Royal Docks Business School

UEL University of East London

MKM227 Postgraduate Dissertation

Student Number: U1221980

	Comments	Max Mark	Actual Mark
Introduction *Identification of a valid topic, research question and objectives framed to Masters Level standard with academic rationale developed, clear industry contextualisation of the research topic*	**Supervisor Comments:**	10%	
	2nd marker Comments:		
Critical Literature Review *Depth and breadth of literature search, engagement with seminal authors and papers, evidence of a critical approach toward the scholarly literature*	**Supervisor Comments:**	25%	
	2nd marker Comments:		

Research Methodology *Evaluation of research philosophies and perspectives. Justification of methodological approach, sampling strategy, data analysis and reliability and validity measures as applicable*	**Supervisor Comments:** **2nd marker Comments:**	**15%**	
Data Analysis and Interpretation *Evidence of rigor in data analysis and*	**Supervisor Comments:**	**35%**	

interpretation procedures, identification of key patterns and themes in the research data, integration of academic theory into explanation of findings	**2nd marker Comments:**		
Conclusions and Recommendations *Research question and objectives addressed with implications to theoretical and managerial concepts considered. Recommendations provided for theory, practice and future research*	**Supervisor Comments:**	**10%**	
	2nd marker Comments:		
	Supervisor Comments:	**5%**	

Organisation, presentation and references. *Well structured and ordered dissertation with correct use of grammar and syntax. In-text citation and bibliography conforming to "Cite Them Right"*	 **2nd marker Comments:**		
Total	**First Marker Total** **Second Marker Total**	**100%**	
Supervisor General Comments:		**Agreed Mark:**	

2nd Marker General Comments:	

Supervisor's Name: ……………………………………….. **Signature: …………………………**

2nd Marker's Name: ………………………………………. **Signature: …………………………**

The Forecast of Exchange Rates using Artificial Neural Networks, and their comparison to classic models

A dissertation submitted in partial fulfilment of the requirements of the Royal Docks Business School, University of East London for the degree of **MSc Financial Management**

May 2014

12,018

I declare that no material contained in the thesis has been used in any other submission for an academic award

Student Number:U1221980 Date: 13.04.2014

Dissertation Deposit Agreement

Libraries and Learning Services at UEL is compiling a collection of dissertations identified by academic staff as being of high quality. These dissertations will be included on ROAR the UEL Institutional Repository as examples for other students following the same courses in the future, and as a showcase of the best student work produced at UEL.

This Agreement details the permission we seek from you as the author to make your dissertation available. It allows UEL to add it to ROAR and make it available to others. You can choose whether you only want the dissertation seen by other students and staff at UEL ("Closed Access") or by everyone worldwide ("Open Access").

I DECLARE AS FOLLOWS:

- That I am the author and owner of the copyright in the Work and grant the University of East London a licence to make available the Work in digitised format through the Institutional Repository for the purposes of non-commercial research, private study, criticism, review and news reporting, illustration for teaching, and/or other educational purposes in electronic or print form
- That if my dissertation does include any substantial subsidiary material owned by third-party copyright holders, I have sought and obtained permission to include it in any version of my Work available in digital format via a stand-alone device or a communications network and that this permission encompasses the rights that I have granted to the University of East London.
- That I grant a non-exclusive licence to the University of East London and the user of the Work through this agreement. I retain all rights in the Work including my moral right to be identified as the author.
- That I agree for a relevant academic to nominate my Work for adding to ROAR if it meets their criteria for inclusion, but understand that only a few dissertations are selected.
- That if the repository administrators encounter problems with any digital file I supply, the administrators may change the format of the file. I also agree that the Institutional Repository administrators may, without changing content, migrate the Work to any medium or format for the purpose of future preservation and accessibility.
- That I have exercised reasonable care to ensure that the Work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.
- That I understand that the University of East London does not have any obligation to take legal action on behalf of myself, or other rights holders, in the event of infringement of intellectual property rights, breach of contract or of any other right, in the Work.

I FURTHER DECLARE:

- That I can choose to declare my Work "Open Access", available to anyone worldwide using ROAR without barriers and that files will also be available to automated agents, and may be searched and copied by text mining and plagiarism detection software.
- That if I do not choose the Open Access option, the Work will only be available for use by accredited UEL staff and students for a limited period of time.

/cont

Dissertation Details

Field Name	**Details to complete**
Title of thesis *Full title, including any subtitle*	The Forecast of Exchange Rates using Artificial Neural Networks, and their comparison to classic models
Author *Separate the surname (family name) from the forenames, given names or initials with a comma, e.g. Smith, Andrew J.*	RAFFA, Mohamed Karim
Supervisor(s)/advisor *Format as for author.*	Dr. Karathanasopoulos, Andreas
Author Affiliation *Name of school where you were based*	Royal Docks Business School UEL
Qualification name *E.g. MA, MSc, MRes, PGDip*	MSc
Course Title *The title of the course e.g.*	MSc Financial Management
Date of Dissertation *Date submitted in format: YYYY-MM*	2014-05
Do you want to make the dissertation Open Access (on the public web) or Closed Access (for UEL users only)?	Open ■ Closed □

By returning this form electronically from a recognised UEL email address or UEL network system, I grant UEL the deposit agreement detailed above. I understand inclusion on and removal from ROAR is at UEL's discretion.

Name: Mohamed Karim RAFFA

Signature: .... Date: 13.05.2014

ROYAL DOCKS BUSINESS SCHOOL
UEL

The Forecast of Exchange Rates using Artificial Neural Networks, and their comparison to classic models

Post-Graduate Dissertation

U1221980

12018 words

Due:13.05.2014

ABSTRACT

Predicting Foreign Exchange rates has forever been a task of great importance to any individual, business or organization having to deal with a foreign currency. In the wake of a world where global transactions are an everyday activity, readiness and skill when dealing with the forecasting of international monetary movements is a key factor in the success of any operation; be it that of an individual investor, or that of multi-national index listed company. The motivation behind the desire of conquering the skill of forecasting may range from the simple desire to hedge one's investments and dealings in a foreign currency, to that of a speculative investor, looking for arbitrage opportunities in trading foreign exchange markets.

This paper had for motivation to test and compare various models in their ability to forecast the return generated by price movements of three globally available and traded currencies; notable the Euro – US Dollar, the Euro-Swiss Franc and the Pound Sterling – US Dollar. Recent studies have been showing great promise in the use of Artificial Neural Networks in the field of forecasting exchange traded assets and currencies; which is why this paper has discussed the performance of 4 Learning Machine models in comparison to 3 base models and 2 linear models. The learning machine models being studied are the Multi-Layer Perceptron, the Higher Order Neural Network, Gene Expression and Rolling Genetic-Support Vector Regression. These models were compared using various methods of statistical evaluation, in order to measure the discrepancy of the forecasted values from the actual values, as well as the annualized return and the risk to return ratio.

It was concluded that modern forecasting technique do outweigh the classic base and linear models in terms of forecasting accuracy as well as potential gain and risk to return.

ACKNOWLEDGEMENTS

I would first of all like to thank Dr. Andreas Karathanasopoulos my dissertation advisor, for his dedicated help, advice and guidance in the writing of this paper.

I would also like to thank my parents Dr. Ash Raffa and Irene Raffa, for their love and everlasting support through years of study, which have finally brought me to the completion of this paper.

And finally I would like to generally thank all friends and family member who have been of great help and a source of motivation in keeping me on track and motivated.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1.0 INTRODUCTION

Forecasting foreign exchange rates remains one of the most important tasks in the globalized world, yet one of the most obscure and uncertain. Its importance isn't limited to the speculator, but affects every industry and sector with any international exposure; in a day and age where globalization is no longer an abstract concept, but a part of daily business, it is safe to say that foreign exchange is at the heart of modern markets.

Global foreign Exchange trading accounts for the highest daily trading volume; with an estimated average of $1.5 trillion. This largely being consisted of spot deals, making it deep, very liquid and most importantly very volatile. Of all currency pairs traded this research paper has for aim to forecast the future fluctuations of the EUR/USD; USD/GBP and the EUR/CHF.

The main goal of this paper is to compare various models, more importantly popular linear models and artificial intelligence models, in their ability to accurately forecast foreign exchange markets. These will then be inputted into a trading strategy, and be individually evaluated based on popular ratios, overall return and general efficiency.

It is important to note, that it isn't the actual Fx values that will be forecasted in this paper, but rather the return. As past literature will confirm, the return gives a "normal" base to the calculations, whereas the actual values, being only a sample population, don't have an actual value to "return to."

There are well known factors that affect foreign exchange rates including economic factors such as inflation, growth and interest rates, as well as other political and economic news. The weight at which these factors influence the market are unknown, and vary depending on a multitude of external factors, including but not limited to each traders personal perception, this has led to great skepticism in the ability to predict Fx market with any precision, this is upheld by the Efficient Market Hypothesis; if this were to hold, the market would perfectly reflect all information available, and adjust itself to change too quickly for investors to profit. In essence, prediction becomes useless, as the potential for arbitrage evaporates. EMH is greatly criticized, mostly by investors and traders who have managed to beat the market over a certain length of time, this is mostly attributed to the fact that in "real life markets", the change isn't quite as immediate, leaving a slight window where profitable trading opportunities remain.

The motivation behind forecasting exchange rates is vast, not only for the sake of arbitrage opportunities, but fund managers, traders, importers, exporters, all see the prices of goods and services vary and fluctuate if they do not hedge their exposure efficiently.

Predicting Fx has always been challenging, mainly due to the constantly varying influential factor, with no given weight or importance to any known or unpredicted event. Linear models are greatly available, and have been developed on a multitude of bases, but the apparent nonlinearity of Fx time series complicates these, and results in inconsistent and unreliable results. Facing these problems, non-linear models have been developed, such as Artificial Intelligence models, learning machines and Artificial Neural Networks.

As previously mentioned, Linear Models have the disadvantage of being applicable only with stationary data; with the ever varying time series, a seemingly non-stationary data, linear models tend to provide mixed results when varying the analyzed time frame. This paper will discuss some of the most common linear models used in trading; notably the Auto-Regressive Moving Average (ARMA), Moving Average Convergence-Divergence (MACD), Random Walk and Buy and Hold. The trading strategy implemented based on the calculated forecast will then deliver the benchmark returns; meaning that they will be used in order to compare the efficiency of the learning machine models to common linear models.

Artificial Neural Networks (ANNs) are mathematical models, based on the brains nervous system, and the method of computing various inputs into delivering a desired output.

In practice, Artificial Neural Networks have shown great promise in forecasting foreign exchange rates; Refenes (1992), Weigend (1991) and Zhang (1994) have all developed ANN models that resulted in accurate and profitable forecasting

The Neural Networks being implemented in this paper will be the Multi-layer Perceptron (MLP), and the Higher Order Neural Network (HONN), both of these models have shown promise in the field of foreign exchange forecasting when compared to linear models.

In more recent literature, classic Neural Networks, though effective, have been trumped by learning machines originally developed for categorization problems, notably Support Vector Machines (SVMs). Based on support vector regression and originally developed to solve categorization problems, they have recently been applied more frequently in applied trading. As well as the rise in the use of Genetic Algorithms in the field of forecasting have inspired this paper to compare these most common models, in order to determine the overall profitability of each type of model, as well as their overall risk adjust performance in identical scenarios.

Once all models carried out and statistically evaluated using basic investment evaluation tools, such as the Sharpe ratio, annualized return and error measure; the paper will result in answering the question of which model best outperforms the others, as well as if there is a flaw in the Efficient Market Hypothesis.

2.0 LITERATURE REVIEW

When it comes to economics and financial markets, forecasting and predicting is the main goal of any related institution; be it to give an edge or to adequately prepare for change, accuracy in this field is crucial in the decision making process. Forecasting foreign exchange markets remains a very difficult task, as these behave in a non-linear, chaotic and dynamic fashion (Dunis et al., 2011).

Previous methods such as ARMA, MACD and Naïve strategy have proven to work, yet remain inefficient. These models have been compared and contrasted to artificial neural networks the likes of Ghiassi et al. (2005) measuring the performance of ARMA models against that of ANNs.

The main contender for linear models remains Random walk, notably a fully random succession of steps, first introduced by Karl Pearson (1905). Using Random Walk as a benchmark to other forecast models simultaneously tests the Efficient Market Hypothesis, as originally proposed by Eugene Fama (1965) states that it is impossible to build a model that will outperform the market. Though if the models suggested were to beat the random walk benchmark, it will challenge the Efficient Market Hypothesis. It is argued that random walk outperforms most linear models, as underlined by Messe and Rogoff (1983) after testing various forecasting models against the Random Walk with drift, and find that regardless of which model used, none outperform the random walk model. Karemera and Kim (2006) developed an Auto-Regressive Integrated Moving Average model (ARIMA) that outperformed Random Walk while forecasting currency exchange rates on a monthly closing basis. While carrying out the same model on other time frames, it was found that the success of the model is strongly tied to the time frame on which it is applied. Cheung (1993) detected long run memory in exchange rates, and suggested an Auto Regressive Fractionally Integrated Moving Average model, though with little success.

One of the major disadvantages of Linear models remains their need for consistent statistical measures, notably stationarity. Fx time series seem to be all but stationary; ANNs have the advantage of not requiring this type of data set, but will work without limitation to the type of input. (Kondratenko et al. 2003)

The intense interest on both an economic and financial point of view has attracted various researchers in a multitude of scientific areas. Leading to diverse suggested models for forecasting exchange rates, including a variety of different inputs. Economic indicators and news releases used as inputs in the Generalized Auto Regressive Heteroskedasticity model (GARCH) as well as further development in learning machines, brought in from algorithms modeled according to biological functions, lead to the development of Artificial Neural Networks in the field of exchange rate forecasting. (Semprinis et al., 2013)

2.1 HISTORY OF ANNS

Artificial Neural Networks (ANNs) were derived from brain learning process in biology; forming a mathematical model of the biological nervous system (Abraham, 2005). McCulloch and Pitts (1943) created the first computational model based on mathematics and algorithms, this model at the time referred to as “threshold logic” was a stepping stone for all future neural network models. The concept of learning machine was then introduced by Donald Hebb (1949) applying the basic concept of unsupervised learning. The previously mentioned Frank Rosenblatt (1958) developed the perceptron, a pattern recognition algorithm and the basis for the MLP model. Minsky and Papert (1969) discovered two main issues in the field of neural networks. Primarily, single-layer networks weren’t developed enough to handle and process “exclusive-or” circuits. Secondly, the computational power in the 60s was limited, and usable application of models required more than was available at the time. Werbos (1975) goes into detail of how the back-propagation algorithm was modified and restructured in order to fix the exclusive-or issue, yet major development in the field of neural networks stalled till the computational development caught up with requirements.

Since then, ANNs have been a powerful statistical modeling tool, used to detect the underlying function in the relationship within a data set (Lawrence and Andriola, 1992) the strength of ANNs in predicting future variations based on past data made it a valuable tool in in modeling the movement in the prices of financial instruments and assets. They have proved to be more performing than statistical regression models (Refenes, 1994) as well as discriminant analysis (Yoon, 1993) According to Franses and Grievensen (1998) in their attempt to forecast Foreign exchange rates using Neural Networks for technical trading; they emphasize the advantage of neural networks in adapting and exploiting non-linearity.

After 2009, competitions have been set up in the field of pattern recognition, mainly for practical application as in handwriting recognition and traffic sign recognition, though not applicable in this paper, connections can be made.

In the spirit of past literature we chose to adopt a similar forecast model, in the sense that predictions will be made on a “one day ahead” basis, as suggested by Rime et al. (2010). The reason for this duration of forecast are primarily, because they are implementable, secondly the data is precise and largely available, and finally because in practice traders do use daily forecasts on a frequent basis.

To introduce Neural Networks in more detail; the base of a Network is a neuron, very similar to the real neuron found in the central nervous system, the formal neuron works in three parts: the integration step, where the input variables (x_i) are weighted with the parameters $w_{i,j}$. Followed by the non-linear step during which the weighted sum of the inputs is transformed into a non-linear function relevant to the type of network; and finally a propagation step, where the output is propagated to the following neurons. The output of a single formal neuron is seen as:

$$y = f\left(\sum_{j=0}^{n-1} w_{i,j}x_j - \theta_i\right)$$

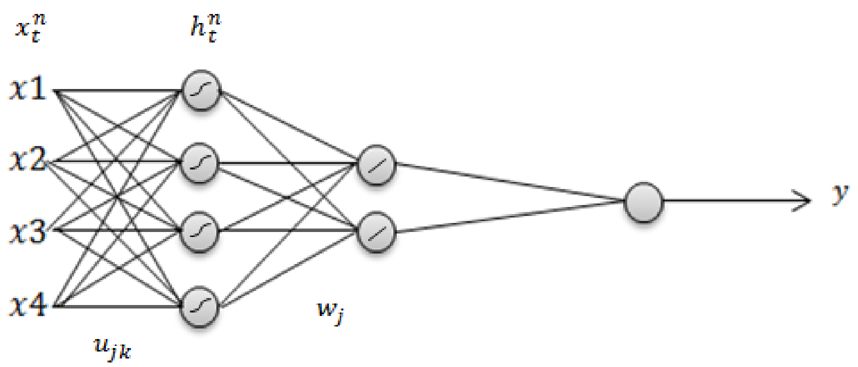
y being the output, $f(x)$ a non-linear function n the number of input lines of neuron i and θ_i being the threshold.

ANNs have the advantage of not requiring stationary statistical measure, as they are optimized by testing mathematical criterion to match past data with future data over time. The motivation of this paper is to compare trading strategies, once used with traditional linear models, and another using types of Artificial Neural Networks. Researchers have been for some time advocating the use and promoting the higher accuracy in forecasting of ANNs; often in other fields than financial, but criticism arises when implementing a trading technique based on these forecasts.

2.2 MULTI-LAYER PERCEPTRONS

Of the various different types of ANNs available, one of the most applicable, according to Yao et al. (1999) remains the sophisticated simplicity of the Multi-Layer Perceptron (MLP). The Perceptron was developed by Frank Rosenblatt in 1957 (Need source), which at the time consisted of one single layer, and was later extended. The MLP can be easily visualized as follow:

FIGURE 1.



Where:

$x_t^n(n = 1,2, \dots, n+1)$ are the model inputs (after the application of the bias node) at a given time t. In this paper, the inputs are the lags in end of day values.

$h_t^n(n = 1,2, \dots, n+1)$ are the hidden nodes (after the application of the bias node) and represent the MLP's output values.

y represents the MLP model output

u_{jk} and $w_j$are the network weights

 Is the transfer sigmoid function $S(x) = \frac{1}{1+e^{-x}}$

 Is a linear function $F(x) = \sum_i x_i$

The goal of the model being to minimize the following error function:

$$E(u_{jk}, w_j) = \frac{1}{T}\sum\left(\ddot{y} - y(u_{jk}, w_j)\right)^2$$

Where $\ddot{y}$ represents the target value.

In the MLP above, each  represents a Neuron; the vertical lines in which these are ordered are layers, the x values are inputs, and y is the output. The connecting lines are the weights w_j.

The previously sited Yao (1999) used this system to predict the USD/CHF exchange rate; using the average of the past 6 weeks closing prices as inputs, his particular MLP had 6 inputs. A very similar approach will be taken in this paper when working with MLPs.

Numerable research has been conducted in the forecasting of time series data using MLPs, though not all successful, Castiglione (2001) has achieved positive results with forecast accuracy of just over 50%. This forecast was carried out on 3 of the major US Indices, notably the S&P500, Nasdaq 100 and the Dow Jones industrial Average. The part failure here may be attributed to the sensitivity to tierce information such as the news or other political incident, which the MLP may not have been trained to use as input. The issue of input selection was attempted to be corrected in the method carried out by Tino et al (2001) where binary signals were inputted into a Recurrent Neural Network (RNN) in order to forecast the DAX and FTSE100. Though the results were disappointing and the author concluded neural networks were no more efficient at forecasting markets than classic models.

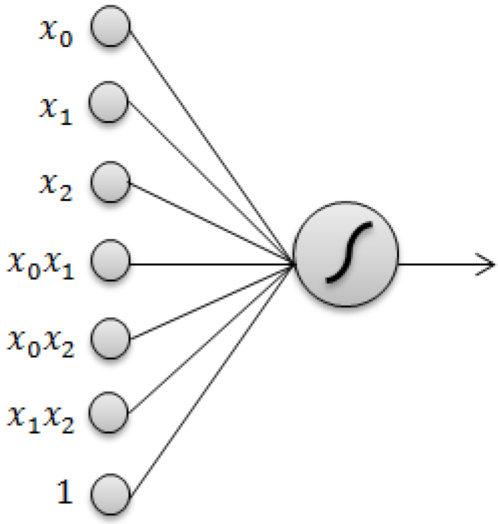
Yao et al (2000) carried out a study more similar to this paper, notably attempting to forecast major currency pairs: AUD, DEM, CHF, GBP and JPY. Two types of data were used in order to attempt forecasting, both using MLP models, but varying on the type of input. First the inputs were exchange rate values with an implemented time lag, and secondly the MLP was fed moving average values of different time values. Both of these were carried out on a weekly basis, meaning one forecast was given for every coming week. Both methods here resulted in good accuracy, essentially showing that artificial neural networks are a viable method for forecasting, though the actual application is limited. The weekly time span reduced the practical use to traders and institutions willing to place more than one trade a week. (V.Kondratenko et al. 2003)

2.3 HIGHER ORDER NEURAL NETWORKS

HONNs were first introduced by Giles and Maxwell in 1987 under the name of Tensor Networks. Similar to Single Layer perceptrons, Higher Order Neural Networks (HONNs) have a structure of neurons with inputs and outputs going through hidden layers; the key difference as explained by A. Ismail (2001) is the way the inputs are combined. As confirmed by Knowels et al (2005) this leads to shorter computational time, but also limits the number of input variables. Zhang et al.(2002) praises HONNs in their ability to "provide superior simulations" when compared to other ANNs, which is underlined by Knowels et al.'s (2005) 8% average increased return compared to MLPs when attempting to forecast the EUR/USD currency pair.

In the similar spirit used when illustrating the MLP, below an illustration of a 3 variable input HONN:

FIGURE 2.



Where is the sigmoid activation function: $S(x) = \frac{1}{(1+e^{-x})}$

As illustrated, the need to establish the relationship between inputs is redundant, the joint activation function also reduced the number of free weights, reducing the training time required in comparison to MLP. This though complicates the use of HONNs with a higher number of inputs, making orders of over 4 rare. Knowels et al. (2005) emphasizes the common issue with MLPs of overfittng and local optima are here greatly avoided.

2.4 SUPPORT VECTOR MACHINES AND SUPPORT VECTOR REGRESSION

Support Vector Regression being an extension of the Support Vector Machine Algorithm, which has been seen to have the ability to forecast exchange rates (Ince and Trafalis, 2005 & Brandl et al. 2009.)

Pattern recognition being heavily discussed in forecasting, brings up Support Vector Machines and Support Vector Regression; primarily used to solve problems within categorizing and classifying pattern recognition frameworks (Vapnik, 1995). In comparison to Neural Networks these also performed well. To apply categorization to variables as volatile and unpredictable as Foreign exchange rates, simplification is necessary. As explained by Zinzalian et al. (2009) determining whether a price will rise or fall can be written out as a simple binary classification problem, with *1* for rise and *0* for fall. Support Vector Regression or Support Vector Machines have gained popularity in the field of forecasting in recent years, though "despite evidence on the non-linear properties of foreign exchange markets" only mixed results have been found when attempting to forecast the price or return of currency pairs (Bahramy et al, 2013).

As described by P. Dharnidharka (2012), the principle behind SVR in a decision determining scenario is fairly simple. Support Vector Machines rely on a concept of decision planes, which define decision boundaries. It separates objects of different class membership from one another. More easily described by illustration, a graph is shown below; on which the objects on the right side of the plane are hollow circles, whereas the objects on the left side of the plane are full.

As described by Plakandaras et al.(2013) Represented in mathematical notation, from the training data set

$$D = [(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)]$$

$$x_i \in R^m$$

$$y_i \in R$$

$$i = 1,2, \ldots n$$

x_i being the observation samples and y_i the dependent variable (the target of the regression system)

The method being to minimize the loss function $\min(\frac{1}{2} \parallel w \parallel^2)$ subject to

$$|y_i - (w^T x_i + b)| \leq \epsilon$$

where

$$i = 1, 2, \ldots n$$

Hereby enforcing and upper deviation threshold and forming an "error tolerance" band.

The previously cited Vapnick et al.(1995) proposed a model which allowed for data exterior to the tolerance zone, though this degraded them proportionally to the edge of the zone; which they fixed by introducing ζ_i and ζ_i^* as slack variables to the function, controlled by a cost parameter C; resulting in the following loss function:

$$\min(\frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{n} (\zeta_i + \zeta_i^*))$$

which yielded the solution:

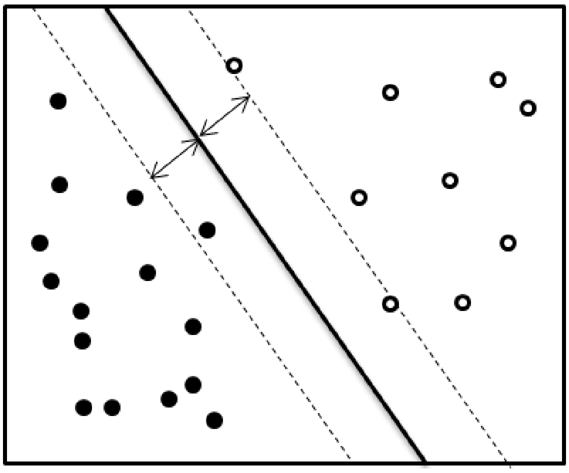
$$w = \sum_{i-1}^{n} (a_i - a_i^*)\, x_i$$

And

$$w = \sum_{i-1}^{n} (a_i - a_i^*)\, x_i^T x$$

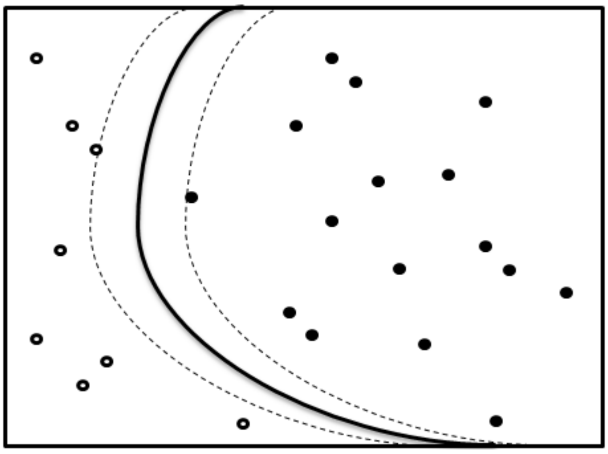
Not all phenomena can be described as linear, therefore the data is initially mapped into a higher dimensional space, where such a function is feasible:

FIGURE 3.



P. Dharnidharka (2012) goes on to explain that this model being solely a "linear classifier", problems tend to be more complex, especially in dealing with data such as Fx time series. More complex structures such as illustrated below tend to be more effective at classifying data sets. Once again the separation is between hollow and full.

FIGURE 4.



The complex curve above is mapped using kernels; mathematical functions used to rearrange and "map" objects.

These kernel functions transform the data into "dots on a map", which is vital to find the relations in the data, hence finding the hyper-plane. The gives the kernel function a basic and critical role in the SVM and its overall performance.

The training set is linearly separable in the feature set, this is called a "kernel trick"

$$K(x, x') = (\phi(x), \phi(x')$$

Some of the more popular kernel functions in Support Vector Regression are:

Linear Kernel function: $K(x_i, x) = x_i^T x$

Polynomial kernel function: $K(x_i, x) = (x_i^T x + 1)^d$

Radial Basis kernel function:$K(x_i, x) = \exp(-\| x - x_i \|_2^2/\sigma^2)$

Sigmoid (MLP): $K(x_1, x) = tanh(\gamma x_1^T x + r)$

Factors d, r, γ represent kernel parameters

As emphasized by Sermpinis et al. (2013) the selection of the kernel parameters is of high importance to the success and accuracy of the SVR. The complexity of the selection of these parameters may lead to a hindrance in the design of such a model. The summary of the selection can be listed in the following way:

1. Kernel function selection
2. Regularization parameter selection
3. Kernel function parameter selection
4. Selection of the size of the error tolerance band

This simplified version of the steps involved in the selection may be more demanding; as the individual optimization of the parameters may not suffice, and the overall performance of the SVR spends on the totality of parameters being set optimally. A variety of approaches have been suggested, such as setting the ε value to a very small constant, as suggested by Trafalis and Ince (2000). A v-SVR approach is also very common, as the ε parameter is more easily controlled using the v parameter. (Scholkopf et al. 1999 and Basak et al. 2007). Though the most popular approach in parameter selection remains the use of a cross-validation technique as applied by Cao et al. (2003) and

Duan et al. (2003), or the use of a grid search algorithm. (Schlokopf and Smola, 2002)

According to Ulrich et al. (2007) as well as Yu et al.(date needed) when compared to other learning algorithms, SVR show great promise in outperforming their more common ANNs. SVRs have the outstanding advantage, of providing a “global and unique solution” without suffering from multiple local minima. In addition they hold the ability of finding balance between model accuracy and model complexity(Kwon and Moon (2007) and Suykens (2005).

This has led to advanced research in Hybrid Support Vector Machines, to which Bahramy et al (2013) and Tan & Wang (2004) narrowed down the success of SVR, to feature and kernel selection. With this in mind, combining Genetic Algorithms with SVMs (GA-SVM) as demonstrated by Bahramy et al. (2013) lead to the optimization of feature and kernel selection through Genetic Algorithms. In this spirit and to verify these claims a rolling Genetic Algorithm- Support Vector Regression Model (GA-SVR) will be tested, using the same input and variables as the other models in order to maintain consistency.

Lee et al (2004) suggest multi-category SVM, built on the base of traditional binary SVM. They carried out their model in two different case studies, both yielding promising results. They make note that their methodology is “a useful addition to the class of nonparametric multi-category classification methods.”

This was later developed further by Liu and Shen (2006) presenting a multi-category ψ-learning methodology, suggesting a non-convex ψ-loss function, reducing the number of support vectors and yielding a more spars solution. Huang et al. (2010 forecasted various currency pairs; all with high liquidity using hybrid chaos-based SVR. when compared to chaos-based neural networks, higher yields were confirmed, as well as outperforming other non-linear based models.

Feature selection is an optimization problem, it involves the search of a feature subset within a space of possibilities in order to find the optimal subsets with respect to the criteria. this process requires a strong search strategy which picks the adequate feature subset and subsequently tests their goodness of fit. Various strategies have been suggested in past literature, though those having showed the most promising and attracting the most attention seem to be the strategies that employ randomized searches in which probabilistic steps are applied (Sun et al., 2004). Genetic Algorithms are most commonly used in such cases (Siedlecki and Sklansky, 1989)

2.5 GENETIC PROGRAMMING AND GENE EXPRESSION

Genetic Programming is a method used to automatically generate programs designed to fulfill specific tasks (Koza, 1992). These Genetic programs use Genetic Algorithms; as previously mentioned, Genetic Algorithms are search algorithms. Having been inspired by the principal of natural selection, they are commonly used if the search space is vast and complicated or if no mathematical analysis is available. In this model, the candidate solutions are referred to as chromosomes, the population of chromosomes is optimized through a number of evolutionary cycles and genetic operations, such as mutations or crossovers. Each Chromosome consists of genes, these genes being the optimization parameters. Each iteration/generation undergoes a fitness function, which evaluated the chromosomes by measuring the precision of the corresponding solution; the “fittest” chromosome is then selected to survive. This evolutionary process is then repeated till criteria are met that will terminate the cycles. GAs have the advantage of being able to deal with large search spaces, without getting trapped in local optimal solutions. (Holland 1995)

Montana (2002) carries out genetic programming and goes through lengths in order to explain the processes. As suggested by Monatana (2002) the programs are usually represented as parse trees. This illustration is used to visually describe the program itself, as well as the mutation and crossover evolutionary processes. The following being a representation of the base program:

FIGURE 5.

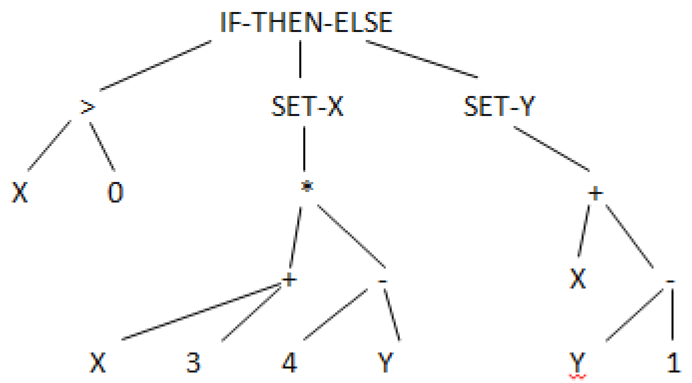
if $x > 0$ then

$x := (x + 3) * (4 - y)$

else

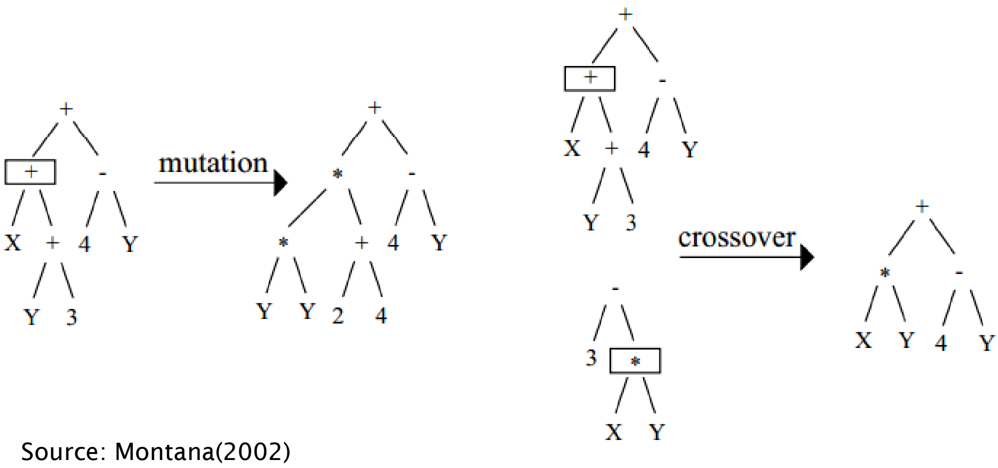
$y := x + y - 1$

end if



Similarly, the illustration of crossovers and mutations using the parse tree structure:

FIGURE 6



Source: Montana(2002)

Illustrations explained:

The crossover follows a series of steps; starting with the random selection of a node within each tree as a crossover point (in the figure the selected node is boxed); secondly the sub-tree rooted at the previously selected node is to be selected in the second parent, and used to replace the sub tree stemming from the node of the first parent in order to generate a "child". Finally determine whether the child is suitable with the limitations set on the program.

Mutation follows a similar set of steps; firstly a random node is selected within the parent tree which will serve as the mutation point. A new tree is then generated, which complies with the parameter set up by the program. Thirdly, the newly generated tree is attached to the parent tree at the selected node; and finally the newly generated tree is checked for program compliance. If all conditions are met the tree is used. (Monatana, 2002)

A very similar approach was used by Allen and Karjalainen (1999) who have chosen to use a Genetic program to learn trading rules applied to the S&P 500 ranging between 1928 and 1995. They have found successful trading results; though once the trading and transaction costs applied, the model didn't earn a consistent return higher than that of a simple buy and hold model. They found that their model showed more accurate results than their benchmark methods when daily returns were positive, and the volatility was low.

2.6 HYBRID GA-SVR MODEL

The combination of the past two presented models yields the Hybrid GA-SVR model to be carried out in this paper, with optimal parameter selection and feature subset combination. Support Vector Regression is favored over classical SVM when applied to end of day forecasts, as SVMs only produce a binary output, whereas in this scenario an exact forecast provides more applicable trading options. Sermipins et al. (2012) suggest applying sophisticated trading strategies to their models in order to better evaluate their performance. Though this is not in the scope of tis paper.

2.7 CONS OF ANNS

A typical criticism of ANNs remains the lack of qualitative input; with past stock price movements being the only variables taken into account. Yoon (1991) suggested and attempted to input qualitative data as well as quantitative data including news reports and other text data. A similar attempt was carried out by Wuthrich (1999).

Although this paper will mainly focus on using end of day closing prices, transaction volumes, interest rates, rates of change and other similar data sources are also commonly used as input for Neural Networks (Saad, 1998 & Cao, 2003).

Increasing the efficiency and accuracy of Neural Networks is key; this is achieved by modifying weights (Yao, 1997). Back-propagation has been the simplest and most common way to develop these efficient models, yet their simplicity leads to slow and not always maximized results. During supervised training of ANNs, genetic algorithms (GAs) were suggested as best method to rearrange and enhance the topology and weights. GAs have been combined to ANNs in order to successfully obtain an efficient model as explained and tested by: Kai (1997), Harrald (1997) & Hayward (2004).

2.8 TRADING APPLICATION

With the aim of this paper being forecasting stock price movements, the ultimate goal here is to produce a model yielding a return exceeding the benchmark methods (Random Walk, Naïve Strategy, MACD and ARMA). When applying ANNs in practice, it has been proven to yield higher returns; though Alexander (1964) has found that if taking into account transaction cost, the return withers and no longer outperforms benchmark returns. His results were concurred by multiple others such as Fama (1970) and Blume (1966) who have tested these models on the Dow 30 Index. Allen (1999), Reay (2002) and Neely (2003) attempted similar models, setting up trading rules in order to maximize profits, yet have all gotten beat by the transaction cost, which lead their results to equalize with Benchmark techniques or fall below them. Nonetheless, some successful studies are present, with models drawn up by Sweeney (1988), as well as a Moving average trading strategy by Bossembinder and Chan (1995) applied to Asian emerging markets. These though emphasize the importance and sensitivity of transaction costs.

2.9 THEORETICAL BACKGROUND

EUR/USD, GBP/USD and EUR/CHF Exchange rates

The importance of Exchange rates on a daily exchange basis cannot be neglected. Major industries, countries and all internationally dealing sectors are affected by the value of their currency with respect to another; this makes the volume of daily trades so large. The applicability of the data in real life trading scenario is also very present; with brokers offering the possibility of opening orders for end of day expiry, options, and even binary options, where a simple “over or under” method is applied.

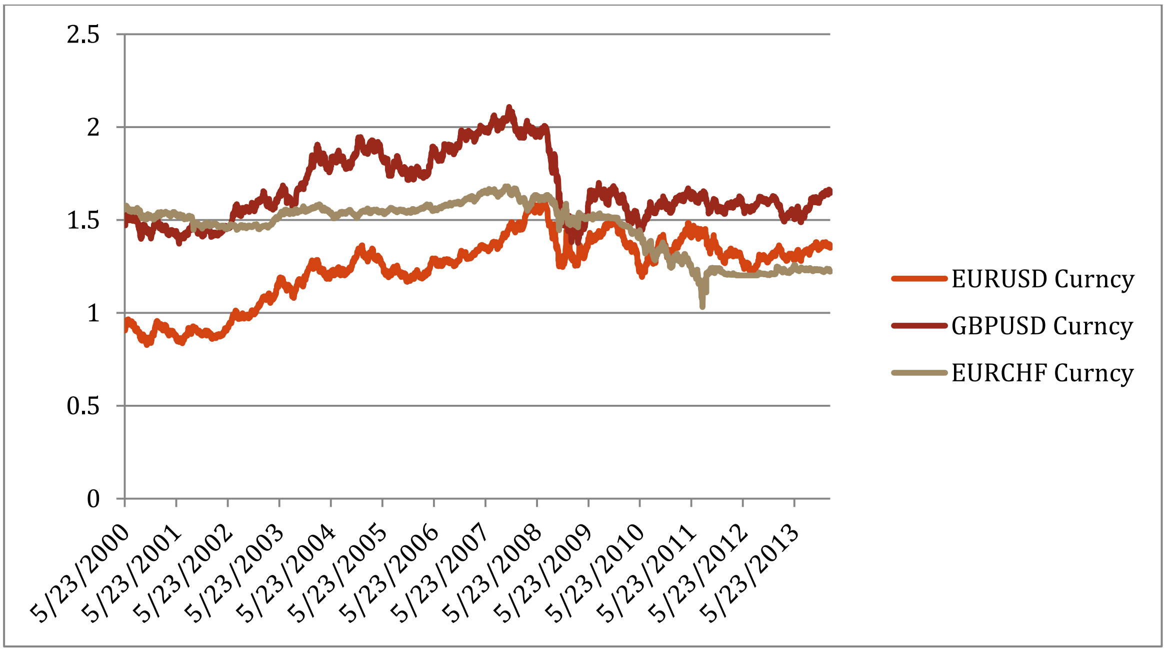
The availability of precise data regarding Exchange rates is greatly available. Though with delicate data sets at hand and precision being a well-regarded factor, importance was set on the origin and more importantly the consistency of data collection, Bloomberg provides a service making historical data sets easily available. These were all imported into workable data sheets on the same day, and treated as a single block unit, before being separated into in sample and out of sample sets. End of day closing prices of the EUR/USD, GBP/USD and EUR/CHF from the 23/05/2000 to the 31/01/2014 were exported, resulting in a total of 3574 data points per currency pair. Demonstrated in the chart below:

Table 1.

Periods	**Trading Days**	**Start date**	**End date**
Total Data set	3564	06/06/2000	31/01/2014
In Sample	2588	06/06/2000	06/05/2010
Out of Sample	976	07/05/2010	31/01/2014

The following Graph shows the total data set plotted over the full period for all three currency pairs:

Figure 7.

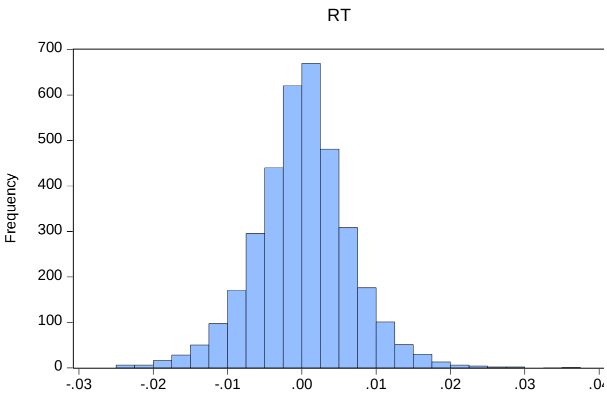


the Aim of this paper is to forecast the Return ($E(R_t)$) of each exchange rate on day ahead, in order to place a trade accordingly.

Below are the histogram distributions as well as basic data relating to the returns (RT) produced by the previously listed exchange rates.

FIGURE 8.

EURUSD distribution



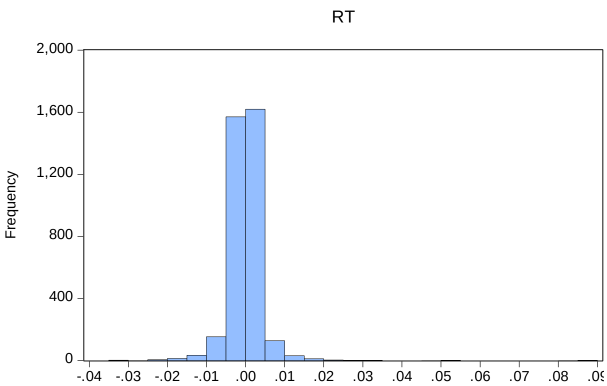
Observations 3573

Mean	0.000132
Median	0.000163
Maximum	0.035262
Minimum	-0.024903
Std. Dev.	0.006427
Skewness	0.060581
Kurtosis	4.327117
Jarque-Bera	264.3902
Probability	0.000000

Series: RT
Sample 5/23/2000 1/31/2014

FIGURE 9.

GBPUSD distribution



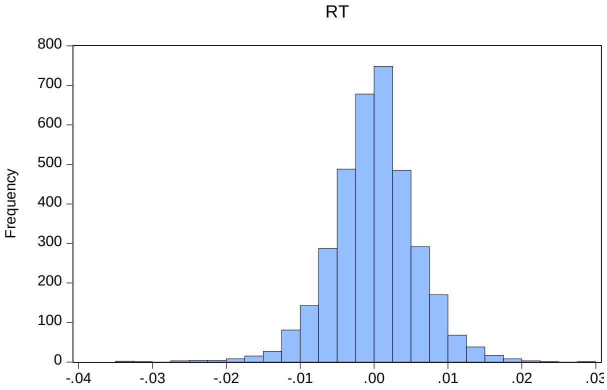
Observations 3573

Mean	-6.01e-05
Median	0.000000
Maximum	0.087536
Minimum	-0.031909
Std. Dev.	0.004121
Skewness	3.083893
Kurtosis	74.43800
Jarque-Bera	765430.3
Probability	0.000000

Series: RT
Sample 5/23/2000 1/31/2014

FIGURE 10.

EURCHF Distribution



Sample 5/23/2000 1/31/2014	
Observations 3573	
Mean	4.65e-05
Median	0.000104
Maximum	0.029684
Minimum	-0.034119
Std. Dev.	0.005739
Skewness	-0.245859
Kurtosis	5.243115
Jarque-Bera	785.0704
Probability	0.000000

Series: RT

The graphs above tend to show a bell shaped distribution, with results converging around a mean value; though according to the Jarque-Bera statistic, the returns are non-normally distributed to 99% confidence interval. Both the EURUSD and the EURCHF have a fairly high kurtosis, and slight skewness. The high level of kurtosis found with the GDPUSD is paired with the high positive skewness.

3.0 METHODOLOGY

To start the 3 currency pairs' data sets will be collected from a reliable source; in this scenario Bloomberg will be used to extract the 3576 historical end of day values of each currency, they are then separated into in sample and out of sample; here the in sample set will have 2588 values, and the out of sample will contain 976 end of day values. As it isn't the actual Fx values that will be forecasted, but rather the return, once the data collected and adequately set up in excel, the daily returns will be calculated using the formula:

$$\mathrm{Rt} = \frac{\mathrm{P_t} - \mathrm{P_{t-1}}}{\mathrm{P_{t-1}}}$$

The first step will be testing the Null Hypothesis, also known as H0; the objective here will be to determine whether or not the data is stationary. If non-stationary, the data sets will be completely uncorrelated, and follow a random walk type distribution, making all other models unreliable. Even if the data were to result in great returns, this would be inconsistent over time.

Stationary data points however have statistical properties that do not change over time, (mean variance) or that follow a certain trend. Meaning that if the process is stationary, it has tendency to return towards a mean, leading to the assumption: "where you are now affects where you will be." Where the assumption lies, that a rapid positive increase in return will yield an equally averaged negative return over time, in fact so that on a larger time scale the mean would stay constant. In this paper, H0 will be tested through the use of EViews software, where the unit root test is readily available and easily applicable to the data sets. This will be measured through the more appropriate Dickey Fuller GLS, where unit root will be tested for both intercept and trend. If the unit root is present, this confirms the null hypothesis.

The transformation of data from non-stationary to stationary requires a few simple steps. Primarily to determine the type of non-stationary data at hand, of which there are 4 main types. Notable:

- Pure Random walk, which assumes that the next value will be equal the previous with the addition of stochastic component referred to as "noise". This independent variable is identically distributed with mean 0 and variance σ^2. The importance of transformation in the case of random walk is due to the fact that it is a non-mean revering process, meaning that can move in a positive or negative direction without returning to the mean. Also causing it to have an ever evolving variance which tends towards infinity over time. It is represented by the following formula: $Y_t = Y_{t-1} + \varepsilon_t$
- Random Walk with Drift, is the same model as pure random walk, though with the addition of a constant, also known as the "drift" (α). The formula goes as follows: $Y_t = \alpha + Y_{t-1} + \varepsilon_t$
- Deterministic trend, is often confused with random walk with drift, as they share all similar components, with the only difference being that the value at time t is regressed to the previous period Y_{t-1} in the case of random walk with drift, and that in the case of a Deterministic Trend, the future value is regressed to a time trend βt. A deterministic trend has a mean which increases around a fixed trend, which is constant and independent of time. The formula is represented by $Y_t = \alpha + \beta t + \varepsilon_t$
- Random Walk with drift and deterministic trend, is a combination of the two previously mentioned types of non-stationary data, and is represented by the following formula: $Y_t = \alpha + Y_{t-1} + \beta t + \varepsilon_t$

A non-stationary data set following any of the types of random walk mentioned above can be transformed by differencing. Subtracting the value of $Y_{t-1}$from Y_t in the case of random walk or random walk with drift. In the case of a deterministic trend, a process referred to as detrending must be carried out, in which βt is subtracted from Y_t resulting in the following formula: $Y_t - \beta t = \alpha + \varepsilon_t$. detrending can be applied to a random walk with drift and deterministic trend, though this will not affect the variance, which will continue to tend towards infinity.

Once the data's stationarity determined, and the transformation applied, or neglected; the data is separated into the previously mentioned in and out of sample sets, and readied for the benchmark strategies.

We start the analysis by calculating the benchmark forecasts for 3 basic strategies; notably the Random Walk, no-change and Buy and Hold. These will be used as comparison to the linear models and learning machines designed at the next stage. The simplicity of these models requires only little calculation, Once the results of these strategies computed through excel, they are then plugged into the following trading strategy:

$$if E(R_{t+1}) > R_t \rightarrow buy$$

$$if E(R_{t+1}) < R_t \rightarrow sell$$

$$if E(R_{t+1}) = R_t \rightarrow hold$$

at which stage the percentage returns will be determined as well as the Mean Average Percentage Error, the Root Mean Squared Error and the Sharpe Ratio. We here expect to see modest gains and potential losses; due to their primitive and simplistic method, are more likely to yield little returns, higher errors and low Sharpe values.

The trading strategy is completely processed through excel; due to the precise yet simplistic nature of the software, a comprehensive set of results can be calculated using only the forecasted return, and the original exchange rates as inputs. Simply put, the trading system determines whether he forecasted return is positive, negative, or zero; turning the result into a trading signal of either "1", "0" or "-1". After which a position is taken, either a buy position, a sell position or a hold; the difference is then taken between the previous value and the new, resulting in the percentage gain of the transaction alone. These are then cumulated and annualized in order to calculate the annualized return as well as the other statistical evaluation measures.

Though both the MACD and the ARMA models have modifiable and "tweakable" parameters such as period length thus allowing for limited optimization. It is therefore to be expected that there will be a small variety of returns presented, all labeled with small applied changes. The ARMA and MACD model will primarily be tested with 4 variations in time period, notably: (1.3.10); (1. 4. 7); (1.6.12); (1.7.9).

The ARMA model follows the principle of two terms, notable the Autoregressive term(p), defined by the following formula:

$$X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t$$

and the moving average term (q) defined by the following formula:

$$X_t = \mu + \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

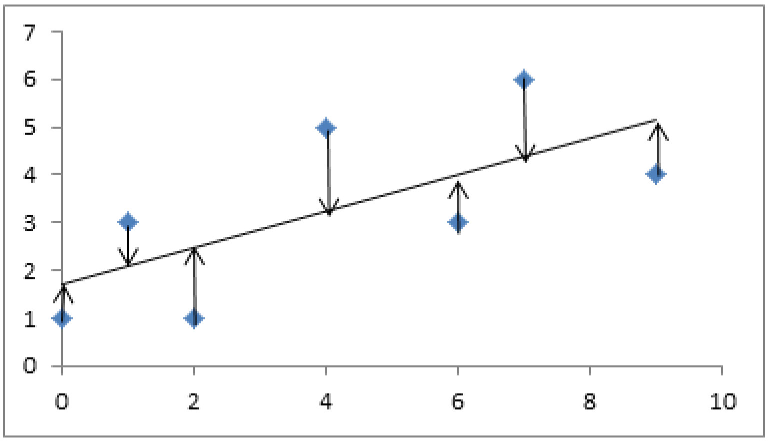
combined they yield the ARMA formula:

$$X_t = x + \varepsilon_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i}$$

Where θ and φ are parameters, c is a constant and ε_t is the error term at time t

This process is carried out on Eviews, where through equation estimation allows for the for the forecasted return to be calculated, once the parameters (period lengths) of the moving averages and auto-regressions is defined. The method employed by the Eviews software is the Least Squared method; being the standard approach to the estimation or approximation of a solution for sets of data. The basis of this model is to reduce the sum of the squared residuals of each data point to a minimum; the residual being the difference between the so called "best fit" and the actual data point. This can be simply illustrated with the graph below:

FIGURE 11.



Where each point represents a data point, and the length of each arrow depicts the minimized residual.

Once these forecasted returns calculated, they are once again inputted into the trading strategy computed on excel. Simply put this trading strategy uses the forecasted value of the following day to determine whether the underlying currency pair should be bought (going long), sold (going short) or held; the results are then computed over time and provide statistical data for total profit, annualized profit, number of transactions completed… etc. of which this paper chose to focus on annualized return, Root Mean Squared Error, Mean Average Percentage Error and Sharpe Ratio.

this completes the linear model section of this paper.

Once our linear model forecasted values calculated, the main aspect of this paper will come to hand; notably the use of Learning Machines. Forecasting Currency prices, or any other exchange traded assets, is particularly difficult due to its volatility, non-stationary and non-linear nature. When the basic statistical measure commonly used in linear models fail to maintain a measurable trend or mean, the model must constantly be adapted. Learning Machines find their place here, as they form regardless of the type of input and adapt through a training process, depending on the expected output.

The process for the MLP and the HONN are very similar in terms of data preparation; the idea of the neural networks is to use a series of past lags as input values, in this paper 9 lags have been chosen, all on a 1 day difference, as illustrate below:

TABLE 2.

OUTPUT	INPUTS								
0.526761									
-1.24854	0.526761								
-0.36685	-1.24854	0.526761							
0.729831	-0.36685	-1.24854	0.526761						
-0.30026	0.729831	-0.36685	-1.24854	0.526761					
-0.17022	-0.30026	0.729831	-0.36685	-1.24854	0.526761				
-0.68861	-0.17022	-0.30026	0.729831	-0.36685	-1.24854	0.526761			
-0.27075	-0.68861	-0.17022	-0.30026	0.729831	-0.36685	-1.24854	0.526761		
-1.78122	-0.27075	-0.68861	-0.17022	-0.30026	0.729831	-0.36685	-1.24854	0.526761	
-4.73269	-1.78122	-0.27075	-0.68861	-0.17022	-0.30026	0.729831	-0.36685	-1.24854	0.526761

This allows the neural network to learn from changed in its historical data movements, with each lag referring to an ideal output from different lengths in the past.

In this format, the values are separated into in sample and out of sample periods as illustrated in table. The training of the Network is of high importance, the in-sample period is used as training, given that the inputs here are the lagged past values of the currency, with target output being the actual value to be forecasted, the forecast is solely based on a "pattern recognition" basis, as no exterior stimulation such as, for example, news announcements or interest rate changes are taken into account here.

The same procedure is then carried out with the out of sample data set. Though this is to simulate the applied efficacy of the learning machine, and the difference between the forecasted values and actual values will be listed as errors, rather than solely as model adjustment parameters. Once both the model optimization completed and the out of sample selection tested, the values are gathered

These steps are equivalent for both the MLP and the HONN. Both models are run though Matlab software, providing flawless modeling and easily understandable output data.

At this stage, Matlab provides two types of output graphs to be utilized as comparison; notably the previously mentioned overall percentage gain, as well as the squared error. In the case of the squared error, it is a value to be minimized, and the percentage gain to be maximized. To find the rest of our statistical evaluators, the forecasted values are exported from excel, and pasted into the same excel based trading strategy previously used for the linear models.

Once a set of values found for all three currency pairs and both ANN models, the following two learning machine models are to be carried out.

The Genetic Algorithm - Support Vector Regression once again requires Matlab to carry out the model. The data is arranged with the same number of lags as all previous models and the same time frame. Matlab provides additional results table: computation time and the prediction for the in-sample and the out-of-sample and the parameters, which will be again copy-pasted into excel in order to evaluate the efficacy and risk adjusted performance of the model. The power of this model, lies in its ability to capture nonlinearities and asymmetries in the past lagged values, and select the optimal feature subset in order to produce solid rolling SVR forecasts for the underlying currency pair. Genetic Algorithms are capable of searching for potential predicators in a much larger pool than used in this paper; past literature has suggested a variety of economic inputs, such as interest rates, unemployment rates and a variety of other available indicators that may have an effect of the exchange rates at hand. Though with only 9 lagged values as inputs, one can only assume a higher efficiency in the parameter selection, and thus a larger potential output.

3.1 STATISTICAL EVALUATION

The previously mentioned statistical evaluation ratios and values are explained as follows:

The primary choice of evaluation is the use of the annualized percentage gain; giving an overall idea of how well the respective model has performed in terms of profits or losses.

Illustrated by the following formula:

$$Annualised\ Return = \frac{\sum_{t=1}^{n}\left(\frac{P_t - P_{t-1}}{P_{t-1}} * 100\right)}{n} * 252$$

Where n is the number of days on which a trade was placed and P_t is price at time t. The annualized return formula provides an indication of how well a model has performed as a time-weighted annual percentage. In this model return in solely calculated by the profits and losses made by each position taken by the model.

Secondly the Root Mean Squared Error (RMSE), illustrated with the following formula:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(\hat{Y}_t - Y_t\right)^2}$$

Where n is the number of data sets, $\hat{Y}_t$ is the forecasted value at time t and Y_t is the actual value at time t.

The root mean squared error, also known as the root mean squared deviation (RMSD) is a measure of differences between values produced by an estimation function and the actual values observed. In more simple terms, the RMSE is the calculation between real values and their estimated counterparts. When used on in-sample data, the difference between these values is referred to as a “residual”, whereas during an out of sample process they are called “prediction errors.” The formula serves to aggregate the magnitude in prediction errors or residuals of various time series into a single measure of accuracy. It is a frequently used and appreciated measure, but can only be used to compare forecasting errors for one particular variable, as it is scale dependent, hence can accept different models, but not different data types. Under ideal unbiased conditions, the RMSE would be equal to the standard deviation, and a perfect fit would leave an RMSE of 0.

As well as the Mean Absolute Percentage Error

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{\hat{Y}_t - Y_t}{Y_t}\right|$$

The Mean average percentage error is a measure of accuracy, frequently used in trend estimation. It expresses the accuracy in the form of a percentage, the absolute value of the difference between the forecasted value and he real value is taken, then divided by the actual price; when the sum of all points is taken, then divided by n and multiplied by 100 yields the percentage error. A major drawback of this method, is although it yields a percentage result, there isn't a relative value; meaning that a perfect fit, would result in a 0% value, though there is no upper limit to the badness of fit; resulting in values over 100%.

With the long run objective of this paper to create a model applicable to trading and investment, performance can be evaluated through a series of ratios and rates.

The Sharpe Ratio, also known as the return to variability ratio, measures the risk adjusted return of an investment. It is one of the most popular and used evaluation ratios among investors and is given by the following formula:

$$Sharpe\ ratio = \frac{r_A}{\sigma_A}$$

Where r_A is the annualized return and σ_A is the standard deviation. To be noted that the previously calculated returns will be used to calculate this ratio, a not the forecasts.

The Sharpe ratio being an adequate and widely used ratio gives a good indication of the solidity of the trading model. Though with the intention being a practically applied model; simulating trading conditions, such as transaction cost may give a more firm approximation to how the model would perform under real conditions. The greater a Sharpe Ratio, the better it's risk adjusted performance. In the case of a negative Sharpe ratio, a risk free asset would have a greater performance.

4.0 ANALYSIS

4.1 TESTING THE NULL HYPOTHESIS

As extensively explained in the methodology section of this paper, the null hypothesis tests the stationarity of the data set. With linear models requiring a fixed set of statistical measures, such as the mean and the standard deviation, data sets may have to be transformed through stationorisation if the Null hypothesis happens to be rejected. Eviews handily provides a unit root testing tool, testing for stationarity using a variety of methods; the results for the full (both in and out of sample) data sets are shown below for the EURUSD currency pair, the results for the EURCHF and GBPUSD can be found in the appendix:

TABLE 3.

EURUSD

Null Hypothesis: RT has a unit root
Exogenous: Constant, Linear Trend
Lag Length: 7 (Automatic - based on SIC, maxlag=29)

		t-Statistic
Elliott-Rothenberg-Stock DF-GLS test statistic		-15.78691
Test critical values:	1% level	-3.480000
	5% level	-2.890000
	10% level	-2.570000

*Elliott-Rothenberg-Stock (1996, Table 1)

DF-GLS Test Equation on GLS Detrended Residuals
Dependent Variable: D(GLSRESID)
Method: Least Squares
Date: 04/09/14 Time: 20:47
Sample (adjusted): 10 3574
Included observations: 3565 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
GLSRESID(-1)	-0.605111	0.038330	-15.78691	0.0000
D(GLSRESID(-1))	-0.362540	0.037312	-9.716442	0.0000
D(GLSRESID(-2))	-0.325914	0.035842	-9.093165	0.0000
D(GLSRESID(-3))	-0.290891	0.033908	-8.578842	0.0000
D(GLSRESID(-4))	-0.230868	0.031290	-7.378405	0.0000
D(GLSRESID(-5))	-0.166388	0.027757	-5.994357	0.0000
D(GLSRESID(-6))	-0.116803	0.023166	-5.042015	0.0000
D(GLSRESID(-7))	-0.056754	0.016692	-3.400047	0.0007

R-squared	0.485002	Mean dependent var	-6.88E-06
Adjusted R-squared	0.483989	S.D. dependent var	0.009123
S.E. of regression	0.006554	Akaike info criterion	-7.215336
Sum squared resid	0.152776	Schwarz criterion	-7.201471
Log likelihood	12869.34	Hannan-Quinn criter.	-7.210392
Durbin-Watson stat	2.000744		

As seen in the 3 Eviews output charts, (one for each of the currency pairs to be forecasted) all have unit root and confirm the null hypothesis. It is therefore safe to assume the linear methods used to forecast will be reliable and consistent without needing to transformation of the data.

4.2 BASE MODELS

The motivation behind the calculation of these models remains the evaluation of popular linear models in comparison with Artificial Neural Network models and Learning Machines. The strategy with which the values for the following models were calculated are based on a simulated end of day trading method, where the forecasted value for the following day will serve as base of action for the trading position to be taken on the current day. The models being tested here are inspired by past literature as are the methods of evaluation used there, (seen in the literature review section of this paper). The models to follow are the random walk, Buy and Hold, and no change models.

4.3 RANDOM WALK

The random walk model drew attention and gained notoriety in 1973, when the author Burton Malkiel published "A Random Walk down Wall Street." The theory itself is reflected in its name, suggesting that all exchange traded assets, currencies or other, all follow a completely random and unpredictable path, making it impossible to outperform the markets without taking on an additional risk. The formula listed below, simply suggest that the evolution from the preceding price is based entirely on the old price, to which a drift constant and an error term is added. The formula is given below:

$$Y_t = \mu + Y_{t-1} + \epsilon_t$$

The main reason for the model's notoriety is due to the simple fact that it has the tendency of outperforming most classical models. It has been used as a benchmark value for multiple linear models, Artificial Neural networks and Learning machines. Though as the model suggest, it has little practical value and technically entices investors to take completely random decisions when making investments.

The following chart shows the results obtained when using this model:

TABLE 4.

	EURUSD	EURCHF	USDGBP
Annualized Rt	28.51%	27.25%	26.90%
RMSE	0.0515	0.0634	0.0515
MAPE	133.78%	135.29%	134.36%
Sharpe	0.75	0.72	0.71

As past literature had foreseen, the Random walk has substantial returns, and acceptable levels of RMSE and RMSPE. Making it a firm contender to the coming forecasting models, both linear and Artificial.

The practicality of the model lacks, as well as the Random walk may perform in statistical terms, the practical application of a fully random selection of investments and weights may not be a very convincing approach to "real life" investments.

4.4 BUY AND HOLD

Supporters of the Buy and Hold strategy follow the assumption that stocks will go up in the long term, and that investors will find themselves making profit if they manage to hold out the volatility of the investment. This directly contradicts the day trader or the momentum trader, where small movements in the stock market are to be exploited in order to profit in rapid buying and selling in the short term.

below is a table illustrating the results found using the buy and hold strategy for each one of the currency pairs, once again being analyzed using the annualized return, the root mean squared error, the mean average percentage error and the Sharpe ratio.

TABLE 5.

	EURUSD	EURCHF	GBPUSD
Annualized Rt	1.92%	-3.00%	3.72%
RMSE	0.1	0.1003	0.1002
MAPE	8900.02%	45580.80%	8990.35%
Sharpe	0.2	-0.23	0.26

As was to be expected, the results are minimal, yet the attention is drawn to the substantially large percentage error; this too was to be expected, as the buy and hold strategy assumes that the best forecast is the previous price, any deviation of the price, be it positive or negative will result in an increased error term. Though due to the assumption of the price going up in the long run, it is safe to neglect the error terms when adopting this strategy. This strategy may be used as a safer bet, in order to secure a certain percentage of a total portfolio in a long term-investment and reduce exposure to overall market fluctuations.

4.5 NO CHANGE

No change theory is based on the Random Walk Model, in the sense that

$$Y_t = Y_{t-1}$$

Which is a simple simplification of the random walk model:

$$Y_t = \mu + Y_{t-1} + \epsilon_t$$

As illustrated above, the key difference between the no-change and the random walk model, is the suppression of the drift constant and the error term, quite literally following what the model's name suggests: no change.

Implementing the idea that the previous price is the best indicator of the current price. This approach was used to benchmark GDP growth and inflation models by Marellino (2006). In applicable terms, this model suggests a buy and hold strategy, yet resulting in no expected return. For instance if the return were to be positive, the deviation from the original point will be classified as an error.

Illustrated in the table below:

TABLE 6.

	EURUSD	EURCHF	USDGBP
Annualized Rt	0.00%	0.00%	0.00%
RMSE	0.0062	0.0077	0.009
MAPE	99.49%	99.69%	99.08%
Sharpe	N/A	N/A	N/A

With the assumption of no price change, no trade would be placed when starting the investment, hence the return here is 0%, and the percentage error approaches 100%.

The Sharpe ratio isn't applicable to this model, as there is no return, hence producing an equation with a division by zero.

$$Sharpe = \frac{annualised\ return}{Annualised\ Volatility}$$

$$Sharpe = \frac{0.00\%}{0.00\%}$$

An arguable critique to this model, is the simple fact that no return is achieved or expected, meaning that a risk free investment with a fixed return would perform better on all fronts.

5.0 LINEAR MODELS

The following models are the linear models, notably the ARMA and MACD. These models are frequently used in the taking of practical trading decisions on a daily basis. Their input is limited to the lengths of period to be analyzed, they are based on moving averages and auto regressive values taken from the specified periods. These periods are selected on a trial and error basis, hence their limited effective utility, and low expected precision.

5.1 AUTO-REGRESSIVE MOVING AVERAGE

ARMA remains one of the most configurable and optimizable linear models in this paper; with which some impressive results have been calculated. As previously mentioned, the periods for the auto regressive data and the moving averages were: (1.3.10); (1. 4. 7); (1.6.12); (1.7.9).

The periods having yielded the highest result both in sample and out differed from one currency pair to the next. It's safe to say again that linear models will not adapt to the underlying data set, only simple patterns.

Below are the results of the most performing ARMA models: (only out of sample)

TABLE 7.

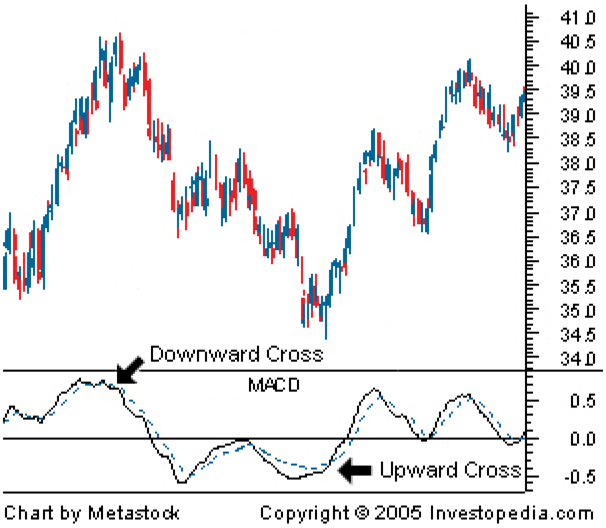
Periods	1.3.10	1.7.9	1.6.12
Annualized Rt	6.01%	17.19%	9.43%
RMSE	0.0705	0.0634	0.0771
MAPE	1598.72%	8287.13%	1891.54%
Sharpe	0.62	1.79	1.23

The results were satisfying, even more than expected in the case of EUR/CHF, yielding a 17.19% return using the 1.7.9 periods. With a Sharpe ratio of 1.79, the model yields excellent risk adjusted returns, though with an error of 8287.13%, the consistency of the model over time may be questioned.

5.2 MOVING AVERAGE CONVERGENCE DIVERGENCE

MACD involves placing a trade based on the overlap of moving averages of different periods, theoretically giving the indication of a change in trend, as illustrated below:

FIGURE 12.



The same periods are to be used as for the previously calculated ARMA model, notably: (1.3.10); (1. 4. 7); (1.6.12); (1.7.9).

Once again, the most performing periods may vary from one currency pair to the next. With different mean values, distribution and volatility, this is to be expected.

The most performing results of the out-of-sample period are given below:

TABLE 8.

	EURUSD	EURCHF	USDGBP
Periods	1.7.9	1.6.12	1.7.9
Annualized Rt	1.92%	4.28%	3.01%
RMSE	0.0098	0.0139	0.0115
MAPE	690.38%	5736.72%	1050.53%
Sharpe	0.2	0.44	0.38

The results yielded by the MACD model were disappointing in all measures, with a very low annualized return, a risk free investment would have yielded higher results, as shown by the Sharpe ratio, when calculating risk adjusted performance, with all three currency pairs scoring under 0.5 in ratio. Once again an outstandingly large Mean Average Percentage Error and Root mean squared error.

As previously stated, and confirmed by past literature, the sensitivity and inconsistency of linear model such as the MACD make them impractical for "real life" use, though they are surprisingly popular in day trading and a very commonly used tool to the everyday trader.

6.0 ARTIFICIAL NEURAL NETWORKS

6.1 MULTI-LAYER PERCEPTION

The multi-layer perceptron, as previously introduced in the literature review is based on a system of nodes, with the input nodes containing the 9 lagged values, which are then transmitted to the hidden layer, as the weighted sum of its inputs, the hidden layer then passes on the information to the output layer through a non-linear activation function. A simple illustration is given in graph in the literature review

After computing the data through Matlab, the following results were found for all currency pairs:

TABLE 9.

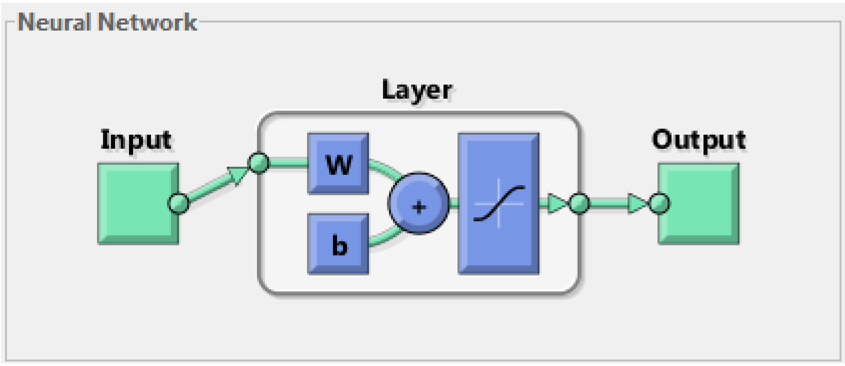
Annualized Rt	31.79%	26.01%	23.81%
RMSE	0.024	0.0162	0.0169
MAPE	107.06%	106.97%	107.13%
Sharpe	0.83	0.68	0.62

The simple setup of an MLP model makes it the go-to model when referring to Artificial Neural Networks. The method of transmitting the weighted input variable to a hidden layer, passing through a non-linear activation function, then processing the data as in or out of threshold, before reaching the final output, adjusting its weights during the training process, it produces a very simple and understandable graph, as illustrated in the literature review. The results produced by this model are solid building blocks for any trading strategy, with annualized returns between 23% and 31%, the risk factor represented by the Sharpe ratio being acceptable, and the Mean Average Percentage error at an average level for this type of model, the MLP produces results that outweigh the previous linear models, and are on a similar level with the random walk. In terms of practical application, this model may very well be used as part of an overall strategy.

6.2 HIGHER ORDER NEURAL NETWORKS

The higher order neural network implements a very similar system to the MLP, with the difference of inputs; explained in more detail in the methodology and the literature review.

Matlab illustrates the process as follows:



the input on the left depicting the lagged values, which is then passed on to the hidden layer, at which point a bias node *b* is added, their sum is passed onto a sigmoid function, and processed to form an adequate output.

Yielding the forecasted results in the same format as the previously demonstrated MLP:

TABLE 10.

Annualized Rt	32.38%	24.37%	22.16%
RMSE	0.024	0.024	0.024
MAPE	107.10%	107.03%	107.21%
Sharpe	0.85	0.64	0.58

As is to be expected with such a large data set, using the HONN yields less satisfying results than the MLP. The extensive literature to this topic did suggest that the higher number of inputs result in lower forecasts. As the total sample size in this paper contains 3564 end of day closing prices, which were distributed over 9 lags, the recommended maximum of 4 inputs was here surpassed. It was hypothesized that the results wouldn't yield as high a total percentage gain, or as low a squared error, though the program did nonetheless yield results that surpassed the ARMA and MACD models, with a slightly higher accuracy than the Random Walk, this model remains a strong forecasting tool to be used with Foreign Exchange rate prediction.

6.3 GENE EXPRESSION AND GENETIC ALGORITHMS

GAs are problem solving methods that mimic the process of natural evolution.
They differ from ANNs in the sense that they use the concept of natural selection in order to fit the best solution to a problem. they tend to be used in order to maximize or minimize a certain feedback measure. They can be used within or exterior to the construction of an ANN. The main goal of the Genetic algorithm is to form an ideal program, based on failures; just like natural selection, unwanted results that deviate too far from the desired or expected result will "die" and be discontinued in the evolution of the program. The overall past literature has a satisfying approach to genetic algorithms, with tested models either breaking even with benchmark models, or outperforming them under the correct circumstances.

TABLE 11.

Annualized Rt	34.75%	37.23%	36.31%
RMSE	0.024	0.024	0.024
MAPE	103.56%	103.48%	103.60%
Sharpe	0.91	0.98	0.95

The results have shown the highest return so far, with a Sharpe ratio approaching 1.0 it has the traits of a good risk adjusted investment.

6.4 ROLLING GENETIC ALGORITHM - SUPPORT VECTOR REGRESSION

Support Vector Regression has shown to be the most promising of all singular models rested in past literature, also the model to have the most recent trials and papers. As suggested in the literature review, the application of Learning Machines in the fields of forecasting has only recently attracted as much attention; mainly due to the largely available computational power. With the restrictions lifted and the birth of supercomputers, offering the opportunity for Artificial intelligence models to thrive among vast fields, not limited to Foreign exchange, but including biology, chemistry, cancer research and all other sciences, where an input relates to an output in any way.

With the two most performing models listed in past literature combined into a rolling hybrid model, only success is to be expected; with past papers reviewed in the literature review suggesting the combination of Support Vector Regression with other models in order to better its feature subset selection. Gene Expression being one of the most popular and efficient models in the field of program selection seems like a natural choice in bettering an already efficient model. In more simplistic terms, the feature selection is here “outsourced” to the Gene expression model.

The results are once again listed in the same format as the previous models below:

TABLE 12

Annualized Rt	30.12%	36.73%	33.70%
RMSE	0.0189	0.0189	0.0189
MAPE	135.09%	134.55%	134.79%
Sharpe	0.79	0.96	0.89

The results are impressive, with returns over 30% for all three currency pairs, and an error term in line with expectations. The GA-SVR has proven its efficiency in the field of forecasting.

6.5 STATISTICAL EVALUATION OF ALL MODELS

TABLE 13

		EURUSD	EURCHF	USDGBP
Random Walk				
	Annualized Rt	28.51%	27.25%	26.90%
	RMSE	0.0515	0.0634	0.0515
	MAPE	133.78%	135.29%	134.36%
	Sharpe	0.75	0.72	0.71
no change				
	Annualized Rt	0.00%	0.00%	0.00%
	RMSE	0.0062	0.0077	0.009
	MAPE	99.49%	99.69%	99.08%
	Sharpe	N/A	N/A	N/A
Buy and Hold				
	Annualized Rt	1.92%	-3.00%	3.72%
	RMSE	0.1	0.1003	0.1002
	MAPE	8900.02%	45580.80%	8990.35%
	Sharpe	0.2	-0.23	0.26
MACD	Periods	1.7.9	1.6.12	1.7.9
	Annualized Rt	1.92%	4.28%	3.01%
	RMSE	0.0098	0.0139	0.0115
	MAPE	690.38%	5736.72%	1050.53%
	Sharpe	0.2	0.44	0.38
ARMA	Periods	1.3.10	1.7.9	1.6.12
	Annualized Rt	6.01%	17.19%	9.43%
	RMSE	0.0705	0.0634	0.0771
	MAPE	1598.72%	8287.13%	1891.54%
	Sharpe	0.62	1.79	1.23
MLP				
	Annualized Rt	31.79%	26.01%	23.81%
	RMSE	0.024	0.0162	0.0169
	MAPE	107.06%	106.97%	107.13%
	Sharpe	0.83	0.68	0.62
HONN				
	Annualized Rt	32.38%	24.37%	22.16%
	RMSE	0.024	0.024	0.024
	MAPE	107.10%	107.03%	107.21%
	Sharpe	0.85	0.64	0.58
Gene Expression				
	Annualized Rt	34.75%	37.23%	36.31%
	RMSE	0.024	0.024	0.024
	MAPE	103.56%	103.48%	103.60%
	Sharpe	0.91	0.98	0.95
GA-SVR				
	Annualized Rt	30.12%	36.73%	33.70%
	RMSE	0.0189	0.0189	0.0189
	MAPE	135.09%	134.55%	134.79%
	Sharpe	0.79	0.96	0.89

When observing the totality of the results as listed above, their forecasting performance can more easily be ranked in terms of different statistical measures.

Due to the general similarity of percentages across currencies, these measures were averaged across all three rates, in order to produce an overall performance measure of the given model.

With regard to annualized percentage gain, the top three performing models are:

1. Genetic Programming
2. GA-SVR
3. Random Walk

Which favors the goal of this paper in showing the higher performance of Genetic feature selection in contrast to more simple neural networks.

With regard to Root mean square Error and Mean Average Percentage Error:

1. No change
2. Gene Expression
3. Higher Order Neural Network

With regard to the Sharpe ratio:

1. ARMA
2. Gene Expression
3. GA-SVR

Overall the results didn't defy the expectations, with the neural networks and the random walk proving to be the highest profitable and accurate models, the only surprise here was the Sharpe ratio of the ARMA model, though the lower profitability and record high Mean Average Percentage Error deter the use of this model in practical situations.

7.0 CONCLUSION

The goal of this paper, was to determine whether learning machines and artificial neural networks were in fact more efficient and precise at forecasting future foreign exchange rate movements. The time span of this paper yielded "one day ahead" forecasts, based on patterns found in 9 lagged values all on a one day interval. The span seemed as an adequate time frame, as it is applicable in real trading scenarios, and didn't require any other inputs than past historical closing prices. In practice, it isn't uncommon to see traders use linear models when carrying out technical analysis, though according to this paper, their choice of tool may be misplaced.

The linear models did perform relatively well to more common base strategies such as the no-change and the buy and hold strategy. Though the Random Walk did outperform the linear models by a substantial margin; just as suggested in past literature.

The Multi-Layer Perceptron outperformed expectancies, and did justice to the various researchers having mentioned it in past papers; with annualized returns averaging at 25% for all three currencies, surpassing the more simple HONN both in terms of annualized percentage gain as well as Error measure, more than likely due to the unspecified relationship between inputs in Higher Order Neural Networks.

Finally the Genetic algorithm and the Hybrid GA-SVR outperformed all other models, yielding results as high as 37% annualized return, and no lower than 30%. With a fairly acceptable Sharpe ratio, considering the high return, this paper has successfully proven the superiority of all learning machine models to base models, and more importantly the efficiency and high potential of Genetic algorithms in the field of Forecasting as well as the optimized features of the GA-SVR.

8.0 RECOMMENDATIONS

Over the course of this paper, a variety of models have been tested, ranging from classic buy and hold models to the notorious Random Walk, which against odds seems to beat most models based entirely on random factors and inputs. The more classic ARMA and MACD models were equally tested, given their popularity with the day-trader, and their presence in many trading tutorials scattered around the web, their results have been mostly disappointing over a simulated trading period.

Artificial neural networks were tested and optimized over time, in order to attempt the outsourcing of market adaptation to strict formula, producing models that adapt over time, finding patters solely based on historical price movements over a theoretically infinite amount of time. Time here being one of the main factors with which ANNs and Learning machines differ from classic linear models; as the linear counterparts usually only base themselves on a sample of an overall population period to make their predictions, specified in the case of ARMA and MACD by the periods. The ANNs take into account the whole studied population and adapt according to all of these inputs. Of the two basic ANN models, the MLP outperformed the HONN by a slight amount; as the literature suggested, the number of inputs to outputs ratio may have been too high, and a lower number of lags may have yielded a better result using the HONN. The MLP seems to have achieved such great results through its simplicity, though limitations may lie when applying the model to a smaller time frame.

The GA proved successful in outperforming most of the benchmark models, though not by an astounding margin, resembling the results of the SVR, the Hybrid model is expected to yield better results. The main difference between these two models seems to be the risk adjusted performance; notably the Sharpe ratio, which is slightly higher in the Gene Expression model than the SVR, though not by a substantial amount. The Mean Average Percentage Gain is substantially higher in the SVR model, by an average of 30% across he different currency pairs.

As hypothesized in earlier stages, the GA-SVR model yielded the most promising results, with consistent profits to be expected at a future time, even with the length of the study were to be changed. The computation time on this model may cause problems if applied to too short time frames, though in applicable terms, the model appears to be profitable when trading on an end of day basis.

considering all this information, it is naturally recommended to use the finally suggested hybrid GA-SVR model as a primary forecasting tool; the availability of trading Foreign exchange on an end of day basis is available among a vast series of brokers, all providing a variety of spreads, leveraging options and transaction costs. In order to fully determine the viability of these models in a real life scenario, a simulation including leverage and transaction costs taken from a series of brokers would be necessary.

It is nonetheless necessary to emphasize, that similarly to any investment, a combination of strategies and investments would be recommended in order to secure a long term gain. With the models tested in this paper, a combination of the Rolling Genetic – Support Vector Regression, the Gene Expression model, the Multi-Layer Perceptron and the Buy and hold strategy would seem to provide an excellent basis for a trading strategy. Though once again further research would be required, including simulated trading using a portfolio of strategies, including transaction cost as well as leverage in order to determine with accuracy how profitable these models would be in a "real life" scenario.

On a side-note, past literature has brought up the topic of Efficient Market Hypothesis, credited to Eugene Fama in the early 1960s; and the idea of disproving the hypothesis if any of the models succeed in outperforming the benchmark models; notably the random walk model. As the theory states that all information available in the markets is perfectly reflected in the underlying asset, stock, or currency, it is impossible to make a profitable investment based on the undervalued or overvalued price of an asset. The linear, Artificial Neural Network and learning machine models employed in this paper all solely have historical prices as input values; whereas these models have succeeded in outperforming the benchmark values based solely on technical analysis. It is therefore safe to assume, that under the conditions offered to the models in this paper, the Efficient Market Hypothesis does not hold. This assumption though cannot be made on the market as a whole, and is limited to this papers choice of currencies, time frame and input types, though it does not a flaw in the hypothesis, leading the author to assume the market is most definitely not strong form of EMH, but further research would be needed to prove that the market be semi-strong or weak.

SOURCES

Alexander, S. S. (1964) *The Random Character of Stock Market Prices.* Cambridge, MA: MIT Press.

Allen, F. and Karjalainen, R. (1999) 'Using genetic algorithms to find technical trading rules', *J. Finan. Econ.*, 51, pp. 245–271.

Allen, F., and Karjalainen, R. (1999) 'Using genetic algorithms to find technical trading rules', *Journal of financial Economics*, *51*(2), pp. 245-271.

Bahramy, F. and Crone, S. F. (2013) 'Forecasting foreign exchange rates using Support Vector Regression', *Computational Intelligence for Financial Engineering & Economics (CIFEr), 2013 IEEE Conference,* pp. 34-41.

Basak, D., Pal, S. and Patranabis, D. C. (2007) 'Support Vector Regression', *Neural Information Processing - Letters and Reviews, 10 (10), pp. 203-224.*

Bauer, R. J. (1994) *Genetic algorithms and investment strategies*. New York, Wiley.

Bessembinder, H and Chan, K. (1995) 'The profitability of technical trading rules in the Asian stock markets', *Pacific-Basin Finance J.*, 3, pp.257–284.

Cao, L. J. and Tay, F. E. H. (2003) 'Support vector machine with adaptive parameters in financial time series forecasting', *IEEE Trans. Neural Netw.*, 14(6), pp. 1506–1518.

Castiglione, F. (2001) 'Forecasting Price Increments Using an Artificial Neural Network' in *Complex Dynamics in Economics: A Special Issue of Advances in Complex Systems*, 3(1), pp. 45-56.

Cheung, Y. W. (1993) 'Long memory in foreign-exchange rates', *Journal of Business and Economic Statistics*, 11, pp. 93- 101.

Donald, H. (1949) *The Organization of Behavior*. New York: Wiley.

Duan, K., Keerthi, S. and Poo, A. N. *(2003)* 'Evaluation of simple performance measures for tuning SVM hyperparameters', *Neurocomputing", 51, pp. 41-59.*

Dunis, C. L., Laws, J. and Karathanasopoulos, A. (2011) 'Modelling and trading the Greek stock Market with mixed neural network models', *Working Paper Series No. 15*, Centre for EMEA Banking, Finance & Economics, London Metropolitan Business School.

Fama, E (1965) 'The Behavior of Stock Market Prices', *Journal of Business,* 38, pp. 34–105.

Fama, E. F and Blume, M. E . (1966) 'Filter rules and stock market trading', *J. Business*, 39, pp. 226–241.

Fama, E. F. (1970) ‘Efficient capital markets: A review of theory an empirical work’, *J. Finance*, 25, pp. 383–417.

Farley, B.G.and Clark, W.A. (1954) ‘Simulation of self-organizing systems by digital computer’, *IRE Transactions on Information Theory*, 4 (4), pp. 76–84.

Giles, L. and Maxwell, T. (1987) ‘Learning invariance and generalization in high-order neural networks’, *Applied Optics*, 26(23) 4972-4978.

Hans Frances, P. and van Griensven, (1998) ‘Forecasting exchange rates using neural networks for technical trading rules’, *Studies in Nonlinear Dynamics and Econometrics*, 2(4), pp. 109-114.

Harrald, P. G. and Kamstra, M. (1997) ‘Evolving artificial neural networks to combine financial forecasts’, *IEEE Trans. Evol. Comput.*, 1(1), pp. 40–52.

Hayward, S. (2004) ‘Simulating profitable stock trading strategies with an evolutionary artificial neural network’, *in Proc. Int. Workshop Intell. Finance*, pp. 108–117.

Holland, J. (1995) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* Cambridge, Mass: MIT Press.

Huang, C.L. and Wang, C.J. (2006) ‘A GA-based feature selection and parameters optimization for support vector machines’, *Expert Systems with Applications*, 31(2), pp. 231-240.

Kai, F. and Wenhua, X. (1997) ‘Training neural network with genetic algorithms for forecasting the stock price index’, *in Proc. IEEE Int. Conf. Intell. Process. Syst.*, pp. 401–403.

Kanungo, R. P. (2004) ‘Genetic Algorithms: Genesis of Stock Evaluation’,*Economics WPA Working Paper*.

Karemera, D. and Kim, B. (2006) ‘Assessing the forecasting accuracy of alternative nominal exchange rate models: the case of long memory’*, Journal of Forecasting,* 25 (5), pp 369–380.

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

Koza, J. R. (1992) *Genetic Programming*. Cambridge, MA: MIT Press/Bradford Books.

Kwon, Y.K. and Moon, B.R. (2007) ‘A Hybrid Neurogenetic Approach for Stock Forecasting’, *IEEE Trans. Neural Netw.,* 18, pp. 851-864.

Lawrence J. and Andriola, P. (1992) 'Three-step method evaluates neural networks for your application', pp. 93–100.

Lee, Y., Lin, Y. and Wahba, G. (2004) 'Multicategory Support Vector Machines', *Journal of the American Statistical Association*, 99(465), pp. 67-81.

Lin, L., Cao, L., Wang, J. and Zhang, C. (2004) 'The Applications of Genetic Algorithms in Stock Market Data Mining Optimization', *Information and Communication Technologies*, 33, pp. 8-16.

Liu, Y. and Shen, X. (2006) 'Multicategory E-Learning', *Journal of the American Statistical Association*, 101(474), pp. 500-509.

Lo, A. (2000) 'Finance: A Selective Survey', *Journal of the American Statistical Association*, 95(450), pp. 629-635

Malkiel, B. G. (1973) *A random walk down Wall Street: The time-tested strategy for successful investing*. New York, W.W. Norton.

Marcellino, M. (2006) 'A simple benchmark for forecasts of growth and inflation'

McCulloch, Warren, M and Pitts, W. (1943) 'A logical calculus of ideas immanent in nervous activity', *Bulletin of Mathematical Biophysics*, 5 (4), pp 115–133.

Meese, R. and Rogoff K. (1983) 'Empirical Exchange Rate Models of the Seventies: Do they Fit out of Sample?, *Journal of International Economics*, 14, pp. 3–24

Minsky, M. and Papert, S. (1969) *An Introduction to Computational Geometry*. MIT Press.

Montana, D. J. (1995) 'Strongly typed genetic programming' *Evolutionary computation*, *3*(2), pp. 199-230.

Neely, C. J. (2003) 'Risk-adjusted, ex ante, optimal technical trading rules in equity markets', *Proc. Int. Rev. Econom. Finance*, 12 (1), pp. 69–87.

Pearson, K. (1905) 'The problem of the random walk', Nature. 72, 294-310.

Plakandaras, V., Papadimitriou, T. and Gogas, P. (2013) 'Forecasting daily and monthly exchange rates with machine learning techniques' Democritus University of Thrace, Department of Economics in its series DUTH Research Papers in Economics with number 3-2013.

Ready, M. J. (2002) 'Profits from technical trading rules', *Finan. Manage.*, 31, pp. 43–61.

Refenes, A,N. (1992) ‘Managing Exchange rate prediction strategies with neural networks’, *Technique and application of Neural Networks*, Sept, 1992. Liverpoool, UK.

Refenes, A.N., Zapranis, A. and Francis, G. (1994) ‘Stock performance modeling using neural networks: A comparative study with regression models’, *Neural Networks*, 7(2), pp. 375–388.

Rime, D., Sarno, L. and Sojli E. (2010) ‘Exchange rate forecasting, order flow and macroeconomic information’, *Journal of International Economics*, 80, pp. 72-88

Rochester, N., Holland, J.H., Habit, L.H. and Duda, W.L. (1956) ‘Tests on a cell assembly theory of the action of the brain, using a large digital computer’. *IRE Transactions on Information Theory*, 2 (3), pp. 80–93.

Saad, E.W., Prokhorov, D.V. and Wunsch, D.C. (1998) ‘Comparative study of stock trend prediction using time delay, recurrent and proba- bilistic neural networks’, *IEEE Trans. Neural Netw.*, 9(6), pp. 1456–1470.

Scholkopf, B. and Smola, A. *(2002*) *Learning with kernels. Cambridge: MIT Press.*

Scholkopf, B., Smola, A., Williamson, R. and Bartlett, P. *(2000)* “New Support Vector Machines*”, Neural Computation, 10(5), pp. 335-347.*

Sermpinis G., Theofilatos K., Karathanasopoulos A., Georgopoulos F. E., and Dunis C. (2013) ‘Forecasting foreign exchange rates with adaptive neural networks using radial-basis functions and Particle Swarm Optimization’, *European Journal of Operational Research*, 225, pp. 528–540.

Sermpinis, G. Stasinakis, C., Theofilatos, K. and Karathanasopoulos, A. (*2013) ‘*Modeling, Forecasting and Trading the EUR Exchange Rates with Hybrid Rolling Genetic Algorithms – Support Vector Regression Forecast Combinations’

Siedlecki, W. and Sklansky, J. (1989) ‘A note on genetic algorithms for large-scale feature selection’, *Pattern Recognition Letters*, 10(5), pp. 335-347.

Sun., Z., Bebis, G. and Miller, R. (2004) ‘Object detection using feature subset selection’, *Pattern Recognition*, 37 (11), pp. 2165–2176.

Suykens, J. A. K., Brabanter, J. D., Lukas, L. and Vandawalle, L. (2002) ‘Weighted least squares support vector machines: robustness and sparse approximation’, *Neurocomputing,* 48 (1-4), pp. 85-105.

Sweeney, R. J. (1988) ‘Some new filter rule test: Methods and results’*, J. Finan. Quantitative Anal*., 23, pp. 285–300.

Tan, T. and Wang, J. (2004) 'Support Vector Machine with Hybrid Kernel and Minimal Vepnik-Chervonenskis Dimension', *Transaction on Knowledge and Data Engeneering*, 16 (4), pp.385 – 395.

Trafalis, T. B. and Ince, H. *(2000)* 'Support vector machine for regression and applications to financial forecasting', *Neural Networks, 1(6), pp. 348-353.*

Ullrich C., Seese D., Chalup S. (2006) 'Foreign Exchange Trading with Support Vector Machines', İn Decker R. and Lenz H. J. (eds.) *Advances in Data Analysis: Studies in Classification, Data Analysis, and Knowledge Organization*. Springer: Berlin, pp. 539-546.

Weigend, A.S. (1991) 'Generalisation by weight-elimination applied to currency exchange rate prediction', *IEEE International Joint Conference on Neural Networks,* Nov., 1991, Singapore.

Werbos, P.J. (1975*). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*

Wuthrich, B., Cho, V., Leung, S., Permunetilleke, D., Sankaran, K. and Zhang, J. (1999) 'Daily stock market forecast from textual web data', *in Proc. IEEE Int. Conf. Syst., Man, Cybern.*, pp. 2720–2725.

Xiru, Z (1994) 'Non-linear intraday prediction models for intraday exchange rate trading', *International Journal of Intelligent Systems in Accounting and Finance Management*, 54(3), pp. 234-252.

Yao, J. and Lim Tan, C. (2000) 'A case study on using neural networks to perform technical forecasting of Forex' Neurocomputing, 34(1-4), pp. 79-98.

Yao, J. and Tan, C. L. (2000) 'A case study on using neural networks to perform technical forecasting of forex', *Neurocomputing*, *34*(1), pp. 79-98.

Yao, X. and Liu, Y. (1997) 'A new evolutionary system for evolving artificial neural networks', *IEEE Trans. Neural Netw.*, 8(3), pp. 694–713.

Yoon, Y. and Swales, G. (1991) 'Predicting stock price performance: A neural network approach', *in Preceedings of the 24th Annu. Hawaii Int. Conf. Syst. Sci.*, 4, pp. 156–162.

Yoon, Y., Swales, G. and Margavio, T.M. (1993) 'A comparison of discriminant analysis versus artificial neural networks', *J. Oper. Res. Soc.*, 44(1), pp. 51–60.

Yu, L., Wang, S. and Lai, K. K. (2007) Foreign-exchange-rate forecasting with artificial neural networks, New York: Springer.

Yu,K and Kuperin, A. (2003) 'Using Recurrent Neural Networks To Forecasting of Forex' St.Petersburg State University

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

ZINZALIAN, A. SIMEONOV, D. & ROBEVA, E. (2009) FX FORECASTING WITH HYBRID SUPPORT VECTOR MACHINES.

APPENDIX

STATISTICAL EVALUATION OF ALL MODELS

		EURUSD	EURCHF	USDGBP
Random Walk				
	Annualized Rt	28.51%	27.25%	26.90%
	RMSE	0.0515	0.0634	0.0515
	MAPE	133.78%	135.29%	134.36%
	Sharpe	0.75	0.72	0.71
no change				
	Annualized Rt	0.00%	0.00%	0.00%
	RMSE	0.0062	0.0077	0.009
	MAPE	99.49%	99.69%	99.08%
	Sharpe	N/A	N/A	N/A
Buy and Hold				
	Annualized Rt	1.92%	-3.00%	3.72%
	RMSE	0.1	0.1003	0.1002
	MAPE	8900.02%	45580.80%	8990.35%
	Sharpe	0.2	-0.23	0.26
MACD	Periods	1.7.9	1.6.12	1.7.9
	Annualized Rt	1.92%	4.28%	3.01%
	RMSE	0.0098	0.0139	0.0115
	MAPE	690.38%	5736.72%	1050.53%
	Sharpe	0.2	0.44	0.38
ARMA	Periods	1.3.10	1.7.9	1.6.12
	Annualized Rt	6.01%	17.19%	9.43%
	RMSE	0.0705	0.0634	0.0771
	MAPE	1598.72%	8287.13%	1891.54%
	Sharpe	0.62	1.79	1.23
MLP				
	Annualized Rt	31.79%	26.01%	23.81%
	RMSE	0.024	0.0162	0.0169
	MAPE	107.06%	106.97%	107.13%
	Sharpe	0.83	0.68	0.62
HONN				
	Annualized Rt	32.38%	24.37%	22.16%
	RMSE	0.024	0.024	0.024
	MAPE	107.10%	107.03%	107.21%
	Sharpe	0.85	0.64	0.58
Gene Expression				
	Annualized Rt	34.75%	37.23%	36.31%
	RMSE	0.024	0.024	0.024
	MAPE	103.56%	103.48%	103.60%
	Sharpe	0.91	0.98	0.95
GA-SVR				
	Annualized Rt	30.12%	36.73%	33.70%
	RMSE	0.0189	0.0189	0.0189
	MAPE	135.09%	134.55%	134.79%
	Sharpe	0.79	0.96	0.89

NULL HYPOTHESIS TESTING RESULTS

EURUSD

Null Hypothesis: RT has a unit root

Exogenous: Constant, Linear Trend

Lag Length: 7 (Automatic - based on SIC, maxlag=29)

		t-Statistic
Elliott-Rothenberg-Stock DF-GLS test statistic		-15.78691
Test critical values:	1% level	-3.480000
	5% level	-2.890000
	10% level	-2.570000

*Elliott-Rothenberg-Stock (1996, Table 1)

DF-GLS Test Equation on GLS Detrended Residuals

Dependent Variable: D(GLSRESID)

Method: Least Squares

Date: 04/09/14 Time: 20:47

Sample (adjusted): 10 3574

Included observations: 3565 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
GLSRESID(-1)	-0.605111	0.038330	-15.78691	0.0000
D(GLSRESID(-1))	-0.362540	0.037312	-9.716442	0.0000
D(GLSRESID(-2))	-0.325914	0.035842	-9.093165	0.0000
D(GLSRESID(-3))	-0.290891	0.033908	-8.578842	0.0000
D(GLSRESID(-4))	-0.230868	0.031290	-7.378405	0.0000
D(GLSRESID(-5))	-0.166388	0.027757	-5.994357	0.0000
D(GLSRESID(-6))	-0.116803	0.023166	-5.042015	0.0000
D(GLSRESID(-7))	-0.056754	0.016692	-3.400047	0.0007

R-squared	0.485002	Mean dependent var	-6.88E-06
Adjusted R-squared	0.483989	S.D. dependent var	0.009123
S.E. of regression	0.006554	Akaike info criterion	-7.215336
Sum squared resid	0.152776	Schwarz criterion	-7.201471
Log likelihood	12869.34	Hannan-Quinn criter.	-7.210392
Durbin-Watson stat	2.000744		

EURCHF

Null Hypothesis: RT has a unit root

Exogenous: Constant, Linear Trend

Lag Length: 7 (Automatic - based on SIC, maxlag=29)

		t-Statistic
Elliott-Rothenberg-Stock DF-GLS test statistic		-15.78691
Test critical values:	1% level	-3.480000
	5% level	-2.890000
	10% level	-2.570000

*Elliott-Rothenberg-Stock (1996, Table 1)

DF-GLS Test Equation on GLS Detrended Residuals

Dependent Variable: D(GLSRESID)

Method: Least Squares

Date: 04/09/14 Time: 20:50

Sample (adjusted): 10 3574

Included observations: 3565 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
GLSRESID(-1)	-0.605111	0.038330	-15.78691	0.0000
D(GLSRESID(-1))	-0.362540	0.037312	-9.716442	0.0000
D(GLSRESID(-2))	-0.325914	0.035842	-9.093165	0.0000
D(GLSRESID(-3))	-0.290891	0.033908	-8.578842	0.0000
D(GLSRESID(-4))	-0.230868	0.031290	-7.378405	0.0000
D(GLSRESID(-5))	-0.166388	0.027757	-5.994357	0.0000
D(GLSRESID(-6))	-0.116803	0.023166	-5.042015	0.0000
D(GLSRESID(-7))	-0.056754	0.016692	-3.400047	0.0007

R-squared	0.485002	Mean dependent var	-6.88E-06
Adjusted R-squared	0.483989	S.D. dependent var	0.009123
S.E. of regression	0.006554	Akaike info criterion	-7.215336
Sum squared resid	0.152776	Schwarz criterion	-7.201471
Log likelihood	12869.34	Hannan-Quinn criter.	-7.210392
Durbin-Watson stat	2.000744		

GBPUSD

Null Hypothesis: RT has a unit root
Exogenous: Constant, Linear Trend
Lag Length: 7 (Automatic - based on SIC, maxlag=29)

		t-Statistic
Elliott-Rothenberg-Stock DF-GLS test statistic		-15.78691
Test critical values:	1% level	-3.480000
	5% level	-2.890000
	10% level	-2.570000

*Elliott-Rothenberg-Stock (1996, Table 1)

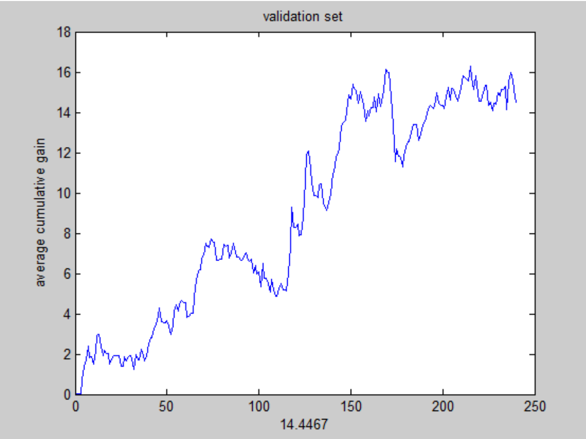
DF-GLS Test Equation on GLS Detrended Residuals
Dependent Variable: D(GLSRESID)
Method: Least Squares
Date: 04/09/14 Time: 20:51
Sample (adjusted): 10 3574
Included observations: 3565 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
GLSRESID(-1)	-0.605111	0.038330	-15.78691	0.0000
D(GLSRESID(-1))	-0.362540	0.037312	-9.716442	0.0000
D(GLSRESID(-2))	-0.325914	0.035842	-9.093165	0.0000
D(GLSRESID(-3))	-0.290891	0.033908	-8.578842	0.0000
D(GLSRESID(-4))	-0.230868	0.031290	-7.378405	0.0000
D(GLSRESID(-5))	-0.166388	0.027757	-5.994357	0.0000
D(GLSRESID(-6))	-0.116803	0.023166	-5.042015	0.0000
D(GLSRESID(-7))	-0.056754	0.016692	-3.400047	0.0007

R-squared	0.485002	Mean dependent var	-6.88E-06
Adjusted R-squared	0.483989	S.D. dependent var	0.009123
S.E. of regression	0.006554	Akaike info criterion	-7.215336
Sum squared resid	0.152776	Schwarz criterion	-7.201471
Log likelihood	12869.34	Hannan-Quinn criter.	-7.210392
Durbin-Watson stat	2.000744		

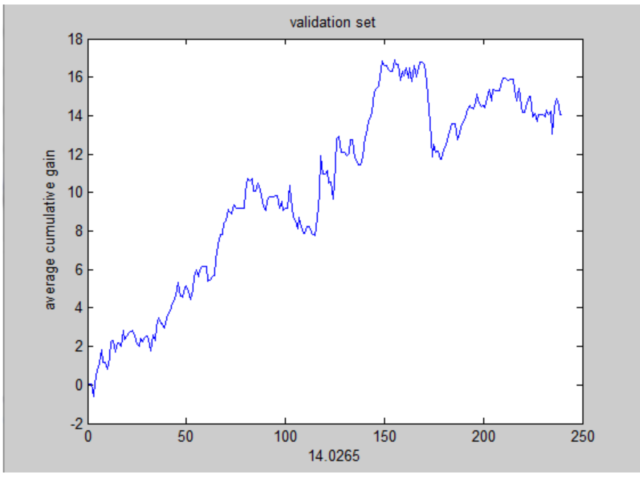
MATLAB GRAPH OUTPUT FOR MLP FOR ALL THREE CURRENCY PAIRS

EURUSD

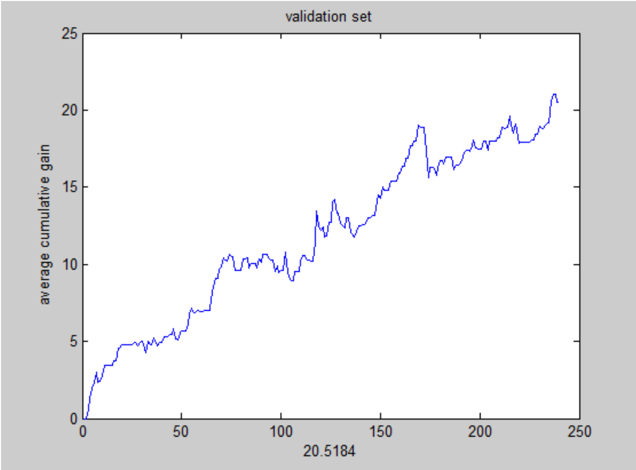


EURCHF

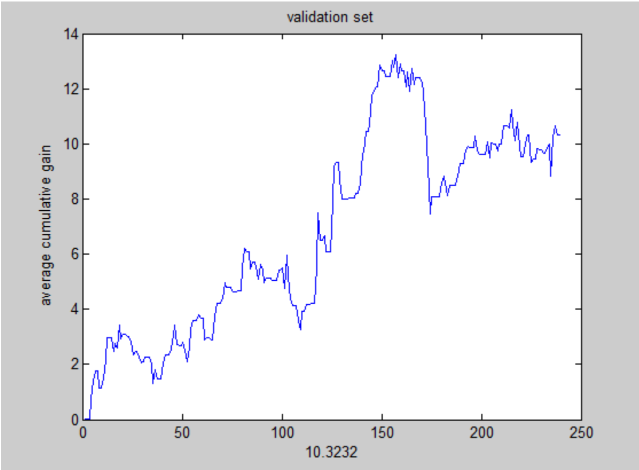

red: Training data green: Test data
Squared error
training blocks
validation set
cumulative squared error
71.3714

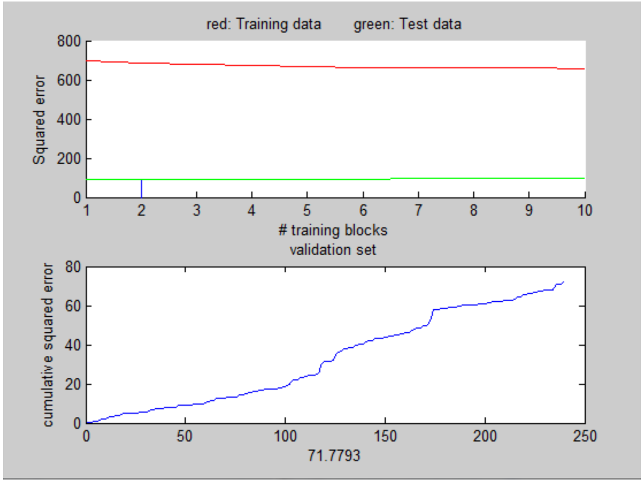

validation set
average cumulative gain
14.0265

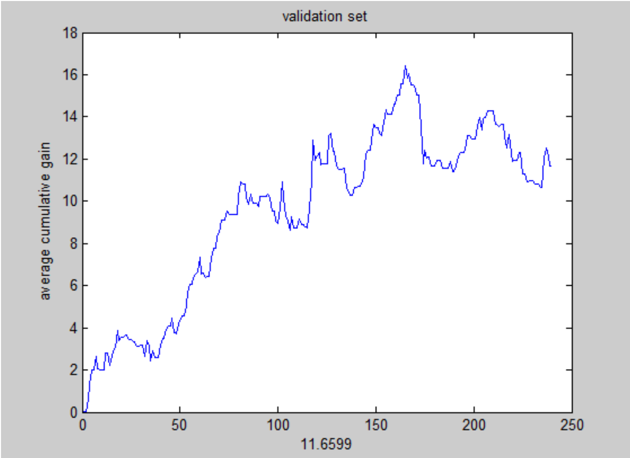

red: Training data green: Test data
gain (%)
training blocks
validation set
cumulative gain
17.1443


validation set
average cumulative gain
0
5
10
15
20
25
0
50
100
150
200
250
20.5184

GBPUSD


validation set
average cumulative gain
0
2
4
6
8
10
12
14
0
50
100
150
200
250
10.3232


red: Training data green: Test data
Squared error
0
200
400
600
800
1
2
3
4
5
6
7
8
9
10
training blocks
validation set
cumulative squared error
0
20
40
60
80
0
50
100
150
200
250
71.7793


validation set
average cumulative gain
11.6599


red: Training data
green: Test data
gain (%)
training blocks
validation set
cumulative gain
10.3808