Kalman Filters and Neural Networks in Forecasting and Trading

Georgios Sermpinis[1], Christian Dunis[2], Jason Laws[3], and Charalampos Stasinakis[4]

[1] University of Glasgow Business School
georgios.sermpinis@glasgow.ac.uk
[2] Liverpool Business School
cdunis@tiscali.co.uk
[3] University of Liverpool Management School
J.Laws@liverpool.ac.uk
[4] University of Glasgow Business School
c.stasinakis.1@research.gla.ac.uk

Abstract. The motivation of this paper is to investigate the use of a Neural Network (NN) architecture, the Psi Sigma Neural Network, when applied to the task of forecasting and trading the Euro/Dollar exchange rate and to explore the utility of Kalman Filters in combining NN forecasts. This is done by benchmarking the statistical and trading performance of PSN with a Naive Strategy and two different NN architectures, a Multi-Layer Perceptron and a Recurrent Network. We combine our NN forecasts with Kalman Filter, a traditional Simple Average and the Granger- Ramanathan's Regression Approach. The statistical and trading performance of our models is estimated throughout the period of 2002-2010, using the last two years for out-of-sample testing. The PSN outperforms all models' individual performances in terms of statistical accuracy and trading performance. The forecast combinations also present improved empirical evidence, with Kalman Filters outperforming by far its benchmarks.

Keywords: Psi Sigma Network, Recurrent Network, Forecast Combinations, Kalman Filter.

1 Introduction

The term of Neural Network (NN) originates from the biological neuron connections of human brain. The artificial NNs are computation models that embody data-adaptive learning and clustering abilities, deriving from parallel processing procedures [34]. The NNs are considered a relatively new technology in Finance, but with high potential and an increasing number of applications. However, their practical limitations and contradictory empirical evidence lead to skepticism on whether they can outperform existing traditional models.

The motivation of this paper is to investigate the use of a Neural Network (NN) architecture, the Psi Sigma Neural Network (PSN), when applied to the task of forecasting and trading the Euro/Dollar (EUR/USD) exchange rate and to explore the

C. Jayne, S. Yue, and L. Iliadis (Eds.): EANN 2012, CCIS 311, pp. 433–442, 2012.

utility of Kalman Filters in combining NN forecasts. This is done by benchmarking the statistical and trading performance of PSN with a Naive Strategy and two different NN architectures, a Multi-Layer Perceptron (MLP) and a Recurrent Network (RNN). We combine our NN forecasts with Kalman Filter, a traditional Simple Average and the Granger- Ramanathan's Regression Approach (GRR). The statistical and trading performance of our models is estimated throughout the period of 2002-2010, using the last two years for out-of-sample testing. In terms of our results, the PSN outperforms all models' individual performances in terms of statistical accuracy and trading performance. The forecast combinations also present improved empirical evidence, with Kalman Filters outperforming by far its benchmarks.

Section 2 is a short literature review and Section 3 follows with the description of the EUR/USD ECB fixing series, used as our dataset. Sections 4 and 5 give an overview of the forecasting models and the forecast combination methods we implemented respectively. The statistical and trading performance of our models is presented in Section 6. Finally, some concluding remarks are summarized in Section 7.

2 Literature Review

The most common NN architecture is the MLP and seems to perform well at time-series financial forecasting [16], although the empirical evidence can be contradictory in many cases. Ince and Trafalis [14] forecast the EUR/USD, GBP/USD, JPY/USD and AUD/USD exchange rates with MLP and Support Vector Regression and their results show that MLP achieves less accurate forecasts. On the other hand, Tenti [21] and Dunis and Huang [3] achieved encouraging results also by using RNNs to forecast the exchange rates. PSNs were first introduced by Ghosh and Shin [7] as architectures able to capture high-order correlations. Ghosh and Shin [7, 8] also present results on their forecasting superiority in function approximation, when compared with a MLP network and a Higher Order Neural Network (HONN). Satisfactory forecasting results of PSN were presented by Hussain *et al.* [13] on the EUR/USD, the EUR/GBP and the EUR/JPY exchange rates using univariate series as inputs in their networks. Bates and Granger [1] and Newbold and Granger [17] suggested combining rules based on variances-covariances of the individual forecasts, while Granger and Ramanathan [10] presented a regression combination forecast framework with encouraging results. According to Palm and Zellner [18], it is sensible to use simple average for combination forecasting, while Deutsch *et al.* [2] achieved substantially smaller squared forecasts errors combining forecasts with changing weights. Time-series analysis is often based on the assumption that the parameters are fixed. However, Harvey [12] and Hamilton [11] both suggest using state space modeling, such as Kalman Filter, for representing dynamic systems where unobserved variables (so-called 'state' variables) can be integrated within an 'observable' model. According to Goh and Mandic [9] the recursive Kalman Filter is suitable for processing complex-valued nonlinear, non-stationary signals and bivariate signals with strong component correlations. Kalman Filter is also considered an optimal time-varying financial forecast for financial markets [4]. Terui and van Dijk [22] also suggest that the combined forecasts perform well, especially with time varying coefficients.

3 The EUR/USD Exchange Rate and Related Financial Data

The European Central Bank (ECB) publishes a daily fixing for selected EUR exchange rates: these reference mid-rates are based on a daily concentration procedure between central banks within and outside the European System of Central Banks, which normally takes place at 2.15 p.m. ECB time. The reference exchange rates are published both by electronic market information providers and on the ECB's website shortly after the concentration procedure has been completed. Although only a reference rate, many financial institutions are ready to trade at the EUR fixing and it is therefore possible to leave orders with a bank for business to be transacted at this level.

In this paper, we examine the EUR/USD over period 2002 -2010, using the last two years for out-of-sample.

Table 1. The EUR/USD Dataset - Neural Networks' Training Dataset

PERIODS	TRADING DAYS	START DATE	END DATE
Total Dataset	2295	3/01/2002	31/12/2010
Training Dataset *(In-sample)*	1270	3/01/2002	29/12/2006
Test Dataset *(In-sample)*	511	02/01/2007	31/12/2008
Validation Dataset *(Out-of-sample)*	514	02/01/2009	31/12/2010

The graph below shows the total dataset for the EUR/USD and its volatile trend since early 2008.

Fig. 1. EUR/USD Frankfurt daily fixing prices

To overcome the non-stationary issue, the EUR/USD series is transformed into a daily series of rate returns. So given the price level P_1, P_2,…, P_t, the return at time t is calculated as:

$$R_t = \left(\frac{P_t}{P_{t-1}}\right) - 1 \tag{1}$$

In the absence of any formal theory behind the selection of the inputs of a neural network, we conduct some neural networks experiments and a sensitivity analysis on a pool of potential inputs in the training dataset in order to help our decision. Our aim is to select the set of inputs for each network which is the more likely to lead to the best trading performance in the out-of-sample dataset. In our application, we select as inputs sets of autoregressive terms of the EUR/USD, EUR/GBP and the EUR/JPY exchange rates, based on the higher trading performance for each network in the test sub-period.

4 Forecasting Models

4.1 Naive Strategy

In this paper we use the Naive Strategy in order to benchmark the efficiency of the NNs' trading performance. The Naive Strategy is considered to be the simplest strategy to predict the future. That is to accept as a forecast for time *t+1,* the value of time *t,* assuming that the best prediction is the most recent period change. Thus, the model takes the form: $\hat{Y}_{t+1} = Y_t$ (2) where Y_t is the actual rate of return at time t and $\hat{Y}_{t+1}$ is the forecast rate of return at time *t+1.*

4.2 Neural Networks (NNs)

Neural networks exist in several forms in the literature. The most popular architecture is the Multi-Layer Perceptron (MLP). A standard neural network has at least three layers. The first layer is called the input layer (the number of its nodes corresponds to the number of explanatory variables). The last layer is called the output layer (the number of its nodes corresponds to the number of response variables). An intermediary layer of nodes, the hidden layer, separates the input from the output layer. Its number of nodes defines the amount of complexity the model is capable of fitting. In addition, the input and hidden layer contain an extra node called the bias node. This node has a fixed value of one and has the same function as the intercept in traditional regression models. Normally, each node of one layer has connections to all the other nodes of the next layer.

The network processes information as follows: the input nodes contain the value of the explanatory variables. Since each node connection represents a weight factor, the information reaches a single hidden layer node as the weighted sum of its inputs. Each node of the hidden layer passes the information through a nonlinear activation function and passes it on to the output layer if the calculated value is above a threshold.

The training of the network (which is the adjustment of its weights in the way that the network maps the input value of the training data to the corresponding output value) starts with randomly chosen weights and proceeds by applying a learning algorithm called backpropagation of errors [19].The learning algorithm simply tries to find those weights which minimize an error function (normally the sum of all squared differences between target and actual values). Since networks with sufficient hidden nodes are able to learn the training data (as well as their outliers and their noise) by heart, it is crucial to stop the training procedure at the right time to prevent overfitting (this is called 'early stopping'). This can be achieved by dividing the dataset into 3 subsets respectively called the training and test sets used for simulating the data currently available to fit and tune the model and the validation set used for simulating future values. The training of a network is stopped when the mean squared forecasted error is at minimum in the test-sub period. The network parameters are then estimated by fitting the training data using the above mentioned iterative procedure (backpropagation of errors). The iteration length is optimised by maximising the forecasting accuracy for the test dataset. Then the predictive value of the model is evaluated applying it to the validation dataset (out-of-sample dataset).

4.2.1 The Multi-Layer Perceptron Model (MLP)

MLPs are feed-forward layered NN, trained with a back-propagation algorithm. According to Kaastra and Boyd [15], they are the most commonly used types of artificial networks in financial time-series forecasting. The training of the MLP network is processed on a three-layered architecture, as described above.

4.2.2 The Recurrent Neural Network (RNN)

The next NN architecture used in this paper is the RNN. For an exact specification of recurrent networks, see Elman [6]. A simple recurrent network has an activation feedback which embodies short-term memory. The advantages of using recurrent networks over feed-forward networks for modeling non-linear time series have been well documented in the past. However, as mentioned by Tenti [21], "the main disadvantage of RNNs is that they require substantially more connections, and more memory in simulation than standard back-propagation networks" (p. 569), thus resulting in a substantial increase in computational time.

4.2.3 The Psi-Sigma Neural Network (PSN)

The PSNs are a class of Higher Order Neural Networks with a fully connected feed-forward structure. Ghosh and Shin [7] were the first to introduce the PSN, trying to reduce the numbers of weights and connections of a Higher Order Neural Network. Their goal was to combine the fast learning property of single-layer networks with the mapping ability of Higher Order Neural Networks and avoid increasing the required number of weights. The price for the flexibility and speed of Psi Sigma networks is that they are not universal approximators. We need to choose a suitable order of approximation (or else the number of hidden units) by considering the estimated function complexity, amount of data and amount of noise present. To overcome this, our code runs simulations for orders two to six and then it presents the best network. The evaluation of the PSN model selected comes in terms of trading performance.[1]

[1] For a complete description of all the neural network models we used and their complete specifications see Sermpinis et al. [20].

5 Forecasting Combination Techniques

In this section we present the techniques that we used to combine our NNs forecasts. It is important to outline that a forecast combination targets either to follow the trend of the best individual forecast (*'combining for adaptation'*) or to significantly outperform each one of them (*'combining for improvement'*) [23].

5.1 Simple Average

The first forecasting combination technique used in this paper is Simple Average, which can be considered a benchmark forecast combination model. Given the three NNs' forecasts $f^t_{MLP}, f^t_{RNN}, f^t_{PSN}$ at time *t*, the combination forecast at time *t* is calculated as:

$$f^t_{c_{NNs}} = (f^t_{MLP} + f^t_{RNN} + f^t_{PSN}) / 3 \tag{3}$$

5.2 Granger and Ramanathan Regression Approach (GRR)

According to Bates and Granger [1], a combining set of forecasts outperforms the individual forecasts that the set consists of. Based on Granger and Ramanathan [10] we combine our forecasts as follows:

$$f_c = a_0 + \sum_{i=1}^{n} a_i f_i + \varepsilon_1 \tag{GRR}$$

Where:

- f_i, $i=1,\ldots,n$ are the individual one-step-ahead forecasts,
- f_{c1}, f_{c2}, f_{c3} are the combination forecast of each model,
- α_0 is the constant term of the regression
- α_i are the regression coefficients of each model
- $\varepsilon_1, \varepsilon_2, \varepsilon_3$ are the error terms of each regression model

The GRR model at time *t* used in this paper is specified as shown below:

$$f^t_{c_{NNs}} = 0.0422 + 35.023 f^t_{MLP} + 13.461 f^t_{RNN} + 56.132 f^t_{PSN} + \varepsilon_t \tag{4}$$

From (4) it is obvious that GRR favors PSN forecasts.

5.3 Kalman Filter

Kalman Filter is an efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noisy measurements. The time-varying coefficient combination forecast suggested in this paper is shown below:

$$\text{Measurement Equation:} \quad f^t_{c_{NNs}} = \sum_{i=1}^{3} a^t_i f^t_i + \varepsilon_t, \quad \varepsilon_t \sim NID\left(0, \sigma^2_\varepsilon\right) \tag{5}$$

$$\text{State Equation:} \quad a^t_i = a^{t-1}_i + n_t, \quad n_t \sim NID(0, \sigma^2_n) \tag{6}$$

Where:

- $f_{c_{NNs}}^{t}$ is the dependent variable (combination forecast) at time t
- f_i^t $(i = 1, 2, 3)$ are the independent variables (individual forecasts) at time t
- a_i^t $(i = 1, 2, 3)$ are the time-varying coefficients at time t for each NN
- ε_t, n_t are the uncorrelated error terms (noise)

The alphas are calculated by a simple random walk and we initialized $\varepsilon_1 = 0$. Based on the above, our Kalman Filter model has as a final state the following:

$$f_{c_{NNs}}^{t} = 5.80 f_{MLP}^{t} + 1.16 f_{RNN}^{t} + 75.89 f_{PSN}^{t} + \varepsilon_t \tag{7}$$

From the above equation we note that the Kalman filtering process also favors the PSN model. This is what one would expect, since it is the model that performs best individually.

6 Statistical and Trading Performance

As it is standard in literature, in order to evaluate statistically our forecasts, the RMSE, the MAE, the MAPE and the Theil-U statistics are computed (see Dunis and Williams [5]). For all four of the error statistics retained the lower the output, the better the forecasting accuracy of the model concerned. In Table 2 we present the statistical performance of all our models in the out-of-sample period.

Table 2. Summary of the Out-of-sample Statistical Performance

	NAIVE	**MLP**	**RNN**	**PSN**	**Simple Average**	**GRR**	**Kalman Filter**
MAE	0.0084	0.0058	0.0056	0.0048	0.0048	0.0047	0.0044
MAPE	405.62%	112.37%	105.97%	97.88%	94.07%	92.83%	88.37%
RMSE	0.0107	0.0061	0.0060	0.0054	0.0053	0.0049	0.0043
Theil-U	0.7958	0.7301	0.6001	0.4770	0.5672	0.5297	0.5212

We note that from our individual forecasts, the PSN statistically outperformed all other models. Similarly, for our forecast combinations methodologies the Kalman Filter beat its benchmarks for the four statistical criteria retained in the out-of-sample period.

The trading strategy applied in this paper is to go or stay ‘long’ when the forecast return is above zero and go or stay ‘short’ when the forecast return is below zero.[2] In Table 3 below we present the out-of-sample trading performance of our models before and after transaction costs.

[2] The transaction costs for a tradable amount, say USD 5-10 million, are about 1 pip (0.0001 EUR/USD) per trade (one way) between market makers. But since we consider the EUR/USD time series as a series of middle rates, the transaction costs is one spread per round trip. With an average exchange rate of EUR/USD of 1.369 for the out-of-sample period, a cost of 1 pip is equivalent to an average cost of 0.007% per position.

Table 3. Summary of Out-of-Sample Trading Performance

	NAIVE	MLP	RNN	PSN	Simple Average	GRR	Kalman Filter
Annualised Return (excluding costs)	-4.80%	14.80%	16.07%	18.37%	16.37%	16.99%	28.79%
Annualised Volatility	12.03%	11.83%	11.02%	10.89%	10.85%	11.02%	10.92%
Information Ratio (excluding costs)	-0.4	1.25	1.46	1.69	1.51	1.54	2.64
Maximum Drawdown	-6.41%	-6.23%	-6.23%	-6.31%	-6.31%	-6.31%	-6.31%
Annualized Transactions	77	71	71	76	70	63	73
Transaction Costs	0.54%	0.50%	0.50%	0.53%	0.49%	0.44%	0.51%
Annualised Return (including costs)	**-5.34%**	**14.30%**	**15.57%**	**17.84%**	**15.88%**	**16.55%**	**28.28%**
Information Ratio (including costs)	**-0.44**	**1.21**	**1.41**	**1.64**	**1.46**	**1.50**	**2.59**

From the last two rows of Table 3, we note that the PSN continues to outperform all other single forecasts in terms of trading performance, coinciding with its statistical superiority. From our forecast combinations, only the Kalman Filter beats our best single forecast. The Simple Average and GRR methods seem unable to outperform PSN in the out-of-sample period. On the other hand, the GRR strategy still outperforms the MLP and the RNN models in terms of annualised return and information ratio. That could be thought as a trend to adapt to the best individual performance (*'combining for adaptation'* [23]). It seems that the ability of Kalman Filter to provide efficient computational recursive means to estimate the state of our process, gives it a considerable advantage compared to our fixed parameters combination models.[3]

7 Concluding Remarks

The motivation of this paper is to investigate the use of a Neural Network (NN) architecture, the Psi Sigma Neural Network (PSN), when applied to the task of forecasting and trading the Euro/Dollar (EUR/USD) exchange rate and to explore the utility of Kalman Filters in combining NN forecasts. This is done by benchmarking the statistical and trading performance of PSN with a Naive Strategy and two different NN architectures, a Multi-Layer Perceptron (MLP) and a Recurrent Network (RNN). We combine our NN forecasts with Kalman Filter, a traditional Simple Average and the Granger- Ramanathan's Regression Approach (GRR). The statistical and trading performance of our models is estimated throughout the period of 2002-2010, using the last two years for out-of-sample testing.

As it turns out, the PSN outperforms its benchmarks models in terms of statistical accuracy and trading performance. It is also shown that all the forecast combinations outperform in the out-of-sample period all our single models, except the PSN, for the

[3] The in-sample statistical and trading performances of our models are not presented for the sake of space. Nonetheless, the ranking of the models does not change considerably compared with the out-of-sample period.

statistical and trading terms retained. Simple Average and GRR do not beat PSNs' best individual performance, but are better than MLP and RNN, while Kalman Filter presents the best results. It seems that the ability of Kalman Filter to provide efficient computational recursive means to estimate the state of our process gives it a considerable advantage compared to our fixed parameters combination models. The remarkable trading performance of Kalman Filter allows us to conclude that it can be considered as an optimal forecast combination for the models and time-series under study. Our results should also go some way towards convincing a growing number of quantitative fund managers to experiment beyond the bounds of traditional models and trading strategies.

References

[1] Bates, J.M., Granger, C.W.J.: The Combination of Forecasts. Operational Research Society 20(4), 451–468 (1969)

[2] Deutsch, M., Granger, C.W.J., Teräsvirta, T.: The combination of forecasts using changing weights. International Journal of Forecasting 10(1), 47–57 (1994)

[3] Dunis, C.L., Huang, X.: Forecasting and trading currency volatility: an application of recurrent neural regression and model combination. Journal of Forecasting 21(5), 317–354 (2002)

[4] Dunis, C.L., Shannon, G.: Emerging Markets of South-East and Central Asia: Do they Still Offer a Diversification Benefit? Journal of Asset Management 6(3), 168–190 (2005)

[5] Dunis, C.L., Williams, M.: Modelling and Trading the EUR/USD Exchange Rate: Do Neural Network Models Perform Better? Derivatives Use, Trading and Regulation 8, 211–239 (2002)

[6] Elman, J.L.: Finding Structure in Time. Cognitive Science 14(2), 179–211 (1990)

[7] Ghosh, J., Shin, Y.: The Pi-Sigma Network: An efficient Higher-order Neural Networks for Pattern Classification and Function Approximation. In: Proceedings of International Joint Conference of Neural Networks, vol. 1, pp. 13–18 (1991)

[8] Ghosh, J., Shin, Y.: Efficient Higher-Order Neural Networks for Classification and Function Approximation. International Journal of Neural Systems 3(4), 323–350 (1992)

[9] Goh, S.L., Mandic, D.P.: An Augmented Extended Kalman Filter Algorithm for Complex-Valued Recurrent Neural Networks. Neural Computation 19(4), 1039–1055 (2007)

[10] Granger, C.W.J., Ramanathan, R.: Improved methods of combining forecasts. Journal of Forecasting 3(2), 197–204 (1984)

[11] Hamilton, J.D.: Time series analysis. Princeton University Press, Princeton (1994)

[12] Harvey, A.C.: Forecasting, structural time series models and the Kalman filter. Cambridge University Press, Cambridge (1990)

[13] Hussain, A.J., Ghazali, R., Al-Jumeily, D., Merabti, M.: Dynamic Ridge Polynomial Neural Network for Financial Time Series Prediction. In: IEEE International Conference on Innovation in Information Technology, Dubai, pp. 1–5 (2006)

[14] Ince, H., Trafalis, T.B.: A hybrid model for exchange rate prediction. Decision Support Systems 42(2), 1054–1062 (2006)

[15] Kaastra, I., Boyd, M.: Designing a Neural Network for Forecasting Financial and Economic Time Series. Neurocomputing 10(3), 215–236 (1996)

[16] Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., Newton, J., Parzen, E., Winkler, R.: The accuracy of extrapolation (time series) methods: Results of a forecasting competition. Journal of Forecasting 1(2), 111–153 (1982)
[17] Newbold, P., Granger, C.W.J.: Experience with Forecasting Univariate Time Series and the Combination of Forecasts. Journal of the Royal Statistical Society 137(2), 131–165 (1974)
[18] Palm, F.C., Zellner, A.: To combine or not to combine? issues of combining forecasts. Journal of Forecasting 11(8), 687–701 (1992)
[19] Shapiro, A.F.: A Hitchhiker's guide to the techniques of adaptive nonlinear models. Insurance: Mathematics and Economics 26(2-3), 119–132 (2000)
[20] Sermpinis, G., Laws, J., Dunis, C.L.: Modelling and trading the realised volatility of the FTSE100 futures with higher order neural networks. European Journal of Finance, 1–15 (2012)
[21] Tenti, P.: Forecasting foreign exchange rates using recurrent neural networks. Applied Artificial Intelligence 10(6), 567–581 (1996)
[22] Terui, N., Van Dijk, H.K.: Combined forecasts from linear and nonlinear time series models. International Journal of Forecasting 18(3), 421–438 (2002)
[23] Yang, Y.: Combining Forecasting Procedures: Some theoretical results. Econometric Theory 20(1), 176–222 (2004)