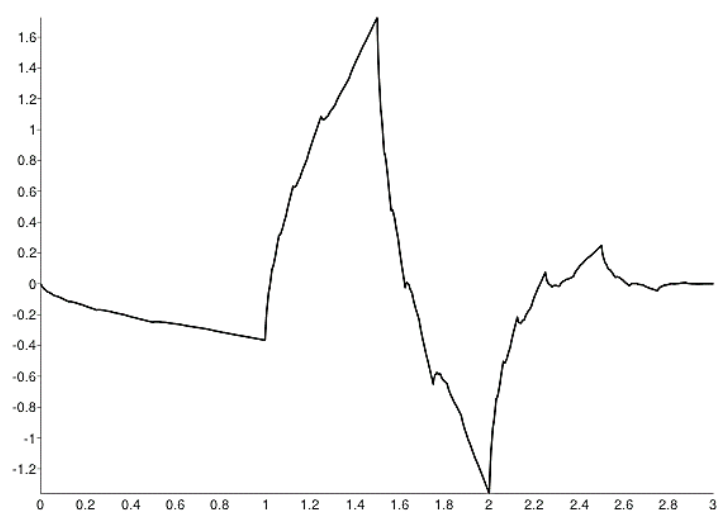

Figure 3.2- Debauchees 2 wavelet

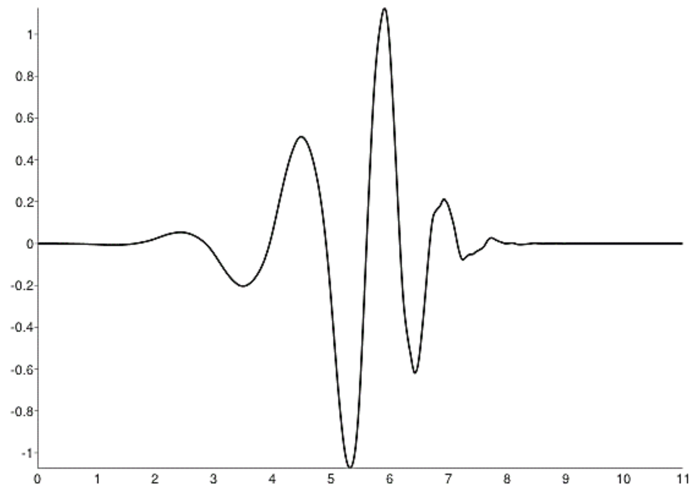

Figure 3.3-Debauchees 6 wavelet

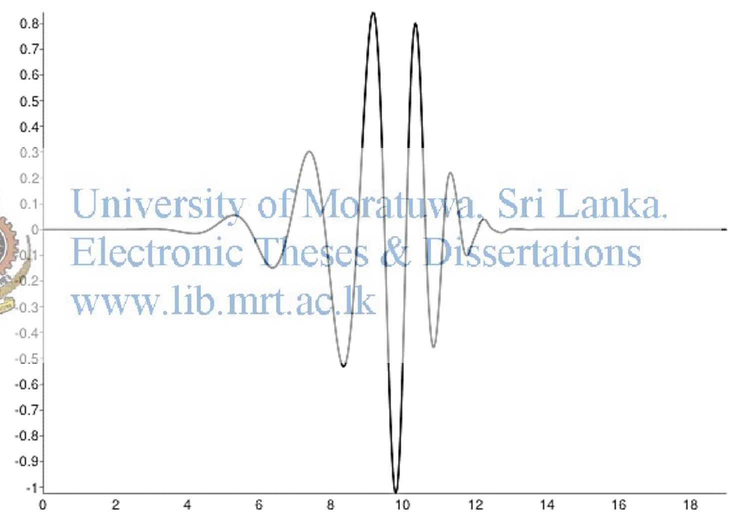

Figure 3.4- Debauchees 10 wavelet

Figure 3.8 and Figure 3.9 shows the Debauchee's wavelets for EUR/USD denoising with level 2 wavelet decomposition. *Db1* has identical response to Haar wavelet and the noise removal is also the same with *Db1* and *Haar*. Therefore, noise removal for *Haar* wavelet has not been considered. According to the Figure 3.8 and Figure 3.9, Debauchee's wavelet shows higher capability to remove the noise. Smoother time series and more noise removal can be achieved with higher levels of Debauchee's wavelets in DWT. However the use of higher Debauchee's wavelet for denoising is limited with time lag involved within. Time lag reduces the capability to identify the correct time to buy or sell. In SWT, noise removal is more consistent with increasing the Debauchee's wavelet level. Low levels of Debauchee's wavelets in DWT with hard threshold can capture the sharp peaks of the time series. Results show the property is lost with SWT for both soft and hard thresholding.

Selected Coiflets wavelets are shown in Figure 3.5 to Figure 3.7.

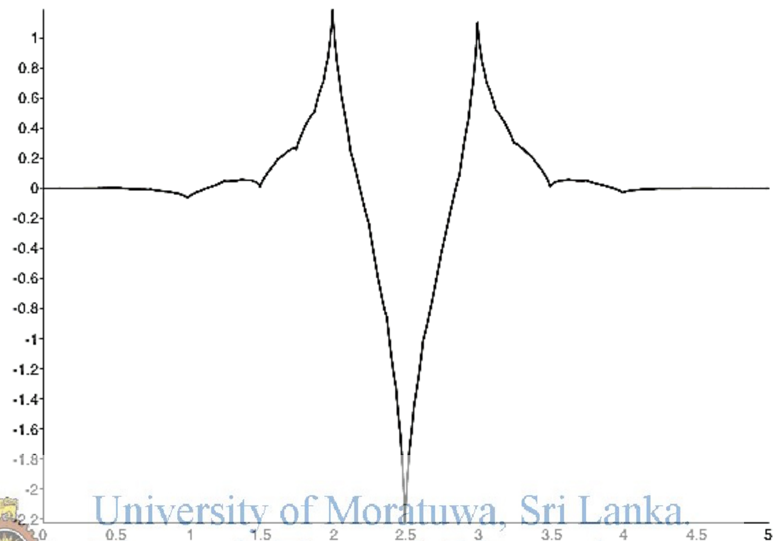

Figure 3.5- Coiflet 1 wavelet

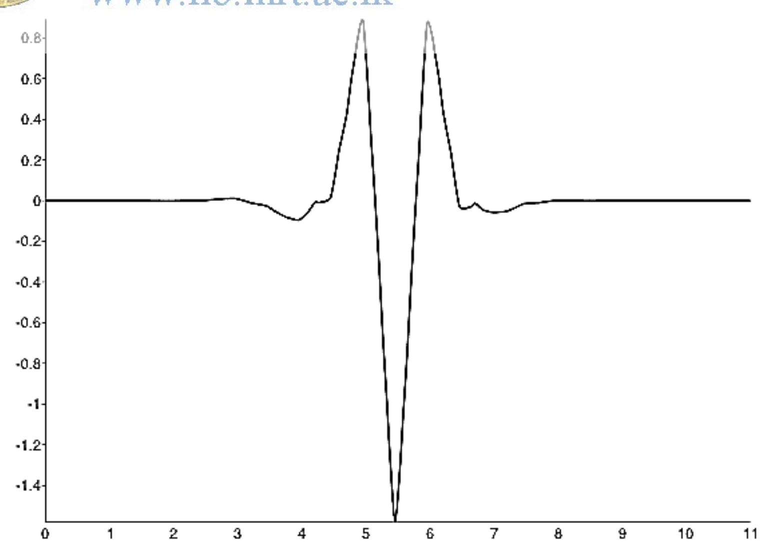

Figure 3.6- Coiflet 2 wavelet

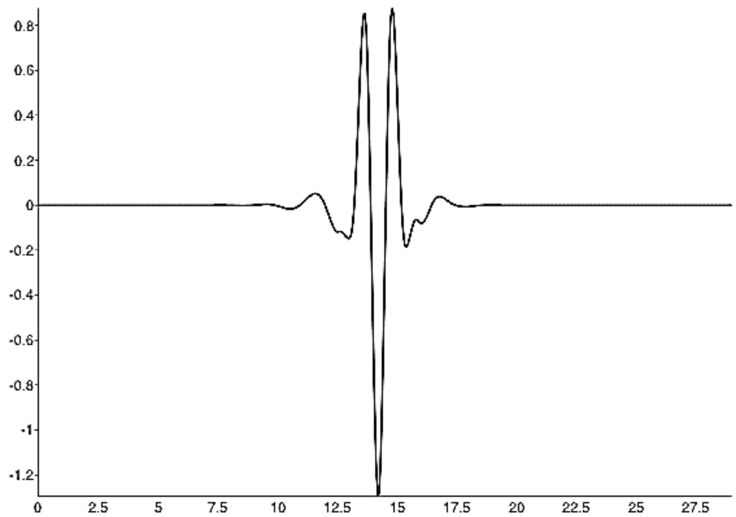

Figure 3.7-Coiflet 5 wavelet

Results of the Coiflet wavelet analysis for denoising capabilities are shown in Figure 3.10 and Figure 3.11. Coiflet 1, 2 and 5 have been used for the comparison. DWT hard thresholding with Coiflet1 is shown the better edge detection capability than the Haar and Debauchee's wavelets. Unfortunately, level 1 Coiflet wavelet does not remove the noise effectively. However, higher levels of Coiflet wavelets denoising capabilities are similar with higher levels of Debauchee's wavelets.

Comparison of the three kinds of wavelets gave the idea that low level wavelets are better for understanding the turning points of the EUR/USD exchange rates and high level wavelets are capable of creating a better denoised time series. Trading in real times needs denoised data those are not lagging behind in order to make the forecast which leads to profit. Therefore, intermediate wavelet is better for reducing time lag incorporated.

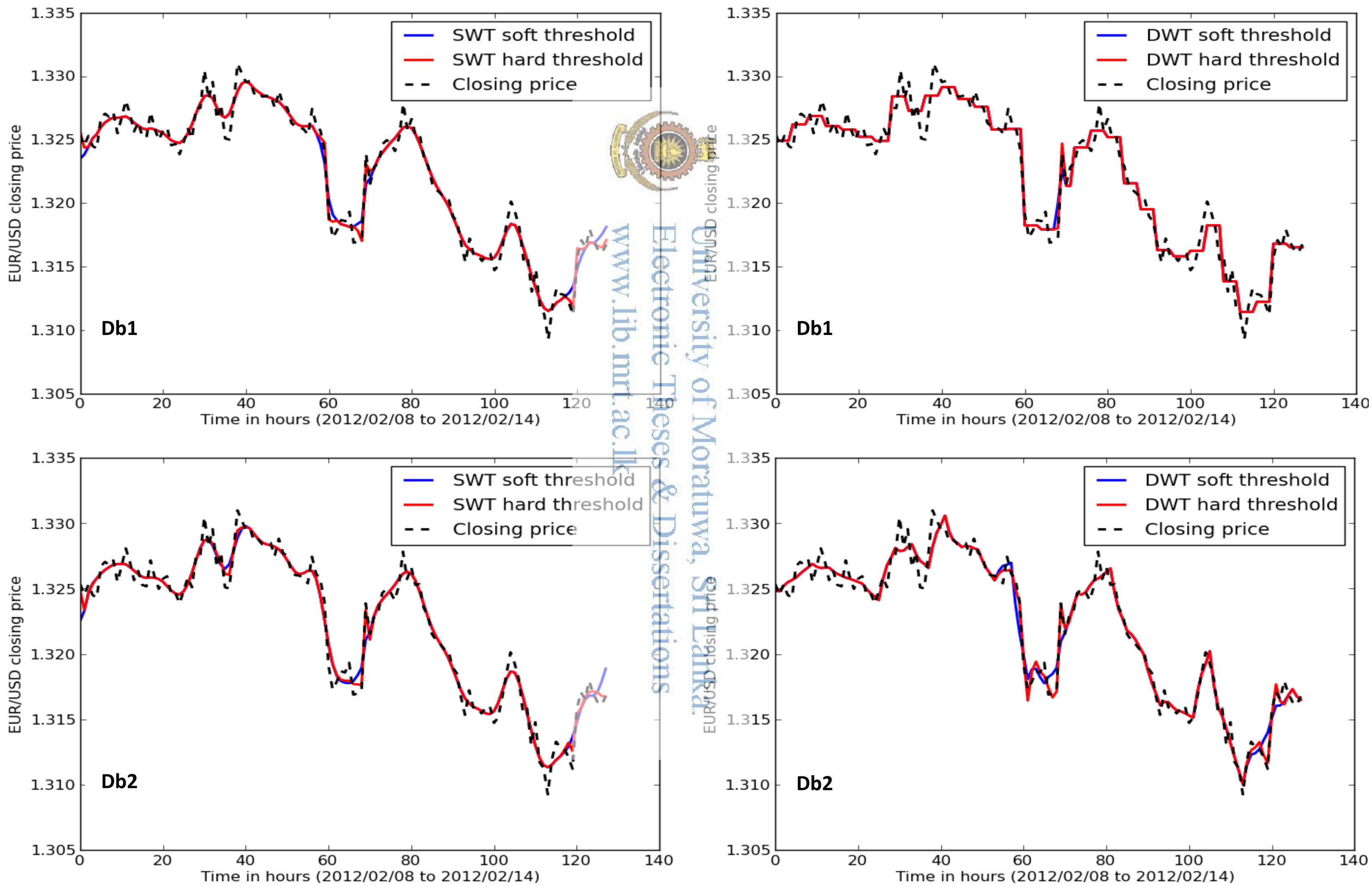

Figure 3.8 - Debauchee's wavelet transform for EUR/USD. Db1 and Db2 wavelets combined with SWT and DWT denoising is shown in the figure. Soft and had thresholding have been used for both SWT and DWT for comparison.

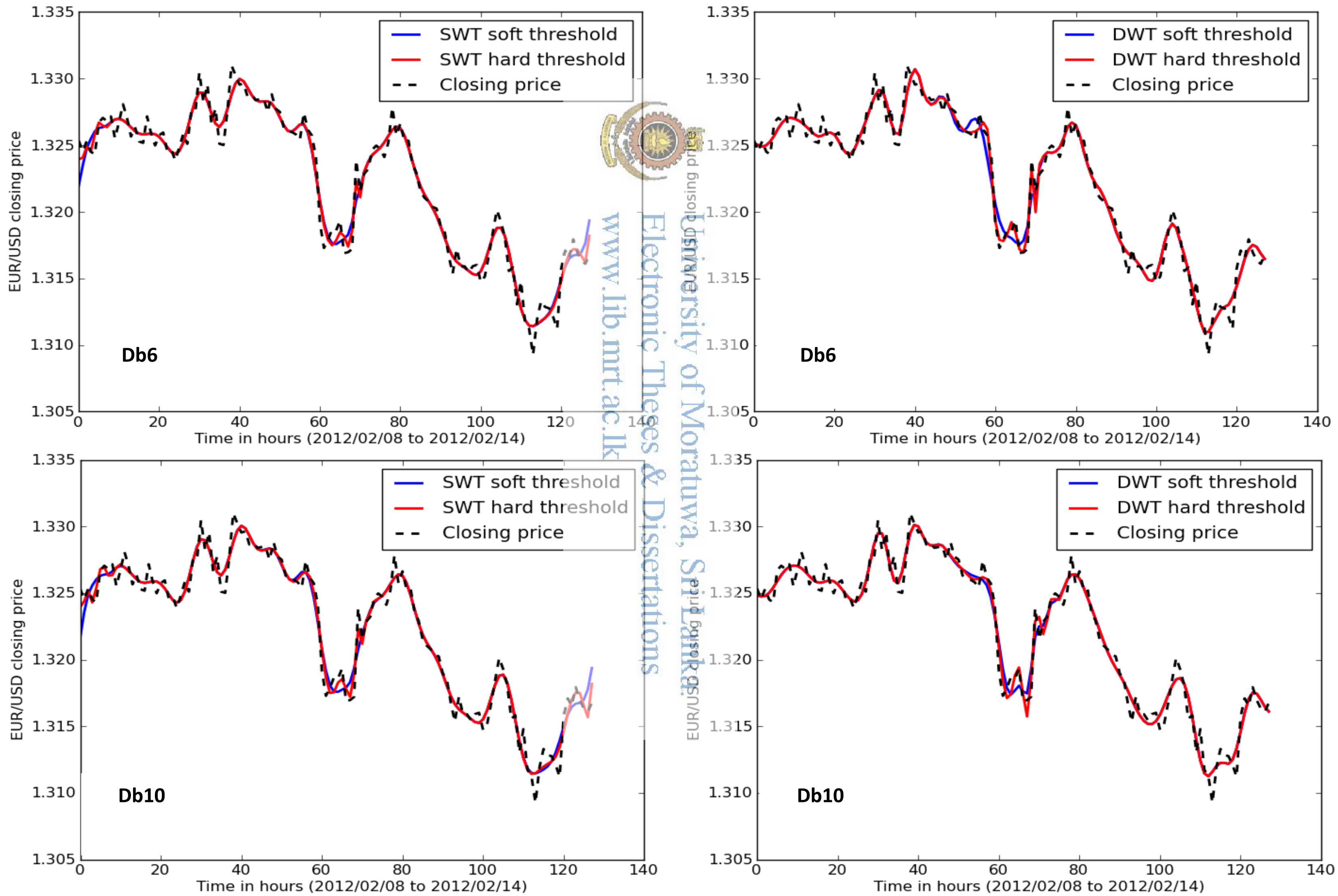

Figure 3.9 - Debauchee’s wavelet transform for EUR/USD. Db6 and Db10 wavelets combined with SWT and DWT denoising is shown in the figure. Soft and had thresholding have been used for both SWT and DWT for comparison.

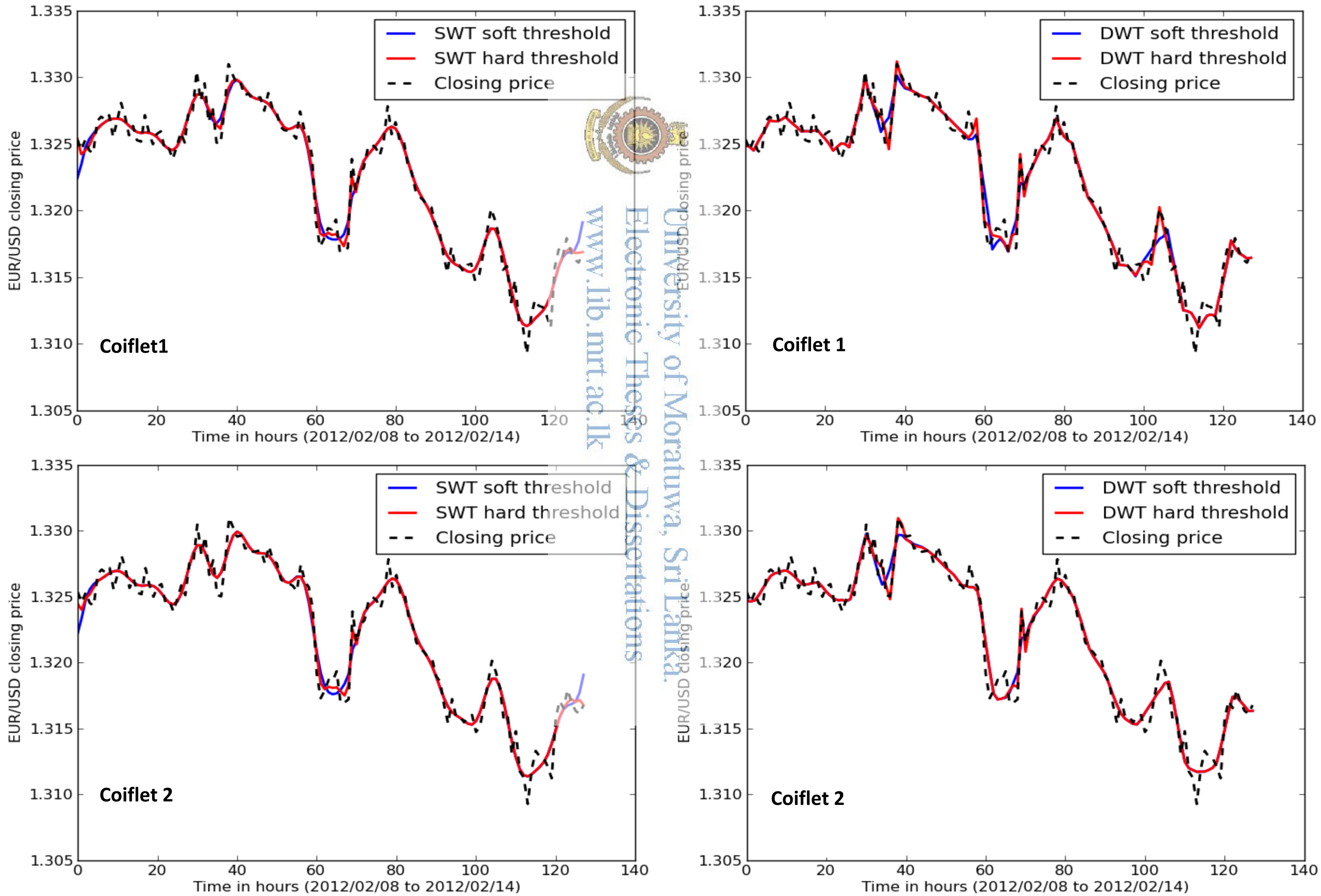

Figure 3.10 - Coiflet wavelet transform for EUR/USD. Coiflet 1 and coiflet 2 wavelets combined with SWT and DWT de-noising is shown in the figure. Soft and had thresholding have been used for both SWT and DWT for comparison.

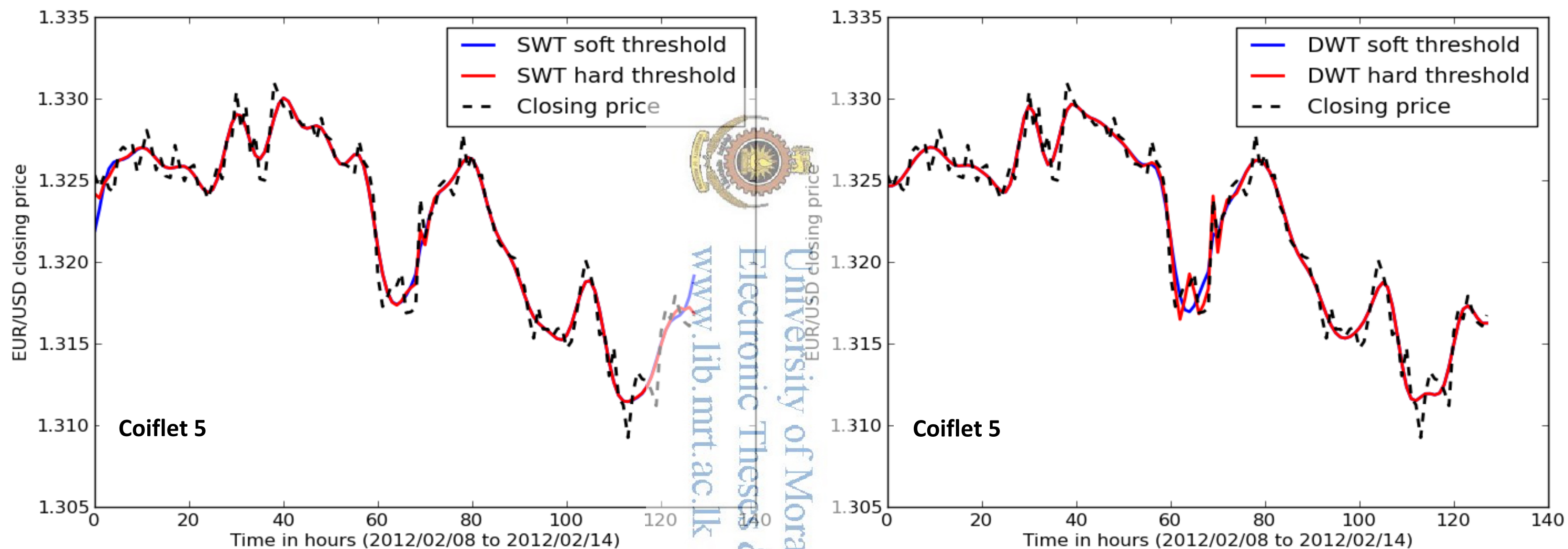

Figure 3.11 - Coiflet wavelet transform for EUR/USD. Coiflet 5 wavelets combined with SWT and DWT denoising is shown in the figure. Soft and had thresholding have been used for both SWT and DWT for comparison.

3.1.1 Wavelet based denoising issue

Denoising accuracy of the time series has been discussed in this section. Reason for the accuracy check is because when trading in a real system, price forecast is based on de-noised values of the edge of the time frame. Therefore, de noised values at the edge should give the actual time direction. This problem is further discussed in section 4.1.

Figure 3.12 and Figure 3.13 shows Haar based stationary wavelet transform and discrete wavelet transform based denoising compared with soft and hard thresholding for accuracy at the edge of the time series. Figure 3.14 and Figure 3.15 shows the comparison of de-noising performance of the Coiflet3 wavelet. Figure 3.16 and Figure 3.17 shows the comparison of denoising performance of the Debauchee's 6 wavelet.

Even though many suggest that stationary wavelet transform has better denoising capability compared with discrete wavelet transform, results of the Table 3.1 have shown that discrete wavelet transform with soft thresholding have the better accuracy for EUR/USD denoising. Detailed comparison has been done for Debauchee's wavelet family and coiflet wavelet family. Details of the comparison are shown at Table 3.1 Performance of the comparison is evaluated by taking mean squared error (MSE) and directional accuracy between the denoised edge and the real trend direction. Mean error and directional accuracy (D %) is calculated for 400 hourly data points. MSE is calculated as in (3.7). Finally the Debauchee's wavelet has been selected for further testing from the results of Table 3.1.

$$MSE = \frac{1}{10}\sum_{j=1}^{400}\sum_{i=1}^{10}(x(j)_{(128-i)} - d_{128-i+j})^2 \quad \text{.....................................(3.7)}$$

Here, d is the actual trend denoised with the wavelet transform over entire set and $x(j)$ is the denoised series of 128 data. $x(j)$ is shifting with increment of j

Directional statistics are calculated as given in (3.8)

$$D\% = \frac{1}{400} * (\frac{correct}{correct + wrong})100\% \quad \text{..(3.8)}$$

Here $Correct$ and $Wrong$ are calculated as flows. At the initial conditions, both $Correct$ and $Wrong$ are equal to zero.

$$\sum_{j=1}^{400}\sum_{i=1}^{10}(x(j)_{128-i} - x(j)_{128-(i+1)})(d_{128+j-i} - d_{128+j-(i+1)})\begin{cases} if \geq 0 \; correct + 1 \\ if < 0 \;\; wrong + 1 \end{cases} \quad \text{….(3.9)}$$

Table 3.1- Error comparison with level 2 decomposition

Error		**DB1**	**DB2**	**DB3**	**DB4**	**DB5**	**DB6**	**DB7**	**DB8**	**DB9**	**DB10**
SWT soft	MSE	55.4	90.1	107.7	143.1	165.5	130.0	231.7	253.6	264.2	236.57
	D%	85	81	78	77	75	75	75	74	74	73
DWT soft	**MSE**	**2.7**	**11.51**	**12.1**	**11.7**	**7.8**	**15.2**	**9.6**	**10.3**	**12.3**	**10.8**
	D%	**99**	**94**	**94**	**92**	**91**	**92**	**93**	**90**	**91**	**91**
SWT hard	MSE	14.3	21.1	37.1	48.1	64.2	61.1	88.3	102.4	116.9	101.3
	D%	92	88	84	81	79	79	78	75	73	73
DWT hard	MSE	36.1	30.7	27.9	28.0	20.6	36.1	27.4	27.5	18.9	26.8
	D%	99	94	92	91	91	89	91	88	90	90

Error		**Coiflet1**	**Coiflet2**	**Coiflet3**	**Coiflet4**	**Coiflet5**
SWT soft	MSE	111.8	153.3	190.8	216.6	235
	D%	81	76	74	73	73
DWT soft	MSE	**9.33**	16.5	14.3	18.8	17.8
	D%	**92**	88	90	89	89
SWT hard	MSE	45.4	55.5	77.6	96.3	106.8
	D%	83	82	79	77	75
DWT hard	MSE	26.6	31.9	29	31.4	30.0
	D%	90	87	89	87	88

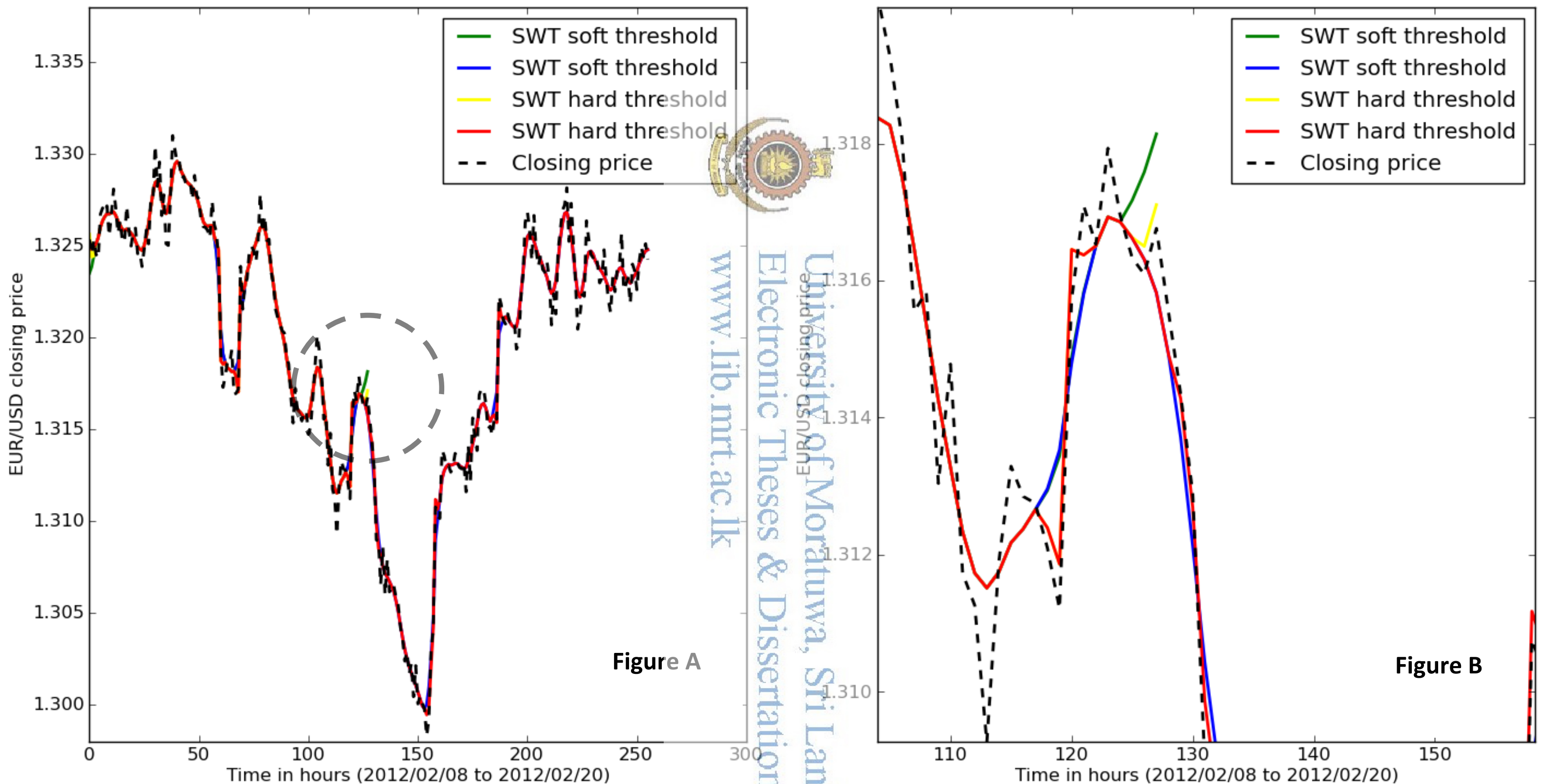

Figure 3.12- Comparison of the accuracy of Haar wavelet in SWT denoising. Figure 3.12 (A) shows the denoised results of two timeframes superimposed. Figure 3.12 (B) zoomed into selected area of Figure 3.12 (A). Yellow and Green lines of Figure 3.12 (B) shows the wrong trend direction generated from the Haar based SWT soft and hard thresholding.

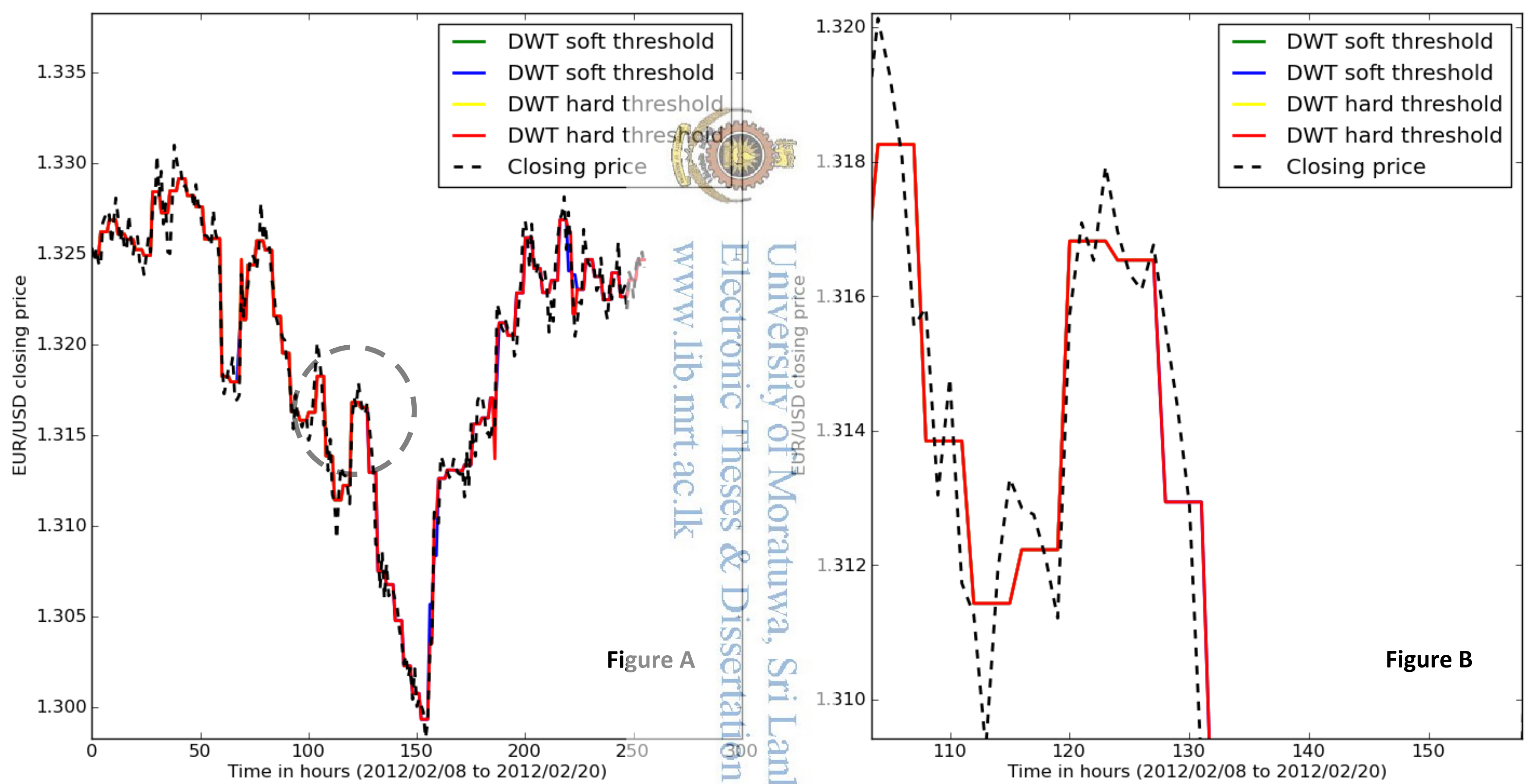

Figure 3.13 - Comparison of the accuracy of Haar wavelet in DWT denoising. Figure 3.13 (A) shows the denoised results of two timeframes superimposed. Figure 3.13 (B) zoomed into selected area of Figure 3.13 (A). All the lines of Figure 3.13 (B) super imposed with one another indicate that the noise removal at the edge for Haar based denoising is accurate.

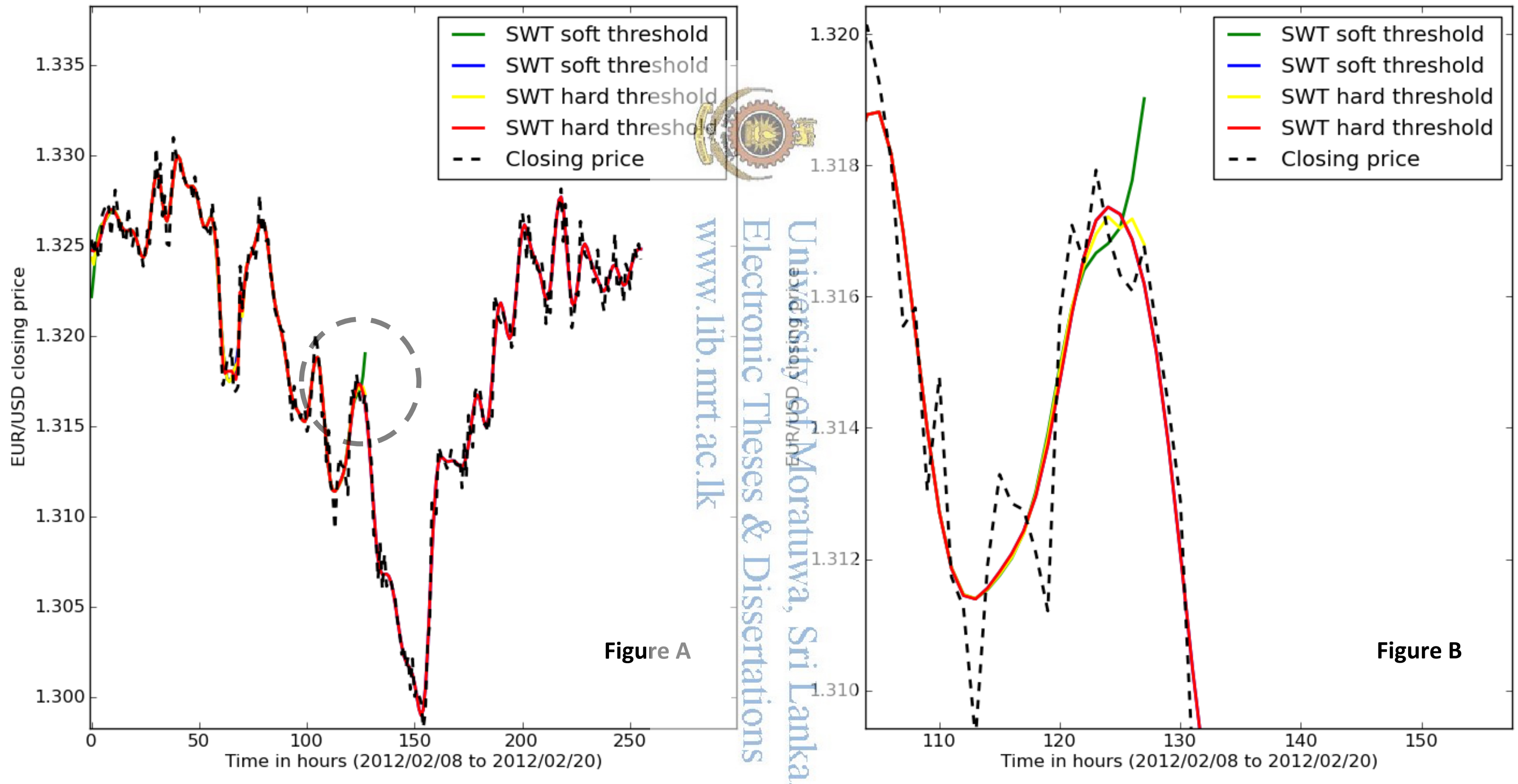

Figure 3.14 Comparison of the accuracy of Coiflet 3 in SWT denoising Figure 3.14 (A) shows the denoised results of two timeframes superimposed. Figure 3.14 (B) zoomed into selected area of Figure 3.14 (A). Yellow and Green lines of Figure 3.14 (B) shows the wrong trend direction generated from the Coiflet 3 based SWT soft and hard thresholding

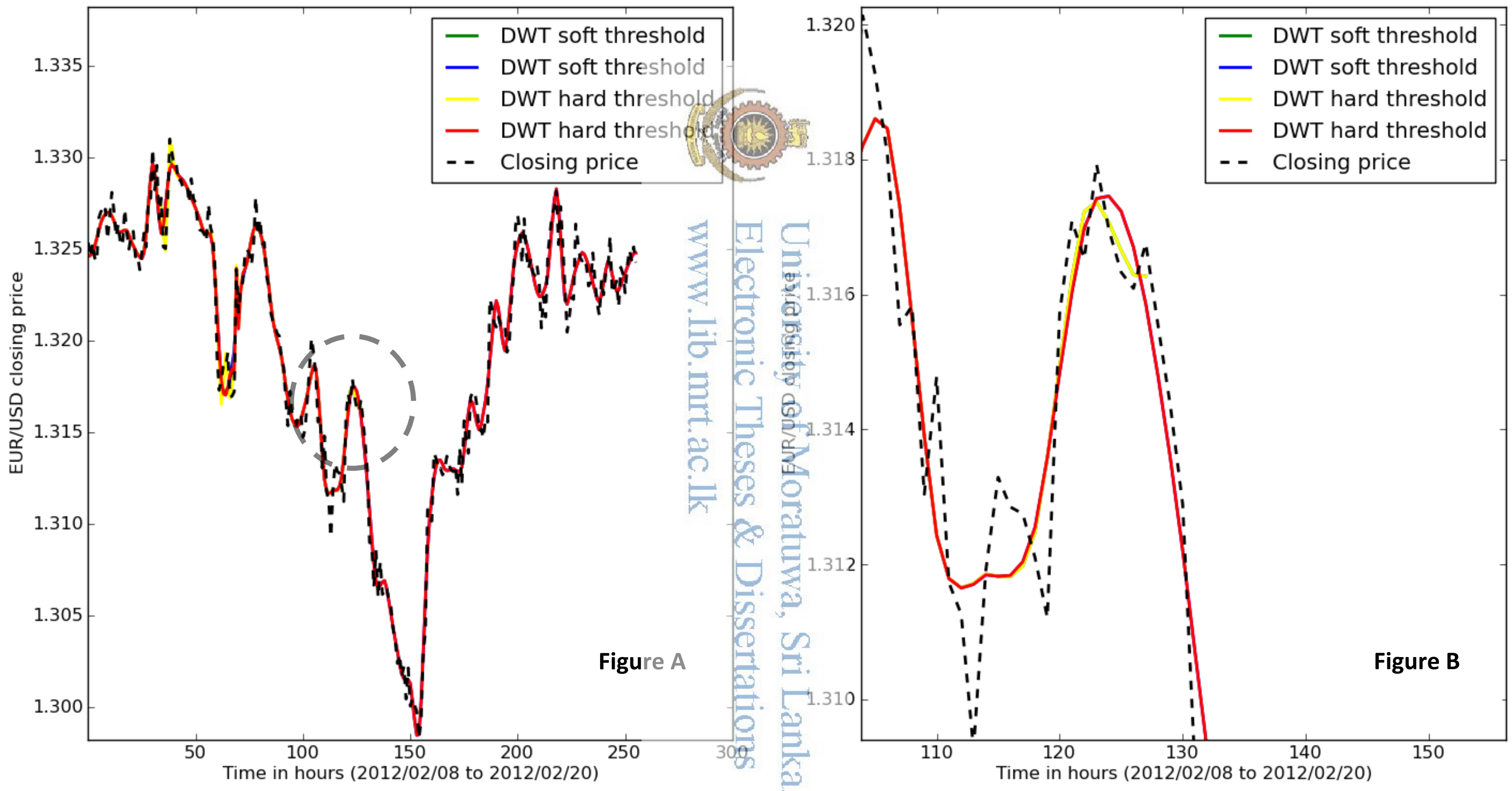

Figure 3.15- Comparison of the accuracy of Coiflet3 in DWT denoising. Figure 3.15 (A) shows the denoised results of two timeframes superimposed. Figure 3.15 (B) zoomed into selected area of Figure 3.15 (A). Yellow and Green lines of Figure 3.15 (B) shows the correct trend direction generated from the Coiflet3 based SWT soft and hard throsholding. Accuracy is less than Haar based DWT denoising

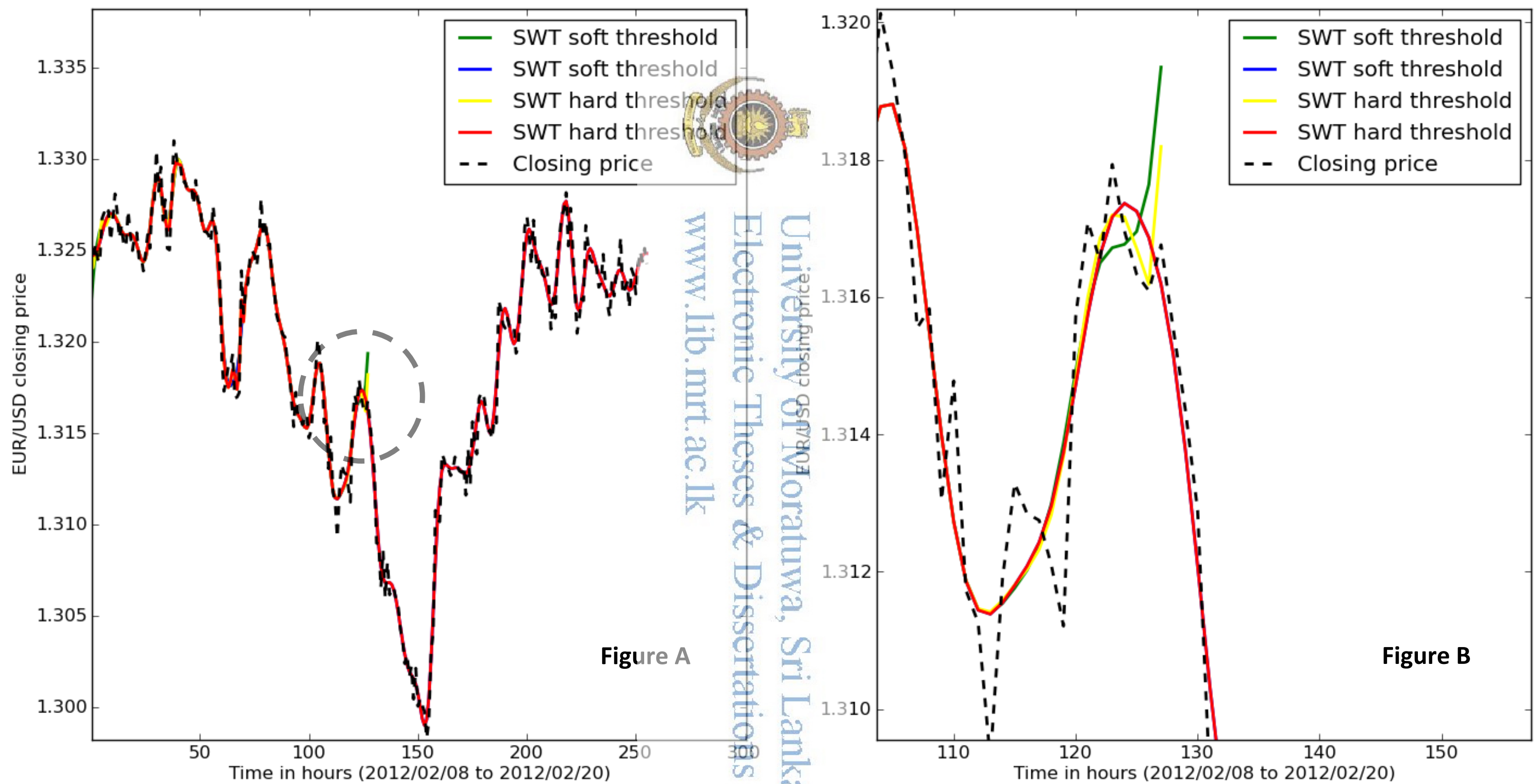

Figure 3.16- Comparison of the accuracy of Db6 wavelet in SWT denoising. Figure 3.16 (A) shows the denoised results of two timeframes superimposed. Figure 3.16 (B) zoomed into selected area of Figure 3.16 (A). Yellow and Green lines of Figure 3.16 (B) shows the wrong trend direction generated from the Db6 based SWT soft and hard throsholding

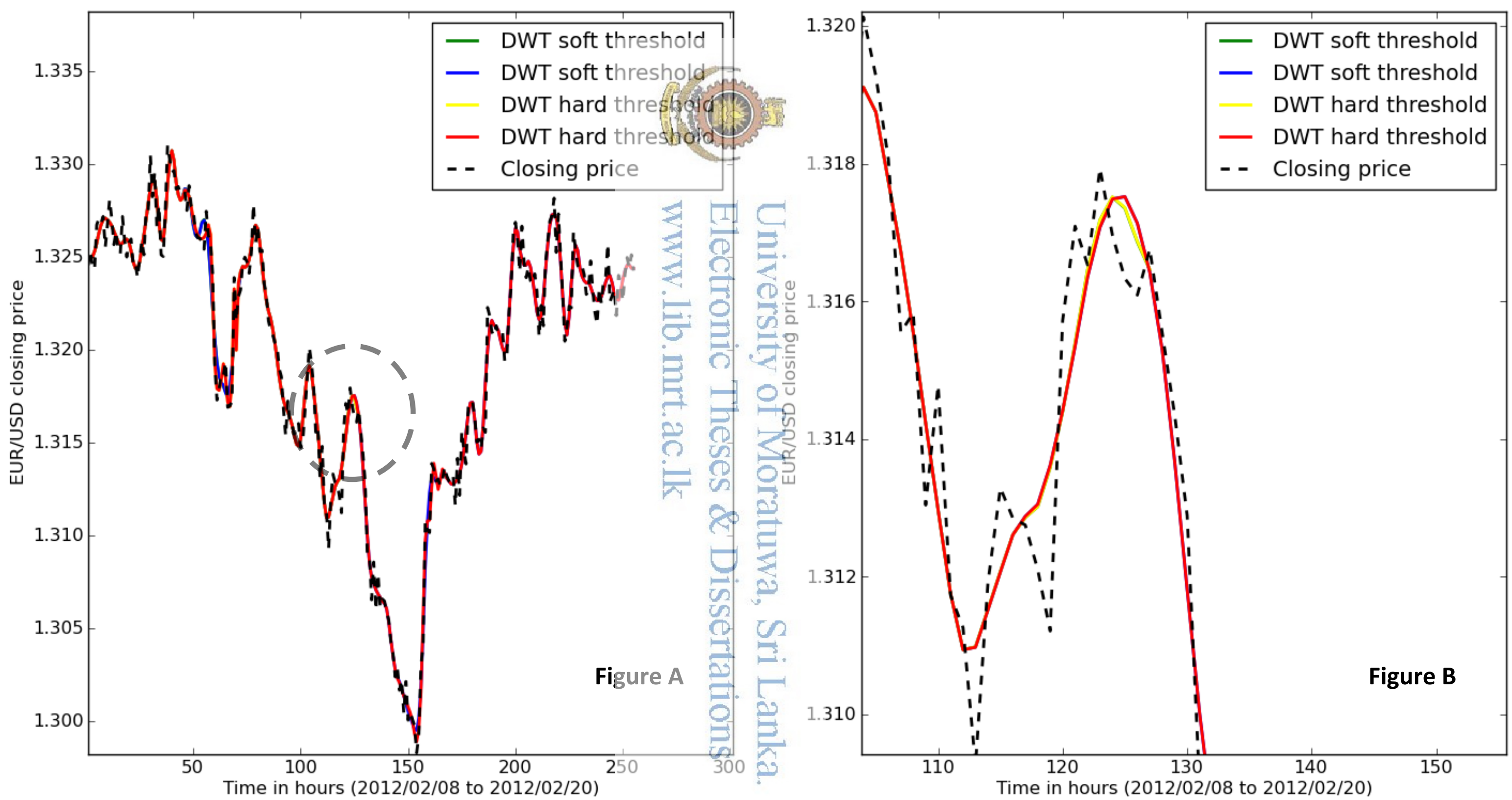

Figure 3.17- Comparison of the accuracy of Db6 wavelet in DWT denoising. Figure 3.17 (A) shows the denoised results of two timeframes superimposed. Figure 3.17 (B) zoomed into selected area of Figure 3.17 (A). Yellow and Green lines of Figure 3.17 (B) shows the correct trend direction generated from the Db6 based SWT soft and hard throsholding which is more accurate

In order to compare the different levels of wavelet denoising performance on EUR/USD exchange rates, noise removal of selected Debauchee's wavelets have been used with level 1 to 5 decompositions. Results are shown in Table 3.2 and fixed decomposition level was chosen to be level 2 according to the comparison.

Table 3.2- Debauchee's denoising for DWT with soft thresholding (MSE- Mean Squared error, D% - Directional Statistics)

Level		Db1	Db2	Db3	Db4	Db5	DB6	Db7	Db8	Db9	Db10
1	MSE	0.4	1.14	1.41	1.4	2.1	2.8	3.4	3.6	3.7	3.6
	D%	99	98	96	95	95	94	95	94	95	94
2	MSE	2.7	11.5	12.1	11.7	7.8	15.2	9.6	10.3	12.3	10.8
	D%	99	94	94	92	91	92	93	90	91	91
3	MSE	7.9	62.3	47.9	51.8	43.4	50.0	61.0	45.7	61.4	54.5
	D%	99	94	85	90	88	91	88	89	88	88
4	MSE	56.1	196	245.9	165.1	333.1	211.7	169.6	204.3	188.1	157.8
	D%	98	94	85	82	85	84	86	87	84	85
5	MSE	322.9	460.1	1300.8	730.0	348.3	948.6	885.8	403.0	641.9	1126.3
	D%	98	94	85	80	79	81	82	79.6	80	85

It can be seen that noise on the input data does affect the wavelet denoising performance. Accuracy of the noise removal of the edge reduces with the higher levels of wavelets because higher levels try to achieve more smoothing. If the lower level wavelets are taken, learning model cannot be able to forecast accurately because the noise left in the model is affect the performance. This problem is reduced by taking out some amount of noise from the original time series by averaging. There are few moving averages those try to reduce this effect by implanting new averaging methods. Simple moving average, weighted moving average, exponential moving average, hull moving average and triple exponential moving average are popular averaging methods. Among various kinds of available moving average indicators, Triple exponential moving average (TEMA) is taken. It is a combination of exponential, double exponential and triple exponential moving averages. Calculation of the exponential moving average is given in equation (3.10).

$$\mathrm{TEMA} = 3\mathrm{xEMA} - 3\mathrm{xEMA(EMA)} + \mathrm{EMA(EMA(EMA))} \quad \ldots\ldots\ldots\ldots(3.10)$$

Here EMA is exponential moving average. The EMA for a series Y can be calculated recursively

For $t > 1$,

$$S_t = \alpha . Y_t + (1 - \alpha) . S_{t-1} \quad \text{……………..…………………...............................(3.11)}$$

Here, $S_1 = Y_1$

The coefficient α represents the degree of weighting decrease

Y_t is the value at a time period t.

S_t is the value of the EMA at any time period t.

Main reason for TEMA selection is that it does not introduce lag for considerable averaging period as compared with other moving averaging methods. TEMA with period 5, 7, 10, 14 and 21 is used with Debauchee's family wavelets as shown in Table 3.3. Here the comparison was done for Debauchee's wavelets according to the denoising performance comparison of the Table 3.2. Accuracy of the noise removal of the edge is increased with the increasing of TEMA period. Therefore, higher level of TEMA can be used along with wavelet denoising. Contritely lag of the TEMA increases with the averaging period. Then again the final forecast of the model will be affected with this lag and performance will be reduced. Unfortunately this dilemma cannot be removed due to uncertainty associated in the time series. Experimental results have shown that it cannot be guaranteed that TEMA based wavelet denoising method performs better than regular wavelet based denoising even though its accuracy and mean squared error is less. Therefore both methods are taken for model testing. Discrete wavelet transform based Db1, Db3 and Db10 are selected from the trial and error results of Table 3.3.

3.2 Learning Algorithm Selection

According to Cover's theorem [64] "A complex pattern-classification problem cast in a high-dimensional space non-linearly is more likely to be linearly separable than in low dimensional space". If the input space is transformed into higher dimensional space according to Cover's theorem, it is easy to linearly separate those patterns and it is advisable to choose higher number of hidden neurons than number of input nodes. Higher number of hidden nodes than the input nodes has been found in [35] [36] [37] [40] [41] and [22] while [23] claimed best neural model as 9-3-1 architecture. Radial basis functions (RBF) [44] and SVM have more principal basis for construction and often create their architecture according to Cover's theorem. In the case of support vector machine it has capability to map the input into much higher (infinite with RBF kernel) dimensional space and thus have higher probability to identify hidden patterns according to Cover's theorem. Basically, the support vector machine is linear kernel based learning machine which eliminates certain shortcomings of vastly used neural network models. Main idea of the support vector machine is to construct a hyper plane that maximizes the separation between positive and negative examples. Figure 3.18 shows the optimal hyper-plane for set of data points

Table 3.3- Error with levels of triple exponential moving average

TEMA period		Db1	Db2	Db3	Db4	Db5	DB6	Db7	Db8	Db9	Db10
Without TEMA	MSE	**2.7**	11.51	**12.1**	11.7	7.8	15.2	**9.6**	10.3	12.3	10.8
	D%	**99**	94	**94**	92	91	92	**93**	90	91	91
5	MSE	2.0	11.1	6.6	9.7	6.4	9.1	8.2	8.9	10.0	9.7
	D%	99	95	96	92	92	93	93	91	92	93
7	MSE	**1.5**	10.14	**4.41**	8.4	4.8	6.2	**6.6**	6.8	7.4	7.5
	D%	**99**	96	**97**	93	93	94	**94**	93	93	93
10	MSE	1.4	8.4	2.7	6.8	3.4	3.7	4.7	4.6	5	5
	D%	99	96	97	93	94	96	95	94	94	94
14	MSE	1.2	6.4	1.7	5.5	2.1	2.3	3.2	3.1	3.1	3.1
	D%	99	97	97	93	95	96	95	94	95	95
21	MSE	0.65	4.3	1.1	3.8	1.2	1.3	2.0	1.8	1.8	1.7
	D%	99	96	98	94	95	96	95	95	96	95

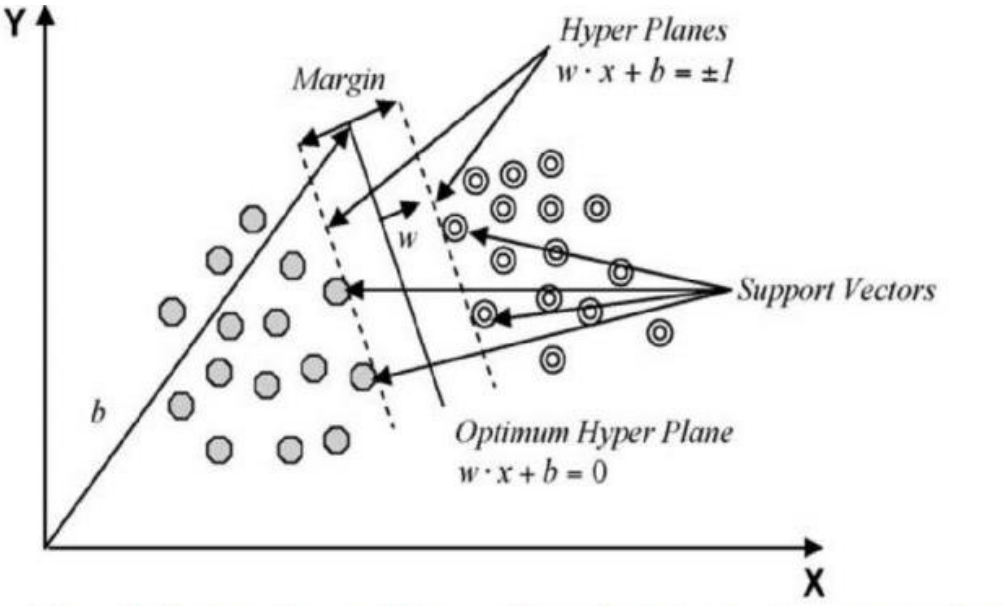

Figure 3.18- Optimal hyper-plane for set of data points

Training sample $\{(x_i, d_i)\}_{i=1}^{N}$, where x_i is the input vector of the *i*th example and d_i is the output for a given x_i. Assume the response d_i belongs to two classes that are linearly separable; the optimal hyper plane for linearly separable patterns can be formed as follows. Here w_0^T and b_0 are optimal weight vector and bias.

$$w_0^T x + b_0 = 0 \quad \text{.....................................(3.12)}$$

Optimal hyper plane calculation is a quadratic optimization problem. It is subjected to constraints $d_i(w^T x_i + b) \geq 1$ for all *i* and the weight vector minimizes the cost function $\phi(w) = \frac{1}{2} w^T w$. This constrained optimization problem is called as Primal problem and it can be stated as follows for linearly separable patterns.

$$J(w, b, \alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i \left[d_i(w^T x_i + b) - 1\right] \quad \text{................................ (3.13)}$$

Auxiliary non negative variable α_i is called Lagrange multipliers. Solutions for primal problem can be given as,

$$w = \sum_{i=1}^{N} \alpha_i d_i x_i \quad \text{.....................................(3.14)}$$

and

$$\sum_{i=1}^{N} \alpha_i d_i = 0 \quad \text{..(3.15)}$$

According to duality theorem, dual problem for support vector machine, given the training sample $\{(x_i, d_i)\}_{i=1}^{N}$, find the Lagrange multipliers $\{\alpha_i\}_{i=1}^{N}$ that maximize the objective function

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{J=1}^{N} \alpha_i \alpha_j d_i d_j K(x_i, x_j) \quad \text{..................................(3.16)}$$

$K(x_i, x_j)$ is the inner product kernel defined as $K(x, x_i) = \varphi^T(x)\varphi(x_i)$. $\varphi(x_i)$ is the nonlinear transformation of the input space to higher dimensional feature space.

Subject to constraints:

1. $\sum_{i=1}^{N} \alpha_i d_i = 0$
2. $0 \leq \alpha_i \leq C$ where C is a user specified positive parameter

Lagrange multipliers that are found by solving the dual problem can be used to determine the optimal weight vector W_0. By changing the (3.14) which is used to find the optimal hyper plane for linearly separable patterns, W_0 can be estimated as follows.

$$w_0 = \sum_{i=1}^{N} \alpha_0 d_i \varphi(x_i) \quad \text{..(3.17)}$$

User specified positive parameter C is determined with the accuracy of the forecast. For the case of separable patterns, SVM produces zero training error and reduce the structural risk. Accordingly, SVM can give a better generalization capability because it does not depend upon the problem domain knowledge and it is unique for support vector machine. Nonlinear data mapping is done with radial basis kernel for regression analysis of time series

Structural risk is often measured by using generalization error. Generalization error should be kept in minimum possible value for best performance. Therefore, reduction of upper bound of the generalization error is essential for every forecasting model. The Vapnik–Chervonenkis (VC) dimension, first proposed by [65], is the representation of maximum number of examples that can be learned by a learning network without an error. VC dimension plays major role in finding upper bound of generalization error. They have defined the guaranteed risk (upper bound of the generalization error) V_g as in (3.18), where $\in_0\ (N, h, \alpha)$ is the confidence interval and it depends on training sample size N, VC-dimension h, and probability α.

$$\in_0\ (N, h, \alpha) = \sqrt{\frac{h}{N}\left[log\frac{2N}{h} + 1\right] - \frac{1}{N}log\ \alpha} \qquad \text{.............................(3.18)}$$

$$V_g(w) = V_t(w) + 2\in_0{}^2\ (N, h, \alpha)\left(1 + \sqrt{1 + \frac{v_w}{\in_0{}^2(N,h,\alpha)}}\right) \qquad \text{.............................(3.19)}$$

Here, training error is denoted by V_t and w represents weight vector for the fixed training set. There is a value of h, which reduces the upper bound of generalization error to its minimum possible value. According to [66], VC dimension h of a purely linear network is proportional to number of free parameters W. h of purely threshold neural network is proportional to $W\ log\ W$, while Sigmoidal activation function based feed forward neural architectures has h proportional to W^2. Going through above-mentioned proofs, upper bound of generalization error can be varied by varying number of hidden neurons. Due to the fact most articles related to feed forward neural networks have used trial and error methods to find the architecture with best generalizing capability. Results have shown significant improvement of the generalization capability [23] [52]. Unlike feed forward neural networks and RBF networks, SVM has interesting property of structural risk minimization. SVM's VC dimension is bounded from above as shown in (3.18). h can be minimized by properly selecting the optimal margin of separation ρ and it is independent of the dimensionality of the input space $m_0$and thus immune to the curse of dimensionality. Let D denote the diameter of the smallest ball containing all the input vectors then h can be written as,

$$h \leq min\left\{\left\lceil\frac{D^2}{\rho^2}\right\rceil, m_0\right\} + 1 \quad \text{...(3.20)}$$

Empirical evidences [5] [26] [43] [6] [54] along with the structural risk minimization property have shown that SVM has achieved good generalization performance than feed forward ANN and RBF networks in their benchmark performance comparisons. Significant comparison was not performed within SVM and recurrent models. However [30] claimed that ensemble recurrent model outperform single SVM.

3.3 Cluster Input Space

Generalization of a forecasting model is still a challenging problem. For a good generalization, the model should give low training error as well as low testing error. Cross validation is the most widely used generalization method. Input data set is divided into three sets called training, validation and testing. Training set is used to train the network. Validation set contains the data that is never seen by the network. Validation set is used in training along with training set to calculate out of sample error. If the output of the sample error increases for consecutive iterations, it is the stopping criteria for network training. Even though using optimized VC dimension to reduce the upper bound of generalization error as presented in Section 3.3, once VC dimension reaches its minimum point, generalization error cannot be reduced further more for a single network. This is due to well-known bias variance dilemma. Average value of the estimation error between the regression function $E[D|X = x]$ and the approximating function $F(x, w)$ evaluated over entire training set τ can be written as follows.

$$L_{av}\big(f(x), F(x,\tau)\big) = B^2(w) + V(w) \quad \text{...(3.21)}$$

$B(w)$ and $V(w)$ are defined by

$$B(w) = E_\tau[F(x,\tau)] - E[D|X = x] \quad \text{...(3.22)}$$

$$V(w) = E_\tau[(F(x,\tau) - E_\tau[F(x,\tau)])^2] \quad \text{...(3.23)}$$

Here, bias $B(w)$ is the representation of inability to approximate the regression function by given network and variance $V(w)$ is the representation of lack of information contained in the training sample. A single neural network achieves low bias result for higher variance and vice versa, unless training set is infinitely large. However, infinitely large training set causes slow convergence due to its size [67]. In a neural architecture design, bias/variance dilemma should be considered seriously. Therefore, both bias and variance should be kept in small values in order to achieve good generalization. According to principal of divide and concur, dividing a computationally complex task into computationally simple tasks and combining its output to take the solution is a better method for solving such task [68] .This method allows reduction in the variance by combining multiple over-trained models, which are trained to zero bias. It will reduce both bias and variance at once for a finite training set, thus remove the bias variance dilemma. Empirical evidence for reducing the bias variance dilemma have found in [34] [24] [25] [46] [48] [53]. In the clustering process K-means clustering [69] was utilized. K-means algorithm is a simple unsupervised learning algorithm based on Euclidean distance that can be used to cluster input into several clusters. Objective function to be minimized in K-means algorithm as follows.

$$J = \sum_{j=1}^{k}\sum_{i=1}^{n}\left\|x_i^{(j)} - c_j\right\|^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.24)$$

Where k is the number of clusters, n is the number of available data points and c_j are the cluster centers.

Inputs $x_i^{(j)}$ are assigned to the nearest cluster j to minimize to objective function. Meanwhile, Bias-variance dilemma is expected to be removed by training separate SVM model for each cluster.

3.4 Proposed Hybrid Model

Forecasting model is divided into two main sections: data preparing and model building. In the data preparing section, input selection and input data preprocessing is considered. Learning algorithm selection, structural risk minimization, generalization and bias variance dilemma are considered in the model building section. Figure 3.19 shows the flow diagram of the proposed model. Proposed model integrates wavelet transform and k-means clustering with SVM.

Most suitable technical indicators are selected in the "Technical indicators" section by trial and error method. Direct closing price values of currency pair are used in comparison with technical indicators. Wavelet transform is used to denoise the input data of the time series. All the inputs are normalized to improve the leaning ability of SVM model. It is shown at the "Remove noise and normalize section". Bias variance dilemma is removed using K-means clustering at the "Cluster input space". Two or three clusters are formed for testing the accuracy. Then the separate SVMs are trained for each cluster. System tries to find best cluster when new input is arrived and then forecast is made with corresponding SVM model.

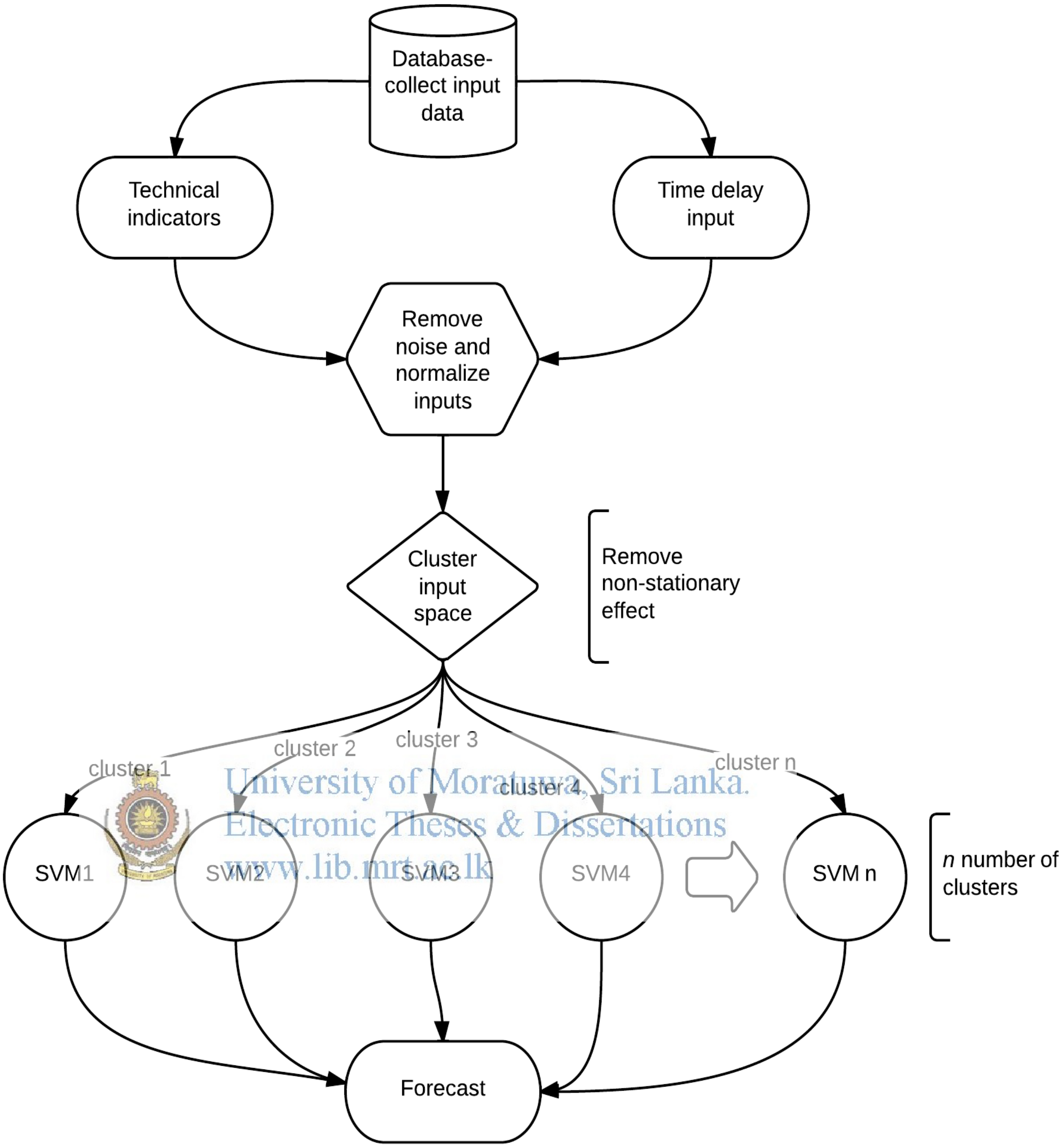

Figure 3.19 - Proposed hybrid model combining wavelet transform, K-means clustering and support vector machine

4. REAL TIME IMPLEMENTATION AND ISSUES

The major problems of previous methods have been identified in this research. The goal of the forecasting system should be to give some idea about market condition in real-time that can help traders to make a profit. Otherwise traders cannot use the forecasted data for the real-time trading. However most of the existing methods have not considered the real-time conditions in their forecasting system and therefore such systems cannot be used for trading in real financial markets. Wavelet based denoising issue, forecasting methodology and performance measure and issues on model testing are discussed in this chapter.

4.1 Wavelet Based Denoising Issue and Errors at the Edge

Almost all architectures found on the literature were tested on past data. Data set is divided into two or three sets namely training set, validation set and test set. Then the model was trained by using training set data. The validation set has been used to choose optimum system parameters. It can be done as in [70].

Initially model is trained using the training set. Then trained model is used to forecast validation set data. Mean error was calculated between actual data and the forecasted data. Then again the model was trained using training set with different parameters and again the mean error is calculated. This is repeated for different parameters for user defined times. The more it is tested; it is more likely to find better parameters. Parameters which give minimum validation error are used to forecast testing data set. If the model continues to give low testing error then it is assumed that the model forecasts in real market with better accuracy. This method is called cross validation and most researchers have used cross validation to prove the performance of the model.

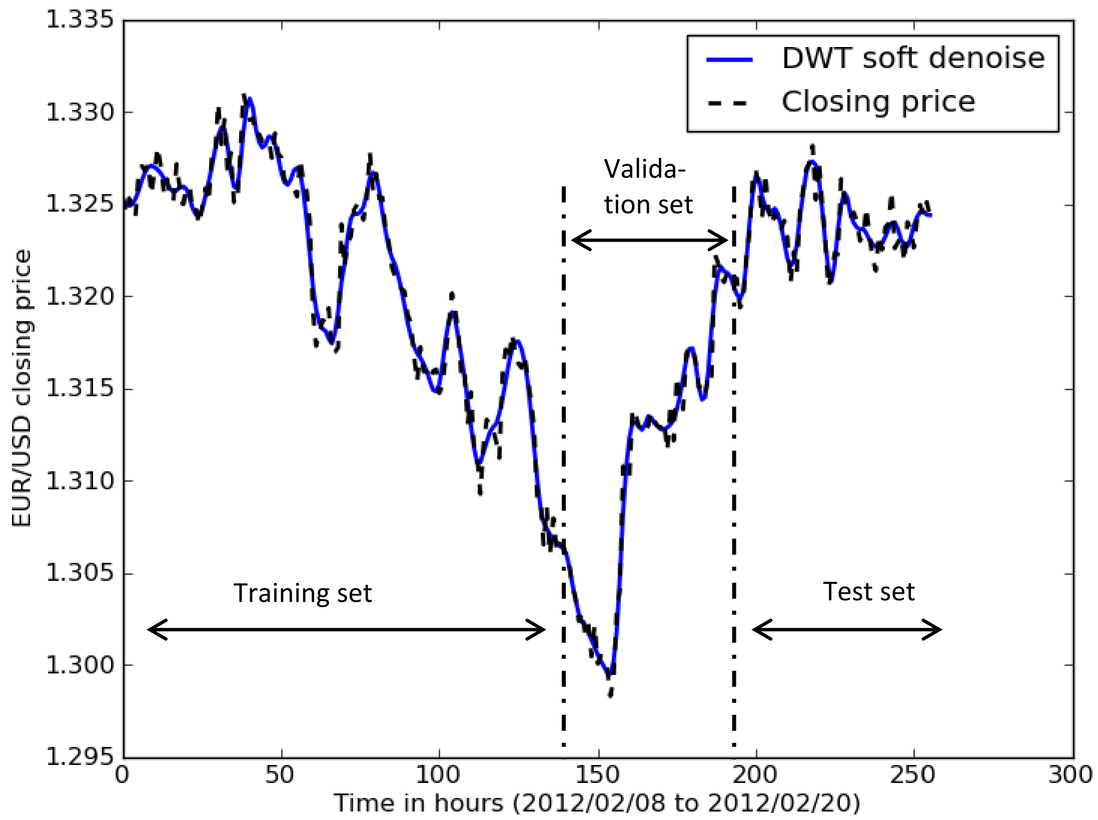

Figure 4.1- EUR/USD closing price denoised with discrete wavelet soft thresholding

However there are certain mistakes found in the literature of cross validation when it is combined to wavelet denoising. Figure 4.1 shows denoised EUR/USD time series that is denoised by Debauchee's wavelet. First, wavelet decomposition is applied to the time series and then small wavelet coefficients are removed by thresholding. Finally, inverse wavelet transform is used with remaining coefficients. The mistake done in here is the training, validation and test sets of whole time series are processed with wavelet transform and noise is removed at once. In that case, wavelet transform uses some amount of future data to find the present values of denoised time series. In a real time system, values of the edge of the denoised time series will often changes with every incoming data. This has been discussed in section 3.1. This issue with real-time data is shown in Table 4.2- Comparison of two graphs on Figure 4.5.

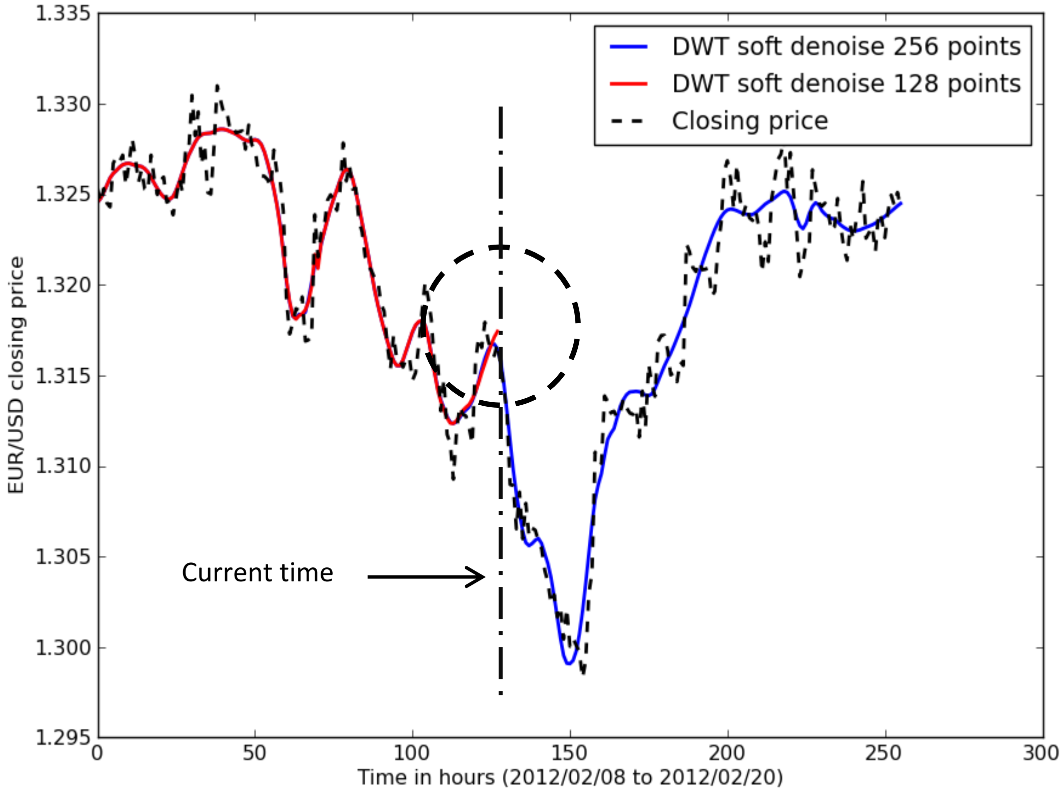

Figure 4.2- Discrete wavelet denoising with soft thresholding is done for EUR/USD time series. Original time series is denoised with 128 data and 256 data. Denoising is accurate for past data. However, denoised data is not always correct for current time frame.

Figure 4.2 shows EUR/USD past data set and recent data set superimposed with each other. It is processed with wavelet transform and noise is removed. Noise removal is done for two stages. First stage half of the time series is taken and noise is removed. At the second stage all the data has taken in to account and noise was removed. Both noise removed graphs have drawn in the same graph with original time series. If one can assume the current time frame is T=0 then available data is first half. Therefore denoised time series is DWT soft denoise 128 points. Forecast of the future values will be done by using recently denoised values. However it can be clearly seen that values at the edge of the time series are changed when arrival of new data. Most often noise at the edge of the time series does not effectively removed by wavelet transform at higher denoising levels as in Figure 4.3.

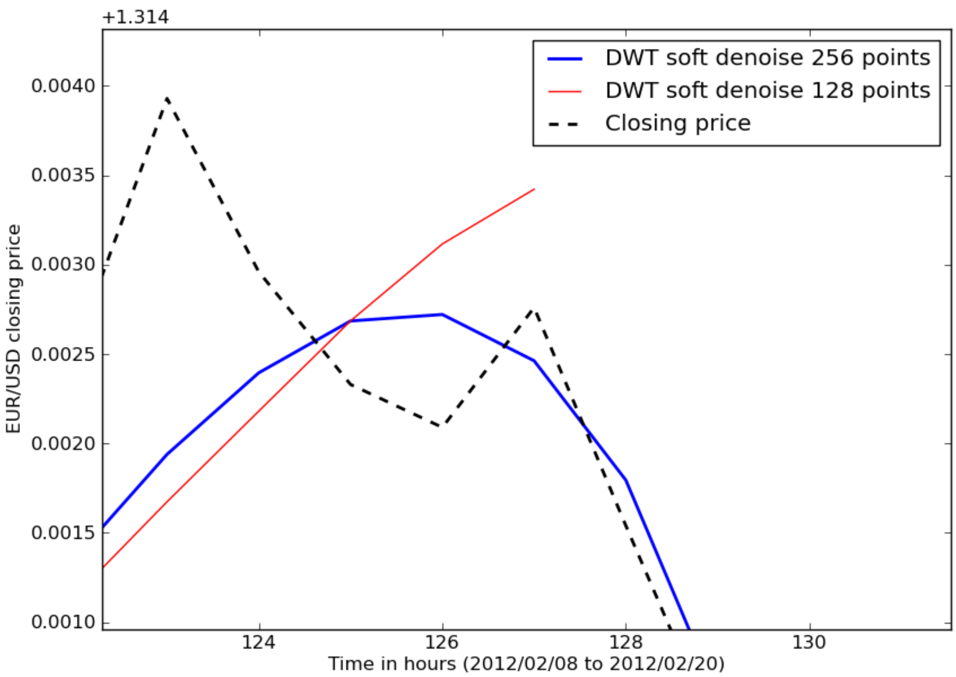

Figure 4.3- Zoomed into current time frame of Figure 4.2

In other sense wavelet denoising sometimes gives the wrong trend at the edge. Therefore, an error correction method should use along with wavelet denoising in order to effectively remove the noise at the edge of the series. In addition to that denoising of the time series should be done at every incoming price values. It is advisable to avoid conventional training testing and validation method with wavelet transform. Having that kind of noise removal method will remove the real time characteristics of the time series. In such a case, even it is performed well in the testing conditions; it will not be able to perform well in the real markets. Therefore, wavelet selection and denoising level selection should be done by properly analyzing the time series.

4.2 Forecasting Methodology and Performance Measure

Error based performance measures are the most widely used performance measures for the forecasting models [37] [41] [31] [24] [28] [47] [34] [42]. Some have proposed the combination of error measure and the directional statistics (which gives the directional accuracy of the forecast) can give more precise understanding of the model performance. Even though directional measure and error measure often give better performance, it is doubtful to use those measures as the performance measure. This is tested using MSE and D% performance measurs. Figure 4.4 shows two forecasts of two models on the same graph.

.

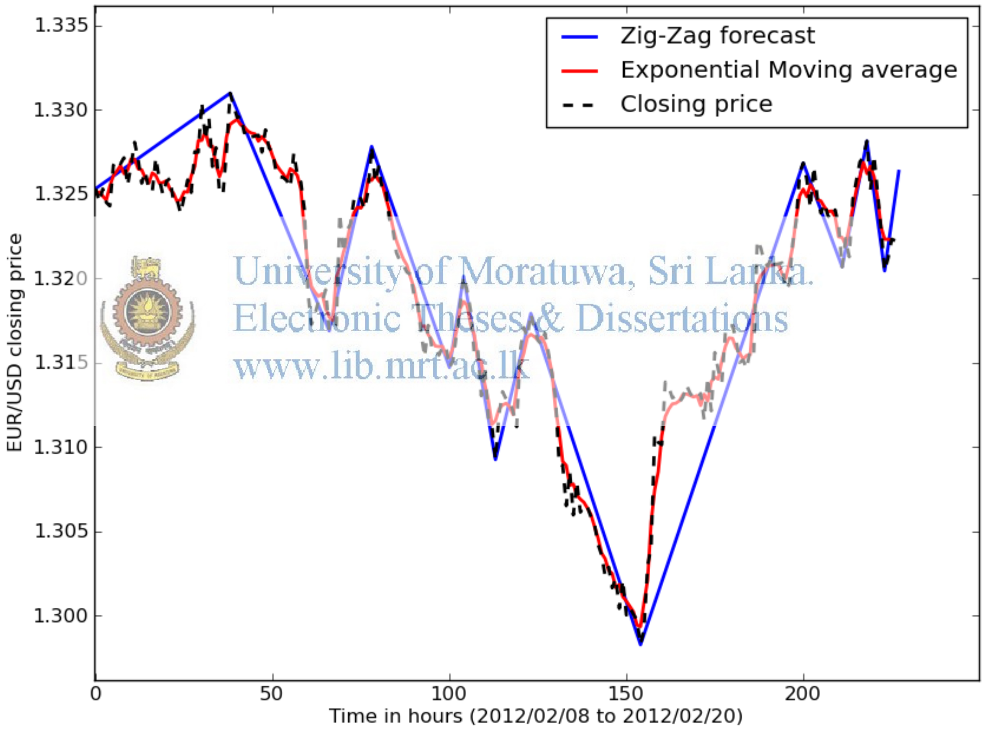

Figure 4.4 - Low mean squared error (MSE) and high directional accuracy (D%) is lead to profit. Higher mean squared error with lower directional accuracy is also leaded to profit.

Table 4.1 - The comparison of two graphs on Figure 4.4

	EMA(4)	Zig Zag	EMA/zig zag
MSE	0.0002273	0.001811	0.125
Dstat	77	65	1.2
Profit(pip)	1970	1078	1.8

Table 4.2- Comparison of two graphs on Figure 4.5

	EMA(30)	Zig Zag	EMA/zig zag
MSE	0.003124	0.00181	1.7
Dstat	64	65	0.98
Profit(pip)	1109	1078	1.028

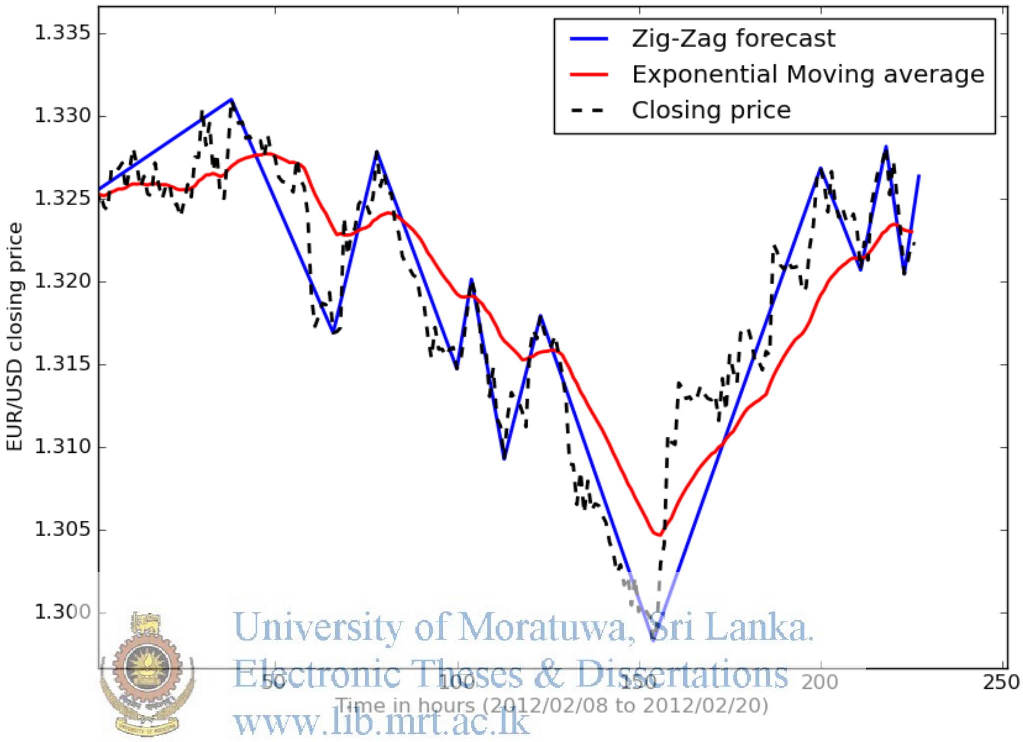

Figure 4.5 – EMA (30) and ideal zig-zag forecast on the same graph.

Table 4.1 summarizes the MSE, Dstat and Profit of the forecast shown in Figure 4.4. Mean error of the Exponential moving average (EMA) forecast is approximately 8 times low compared to Zig-zag forecast. Directional statistics of EMA is 77 against 65 of Zig-Zag forecast. Profit of the EMA forecast is approximately 1.8 times higher than the Zig-Zag forecast. Therefore, better forecasting graph can be selected only using MSE and D% according to Figure 4.4. Table 4.2 summarizes MSE, Dstat and profit for the forecast shown in Figure 4.5. However according to Table 4.2, Zig-Zag forecast should be the better forecast according to two above MSE and D% indicators. However, when concerning profits, ideal EMA (30) forecast is the better one. Results of Figure 4.5 contradict the use of MSE and D% for model selecting performance measures. Therefore profit should be included as the main performance measure along with MSE and D% for any forecasting model where money making is the main concern. This concludes that taking one step ahead forecast does not need to be precise price and direction next to it. It is easy to model a system that can predict the overall direction of the price movement rather than a model predicting the price direction next to it.

5. MODEL TESTING

Model performance testing is done through simulated market conditions. Researchers have tried to imitate the real conditions as much as possible. Metatrader 4 (MT4) platform is incorporated with this study for market testing [71] . It has well developed model testing environment with real market data. Among the available programming languages (C++, java, C#, VB, Oython etc) Python is used to build the forecasting model. Python language is faster than Matlab and has all the required libraries for model building (LIB SVM for SVM model building, *NumPy* for array operations and *Mathplotlib* for plotting graphs). Model is tested for real market conditions by integrating with MT4 platform. Unfortunately there is no way to directly connect Python program with MT4 interface. Therefore integration between MT4 and Python is done through text files. Even though integration worked well, there was a time barrier which limits the speed of testing.

5.1 Meta Trader Expert Advisor

Expert advisor (EA) is a basic program in Meta trader which runs its algorithm for every new incoming price value. Idea is to program an expert advisor which writes past values of given currency pair to a text file. Programing language used is mql4 developed by MetaQuotes Software Corp and it is available with MT4 platform. EA contains three main functions *init(), deinit()* and start(). *init()* function executes the code within the function when it is attached to the chart. Chart can be EUR/USD, USD/JPY or any other currency pair and *init()* executes immediately and once when attached to the chart. When the expert advisor is attached to the chart *start()* function executes its code for every new incoming price value. *deinit()* executes its integrated code when the de-initializing from the chart. *Start()* function contains the whole algorithm for trading from the outputs given by Python program.

EA algorithm for back testing is newly designed for profitable trading by taking proposed model output and RSI values as inputs. Future price direction have taken by model output and further conformed by using RSI values. Finally the EA is designed to trade in the market by using above price direction and random behavior of the market. Trading algorithm has designed in a way that random behavior of the market as an added advantage. Flow diagram can be shown in Figure 5.1

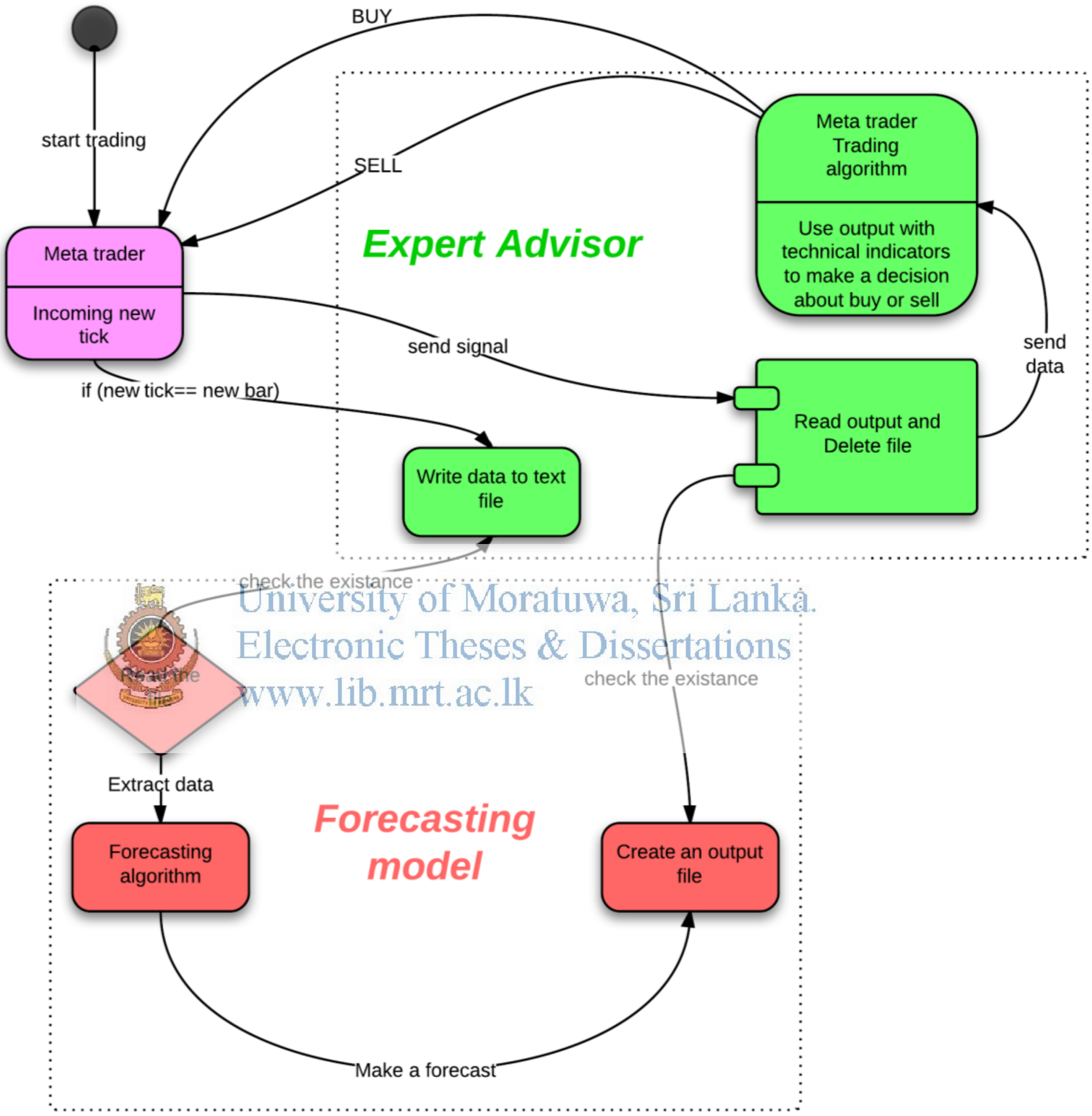

Figure 5.1: Flow diagram of the overall process

Process can be described as follows.

1. Each incoming new price value, EA looks for output file of a Python program. If there is an output file, EA read the output values and make a trading decision. If the new tick equals the time frame value (can be 1 min, 5 min or 15 min etc) then EA creates a new output text file from past data.txt file contains several thousand open, close, high and low past price values of a given time frame.
2. Soon after new text file is created, Python model reads the output, makes a forecast within fraction of a second and writes output value to a retut.txt file.
3. Meanwhile EA is always running and once it is spotted new output file is created. Then it reads the values and deletes it.
4. Buy and sell algorithm is formed for making trades according to the forecast. If the direction of the forecast is changed from previous direction and if it is going down, then it is time to sell and close the buy order. If the direction of the forecast is changed from the previous direction and if it is going up, then it is time to buy and close the sell order.
5. Go to step 1

5.2 Meta Trader Back Testing

Meta trader back testing allows to test whether the new algorithm works on past data by making a simulated environment. There is a considerable debate over the back testing and real time trade executions. MT4 states that the environment is 90% closer to actual system. However, back testing is the only reliable tool available for rapid testing of any new algorithm for considerable amount of time. For example, strategy based on one hour apart past data can be tested for few months backwards within few minutes with the back tester. Or else one has to wait for few months to test the algorithm. Python model is tested upon both real conditions and back tester simulated conditions.

6. SIMULATIONS AND RESULTS

There should be a consistent accuracy on different time frames for a model to succeed in FOREX market. Robustness of the model can be tested by using the capability of model to forecast on different time frames. Forecasting results and its consistency with various market conditions have shown below for 1hour and 5min time frames.

6.1 Forecast for EUR/USD 1 Hr.

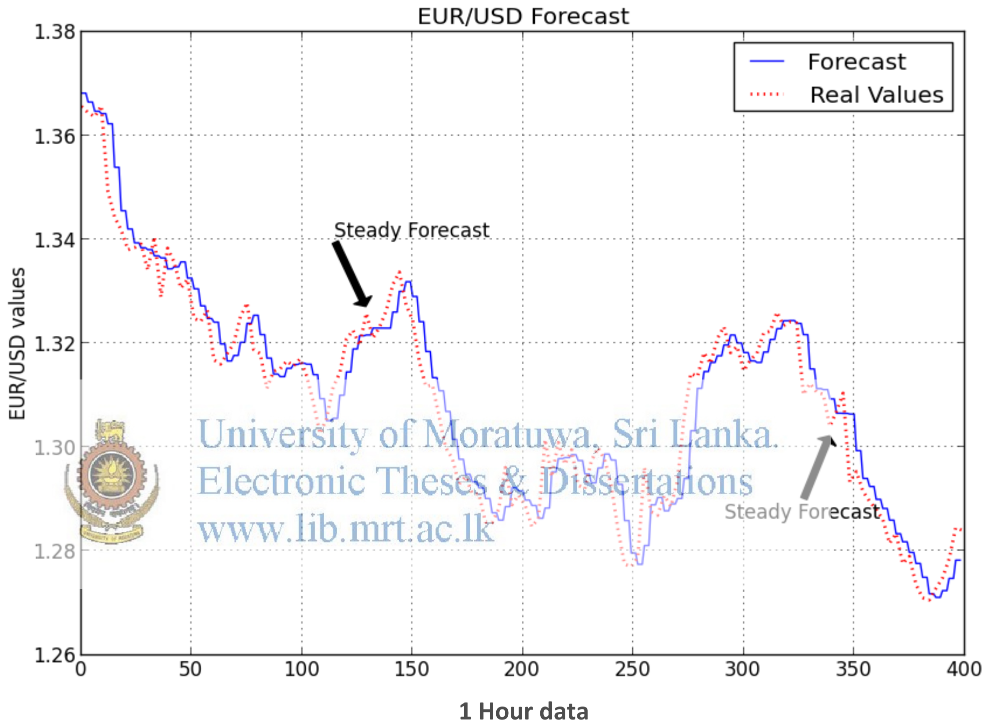

Figure 6.1- Model forecast for 400 one-hour data points

Figure 6.1 shows the forecast for one hour EUR/USD data. Few points have pointed out as steady forecast points to show the capability of the model to forecast overall price direction correctly in high noise conditions. Since the model takes only data that are apart from time frame, forecast does not change with the extra noise lies within the frame. Model forecast will exactly be the same for live trading. This will be an advantage when model test with MT4 back tester. Zoomed positions of steady forecast examples have shown in Figure 6.2 and Figure 6.3

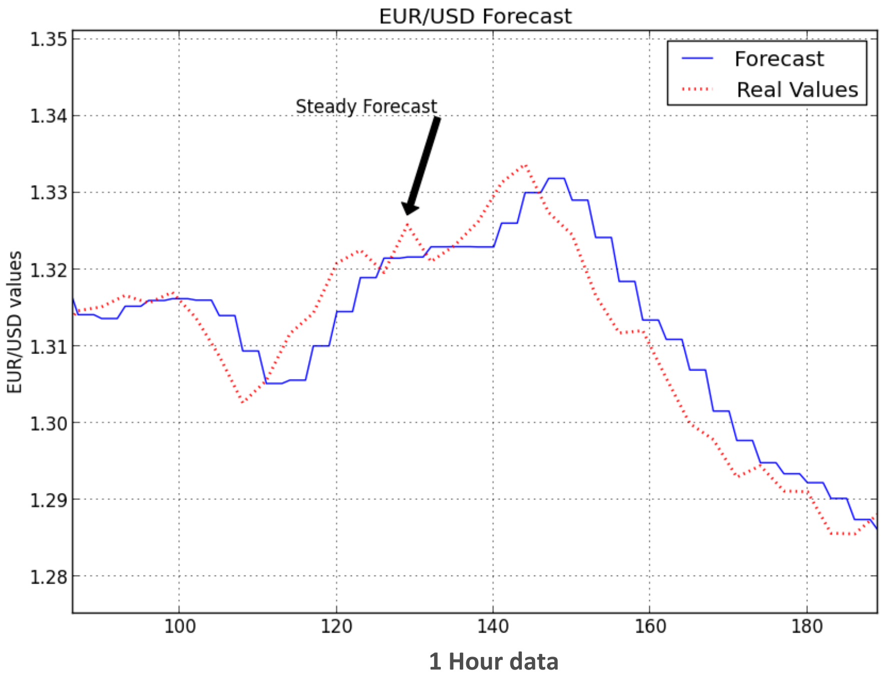

Figure 6.2- Steady forecast for uptrend with noisy environment

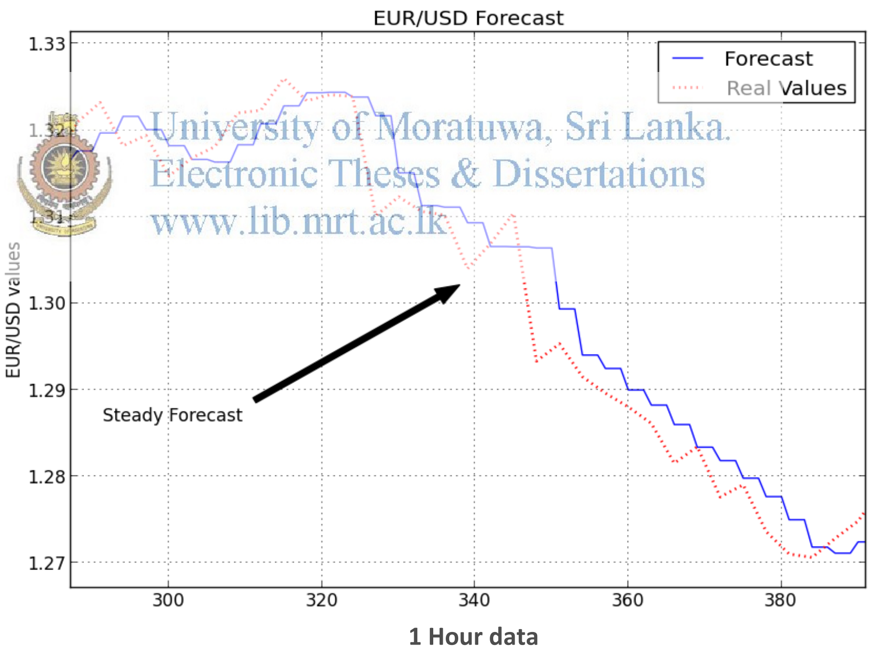

Figure 6.3- Steady forecast for down trend with noisy environment

6.2 Forecast for EUR/USD 5min

EUR/USD 5 min forecast is shown in Figure 6.4. Zoomed area of the forecast is shown in the Figure 6.5 for better visualization purpose. Figure 6.5 shows the capability of the model to accurately forecast the price direction with high noisy environment.

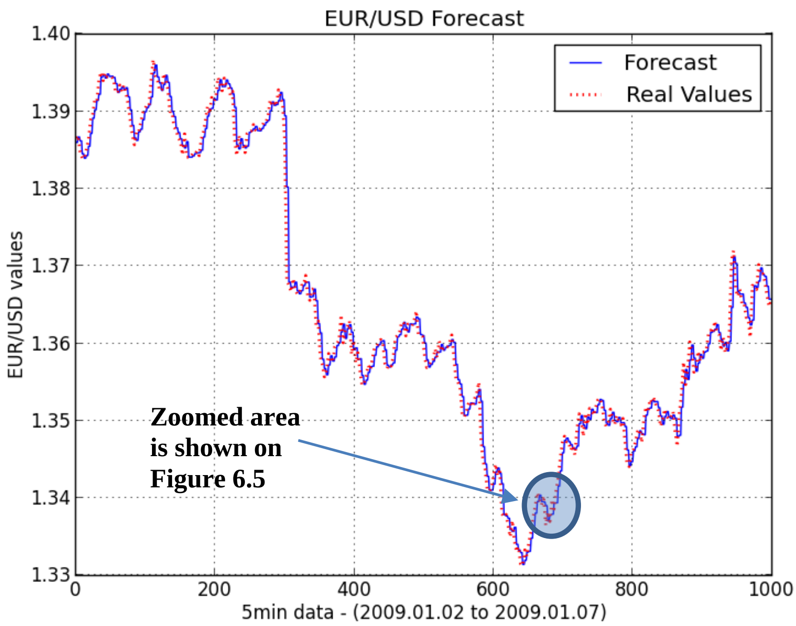

Figure 6.4- Model forecast for 1000 five minute data points.

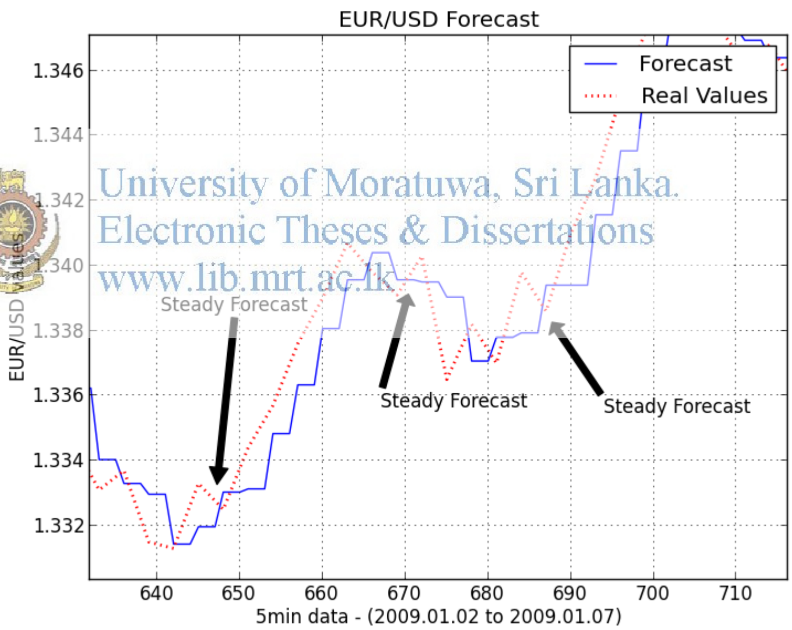

Figure 6.5- Zoomed area for evaluate model robustness under noisy environment.

6.3 Test Results for Real Market Conditions

EUR/USD and USD/JPY currency pairs from 2009 to 2012 have been used for testing the model with MT4. Testing period has been limited to one year without any human supervision and parameter tuning. Robustness of the model for FOREX trading is proven for EUR/USD and USD/JPY by letting the model trade without frequent parameter tuning. When trading in the real markets, there are certain performance measures for consistent performance of the trading model. Which are net profit, maximal drawdown, number of profitable trades and number of loss trades.

Profit is the main target of any trading system. However, even if the trading system is profitable for a year or more, there is no guarantee that the model will gain same profit for the next year. Other performance measures come in handy at this scenario. Even with a high profit model that has shown good results through years; if the model has higher relative drawdown (over 50%-100%), then it cannot be considered as a good trading system for trading further. For example, 50% relative drawdown implies there is a risk that trading account balance can be reduced to half of its initial value at any given moment. Number of profitable and loss trades also give the solid understanding of the model performance. Even if the model has traded profitably, it should be con-sistent throughout the trading period. Same profit can be generated with low number of large profitable trades and higher number of small profitable trades. Model with higher number of small profitable trades does have the higher probability to give a continuous profit over model with low number of large profit trades. Testing results on real conditions have shown the models ability to forecast and trade profitably for consecutive years without frequent parameter tuning.

Section 6.4 presents the results for EUR/USD currency pair. Initial deposit used for trading was 10000$. Table 6.1 summarizes the back testing results for EUR/USD for the period of 2009 January to 2010 January. Model generated the total net profit of 9214.82$ with 21.37% relative drawdown. Number of profitable trades was 176 (93.62%) while the number of lost trades was 12 (6.38%). Figure 6.6 shows the account balance throughout the year 2009.

Table 6.2 summarizes the back testing results for EUR/USD for the period of 2010 January to 2011 January. Model generated the total net profit of 8366.90$ with 33.84% relative drawdown. Number of profitable trades was 193 (100.00%) while the number of lost trades was 0 (0%). Figure 6.7 shows the account balance throughout the year 2010.

Table 6.3 summarizes the back testing results for EUR/USD for the period of 2011 January to 2012 January. Model generated the total net profit of 3543.47$ with 40.53% relative drawdown. Number of profitable trades was 217 (99.54%) while the number of lost trades was 1 (0.46%). Figure 6.8 shows the account balance throughout the year 2011.

Section 6.5 presents the results for USD/JPY currency pair. Initial deposit used for trading was 10000$. Table 6.4 summarizes the back testing results for USD/JPY for the period of 2009 January to 2010 January. Model generated the total net profit of 3180.15$ with 38.92% relative drawdown. Number of profitable trades was 213 (95.09%) while the number of lost trades was 11 (4.91%). Figure 6.9 shows the account balance throughout the year 2009.

Table 6.5 summarizes the back testing results for USD/JPY for the period of 2010 January to 2011 January. Model generated the total net profit of 4281.10$ with 32.12% relative drawdown. Number of profitable trades was 369 (96.60%) while the number of lost trades was 13 (3.40%). Figure 6.10 shows the account balance throughout the year 2010.

Table 6.6 summarizes the back testing results for USD/JPY for the period of 2011 January to 2012 January. Model generated the total net profit of 5664.23$ with 43.73% relative drawdown. Number of profitable trades was 456 (98.49%) while the number of lost trades was 7 (1.51%). Figure 6.11 shows the account balance throughout the year 2011. Summary of the overall results for period of 2009 to 2012 is mentioned in Table 6.7. For EUR/USD, average profit was 7041.73$ and relative drawdown was 31.34%. For USD/JPY, average profit was 4375.16$ and relative drawdown was 38.26%.

6.4 Results for EUR/USD back testing

Table 6.1- Back testing results for the period of 2009 January to 2010 January

Symbol		EURUSD (Euro Vs. US Dollar)			
Period		15 Minutes (M15) 2009.01.02 05:00 - 2009.12.31 17:45 (2009.01.01 - 2010.01.01)			
Model		Every tick (the most precise method based on all available least timeframes)			
Bars in test	25562	Ticks modeled	12305996	Modeling quality	90.00%
Mismatched charts errors	3				
Initial deposit $	10000.00				
Total net profit $	9214.82	Gross profit $	9287.23	Gross loss $	-72.41
Profit factor	128.25	Expected payoff	49.01		
Absolute drawdown	1269.67	Maximal drawdown	3230.80 (21.37%)	Relative drawdown	21.37% (3230.80)
Total trades	188	Short positions (won %)	59 (100.00%)	Long positions (won %)	129 (90.70%)
		Profit trades (% of total)	176 (93.62%)	Loss trades (% of total)	12 (6.38%)
	Largest	profit trade	200.00	loss trade	-27.47
	Average	profit trade	52.77	loss trade	-6.03
	Maximum	consecutive wins (profit in money)	47 (2878.92)	consecutive losses (loss in money)	2 (-3.20)
	Maximal	consecutive profit (count of wins)	2878.92 (47)	consecutive loss (count of losses)	-27.47 (1)
	Average	consecutive wins	15	consecutive losses	1

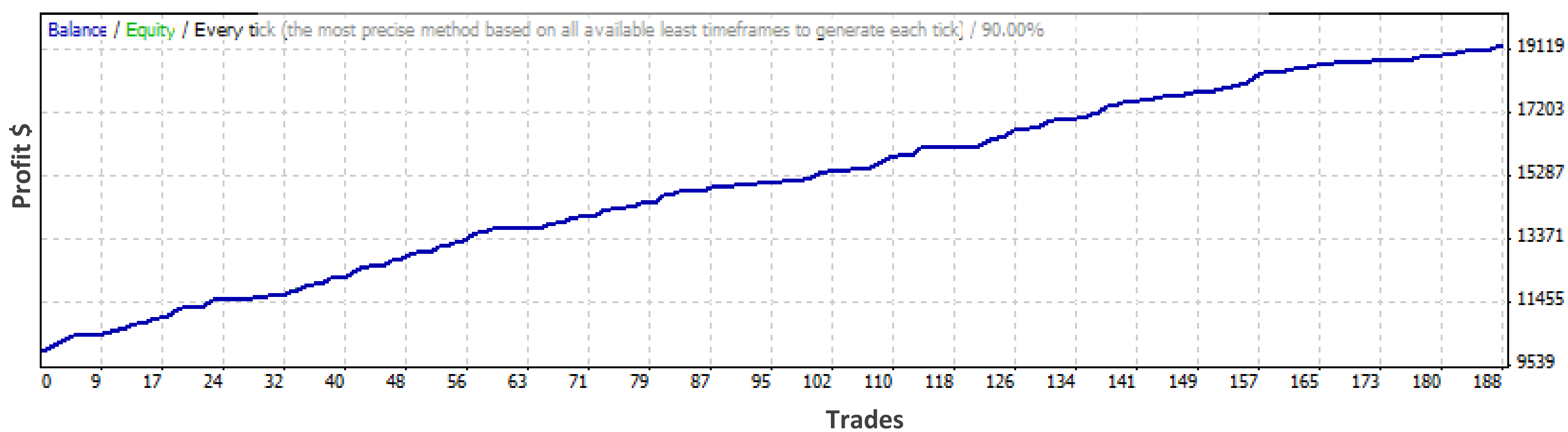

Figure 6.6 - Profit chart for the period of 2009 January to 2010 January

Table 6.2- Back testing results for the period of 2010 January to 2011 January

Symbol		EURUSD (Euro Vs. US Dollar)			
Period		15 Minutes (M15) 2010.01.03 23:00 - 2010.12.31 21:45 (2010.01.01 - 2011.01.01)			
Model		Every tick (the most precise method based on all available least timeframes)			
Bars in test	25720	Ticks modeled	11103057	Modeling quality	90.00%
Mismatched charts errors	5				
Initial deposit	10000.00				
Total net profit	8366.90	Gross profit	8366.90	Gross loss	-0.00
Profit factor		Expected payoff	43.35		
Absolute drawdown	954.00	Maximal drawdown	5965.43 (33.84%)	Relative drawdown	33.84% (5965.43)
Total trades	193	Short positions (won %)	97 (100.00%)	Long positions (won %)	96 (100.00%)
		Profit trades (% of total)	193 (100.00%)	Loss trades (% of total)	0 (0.00%)
	Largest	profit trade	78.00	loss trade	-0.00
	Average	profit trade	43.35	loss trade	-0.00
	Maximum	consecutive wins (profit in money)	193 (8366.90)	consecutive losses (loss in money)	0 (-0.00)
	Maximal	consecutive profit (count of wins)	8366.90 (193)	consecutive loss (count of losses)	-0.00 (0)
	Average	consecutive wins	193	consecutive losses	0

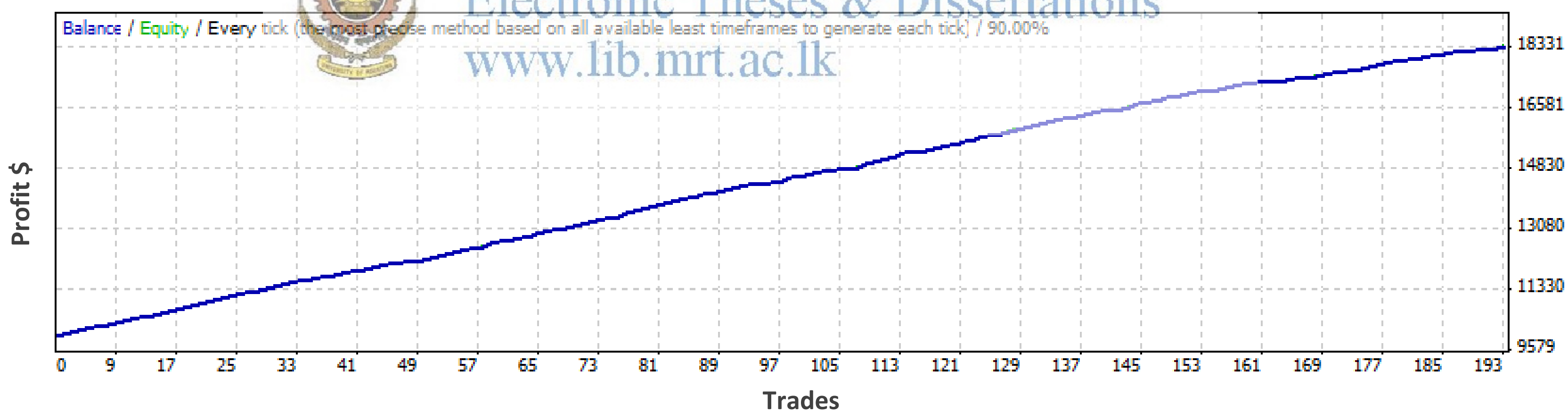

Figure 6.7 - Back testing results for the period of 2010 January to 2011 January

Table 6.3 - Back testing results for the period of 2011 January to 2012 January

Symbol		EURUSD (Euro Vs. US Dollar)			
Period		15 Minutes (M15) 2011.01.02 23:00 - 2011.12.30 22:00 (2011.01.01 - 2012.01.01)			
Model		Every tick (the most precise method based on all available least timeframes)			
Bars in test	25711	Ticks modeled	17362597	Modeling quality	90.00%
Mismatched charts errors	6				
Initial deposit $	10000.00				
Total net profit $	3543.47	Gross profit $	3550.40	Gross loss $	-6.92
Profit factor	512.92	Expected payoff	16.25		
Absolute drawdown	3861.73	Maximal drawdown	4247.63 (37.37%)	Relative drawdown	40.53% (4182.73)
Total trades	218	Short positions (won %)	60 (100.00%)	Long positions (won %)	158 (99.37%)
		Profit trades (% of total)	217 (99.54%)	Loss trades (% of total)	1 (0.46%)
	Largest	profit trade	30.00	loss trade	-6.92
	Average	profit trade	16.36	loss trade	-6.92
	Maximum	consecutive wins (profit in money)	200 (3229.40)	consecutive losses (loss in money)	1 (-6.92)
	Maximal	consecutive profit (count of wins)	3229.40 (200)	consecutive loss (count of losses)	-6.92 (1)
	Average	consecutive wins	109	consecutive losses	1

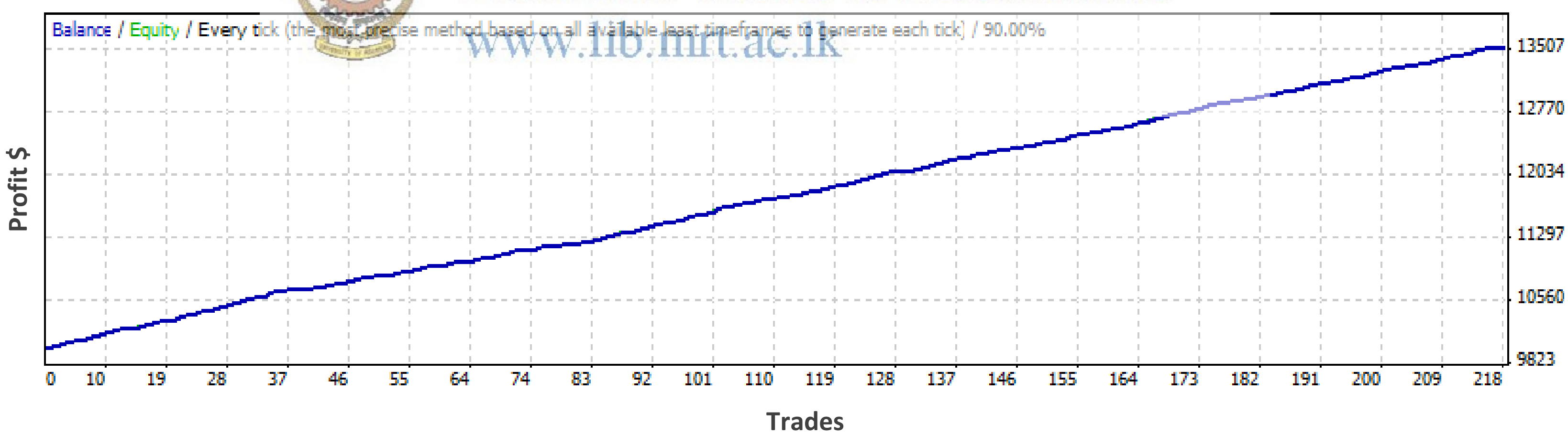

Figure 6.8 - Profit for the period of 2011 January to 2012 January

6.5 Results for USD/JPY back testing

Table 6.4 - Back testing results for the period of 2009 January to 2010 January

Symbol		USDJPY (US Dollar Vs. Japanese Yen)			
Period		15 Minutes (M15) 2009.01.02 09:00 - 2009.12.31 17:45 (2009.01.01 - 2010.01.01)			
Model		Every tick (the most precise method based on all available least timeframes)			
Bars in test	25441	Ticks modeled	6156592	Modeling quality	90.00%
Mismatched charts errors	3				
Initial deposit $	10000.00				
Total net profit $	3180.15	Gross profit $	3191.96	Gross loss $	-11.81
Profit factor	270.28	Expected payoff	14.20		
Absolute drawdown	2060.44	Maximal drawdown	5058.39 (38.92%)	Relative drawdown	38.92% (5058.39)
Total trades	224	Short positions (won %)	151 (96.69%)	Long positions (won %)	73 (91.78%)
		Profit trades (% of total)	213 (95.09%)	Loss trades (% of total)	11 (4.91%)
	Largest	profit trade	30.93	loss trade	-1.76
	Average	profit trade	14.99	loss trade	-1.07
	Maximum	consecutive wins (profit in money)	69 (1073.99)	consecutive losses (loss in money)	1 (-1.76)
	Maximal	consecutive profit (count of wins)	1073.99 (69)	consecutive loss (count of losses)	-1.76 (1)
	Average	consecutive wins	18	consecutive losses	1

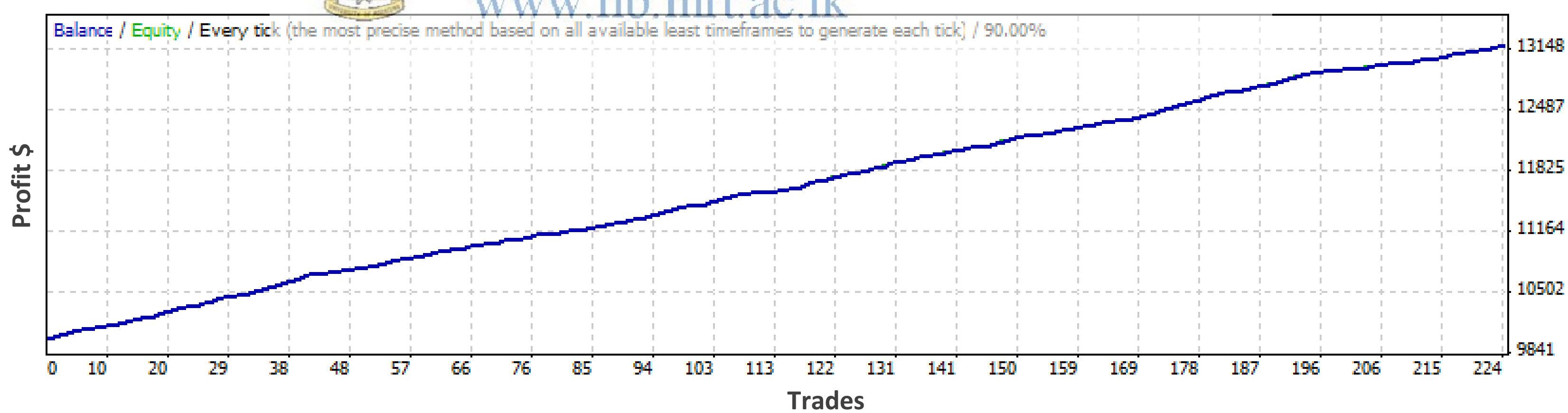

Figure 6.9- Profit for the period of 2009 January to 2010 January

Table 6.5- Back testing results for the period of 2010 January to 2011 January

Symbol		USDJPY (US Dollar Vs. Japanese Yen)			
Period		15 Minutes (M15) 2010.01.03 23:00 - 2010.12.31 23:45 (2010.01.01 - 2010.12.01)			
Model		Every tick (the most precise method based on all available least timeframes)			
Bars in test	21431	Ticks modeled	6736327	Modeling quality	90.00%
Mismatched charts errors	2				
Initial deposit $	10000.00				
Total net profit $	4281.10	Gross profit $	4317.82	Gross loss $	-36.72
Profit factor	117.60	Expected payoff	11.21		
Absolute drawdown	2388.61	Maximal drawdown	3601.37 (32.12%)	Relative drawdown	32.12% (3601.37)
Total trades	382	Short positions (won %)	313 (96.49%)	Long positions (won %)	69 (97.10%)
		Profit trades (% of total)	369 (96.60%)	Loss trades (% of total)	13 (3.40%)
	Largest	profit trade	22.61	loss trade	-29.82
	Average	profit trade	11.70	loss trade	-2.82
	Maximum	consecutive wins (profit in money)	83 (981.87)	consecutive losses (loss in money)	1 (-29.82)
	Maximal	consecutive profit (count of wins)	981.87 (83)	consecutive loss (count of losses)	-29.82 (1)
	Average	consecutive wins	26	consecutive losses	1

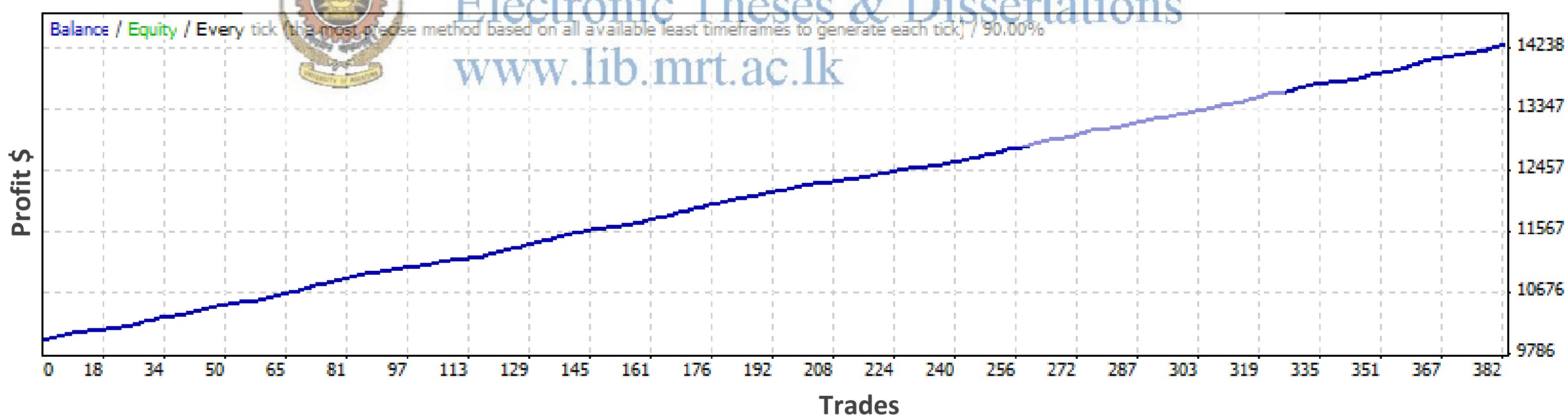

Figure 6.10 - Profit for the period of 2010 January to 2011 January

Table 6.6- Back testing results for the period of 2011 January to 2012 January

Symbol		USDJPY (US Dollar Vs. Japanese Yen)			
Period		15 Minutes (M15) 2011.01.16 23:00 - 2011.12.30 22:00 (2011.01.15 - 2012.01.01)			
Model		Every tick (the most precise method based on all available least timeframes)			
Bars in test	24757	Ticks modeled	10111687	Modeling quality	90.00%
Mismatched charts errors	3				
Initial deposit	10000.00				
Total net profit	5664.23	Gross profit	5806.74	Gross loss	-142.51
Profit factor	40.75	Expected payoff	12.23		
Absolute drawdown	3290.80	Maximal drawdown	5214.59 (43.73%)	Relative drawdown	43.73% (5214.59)
Total trades	463	Short positions (won %)	418 (98.33%)	Long positions (won %)	45 (100.00%)
		Profit trades (% of total)	456 (98.49%)	Loss trades (% of total)	7 (1.51%)
	Largest	profit trade	26.46	loss trade	-102.02
	Average	profit trade	12.73	loss trade	-20.36
	Maximum	consecutive wins (profit in money)	157 (1940.27)	consecutive losses (loss in money)	1 (-102.02)
	Maximal	consecutive profit (count of wins)	1940.27 (157)	consecutive loss (count of losses)	-102.02 (1)
	Average	consecutive wins	65	consecutive losses	1

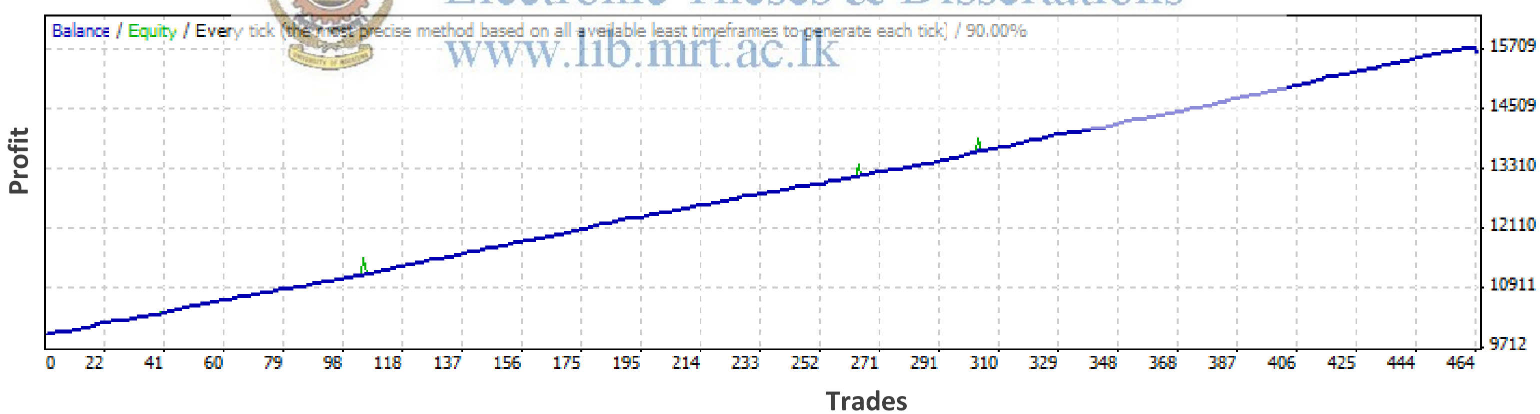

Figure 6.11- Profit for the period of 2011 January to 2012 January

Table 6.7 – Summery of the results

Year	drawdown	Profit trades		Total trades	profit		
		number	%		Gross profit	Gross loss	total
EUR/USD summery							
2009	21.37%	176	93.62%	188	9287.23	-72.41	9214.82
2010	32.12%	193	100%	193	8366.90	0	8366.90
2011	40.53%	217	99.54%	218	3550.4	-6.92	3543.47
average	31.34%	195	97.72%	200	6418.815	-26.44	7041.73
USD/JPY summery							
2009	38.92%	213	95.09%	224	3191.96	-11.81	3180.15
2010	32.12%	369	96.60%	382	4317.82	-36.72	4281.10
2011	43.73%	456	98.49%	463	5806.74	-142.51	5664.23
average	38.26%	346	96.73%	356.3	4438.84	-63.68	4375.16
Overall averages							
average	34.8%	270	97.23%	278	5428	-45	5700

7. CONCLUSION AND FUTURE DIRECTIONS

This thesis presented a profitable forecasting model that combines the wavelet transform k-means clustering and support vector machine. Detailed literature review has been carried out on recent models and promising directions were identified. Proposed model was traded for real market values with simulated conditions of 'metatrader4'.

The following conclusions on the forecasting model design were made.

- The results indicated that the proposed model has the capability to forecast in high noise conditions for different time frames. The wavelet decomposition noise elimination procedure was effective in the noisy input data.
- Debauchee's wavelet combined with discrete wavelet transform and soft thresholding have shown promising results for time series denoising and edge detection.
- K-means clustering has improved the forecasting accuracy by reducing bias variance dilemma.
- SVM structural risk minimization ensured model train to best possible level with available data.
- Combination of chosen techniques has given the steady forecast and sharp peak detection to ensure the correct forecast for auto trading model, which process the forecast to make trading decisions.

Auto trading model was developed with combination of the forecast to trade in simulated environment that is closely related to the real market conditions. Simulated environment used was metatrader4 back testing. Continuous testing of three years has proven the capability of proposed model to trade profitably in two highly volatile currency pairs. Table 6.7 shows the summery of all the forecasts. The proposed model has averaged over 43% payback and 38% relative drawdown for USD/JPY currency pair. It has shown 70% payback and 30% relative drawdown for EUR/USD currency pair which is more volatile. Average payback of 57% for both currencies has shown the gross profit which will be

0.57 times of initial capital at the end of the year for a currency. 35% low drawdown has shown the amount of risk that has taken for achieving that profit. High payback and low drawdown have shown the profitability and low risk that model generates. Average trades per year for USD/JPY were 346 and 195 for EUR/USD. High numbers of average trades per year that are distributed evenly for every month, have proven the profit gained was steady through time. Very low average lost trade amount (-45$) compared to average profit gain (5428$) has proven the accuracy of model's forecast. Therefore, proposed model can be used in live trading account with low risk for profitable trading in FOREX market.

The results presented in this thesis is promising, but can be improved. Instead of depending only on the historical price data, economic factors like inflation, interest rates, current-account deficits, public debt and economic performance should be taken in to account in the future developments. Selection of the forecastable span should be selected more precisely by analyzing correlations. Variable leverage selection can be introduced with the change of profitable trades and level of risk introduced by trading system should be reduced further.

References

[1] "Triennial Central Bank Survey of foreign exchange and derivatives market activity in 2013," Bank For International Settlement, 2013. [Online]. Available: https://www.bis.org/publ/rpfx13.htm.

[2] G. S. Atsalakis and K. P. Valavanis, "Surveying stock market forecasting techniques – Part II: Soft computing methods," *Expert Systems with Applications,* p. 5932–5941, 2009.

[3] K. Y. Huang, T. C. Chang and J. H. Lee, "A hybrid model to improve the capabilities of forecasting based on GRA and ANN theories," in *Proc. of the IEEE Int. Conf. on Grey Systems*, Nanjing, 2009.

[4] J. Yao, C. L. Tan and H. L. Poh, "Neural Networks for technical analysis: a study on KLCI," *Int. Journal of Theoretical and Applied Finance,* vol. 2, no. 2, pp. 221-241, 1999.

[5] L. Yu, S. Wang and K. K. Lai, "A Hybrid GA-Based SVM Model for Foreign Exchange Market Tendency Exploration," in *Foreign-exchange-rate forecasting with artificial neural networks*, F. . S. Hillier, Ed., New York, Springer Science+Business Media, 2007, pp. 155-173.

[6] L. Yu, S. Wang and K. K. Lai, "Forecasting Foreign Exchange Rates with a Multistage Neural Network Ensemble Model," in *Foreign exchange rate forecasting with artificial neural networks*, F. S. Hillier, Ed., New York, Springer Science+Business Media, 2007, pp. 177-202.

[7] L. Yu, S. Wang and K. K. Lai, "A novel nonlinear ensemble forecasting model incorporating GLAR and ANN for foreign exchange rates," *Computers & Operations Research,* vol. 32, no. 10, pp. 2523-2541, October 2005.

[8] B. G. and M. , "efficient market hypothesis," *The New Palgrave: A Dictionary of Economics,* vol. 2, pp. 120-123, 1987.

[9] "Most Traded Pairs," investopedia, 2013. [Online]. Available: http://www.investopedia.com/walkthrough/forex/getting-started/pairs.aspx.

[10] "forex," worldmarkethours, 2013. [Online]. Available: http://www.2013.worldmarkethours.com/Forex/index.htm.

[11] E. Bodt, J. Rynkiewicz and M. Cottrell, "Some known facts about financial data," *European symposium on artificial neural networks,* vol. 25, no. 27, pp. 223-236, 2001.

[12] . E. Fama, " Random Walks In Stock Market Prices," *Financial Analysts Journal,* vol. 21, pp. 55-59, 1965.

[13] C. J. Lu, T. S. Lee and C. C. Chiu, "Financial time series forecasting using Independent Component Analysis and Support Vector Regression," *Decision Support Systems,* vol. 47, p. 115–125, February 2009.

[14] R. Ghazali, A. J. Hussain and P. Liatsis, "Dynamic Ridge Polynomial Neural Network: Forecasting the univariate non-stationary and stationary trading signals," *Expert Systems with Applications,* vol. 38, p. 3765–3776, 2011.

[15] A. Kirman, "Economic theory and the crisis," 14 November 2009. [Online]. Available: http://www.voxeu.org/index.php?q=node/4208.

[16] P. Tenti, "Forecasting foreign exchange rates using recurrent neural networks," *AppliedArtificial Intelligence,* vol. 10, pp. 567-581, 1996.

[17] W. Brock, H. and B. LeBaron, "Nonlinear dynamics, chaos, and instability.," 1991.

[18] P. Grauwe de, H. Dewachter and M. Embrechts, " Exchange rate theory. Chaotic models of foreign exchange markets," *Journal of Economic Behavior & Organization.,* vol. 25, no. 3, p. 473–475, 1994.

[19] H. Fang, L. S. Lai and M. Lai, "Fractal structure in currency futures price dynamics," *The Journal of Futures Markets,* vol. 14, no. 2, pp. 169-181, 1994.

[20] L. Yu, S. Wang, W. Huang and K. K. Lai, "Are foreign exchange rates predictable-a survay from Artificial Neural Network perspective," *Scientific Inquiry,* vol. vol. 8, no. 2, p. 207 – 228, 2007.

[21] [Online]. Available: http://www.metaquotes.net.

[22] C. J. Lu, "Integrating Independent Component Analysis-based denoising scheme with Neural Network for stock price prediction," *Expert Systems with Applications,* vol. 37, p. 7056–7064, 2010.

[23] Z. Yudong and W. Lenan, "Stock market prediction of S&P 500 via combination of improved BCO approach and BP Neural Network," *Expert Systems with Applications,* vol. 36, p. 8849–8854, 2009.

[24] S. H. Hsu, J. J. P. A. Hsieh, T. C. Chih and K. C. Hsu, "A two-stage architecture for stock price forecasting by integrating Self-organizing Map and Support VectorRegression," *Expert Systems with Applications,* vol. 36, p. 7947–7951, 2009.

[25] A. U. Khan, T. K. Bandopadhyaya and S. Sharma, "SOM and Technical Indicators based Hybrid Model gives better returns on investments as compared to BSE-30 Index," in *Proc. of the 3rd Int. Conf. on Knowledge Discovery and Data Mining*, Phuket, 2010.

[26] L. J. Cao and F. E. H. Tay, "Support Vector Machine With adaptive parameters in financial time series forecasting," *IEEE transactions on neural networks,* vol. 14, no. 6, pp. 1506-1519, November 2003.

[27] M. S. C. December 2007. [Online]. Available: http://championship.mql4.com/2007/users/Better.

[28] J. F. Chang, C. W. Chang and W. Y. Tzeng, "Forecasting exchange rates using integration of Particle Swarm Optimization and Neural Networks," in *Proc. of the 4th Int. Conf. on Innovative Computing, Information and Control*, Kaohsiung, 2009.

[29] S. Hayking, "Multilayer perceptrons," in *Neural Networks: A comprehensive foundation, 2nd ed*, NJ, Prentice Hall, 1999, pp. 211-212.

[30] Y. K. Kwon and B. R. Moon, "A hybrid neurogenetic approach for stock forecasting," *IEEE Transactions On Neural Networks, Vol. 18, No. 3, May 2007,* vol. 18, no. 3, pp. 851-864, may 2007.

[31] J. Yao and C. L. Tan, "A case study on using Neural Networks to perform technical forecasting of forex," *Neurocomputing,* vol. 34, pp. 79-98, 2000.

[32] B. Qian and K. Rasheed, "Foreign Exchange Market Prediction with multiple classifiers," *Journal of Forecasting,* vol. 29, p. 271–284, 2010.

[33] E. W. Saad, D. V. Prokhorov and D. C. Wunsch, "Comparative study of stock trend prediction using Time Delay, Recurrent and Probabilistic Neural Networks," *IEEE transactions on neural networks,* vol. 9, no. 6, pp. 1456-1470, November 1998.

[34] H. M. Feng and H. C. Chou, "Evolutional RBFNs prediction systems generation in the applications of financial time series data," *Expert Systems with Applications,* vol. 38, p. 8285–8292, 2011.

[35] F. Hillier , Ed., "Forecasting Foreign Exchange Rates Using an Adaptive Back-Propagation Algorithm with Optimal Learning Rates and Momentum Factors," in *Foreign exchange rate forecasting with artificial neural networks*, New York, Springer Science+Business Media, 2007, pp. 65-84.

[36] L. Yu, S. Wang and K. K. Lai, "An Online BP Learning Algorithm with Adaptive Forgetting Factors for Foreign Exchange Rates Forecasting," in *Foreign exchange rate forecasting with artificial neural networks*, F. S. Hillier, Ed., New York, Springer Science+Business Media, 2007, pp. 87-100.

[37] L. Yu, S. Wang and L. K. K. Kin, "An Improved BP Algorithm with Adaptive Smoothing Momentum Terms for Foreign Exchange Rates Prediction," in *Foreign exchange rate forecasting with artificial neural networks*, F. S. Frederick, Ed., New York, Springer Science+Business Media, 2007, pp. 101-118.

[38] K. J. Kim, "Artificial Neural Networks with evolutionary instance selection for financial forecasting," *Expert Systems with Applications,* vol. 30, p. 519–526, 2006.

[39] P. Wang, "Pricing currency options with Support Vector Regression and Stochastic Volatility Model with jumps," *Expert Systems with Applications,* vol. 38, pp. 1-7, 2011.

[40] M. Mehrara, A. Moeini, M. Ahrari and A. Ghafari, "Using technical analysis with Neural Network for forecasting stock price index in Tehran Stock Exchange," *Middle Eastern Finance and Economics,* no. 6, pp. 51-61, 2010.

[41] Z. Liao and J. Wang, "Forecasting model of global stock index by stochastic time effective Neural Network," *Expert Systems with Applications,* vol. 37, p. 834–841, 2010.

[42] Y. H. Wang, "Nonlinear Neural Network forecasting model for stock index option price: Hybrid GJR–GARCH approach," *Expert Systems,* vol. 36, p. 564–570, 2009.

[43] K. J. Kim, "Financial time series forecasting using Support Vector Machines," *Neurocomputing,* vol. 55, p. 307 – 319, March 2003.

[44] W. Shen, X. Guo, C. Wu and D. Wu, "Forecasting stock indices using radial basis function Neural Networks optimized by Artificial Fsh Swarm Algorithm," *Knowledge Based Systems,* vol. 24, p. 378–385, 2011.

[45] L. P. Ni, Z. W. Ni and Y. Z. Gao, "Stock trend prediction based on fractal feature selection and Support Vector Machine," *Expert Systems with Applications,* 2010.

[46] C. L. Huang and C. Y. Tsai, "A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting," *Expert Systems with Applications,* vol. 36, p. 1529–1539, 2009.

[47] T. J. Hsieh, H. F. Hsiao and W. C. Yeh, "Forecasting stock markets using Wavelet Transforms and Recurrent Neural Networks: An integrated system based on artificial bee colony algorithm," *Applied Soft Computing,* vol. 11, p. 2510–2525, October 2010.

[48] S. C. Hui, M. T. Yap and P. Prakash, "A hybrid time lagged network for predicting stock prices," *Int. Journal of the Computer, the Internet and Management,* vol. 8, 2000.

[49] E. L. de Faria, J. L. Gonzalez, J. T. P. Cavalcante, M. P. Albuquerque and M. P. Albuquerque, "Predicting the Brazilian stock market through Neural Networks and Adaptive Exponential Smoothing methods," *Expert Systems with Applications,* vol. 36, p. 12506–12509, 2009.

[50] L. Yu, S. Wang and K. K. Lai, "Hybridizing BPNN and Exponential Smoothing for Foreign Exchange Rate Prediction," in *Foreign exchange rate forecasting with artificial neural networks*, F. . S. Hillier, Ed., New York, Springer Science+Business Media, 2007, pp. 121-131.

[51] R. Gençay and R. Gibson, "Model risk for European style Stock Index Options," *IEEE transactions on neural networks,* vol. 18, no. 1, pp. 193-202, January 2007.

[52] L. Yu, S. Wang and K. K. Lai, "A Neural Network-based nonlinear metamodeling approach to financial time series forecasting," *Applied Soft Computing,* vol. 9, p. 563–574, August 2009.

[53] S. C. Huang and T. K. Wu, "Integrating Recurrent SOM with Wavelet-based kernel partial least square regressions for financial forecasting," *Expert Systems with Applications,* vol. 37, p. 5698–5705, 2010.

[54] S. C. Huang, P. J. Chuang, C. F. Wu and H. J. Lai, "Chaos based Support Vector Regressions for exchange rate forecasting," *Expert Systems with Applications,* vol. 37, p. 8590–8598, 2010.

[55] J. Kamruzzaman and R. A. Sarker, "Forecasting of currency exchange rates using ANN: a case study," *Neural Networks and Signal Processing,* vol. 1, pp. 793 - 797, April 2004.

[56] X. Ma, C. Zhou and I. J. Kemp, "Automated Wavelet Selection and Thresholding for PD Detection," *IEEE Electrical Insulation Magazine,* vol. 18, no. 2, pp. 37-45, 2002.

[57] C. L. Dunis and M. Williams, "Modelling and Tradingthe EUR/USD Exchange Rate:Do Neural Network Models Perform Better?," 2002.

[58] J. Shlens, "A Tutorial on Principal Component Analysis," New York, 2009.

[59] A. Hyvärinen and E. Oja, "Independent Component Analysis:Algorithms and Applications," *Neural Networks,* vol. 13, pp. 411-430, 2000.

[60] T.-W. Lee, J.-F. . Cardoso, E. Oja and S.-I. Amari, "MISEP – Linear and Nonlinear ICA Based on Mutual Information," *Journal of Machine Learning Research 4,* pp. 1297-1318, 2003.

[61] R. Singh, R. Singh and R. E. Vasquez, "Comparison of Daubechies, Coiflet, and Symlet for edge detection," *Visual Information Processing,* vol. 6, July 1997.

[62] G. and A. , "An introduction to wavelets," *Computational Science & Engineering, IEEE,* vol. 2, no. 2, pp. 50 - 61 , 1995.

[63] J. N. L. and R. W. K. , "Automatic extraction and identification of chart patterns towards financial forecast," *Applied Soft Computing,* vol. 7, no. 4, p. 1197–1208, 2007.

[64] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with application in pattern recognition," *IEEE Transactions on Electronic Computers,* Vols. EC-14, pp. 326-334, 1965.

[65] V. N. Vapnic and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability and its Applications,* vol. 16, no. 2, pp. 264-280, 1971.

[66] S. Hayking, "Learning processes," in *Neural Networks: A comprehensive foundation, 2nd ed*, NJ, Prentice Hall, 1999, pp. 97-98.

[67] S. Geman, E. Bienenstock and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation,* vol. 4, pp. 1-58, 1992.

[68] S. Hayking, "Committee Machines," in *Neural Networks: A comprehensive foundation, 2nd ed*, NJ, Prentice Hall, 1999.

[69] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, 1967.

[70] K. Ron, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* , vol. 2, no. 12, p. 1137–1143, 1995.

[71] [Online]. Available: http://www.metatrader4.com/.

Appendix A

Summary of the all revived articles are presented here

Table A.1 - Summery of the revived articles

Article	Data preprocessing	Network type	Training method	Benchmark models	Performance measures
[3]	Grey Relational Analysis	FFANN	BP	ANN, ANN / Grey model (1, N)	RMSE
[4]	Rescaled range analysis+ Normalizing+ sensitive analysis	FFANN	BP	ARIMA, 3-Benchmark models for profit calculation	NMSE, Sign statistics
[5]	GA	SVM	SVR	RW ,ARIMA , linear discriminant analysis, BPANN, SVM, GA-SVM	Hit ratio
[6]	Normalizing + Interval variation	Ensemble SVM	Proposed multi-stage SVM-based nonlinear ensemble model	Ensemble models trained using simple averaging , Simple MSE, Stacked regression, Variance-based weighting , Artificial neural network	RMSE
[7]	Normalize	GLAR+PCA + FFANN-ensemble	BP	Generalized auto regression(GLAR), ANN, Hybrid, Equal weights, Minimum error	NMSE, DS
[13]	Normalize, ICA	SVM	Support vector regression(SVR)	Random walk (RW), SVR	RMSE, NMSE MAD, DS, correct down trend (CD) , correct up trend (CP)
[14]	Normalized	Dynamic Ridge Polynomial ANN	Constructive learning algorithm	Multilayer perceptron (MLP),Functional Link Neural Network (FLNN), Pi-Sigma Neural Network (PSNN), Ridge Polynomial Neural Network (RPNN)	NMSE, Annualized Return
[23]	-	FFANN	Improved bacterial chemotaxis optimization (IBCO)+BP	BPANN	MSE
[24]	Normalize, Self-organizing feature maps	SVM	SVR	SVR	NMSE,MAE, DS,WDS
[25]	Kohonen self-organizing map	FFANN	BP	BSE-30 SENSEX	Not specified

[26]	Z-score normalization method	SVM	Adaptive SVR	Three-layer BP neural network, Regularized RBF neural network	NMSE, DS,MAE
[28]	Particle swarm optimization + normalize	FFANN	BP	ANN models with different indicators as inputs	MSE, RMSE, MAE
[30]	Euclidean distance based partitioning	Recurrent ensemble model	Genetic algorithm (GA)	Genetic programming prediction, SVM	Frequency of correct prediction
[31]	Rescaled range analysis	FFANN	BP	Auto regression integrated moving average(ARIMA)	NMSE, Gradient
[32]	Normalizing+Hurst exponent	Ensemble	vary	FFANN, K-nearest neighbor, Decision tree, Native Bayesian classifier	Average error rate
[33]	Phase Diagrams, Correlation Dimension, Lyapunov exponent	Probabilistic neural network TDNN,RNN	Temporal BP, Extended Kalman filter,	Fisher linear classifier	False alarms
[34]	Step wise regression analysis +Normalizing+ Dynamic learning algorithm based clustering	RBF	Particle-swarm optimization (PSO)+adaptive recursive least-squares	Type 2 fuzzy time series model, Fuzzy time series model, Fuzzy dual-factor time-series	MAD,MAPE, DS,CP,CD
[35]	Auto-regression testing	FFANN	Adaptive BP	Standard BP, LM-based learning, Extended Kalman filter (EKF),BP with optimal learning rates	NMSE, Directional change statistics (DS)
[36]	Auto-regression testing	FFANN	BP Learning Algorithm with Adaptive, Forgetting Factors	Batch learning, EKF-based learning,Levenberg-Marquardt(LM)based learning, Standard BPNN	NMSE,DS
[37]	Auto-regression testing	FFANN	BP + AdaptiveSmoothing + Momentum Terms	Four different networks with modified back propagation for each	MSE
[38]	GA based instant selection	FFANN	GA	GA-ANN	Hit ratio
[39]	-	StochasticVolatility (SV) model with jumps +SVM	SVR	ANN-SV, Garman–Kohlhagen model	MAE, MAPE
[40]	-	ANN	GMDH algorithm	FFANN	RME, MAPE,DS, Profitability

[41]	Normalize	FFANN	BP+ Stochastic time effective function	-	MSE
[42]	Grey-GJR–GARCH	FFANN	BP	-	RMSE , MAE, MAPE
[43]	Normalize	SVM	SVR	BPANN, Case based reasoning	Hit ratio
[44]	Extract indicators with better performance	RBF	Artificial fish swarm algorithm +K means clustering	RBF optimized by GA ,PSO, ARIMA,ANN,SVM	Error ratio
[45]	Mean removal	SVM	SVR	5 SVM based feature selection methods	DS
[46]	Normalize + Self-organizing feature maps	SVM	SVR + grid search	Single SVR	MSE, MAE, MAPE
[47]	Wavelet transform	RNN	artificial bee colony algorithm(ABC)	BP-ANN, conventional ANN optimized by the ABC algorithm, two conventional fuzzy time-series models	RMSE, MAE, MAPE, Theil'sinequality coefficient
[48]	Normalization + Kohonen SOM	FFANN	temporal BP	FFANN, Highly Granular Unsupervised time Lagged Network	Profit
[49]	-	FFANN	BP	Adaptive exponential smoothing	RMSE, Correct tendencies number
[50]	-	Exponential Smoothing +FFANN	BP	BPNN, Exponential Smoothing Forecast model	RMSE, DS
[51]	GARCH(1,1) Volatility	FFANN	Conjugate gradient based method	BS model with historical volatility ,BS model with GARCH(1,1), SV, SVJ	Average absolute and average squared errors
[52]	Proposed interval sampling method	FFANN	PCA+Meta-modeling	ARIMA,FNN,SVM, 4 meta-models	NRMSE, DS
[53]	Recurrent Self-Organizing Map + Wavelet transform	kernel partial least square regressions	-	ANN, SVMs, Generalized autoregressive conditional heteroskedasticity (GARCH)	RMSE
[54]	Chaos-based delay coordinate embedding	SVM	SVR	Pure SVR, Chaos-BPNN, BPNN	MSE, RMSE, MAE

Appendix B

This includes the first hundred orders of EUR/USD 2009. S/L indicates Stop Loss value and T/P indicates Take Profit level.

Table B.1-First 100 orders of EUR/USD trading for year 2009

#	Time	Type	Order	Size	Price	S / L	T / P	Profit	Balance
1	2009.01.05 20:01	sell	1	1.00	1.35959	0.00000	1.35851		
2	2009.01.05 20:02	t/p	1	1.00	1.35851	0.00000	1.35851	108.00	10108.00
3	2009.01.05 20:02	sell	2	1.00	1.35822	0.00000	1.35712		
4	2009.01.05 20:19	t/p	2	1.00	1.35712	0.00000	1.35712	110.00	10218.00
5	2009.01.12 03:05	sell	3	1.00	1.34207	0.00000	1.34055		
6	2009.01.12 03:29	t/p	3	1.00	1.34055	0.00000	1.34055	152.00	10370.00
7	2009.01.12 07:31	sell	4	1.00	1.34040	0.00000	1.33954		
8	2009.01.12 07:36	t/p	4	1.00	1.33954	0.00000	1.33954	86.00	10456.00
9	2009.01.13 20:16	sell	5	1.00	1.31653	0.00000	1.31617		
10	2009.01.14 13:10	close	5	1.00	1.31650	0.00000	1.31617	4.33	10460.33
11	2009.01.15 01:46	buy	6	1.00	1.31867	0.00000	1.31981		
12	2009.01.15 06:19	close	6	1.00	1.31877	0.00000	1.31981	10.00	10470.33
13	2009.01.16 05:46	buy	7	1.00	1.32276	0.00000	1.32277		
14	2009.01.16 05:46	t/p	7	1.00	1.32277	0.00000	1.32277	1.00	10471.33
15	2009.01.16 05:46	buy	8	1.00	1.32302	0.00000	1.32372		
16	2009.01.16 05:50	t/p	8	1.00	1.32372	0.00000	1.32372	70.00	10541.33
17	2009.01.21 00:15	sell	9	1.00	1.28729	0.00000	1.28665		
18	2009.01.21 00:19	t/p	9	1.00	1.28665	0.00000	1.28665	64.00	10605.33
19	2009.01.22 01:03	buy	10	1.00	1.29838	0.00000	1.29869		
20	2009.01.22 01:49	t/p	10	1.00	1.29869	0.00000	1.29869	31.00	10636.33
21	2009.01.22 01:49	buy	11	1.00	1.29907	0.00000	1.29984		
22	2009.01.22 01:50	t/p	11	1.00	1.29984	0.00000	1.29984	77.00	10713.33
23	2009.01.28 22:46	sell	12	1.00	1.31566	0.00000	1.31477		
24	2009.01.28 22:54	t/p	12	1.00	1.31477	0.00000	1.31477	89.00	10802.33
25	2009.02.02 20:15	buy	13	1.00	1.28410	0.00000	1.28609		
26	2009.02.03 00:53	close	13	1.00	1.28411	0.00000	1.28609	-0.73	10801.60
27	2009.02.04 21:04	sell	14	1.00	1.28468	0.00000	1.28340		
28	2009.02.04 21:17	t/p	14	1.00	1.28340	0.00000	1.28340	128.00	10929.60
29	2009.02.09 03:15	buy	15	1.00	1.29350	0.00000	1.29446		
30	2009.02.09 03:22	t/p	15	1.00	1.29446	0.00000	1.29446	96.00	11025.60
31	2009.02.10 22:18	sell	16	1.00	1.28935	0.00000	1.28762		
32	2009.02.11 00:39	close	16	1.00	1.28933	0.00000	1.28762	3.33	11028.93
33	2009.02.13 01:02	buy	17	1.00	1.28944	0.00000	1.29080		
34	2009.02.13 03:12	t/p	17	1.00	1.29080	0.00000	1.29080	136.00	11164.93
35	2009.02.13 11:02	sell	18	1.00	1.28858	0.00000	1.28742		
36	2009.02.13 11:08	t/p	18	1.00	1.28742	0.00000	1.28742	116.00	11280.93
37	2009.02.16 18:21	sell	19	1.00	1.27626	0.00000	1.27446		
38	2009.02.17 00:26	close	19	1.00	1.27620	0.00000	1.27446	7.33	11288.26
39	2009.02.18 02:19	buy	20	1.00	1.26034	0.00000	1.26228		
40	2009.02.18 06:22	close	20	1.00	1.26037	0.00000	1.26228	3.00	11291.26
41	2009.02.23 06:16	buy	21	1.00	1.28997	0.00000	1.29084		
42	2009.02.23 06:17	t/p	21	1.00	1.29084	0.00000	1.29084	87.00	11378.26
43	2009.02.23 06:17	buy	22	1.00	1.29113	0.00000	1.29264		

44	2009.02.23 06:33	t/p	22	1.00	1.29264	0.00000	1.29264	151.00	11529.26
45	2009.02.23 16:31	sell	23	1.00	1.27260	0.00000	1.27067		
46	2009.02.23 19:02	close	23	1.00	1.27258	0.00000	1.27067	2.00	11531.26
47	2009.02.24 02:48	sell	24	1.00	1.27039	0.00000	1.26849		
48	2009.02.24 15:17	close	24	1.00	1.27038	0.00000	1.26849	1.00	11532.26
49	2009.02.27 03:16	sell	25	1.00	1.27140	0.00000	1.27106		
50	2009.02.27 03:30	t/p	25	1.00	1.27106	0.00000	1.27106	34.00	11566.26
51	2009.03.02 07:32	sell	26	1.00	1.25796	0.00000	1.25795		
52	2009.03.02 07:32	t/p	26	1.00	1.25795	0.00000	1.25795	1.00	11567.26
53	2009.03.02 07:32	sell	27	1.00	1.25775	0.00000	1.25741		
54	2009.03.02 07:33	t/p	27	1.00	1.25741	0.00000	1.25741	34.00	11601.26
55	2009.03.04 04:19	sell	28	1.00	1.24870	0.00000	1.24789		
56	2009.03.05 14:02	close	28	1.00	1.24868	0.00000	1.24789	5.99	11607.25
57	2009.03.05 16:15	sell	29	1.00	1.25379	0.00000	1.25312		
58	2009.03.05 16:21	t/p	29	1.00	1.25312	0.00000	1.25312	67.00	11674.25
59	2009.03.09 17:46	buy	30	1.00	1.26364	0.00000	1.26378		
60	2009.03.09 17:50	t/p	30	1.00	1.26378	0.00000	1.26378	14.00	11688.25
61	2009.03.09 17:50	buy	31	1.00	1.26404	0.00000	1.26602		
62	2009.03.10 00:41	close	31	1.00	1.26410	0.00000	1.26602	4.27	11692.52
63	2009.03.10 07:47	buy	32	1.00	1.27135	0.00000	1.27209		
64	2009.03.10 07:59	t/p	32	1.00	1.27209	0.00000	1.27209	74.00	11766.52
65	2009.03.11 17:32	buy	33	1.00	1.27760	0.00000	1.27867		
66	2009.03.11 17:40	t/p	33	1.00	1.27867	0.00000	1.27867	107.00	11873.52
67	2009.03.12 00:15	buy	34	1.00	1.28140	0.00000	1.28207		
68	2009.03.12 00:19	t/p	34	1.00	1.28207	0.00000	1.28207	67.00	11940.52
69	2009.03.16 07:46	buy	35	1.00	1.29476	0.00000	1.29543		
70	2009.03.16 07:52	t/p	35	1.00	1.29543	0.00000	1.29543	67.00	12007.52
71	2009.03.17 20:01	buy	36	1.00	1.30216	0.00000	1.30251		
72	2009.03.17 22:18	close	36	1.00	1.30220	0.00000	1.30251	4.00	12011.52
73	2009.03.18 01:33	buy	37	1.00	1.30277	0.00000	1.30459		
74	2009.03.18 03:35	t/p	37	1.00	1.30459	0.00000	1.30459	182.00	12193.52
75	2009.03.19 21:00	buy	38	1.00	1.36682	0.00000	1.36713		
76	2009.03.19 21:55	t/p	38	1.00	1.36713	0.00000	1.36713	31.00	12224.52
77	2009.03.20 00:03	buy	39	1.00	1.36673	0.00000	1.36782		
78	2009.03.20 03:13	close	39	1.00	1.36674	0.00000	1.36782	1.00	12225.52
79	2009.03.20 13:00	sell	40	1.00	1.35753	0.00000	1.35593		
80	2009.03.20 13:10	t/p	40	1.00	1.35593	0.00000	1.35593	160.00	12385.52
81	2009.03.23 01:15	buy	41	1.00	1.36430	0.00000	1.36565		
82	2009.03.23 01:25	t/p	41	1.00	1.36565	0.00000	1.36565	135.00	12520.52
83	2009.03.23 05:16	buy	42	1.00	1.36693	0.00000	1.36713		
84	2009.03.23 05:41	t/p	42	1.00	1.36713	0.00000	1.36713	20.00	12540.52
85	2009.03.23 23:05	buy	43	1.00	1.36288	0.00000	1.36308		
86	2009.03.23 23:07	t/p	43	1.00	1.36308	0.00000	1.36308	20.00	12560.52
87	2009.03.23 23:07	buy	44	1.00	1.36349	0.00000	1.36361		
88	2009.03.23 23:12	t/p	44	1.00	1.36361	0.00000	1.36361	12.00	12572.52
89	2009.03.25 04:32	sell	45	1.00	1.34771	0.00000	1.34582		
90	2009.03.25 05:31	t/p	45	1.00	1.34582	0.00000	1.34582	189.00	12761.52
91	2009.03.26 07:17	buy	46	1.00	1.35876	0.00000	1.36008		
92	2009.03.26 11:20	close	46	1.00	1.35878	0.00000	1.36008	2.00	12763.52
93	2009.03.31 00:00	buy	47	1.00	1.31902	0.00000	1.32021		

94	2009.03.31 00:42	t/p	47	1.00	1.32021	0.00000	1.32021	119.00	12882.52
95	2009.04.01 17:47	sell	48	1.00	1.32041	0.00000	1.31936		
96	2009.04.01 18:12	t/p	48	1.00	1.31936	0.00000	1.31936	105.00	12987.52
97	2009.04.02 19:17	buy	49	1.00	1.34604	0.00000	1.34696		
98	2009.04.02 21:40	close	49	1.00	1.34608	0.00000	1.34696	4.00	12991.52
99	2009.04.06 02:31	buy	50	1.00	1.35677	0.00000	1.35813		
100	2009.04.06 04:45	close	50	1.00	1.35683	0.00000	1.35813	6.00	12997.52