A FOREIGN EXCHANGE PORTFOLIO MANAGEMENT MECHANISM BASED ON FUZZY NEURAL NETWORKS

Shuo Yao, Michel Pasquier, Chai Quek
Centre for Computational Intelligence, College of Engineering, School of Computer Engineering
Nanyang Technological University, Nanyang Avenue, Singapore 639798

Abstract – The key in foreign exchange (Forex) trading is to pick the right currency to trade at the right time, primarily based on accurate forecast of future exchange rates. This paper presents a novel neuro-fuzzy approach in foreign exchange (Forex) portfolio management to pick the right pairs of currencies to buy and sell with optimized market timing. The proposed mechanism forecasts future BUY/SELL signals before matching these offsetting signals across different currencies to maximize trade returns. This mechanism makes use of fuzzy neural network (FNNs) as a forecasting tool, technical indicators such as moving averages and a novel Portfolio Trade Timing Optimization (PTTO) algorithm to produce an optimized BUY-SELL schedule for the Forex portfolio under management. Experimental results on real world Forex market data shows that the proposed mechanism yields significantly higher profits against various popular benchmarks.

1. Introduction

A Forex transaction is a simultaneous buying of one currency and selling of another. The Forex market is the largest financial market in the world. At its core are exchange rates and market timing. Based on the two, a large variety of financial instruments have been created. Forward, Futures and Swap are the three major Forex instruments. Exchange rates are under the influence of a myriad of factors. To predict exchange rates based on fundamental analysis, which studies all relevant economic and financial indicators, is overwhelmingly complicated. Therefore, technical analysis is chosen as a sound alternative to forecast short-term Forex market movements. Common technical indicators such as double/triple simple/exponential moving averages are used to detect the turning points, or trade signals, in the market trend line. These signals help a Forex portfolio manager pick the most suitable pair of currencies to make a trade at the best timing, so as to achieve maximized portfolio growth.

In this paper, a novel portfolio management mechanism is proposed to sequentially perform forecasting of exchange rate movements using FNNs, detecting trading signals using moving averages, and organizing these signals into an optimized portfolio trading schedule. The three keys to achieve superb trade returns are 1) quality prediction tools, 2) well chosen technical indicators and 3) sound trading strategies to best utilize outputs of the previous two. However, previous studies are either entirely based on prediction tools that lack the sophistication of today's advanced FNNs [1-2], or short of a sound strategy to make use of the outputs from their prediction tools [3]. The mechanism proposed in this paper significantly improves over established portfolio trading models.

The paper is organized as follows. A brief introduction is provided in section 1, section 2 describes the dataset and the FNN prediction tools employed for the experiments presented in this paper. The major steps in the proposed Forex portfolio management mechanism, including the PTTO algorithm is discussed in section 3. The experiments and the results analysis are presented in section 4. Finally, section 5 concludes the paper.

2. Data and Tools

Eight raw time series of historical exchange rates are sourced from online financial data vendor www.oanda.com. Each times series consists of daily average exchange rates of a single currency pair for the period from Sep. 1, 1999 to Sep.1, 2006. The eight raw time series are EUR/USD, JAP/USD, GBP/USD, CHF/USD, HKD/USD, AUD/USD, MYR/USD and SGD/USD. USD is assumed to be the "home currency" in this portfolio. The home currency is usually the currency in which profit or loss is computed. The period of study, from Sep. 1999 to Sep. 2006, is selected to acknowledge the importance of EUR in today's Forex market, as Euro was officially launched in to global circulation on Jan. 1, 1999.

Table 1: Participating Currency in This Project and Their Regimes

Currency	ISO 4217 Code	Regime
United States dollar	USD	Float
Euro zone Euro	EUR	Float
Japanese yen	JPY	Managed Float
British pound sterling	GBP	Float
Swiss franc	CHF	Float
Australian dollar	AUD	Float
Hong Kong Dollar	HKD	Pegged to USD
Singapore Dollar	SGD	Managed float
Malaysia Ringgit	MYR	Pegged to USD

The nine currencies are chosen to be studied in this work not only because of their heavy weights in global Forex trading, but also their suitability in representing all currencies regimes as well as all prominent Forex markets. There are multiple types of currency regimes in use. Not a single currency regime is right for all countries, or for all times. In this study on Forex

1-4244-1340-0/07/$25.00 ©2007 IEEE

Portfolio Management Strategy, the nine sample currencies have been selected to cover three main categories of currency regimes (Table 1); namely: USD peg, float and managed float. This selection of currencies allows a wide applicability of the results in this study, However, this is probably less optimal due to the inclusion of some hard-pegged and closely managed currencies such as HKD and MYR.

Table 2: Detailed Test Time Series Segmentation

Set	Training Set Period	Test Set Period
1	Sep. 99 to Sep. 03	Sep. 03 to Sep. 04
2	Sep. 00 to Sep. 04	Sep. 04 to Sep. 05
3	Sep. 01 to Sep. 05	Sep. 05 to Sep. 06

As illustrated in Table 2, three sets of training-testing segmentation are constructed from on each of the 7-year time series. This approach of segmentation is used to deal with the non-stationary nature of the signals in this work. The difficulty with this approach is the reduction in the already small number of training data points. The size of the training set is a tradeoff between the level of noise and non-stationarity nature of the data. If the training set is too small, the noise level makes it harder to estimate appropriately. If the training set is too large, the non-stationarity of the data will mean that less relevant data will be used to generate the estimator. In this work, the size of training set is fixed at four times the size of the testing set, meaning 80% of the data is used for training and the remaining 20% for testing. In addition, the window size is another important factor to be considered. Common window sizes for short, medium and long term predictions are 10, 40 and 200 days respectively. Prediction based on technical analysis is most helpful in the short term. Therefore, the window size of 10 is used in this work.

There are many different neural networks available as the function approximator in the trading system. The fuzzy cerebellar model articulation controller (FCMAC) [4] is selected as the default FNN prediction tool in this study. The CMAC architecture is chosen due to its strengths in localized generalization, rapid algorithmic computation, incremental training and good functional representation [5]. FCMAC involves the mapping of various fuzzy inference schemes onto the CMAC architecture transforming it from a “black box” into a “white box” [6]. Two updated fuzzy inference schemes, namely, Yager (Yager) [6] and Compositional Rule of Inference (CRI) [7] are explored and compared. The two schemes are found to be of similar standard of performance through experimentation. Yager is chosen as the default fuzzy inference scheme in this paper.

Inspired by the neurophysiologic theory of the cerebellum, Albus developed a mathematical model called Cerebellar Model Articulation Controller (CMAC) in an attempt to derive an efficient computational algorithm for use in manipulator control [4]. Different from the global weight adjustment of the back-propagation neural network, the main advantage of the CMAC network is its ability to generalize with good learning behavior due to its localized weight adjustment that exhibits faster learning speed and simpler hardware implementation. The local minimum problem that exists in multilayered neural network learning does not exist in CMAC. In addition, the CMAC network requires only a small number of computations per output as compared to other well-known neural networks.

Contemporary machine learning involves a dichotomy in choosing whether to use global learning or local learning. Global learning captures global characteristics of the data. Local learning, on the other hand only focuses on useful local information from the observed data. This approach is adopted by the CMAC family. Both approaches have serious limitations; namely: global learning is often computationally inefficient and local learning does not grasp the overall structure of the data, which may prove to be critical for guaranteeing better performance. To combine complementary advantages of them, a novel learning paradigm, called regional learning, is proposed in this report. A novel algorithm known as Neural Inference Patching *(NIP)* algorithm is designed to optimize regional learning. When an input stimulus is received, NIP checks if the input fails to find any local fuzzy rule with matching antecedent in the rule network (the input has fired into a *hole* in the rule neuron network). When this occurs, NIP performs an expansionary neighborhood search to select the most relevant fuzzy rules in the region. The NIP algorithm is integrated with the FCMAC-Yager architecture [8], resulting in a novel regional-learning fuzzy neural network termed the FCMAC-NIP.

The fuzzy rule base in the FCMAC model is realized as a connectionist structure. The FCMAC model partitions the input and the output spaces into fuzzy sets (clusters). The structure of the model is usually arranged in the form of a multi-dimensional look-up table based on the centroids of all the computed input clusters – this is essentially a hypercube based semantic associative memory implementing the Yager fuzzy rule inference. Each of the fuzzy rules generated is identified by a unique index pattern and the cell content is addressed by the index pattern. During the training phase, each cell selects the most appropriate fuzzy output cluster to store as the output association. Let I denote the total number of input dimensions for a given problem, U_1, U_2, ...,U_i, …, U_I be the universe of discourses for the respective input dimensions and $|U_1|$,…, $|U_2|$,…,$|U_i|$,…, $|U_I|$ are the total number of clusters in each input dimension. Therefore, the maximum number of identified fuzzy rules is

$$|U|= |U_1|\times|U_2|\times\ldots\times|U_i|\times\ldots\times|U_I|.$$

Due to limited training data or the rule reduction function embedded within the local-learning neural network, the actual number of fuzzy rules identified, μ, are often less than the maximum number of possible fuzzy rules. Let

$$\rho = \frac{\mu}{|U|}, \rho \in [0,1]$$

If $\rho < 1$, there are *holes* (memory cells that do not infer or are not associated with any cluster in the output dimension) in the rule neuron network. For example, in a six-input-one-output network, there are 5 clusters found on each input dimension and therefore $|U| = 5^6$. If the network only identifies 1000 fuzzy rules (an amount of rules that is already humanly not interpretable), $\rho = \frac{1000}{5^6} = 6.4\%$. This means only 6.4% of the potential fuzzy rule neurons in the memory are identified. The remaining 93.6% form the *holes* in the rule-base that can be potentially *patched* by the proposed NIP algorithm. In another words, $(1-\rho)$ is a measure of the *sparseness* of the rule network. On average, the probability that a cell is confirmed to be inferring to an identified fuzzy rule is only ρ; and there is a $(1- \rho)$ chance that a memory cell selected actually has no corresponding fuzzy rule. Obviously, our pursuit for higher-dimensionality networks with higher interpretability will result in a diminishing ρ value, indirectly indicating that there are more and more *holes* on which NIP is able to exert its effect. This is based on the premise that the output decision surface is "continuous", i.e., neighboring cells share similar semantic patterns.

A *hole* in an inference rule network is essentially a region of absent rule neurons. Rule neurons surrounding the *hole* are categorized into "orbits" according to their distances to the *hole*. The Expansionary Neighborhood Search designed in NIP searches for neighbor neurons from inner orbits to outer orbits, e.g. the search extends from *Orbit i* to *Orbit i+1* if no rule neurons can be found on *Orbit i*. The innermost orbit is defined as Orbit 0. Intuitively, outer orbits have lower *relevance* than the inner orbits. Therefore, it is wise to keep the maximum number of orbits to search, denoted as M, small. Essentially, this means to limit the search for neighboring neurons in the region, instead of stretching the search too far into areas where neurons are irrelevant and thus insignificant. Empirical results indicate that M = 3 provides an optimal limit on expansion. Moreover, a small M also helps to make NIP fast and computationally efficient. However, as a tradeoff, a small M also disables NIP to patch large *holes*.

Let

Si = the collection of indices of selected clusters in the ith input dimension, $i \in \{1, I\}$

I = is the total number of input dimensions

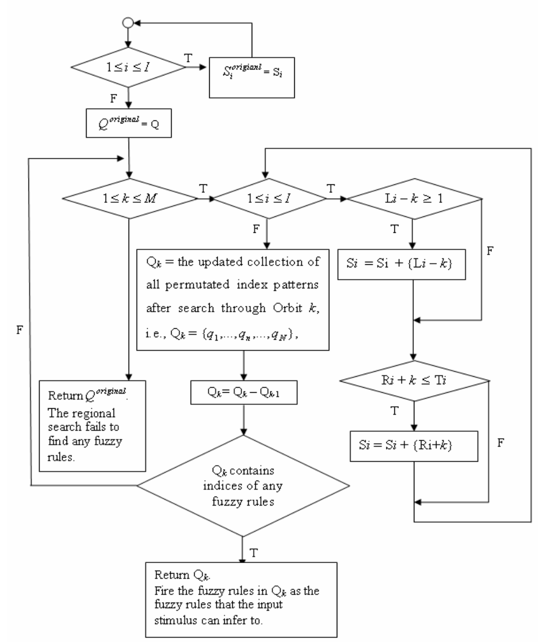
Ti = the total number of clusters in the ith dimension

k = the index of the orbit being searched, $k \in \{0, M\}$ and $k = 0$ initially

M = the maximum number of orbits to search (user-defined parameter)

Q = the original collection of all permutated index patterns, i.e., $Q = \{q_1,...,q_n,...,q_N\}$,

Si contains either one or two clusters initially, i.e., $Si = \{Li, Ri\}$. In cases where $Li \neq Ri$, then $Li + 1 = Ri$ because they are the immediate neighbors. In cases when only one cluster is activated, $Li = Ri$, it has to be the cluster located at one of the two ends of the ith dimension. Specifically, if $Li = Ri = 1$, the activated cluster is at the *low end* or *left end* of the dimension; if $Li = Ri = Ti$, the activated cluster is located at the *high end* or *right end* of the dimension. The flowchart of the proposed expansionary search is as follows:



3. Forex Portfolio Management Mechanism

Performing accurate predictions using a chosen FNN prediction tool is only the first step in a Forex Portfolio Management Mechanism. There are 5 steps in the proposed Portfolio Management Mechanism, as discussed below.

Step 1: Separately train, validate and predict with historical exchange-rate series of currency Xi against the chosen home currency (USD), i={1..8}.

Step 2: Based on the predicted exchange rate series produced by FCMAC, A time-series (for a single currency) of {1, 0,-1} can be derived using trend-following indicators such as SMA&LMA.

Step 3: Assemble the output for each currency from Step 2 into a preliminary buy-sell schedule for the portfolio.

Step 4: Apply the proposed Portfolio Trade Timing Optimization algorithm to optimize the preliminary schedule into the final trading schedule.

Step 5: Evaluate the performance of the portfolio, which depends on 1) the accuracy of predictions produced by FCMAC and 2) the portfolio optimization algorithm. Evaluation is carried out in two methods: rate of return and Sharpe ratio.

Step 1: Apply the historical time series of currency X_i, i = {1… 8} to the FNN prediction tool. From any X_i, three sets of prediction output series will be produced: predicted series for year 2004, 2005 and 2006, respectively. They are referred to as P1, P2 and P3, while the real historical data for the same periods are referred to as R1, R2 and R3. Hence, there are in total 3 * 8 = 24 series of predicted exchange rates in the portfolio.

Step 2: Based on the predicted exchange rate series produced by FCMAC, A series (for a single currency) of {1, 0, -1} can be derived using trend-following indicators such as Short-term Moving Average and Long-term Moving Average. Moving averages are one of the most popular and easiest to use tools available for technical analysis. They are effective low pass filters that smooth a data series and make it easier to sport trends. They also form the building blocks of many other technical indicators and mechanisms, such as the Portfolio Trade Timing Optimization algorithm proposed in this project.

A short-term moving average series (SMA) and a long-term moving average series (LMA) are computed based on the predicted time series produced by FCMAC-NIP. Both SMA and LMA are Simple Moving averages, which are formed by computing the average value over a specified number of past days. In this work, SMA is the simple moving average of the past 5 days whereas LMA is the simple moving average of the past 30 days. The choice of the number of averaging days is more an art than a science. Empirical results obtained in this research effort suggest the judicious choices of 5 and 30 days respectively for SMA and LMA.

Moving averages can be used for trend identification. If the shorter moving average is above the longer moving average, the trend is considered going up. On the other hand, if the shorter moving average is below the longer moving average, the trend is considered downward. The intersections of SMA and LMA are captured as the points where a change in market trend is detected. Therefore, the precise detection of these turning points is crucial in the pursuit of best market timing for Forex transactions.

An intersection of SMA and LMA that signals a downward trend will be seen as an indication of potential upward pressure on the value of the currency against the home currency, which is assumed to be USD in this case, i.e. a BUY signal. By the same token, an intersection that signals an upward trend will be recognized as an indication of potential downward pressure on the value of the currency against USD, i.e. a Sell signal.

A major disadvantage of detecting BUY and SELL signals is the lagging nature of all moving average features – indicators will always be a step behind. More importantly, the sizes of the lag are uncertain and vary across time. Therefore, a parameter named *Lag Corrector* (*lc*), intended to correct lags of moving averages, is employed. Empirical results derived in this study suggest a $lc = 10$ is a good estimate of the optimal Lag Corrector. Thus, if a BUY or SELL signal is detected on Day x of a predefined period, such as a year, ($0<x<365$ or 366), the corrected trading day x' can be determined as $x' = x - lc$.

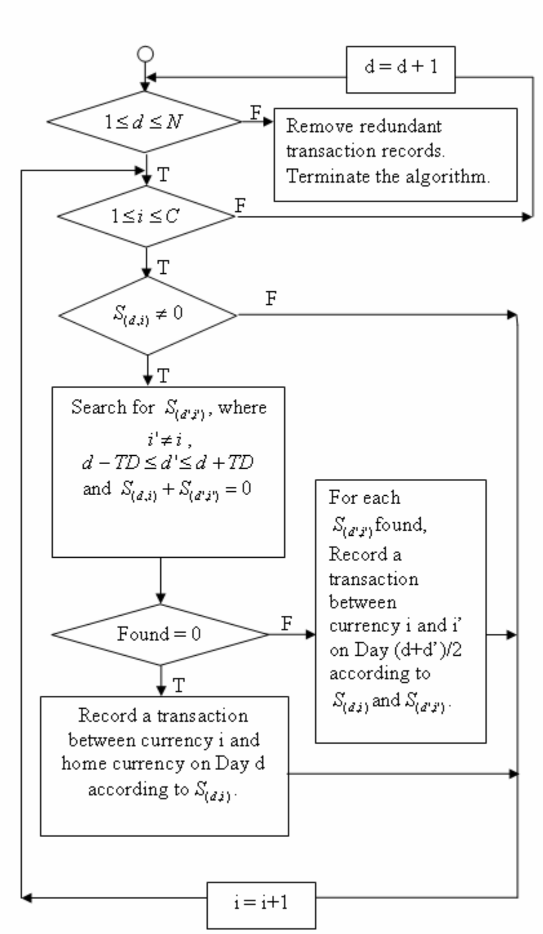
Step 3: Assemble the output for each currency from Step 2 into a preliminary buy-sell schedule for the portfolio. In this preliminary schedule, "1", "-1" and "0" is used to denote BUY, SELL and HOLD signals respectively.

Step 4: Apply the proposed Portfolio Trade Timing Optimization algorithm to optimize the preliminary schedule into an optimized trading schedule. The goal of the optimization algorithm is to match the closest offsetting transactions. For example, if there is a BUY signal in JPY on Day 100, the algorithm will try to find a nearest SELL signal around Day 100 in another currency in the portfolio. Every trade in the optimized trading schedule appears only once in order to avoid double counting. A parameter called Time Discrepancy (TD, defaulted to be 10) can be set by the user to adjust the maximum allowed distance between an offsetting pair of BUY and SELL signals. If no offsetting signal can be found within TD, a transaction will be carried out between that currency and the home currency. Following the previous example, if no matching SELL signal can be found in any other currencies in the portfolio, a transaction will be exercised to buy Japanese Yen with the home currency, which is USD in this case. Moreover, if multiple offsetting trading signals exist within TD, a transaction will be carried out with each of the detected trading signals. A flow chart of the Portfolio Trade Timing Optimization (PTTO) algorithm is shown below:

Let

N = no. of days in the studied period
C = no. of foreign currencies in the portfolio
$S_{(d, i)}$ = the signal detected on Day d for currency i; S = {1, 0, -1}

Found = no. of offsetting trade signals found



Step 5: Evaluate the performance of the portfolio, which depends on 1) the accuracy of predictions produced by FCMAC and 2) the portfolio optimization algorithm. The evaluation is carried out in three methods: rate of return, Sharpe ratio and benchmarking against market indices and basic trading strategy such as Buy-and-Hold.

4. Experiments Results

The output of Step 1 consists of the predicted values of exchange rates of all currencies in the portfolio compared against their corresponding true time series values. For example, as shown in Figure 1a and Figure 1b, the two lines represent the forecast outputs of FCMAC and actual historical exchange rate time series on JPY/USD and GBP/USD within the period of Sep. 03 to Sep. 06.

In Step 2 of the Portfolio Management Mechanism, trading signals are derived from the predicted exchange rate time series with timing-analysis using both the Short Moving Average and Long Moving Average lines. An example of the trading signal capturing technique is demonstrated in Figure 2a. On Day 163, SMA climbs above LMA, signaling a rise in the value of GBP/USD, and thus a depreciation of GBP against USD – a SELL signal is captured for GBP. Similarly, on Day 234, SMA dives below LMA, signaling a drop in the value of GBP/USD, and thus an appreciation of GBP against USD – a BUY signal is captured for GBP. Given the default Log Corrector, *lc* = 10, the corrected timings of the two captured signals are therefore Day 153 and 224.

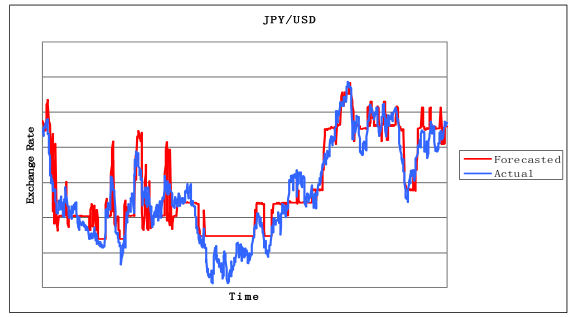


Figure 1a: Output of Step 1 on JPY/USD

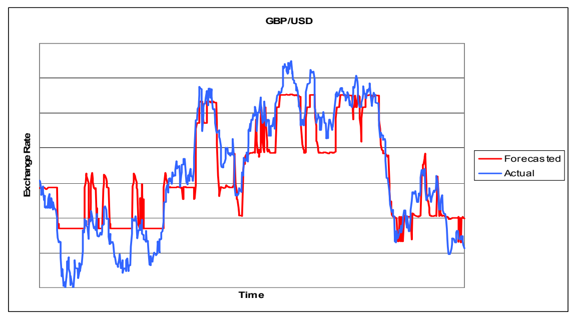


Figure 1b: Output of Step 1 on GBP/USD

To test the accuracy of the captured trading signals, Day 153 and 224 are put onto the true GBP/USD time series of the same period of time. In Figure 2b, GBP/USD rises steadily after the SELL signal on Day 153 and drops sharply after the BUY signal on Day 224. Both the two trading signals are profitable. As shown in Table 3, a portion of the output of Step 2 for GBP/USD is tabulated.

In Step 3, trading signals captured in Step 2 are assembled into preliminary trading schedules for different annual periods. For convenient reference, every relevant currency is given an index (Table 4). Subsequently, a BUY list and SELL list are presented with all trading signals indexed in a chronological order. BUY signals are indexed with positive integers and SELL signals are indexed with negative integers.

Table 3: Trading Signals Captured for GBP/USD

09/03-09/04		09/04-09/05		09/05-09/06	
Date	**BUY/ SELL**	**Date**	**BUY/ SELL**	**Date**	**BUY/ SELL**
30	1	87	1	56	1
65	-1	130	-1	69	-1
77	1	151	1	103	1
104	-1	163	-1	120	-1
171	1	171	1	133	1

In Step 4, the PTTO algorithm is applied to derive the optimized trading schedule from the preliminary trading schedule. An example of the output of PTTO is shown in Table 5. A BUY or SELL signal indexed with "0" indicates that an offsetting trade signal cannot be found among the eight participating currencies in the portfolio, therefore the home currency, USD, is used to trade against this signal.

The first row of Table 5 can be interpreted as follows: The 11th BUY signal detected within this period of time is matched with the 6th SELL signal of the same period. The discrepancy between these two signals is 1 day. GBP (currency index predefined as 4) is bought and AUD (currency index predefined as 1) is sold in this trade. The trade should be carried out on Day 156 of that period.

Table 4: Currency Indexing

Currency	Index	Currency	Index	Currency	Index
AUD	1	GBP	4	MRY	7
CHF	2	HKD	5	SGD	8
EUR	3	JPY	6	USD	9

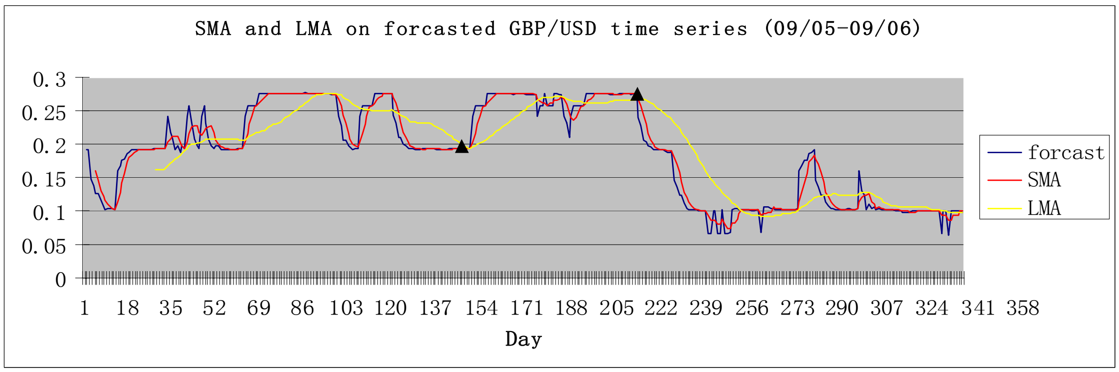


Figure 2a: Detection of Trading Signals using SMA and LMA

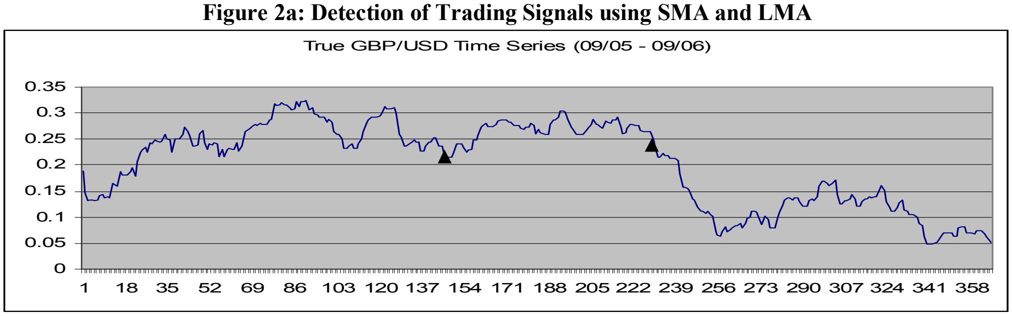


Figure 2b: Examine the Detected Trading Signals on Historical Data

In Step 5, performance analysis and benchmarking are carried out to assess the overall successfulness of the proposed Forex Portfolio Management Mechanism in the following three approaches; namely: rate of return, Sharpe ratio and benchmarking against human experts, market indices and basic trading strategy such as Buy-and-Hold. Detailed figures are available in Table 6. As shown in Figure 3, the managed portfolio outperforms the market indices, especially when the hard-pegged currencies in the portfolio are not considered. In Figure 4, the performance of the proposed mechanism is far more superior in comparison to the simple Buy-And-Hold strategy.

Table 5: Partial output of the PTTO algorithm
(Sep. 03 – Sep. 04)

Buy Signal	Sell Signal	TD	Buy Currency	Sell Currency	Transaction Point
11	-6	1	4	1	156
11	-7	3	4	6	158
16	-11	3	1	6	318
17	-12	4	6	1	339
3	-1	14	3	8	22
4	-1	14	4	8	22
1	0	0	1	9	2
2	0	0	5	9	21

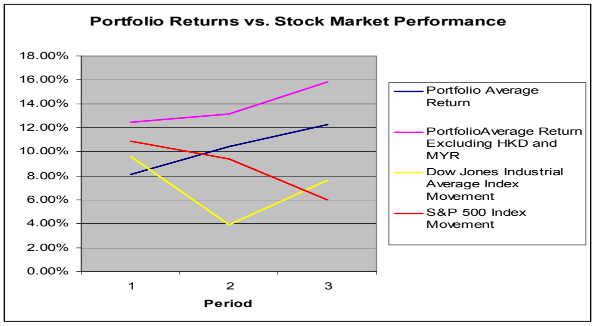


Figure 3: Portfolio Returns Compared Against the General Market

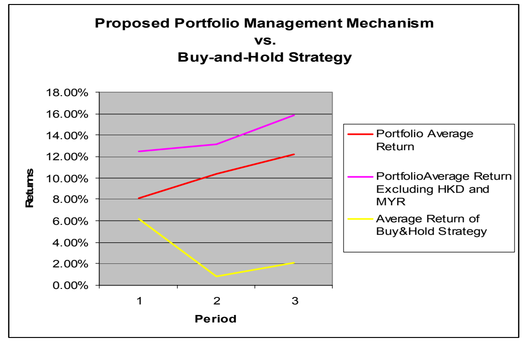


Figure 4: Proposed Portfolio Management Mechanism vs. Buy-and-Hold Strategy

Table 6: Portfolio Performance Evaluation

Portfolio Performance Evaluation			
Year	**2004**	**2005**	**2006**
Average Return	8.10%	10.38%	12.26%
Average Return Excluding HKD and MYR	12.47%	13.16%	15.81%
Average Volatility	2.50%	2.30%	2.22%
Average 1-year T-bill Yield	1.52%	3.02%	4.66%
Average Return of Buy&Hold Strategy	6.14%	0.82%	2.09%
Sharpe Ratio	2.63	3.20	3.42
Dow Jones Industrial Average Index Movement	9.60%	3.90%	7.60%
S&P 500 Index Movement	10.87%	9.40%	6.00%
Compounded Portfolio Return 04-06	**33.95%**		
Compounded Portfolio Return 04-06, Excluding HKD and MYR	**47.39%**		

As illustrated in Table 6, the average annual return of the portfolio is 10.25% and the three-year compounded portfolio return is 33.95%. In practice, currencies under hard-peg regimes, such as HKD and MYR, are less frequently involved in a Forex trader's Portfolio than currencies under float regimes. By excluding HKD and MYR, the average annual return of this portfolio is 13.81% and the three-year compounded portfolio return is 47.39% instead. According to [9], "The minimal acceptable performance level for a serious amateur is a return equal to 2 times the current rate on T-Bills. The goal of a serious amateur is to generate a 20% annual return on equity. For experts, their returns are steadier, but not necessarily higher than those of serious amateurs. Experts have to continue outperforming T-Bills, because to fall behind them would be ridiculous. Trading a serious amount of money year after year, just staying above 20% annual return is a very good performance." Based on the general standards above, the proposed portfolio trading mechanism can be easily qualified as a serious amateur model and is very close to the level of a human expert.

The Sharpe ratio is a measure of risk-adjusted performance of an investment asset, or a trading strategy. It is defined as:

$$S = \frac{E[R - R_f]}{\sigma}$$

where R is the portfolio return, R_f is the risk free rate of return (in this case, the 1-year T-bill rate), and

$$\sigma = \sqrt{Var[R - R_f]}$$

is the standard deviation of the excess return, or volatility. A higher Sharpe ratio gives more return for the same risk, thus, investors are often advised to pick investments with high Sharpe ratio values. The average Sharpe ratio of this portfolio is found to be 3.08, meaning that, on average, for 1 dollar in risk, the proposed portfolio management mechanism is capable of producing 3 dollars of return.

The novel Portfolio Trade Timing Optimization algorithm proposed in this project is core to the success of the entire portfolio management mechanism. As an algorithm, it has a wide range of merits; namely:

1) Applicability: The PTTO algorithm can be applied to a wide range of financial instruments. Most trading activities include a pair of BUY and SELL of assets. PTTO essentially provides a convenient way of matching a BUY with a SELL and vice versa in the time domain.

2) Consistency: Publicly known trading strategies usually perform better in certain situations than in others. Investors often have to take great caution in choosing the most suitable trading strategies. However, PTTO is designed to be consistent regardless of the trends and situations the market is in. As shown in Figure 3, PTTO consistently produce satisfactory returns regardless of the ups and downs in the DJIA and S&P 500 index.

3) Scalability. The number and amount of assets in the portfolio that a trader is able to manage is limited. However, PTTO does not suffer from this problem. It is able to handle any number of assets in a portfolio.

4) Flexibility. PTTO has a critical parameter, Time Discrepancy (TD). The choice of a suitable TD is determined by the user. This feature lends PTTO considerable flexibility. Generally, the larger the TD, the more risk there is to bear for a transaction generated by PTTO.

5) Potential. PTTO can be regarded as a platform from which new features and further enhancement can be built. For example, when a BUY signal is found to be able to match with multiple SELL signals of sufficient proximity in the time domain, the existing version of PTTO matches it against all of the SELL signals indiscriminately. A “selection function” can be added to PTTO to identify the trade signal that results in maximized return, i.e. the gradient at a trading signal may be used as a possible basis of such selection. The empirical results achievable with the existing version of PTTO are already encouraging, there is reason to believe further enhancements can make PTTO an even more attractive algorithm.

Conclusion

In this paper, a Forex portfolio management mechanism that leverages on the strengths of advanced fuzzy neural networks as sound prediction tools and a novel Portfolio Trading Timing Optimization algorithm is proposed. In addition, the proposed mechanism employs the popular technical analysis indicators such as moving averages. Extensive experiments have been carried out to test the performance of the proposed trading mechanism and the results are absolutely encouraging. In comparison to human trading experts, the proposed mechanism can be easily qualified as a serious amateur and is very close to the standard of experts in terms of risk-adjusted investment returns. The proposed PTTO algorithm has many merits, ranging from its consistency, flexibility to its potential to become a good platform from which new functions and enhancements can be built. Future enhancement of PTTO can be directed towards the inclusion of a Generic Algorithm that optimizes important parameters automatically.

References

[1] M. Adya and F. Collopy, “How effective are neural networks at forecasting and prediction? A review and evaluation,” *J. Forecasting*, vol. 17, no. 5–6, pp. 481–495, Sep. 1998.

[2] R. J. Kuo, C. H. Chen, and Y. C. Hwang, “An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network,” *Fuzzy Sets Syst.*, vol. 118, no. 1, pp. 21–45, Feb. 2001.

[3] R. Gencay, “The predictability of security returns with simple technical trading rules,” *J. Empir. Finance*, vol. 5, no. 4, pp. 347–359, Oct. 1998.

[4] J.S. Albus. “A theory of cerebellar function”, Math Biosci, 10:pp.25-61, 1971.

[5] W. T. Miller, F.H. Glanz and L. G. Kraft. “CMAC: An Associative Neural Network Alternative to Back-propagation”, Proc. IEEE, 78:pp.1561-1567, 1990.

[6] Quek and A. Singh “*POP-Yager: A novel self-organizing fuzzy neural network based on the Yager inference*” Expert Systems with Applications, pp. 229-242, Vol. 29, 2005.

[7] A. Zadeh. “Calculus of fuzzy restrictions”, in Fuzzy Sets and Their Applications to Cognitive and Decision Processes, New York: Academic, pp.1-39.[59] C, 1975.

[8] Sim, J. Tung, W. L. Quek, C. *FCMAC-Yager: A Novel Yager-Inference-Scheme-Based Fuzzy CMAC* Neural Networks, IEEE Transactions on Nov.2006, Vol:17, Issue:6 pp: 1394-1410.

[9] “Trading for a living” by Dr. Alexander Elder