Pattern Matching and Neural Networks based Hybrid Forecasting System

Sameer Singh and Jonathan Fieldsend
PANN Research, Department of Computer Science, University of Exeter,
Exeter, UK

Abstract

In this paper we propose a Neural Net-PMRS hybrid for forecasting time-series data. The neural network model uses the traditional MLP architecture and backpropagation method of training. Rather than using the last *n* lags for prediction, the input to the network is determined by the output of the PMRS (Pattern Modelling and Recognition System). PMRS matches current patterns in the time-series with historic data and generates input for the neural network that consists of both current and historic information. The results of the hybrid model are compared with those of neural networks and PMRS on their own. In general, there is no outright winner on all performance measures, however, the hybrid model is a better choice for certain types of data, or on certain error measures.

1. Hybrid Forecasting System

The science of forecasting relies heavily on the models used for forecasting, quantity and quality of data, and the ability to pick the right models for a given data. A number of past publications on pattern matching have argued the need for using historic information through pattern matching for forecasting. A number of these pattern matching techniques, such as our previous PMRS model, have been very successful on a range of economic and financial data. The main philosophy behind these pattern matching techniques is to identify the best matching past trends to current ones and use the knowledge of how the time series behaved in the past in those situations to make predictions for the future. A range of nearest neighbour strategies can be adopted for this matching such as the fuzzy Single Nearest Neighbour (SNN) approach.

Neural networks do not normally use historic matches for forecasting. Instead, their inputs are taken as recent lags [2]. However, their ability to formulate a non-linear relationship between inputs and output has a considerable advantage for producing accurate forecasts [3].

In this paper we combine the pattern matching ability with the determination of non-linear relationship between inputs and output by generating a hybrid model for forecasting. The hybrid model uses the PMRS model [6,7] for determining historic matches used as inputs to a neural network model. The results are shown on a range of scientific time series data and financial market data. We first describe our PMRS and neural network models on a stand-alone basis, and then their combination.

1.1 PMRS component

If we choose to represent a time-series as $y = \{y_1, y_2, ... y_n\}$, then the current state of size one of the time-series is represented by its current value y_n. One simple method of prediction can be based on identifying the closest neighbour of y_n in the past data, say y_j, and predicting y_{n+1} on the basis of y_{j+1}. Calculating an average prediction based on more than one nearest neighbour can modify this approach. The definition of the current state of a time-series can be extended to include more than one value, e.g. the current state s_c of size two may be defined as $\{y_{n-1}, y_n\}$. For such a current state, the prediction will depend on the past state s_p $\{y_{j-1}, y_j\}$ and next series value $y^+{}_p$ given by y_{j+1}, provided that we establish that the state $\{y_{j-1}, y_j\}$ is the nearest neighbour of the state $\{y_{n-1}, y_n\}$ using some similarity measurement. In this paper, we also refer to *states* as *patterns*. In theory, we can have a current state of any size but in practice only matching current states of optimal size to past states of the same size yields accurate forecasts since too small or too large neighbourhoods do not generalise well. The optimal state size must be determined experimentally on the basis of achieving minimal errors on standard measures through an iterative procedure.

We can formalise the prediction procedure as follows:
$\ddot{y} = \phi(s_c, s_p, y^+{}_p, k, c)$

where $\ddot{y}$ is the prediction for the next time step, s_c is the current state, s_p is the nearest past state, $y^+{}_p$ is the series value following past state s_p, k is the state size and *c* is the matching constraint. Here $\ddot{y}$ is a real value, s_c or s_p can be represented as a set of real values, k is a constant representing the number of values in each state, i.e. size of the set, and *c* is a constraint which is user defined for the matching process. We define *c* as the condition of matching operation that series direction change for each member in s_c and s_p is the same.

In order to illustrate the matching process for series prediction further, consider the time series as a vector $y = \{y_1, y_2, ... y_n\}$ where n is the total number of points in the series. Often, we also represent such a series as a function of time, e.g. $y_n = y_t$, $y_{n-1} = y_{t-1}$, and so on. A segment in the series is defined as a difference vector $\delta = (\delta_1, \delta_2, ... \delta_{n-1})$ where $\delta_i = y_{i+1} - y_i$, $\forall i$, $1 \le i \le n-1$. A pattern contains one or more segments and it can be visualised as a string of segments $\rho = (\delta_i, \delta_{i+1}, ... \delta_h)$ for given values of i and h, $1 \le i,h \le n-1$, provided that h>i. In order to define any pattern mathematically, we choose to tag the time series y with a vector of change in direction. For this purpose, a value y_i is tagged with a 0 if $y_{i+1} < y_i$, and as a 1 if $y_{i+1} \geq y_i$. Formally, a pattern in the time-series is represented as $\rho = (b_i, b_{i+1}, \dots b_h)$ where b is a binary value.

The complete time-series is tagged as $(b_1, \dots b_{n-1})$. For a total of k segments in a pattern, it is tagged with a string of k *b* values. For a pattern of size k, the total number of binary patterns (shapes) possible is 2^k. The technique of matching structural primitives is based on the premise that the past repeats itself. It has been noted in previous studies that the dynamic behaviour of time-series can be efficiently predicted by using local approximation. For this purpose, a map between current states and the nearest neighbour past states can be generated for forecasting.

Pattern matching in the context of time-series forecasting refers to the process of matching current state of the time series with its past states. Consider the tagged time series $(b_1, b_i, \dots b_{n-1})$. Suppose that we are at time *n* (y_n) trying to predict y_{n+1}. A pattern of size k is first formulated from the last k tag values in the series, $\rho' = (b_{n-k}, \dots b_{n-1})$. The size k of the structural primitive (pattern) used for matching has a direct effect on the prediction accuracy. Thus the pattern size k must be optimised for obtaining the best results. For this k is increased in every trial by one unit till it reaches a predefined maximum allowed for the experiment and the error measures are noted; the value of k that gives the least error is finally selected. The aim of a pattern matching algorithm is to find the closest match of ρ' in the historical data (estimation period) and use this for predicting y_{n+1}. The magnitude and direction of prediction depend on the match found. The success in correctly predicting series depends directly on the pattern matching algorithm.

The first step is to select a state/pattern of minimal size (k=2). A nearest neighbour of this pattern is determined from historical data on the basis of smallest offset ∇. There are two cases for prediction: either we predict high or we predict low. The prediction $\ddot{y}_{n+1}$ is scaled on the basis of the similarity of the match found. We use a number of widely applied error measures for estimating the accuracy of the forecast and selecting optimal k size for minimal error. The forecasting process is repeated with a given test data for states/patterns of size greater than two and a model with smallest k giving minimal error is selected. In our experiments k is iterated between $2 \leq k \leq 5$.

1.2 Neural Network component

In this paper we use the standard MLP architecture with backpropagation mode of learning. In order to enable consistency between the Neural Network model and the PMRS model, the Neural Network inputs are the 6 most recent lags of the time series (i.e. Y_{t-1}, Y_{t-2}, Y_{t-3}, Y_{t-4}, Y_{t-5} & Y_{t-6}, when the value to be predicted is the actual Y_t). This mimics the PMRS approach of pattern matching of up to 5 historic differences (δ values) which therefore uses the information contained in the most recent 6 lags. (In other circumstances the 6 lags chosen would be those with the highest partial autocorrelation function value when correlated with the actual [5]).

In our study, neural networks have two hidden layers with 5 sigmoidal nodes in each and the networks are fully connected.

The learning rate was set at 0.05 with a momentum of 0.5. The Neural Networks training was stopped when the combined RMSE on the test and training set had fallen by less than 0.025% of their value 5 epochs ago. The inclusion of the validation error prevents over- fitting, however by summing the two errors (as opposed to strictly using the test error on its own), a slight trade-off is permitted between the errors while pushing through local minima. In addition this stopping operator was only used after at least 1000 epochs had passed during training.

1.3 Neural Net-PMRS

Neural Networks with PMRS generated inputs have the same stopping regime as the standard neural network described above with only topological difference in the form of number of inputs. Two hybrid models were trained. The first was training on the matched deltas δ found by the PMRS algorithm, the predicted delta and the lags used to find them, i.e. a PMRS model fitting two historic deltas would use 3 lags. A single asterisk in Table 2 denotes this model (NNPMRS*). The second model used the matched deltas δ found by the PMRS algorithm, the predicted delta and the most recent lag (this model is denoted by a double asterisk in Table 2 - NNPMRS**).

2. Experimental Details

Each data set was partitioned into consecutive segments of 75% (training data) 15% (test data) and 10% (validation data). The Neural Network model weights are adjusted using the Backpropagation algorithm on the training set and stopped (using the stopping method described later) using the root mean square errors on the training set and test set. In the case of PMRS, four different PMRS models with pattern sizes of 2,3,4,and 5 were fitted to the training set and compared on the test set data. The best performing PMRS model on the test set was then chosen for use on the validation set and as the input to the NN-PMRS model. The 'best performing' model was judged as the PMRS lagged model which had the highest number of 'best' error statistics out of the six statistics used: R^2, Percentage direction success, Root Mean Square Error (RMSE), Geometric Mean Relative Absolute Error (GMRAE), Mean Average Percentage Error (MAPE) and Percentage better than random walk (BRW). When two PMRS models performed equally well using this criteria, two NN-PMRS models were fitted.

As the PMRS model needs a store of historical data before it can make predictions (and therefore be of use as an input to a non-linear system such as a Neural Network), the data split is slightly different for the NN-PMRS models. The end 10% of the data is again set aside as an 'unseen' validation set, and of the remaining 90% data, the first 40% is used as the estimation data for the PMRS model to be fitted to. The remaining 50% is split such that the neural network is fitted to the next 40% and tested on the following 10%. This results in a 40/40/10/10 split as opposed to the 75/15/10 split used in the other models.

3. Results

The results for the performance of the two hybrid models, PMRS and Neural networks is shown in Table 2 on a range of error measures recommended by Armstrong and Collopy [1]. It is evident that no single model performs the best on all error measures. In this table, we have picked the best performing PMRS models on the basis of how it performs on test data. For PMRS, we vary the parameter k (pattern size for matching) between 2 and 5 and select those models that perform the best on most number of error measures. In some cases, more than one model is chosen as the best performer as two models can perform equally well on different error measures. Out of the two hybrid models, we find that NNPMRS* model consistently outperforms the second model and therefore we use this as our base hybrid model for comparison with other methods. It is difficult to visualise an overall winner on each measure. In order to interpret results we simplify this process in Figure 1. We find that on most time-series, the hybrid model is the best for generating the lowest GMRAE error and is significantly good on generating high direction rate success. It outperforms the two other models on all error measures in a small proportion of cases, and there is no generic trend to comment on. From these experiments it is abundantly clear that the hybrid model has a considerable future in the forecasting domain. Some of the improvement in results is small, however any improvement is of considerable importance in financial domains, where the capital involved can be extremely large.

4. Conclusions

In this paper we have proposed a novel method of combing pattern matching techniques with neural networks. This hybrid strategy has the advantage that it becomes possible to use historic information efficiently, which is not possible in traditional neural network models. In general, the hybrid model is a better choice in situations where the model selection process relies heavily on low error on GMRAE and high direction success. On other error measures, the hybrid model does not, in the majority of time series, outperform the PMRS and neural net stand-alone models. Our understanding is that the hybrid strategy is of considerable use in a range of prediction domains and it is only through empirical analysis, we can interpret its advantages.

References

1. Armstrong S. and Collopy F. Error measures for generalizing about forecasting methods: Empirical comparisons. International Journal of Forecasting 1992; 8:69-80.
2. Azoff M.E. Neural Network Time Series Forecasting of Financial Markets. John Wiley, 1994.
3. Gately E. Neural Networks for Financial Forecasting. John Wiley, 1996.
4. Peters E. Fractal Market Hypothesis: Applying Chaos Theory to Investment and Economics. John Wiley, 1994.

5. Refenes A.N., Burgess N. and Bentz Y. Neural networks in financial engineering: A study in methodology. IEEE Transactions on Neural Networks 1997; 8:6:1222-1267.
6. Singh S. A long memory pattern modelling and recognition system for financial time-series forecasting. Pattern Analysis and Applications 1999; 2:3:264-273.
7. Singh S. and Fieldsend J.E. Financial Time Series Forecasts using Fuzzy and Long Memory Pattern Recognition Systems. IEEE International Conference on Computational Intelligence for Financial Engineering, New York, (26-28 March, 2000). IEEE Press, 2000.

Data set	Description	Observations
A	Laser generated data	1000
B1, B2, B3	Physiological data, spaced by 0.5 second intervals. The first set is the heart rate, the second is the chest volume (respiration force), and the third is the blood oxygen concentration (measured by ear oximetry).	34000 each
C	tickwise bids for the exchange rate from Swiss francs to US dollars; they were recorded by a currency trading group from August 7, 1990 to April 18, 1991	30000
D1, D2	Computer generated series	50000
E	Astrophysical data. This is a set of measurements of the light curve (time variation of the intensity) of the variable white dwarf star PG1159-035 during March 1989, 10 second intervals.	27204
USDBRL	Daily Exchange Rate of Brazilian Reals to the US Dollar (4 & ¾ years) 22nd Oct 95 to 2nd August 2000	1747
USDCAN	Daily Exchange Rate of Canadian Dollars to the US Dollar (8 years) 3rd August 1992 to 2nd August 2000	2722
USDDEM	Daily Exchange Rate of German Marks to the US Dollar (8 years) 3rd August 1992 to 2nd August 2000	2722
USDJPY	Daily Exchange Rate of Japanese Yen to the US Dollar (8 years) 3rd August 1992 to 2nd August 2000	2722
USDCHF	Daily Exchange Rate of Swiss Francs to the US Dollar (8 years) 3rd August 1992 to 2nd August 2000	2722
USDGBP	Daily Exchange Rate of Pounds Sterling to the US Dollar (8 years) 3rd August 1992 to 2nd August 2000	2722

Table 1. 1991 Santa Fe Competition data & exchange rate data

Exchange Rates are daily average Interbank rates (where the average is calculated as the mid point of the low and the high of that day), 14 Data sets in all, longest being 50,000 points, shortest 1,000 points.

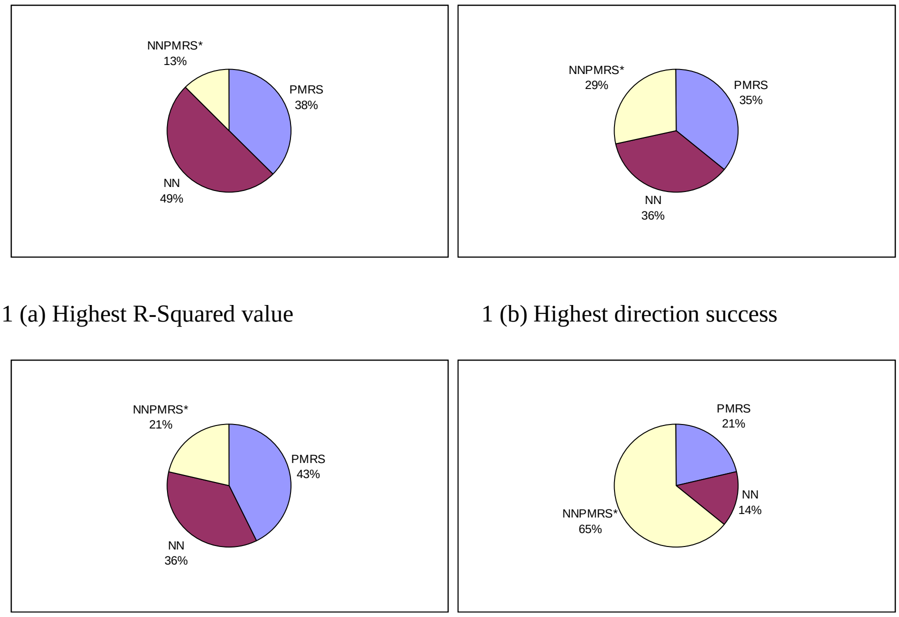
Dataset	Model	k	R^2	% Dir. Suc	RMSE	GMRAE ($\times10^{-3}$)	MAPE	BRW
A	PMRS	4	0.98326	**97.000**	0.98933	1.635	10.28	**95.000**
	NN (bp)	-	**0.99707**	94.949	**0.39313**	**1.2764**	**7.0041**	92.929
	nn-pmrs*	4	0.9845	93.137	0.86214	1.6978	11.287	93.137
	nn-pmrs**	4	0.9826	95.098	0.90955	1.7974	12.146	88.235
B1	PMRS	2	**0.99366**	70.647	**0.077997**	0.29556	**11.024**	35.794
	PMRS	5	0.98997	63.471	0.098606	0.26968	14.356	44.735
	NN (bp)	-	0.98204	**78.464**	0.131870	0.25546	847.26	38.364
	nn-pmrs*	2	0.97110	78.182	0.15517	**0.25512**	10011	47.636
	nn-pmrs*	5	0.94505	76.125	0.21848	0.2678	1446.6	**50.346**
	nn-pmrs**	2	0.98657	75.577	0.10449	0.26309	646.56	22.405
	nn-pmrs**	5	0.97112	76.009	0.15474	0.26463	1007.6	38.322
B2	PMRS	2	0.83335	60.147	72.295	0.50022	67.339	41.647
	PMRS	5	0.77703	67.412	90.893	**0.44723**	81.976	49.324
	NN (bp)	-	**0.91510**	**70.315**	44.175	0.84700	**58.310**	**56.899**
	nn-pmrs*	2	0.90470	65.542	44.439	1.09990	59.607	52.422
	nn-pmrs*	5	0.91469	67.734	**42.434**	0.88111	59.270	53.749
	nn-pmrs**	2	0.89625	60.438	46.421	0.94134	64.565	44.867
	nn-pmrs**	5	0.9028	64.475	44.652	0.95816	61.165	50.865
B3	PMRS	3	0.98993	51.029	21.219	0.43080	1.3969	30.882
	PMRS	4	0.99009	54.529	21.052	0.40880	**1.3522**	33.735
	PMRS	5	0.98993	54.647	21.229	0.39971	1.4022	35.588
	NN (bp)	-	0.98660	55.075	23.318	4.16300	4.6472	51.368
	nn-pmrs*	3	0.82640	49.683	69.114	1.9272	7.1579	41.522
	nn-pmrs*	4	0.989100	65.484	22.449	1.4900	2.5309	**60.467**
	nn-pmrs*	5	0.989100	65.484	22.449	1.4900	2.5309	**60.467**
	nn-pmrs**	3	0.021970	**69.550**	**5.9163**	**0.2390**	4.51×10^{13}	49.510
	nn-pmrs**	4	**0.99040**	48.875	20.420	1.2641	2.1728	43.080
	nn-pmrs**	5	0.99033	48.320	20.25	1.4067	2.2562	42.762
C	PMRS	2	**1.00000**	**47.800**	**2.15×10^{-5}**	0.19019	**0.0485**	36.333
	NN (bp)	-	0.99950	43.481	0.000524	0.19000	1.9894	**43.481**
	nn-pmrs*	2	0.99820	43.399	0.000873	**0.18932**	3.3581	43.431
	nn-pmrs**	2	0.99829	43.399	0.000854	**0.18932**	3.2846	43.431

Dataset	Model	k	R^2	% Dir. Suc	RMSE	GMRAE (×10^{-3})	MAPE	BRW
D1	PMRS	4	0.98671	77.660	0.000931	0.19888	10.457	63.400
	PMRS	5	0.98787	79.540	0.000894	0.19884	10.069	66.120
	NN (bp)	-	**0.99640**	**86.737**	**0.000489**	0.19900	**6.3906**	**70.454**
	nn-pmrs*	4	0.98973	80.804	0.000709	0.19485	8.3753	64.804
	nn-pmrs*	5	0.99416	85.275	0.000549	**0.19477**	6.7570	69.647
	nn-pmrs**	4	0.99204	81.961	0.000647	0.19486	7.7160	61.941
	nn-pmrs**	5	0.99287	82.627	0.000617	0.19483	7.5616	63.882
D2	PMRS	4	0.98535	79.680	0.000929	0.19885	11.869	64.500
	NN (bp)	-	0.99230	83.957	0.000644	0.19900	8.8072	65.913
	nn-pmrs*	4	**0.99440**	**86.863**	**0.000542**	**0.19469**	**8.2082**	**72.314**
	nn-pmrs**	4	0.99173	83.980	0.000654	0.19481	9.2745	63.49
E	PMRS	4	0.57996	56.487	0.003196	0.36995	7.929 $\times10^{17}$	**41.455**
	NN (bp)	-	**0.59060**	**67.929**	0.001641	0.36600	**2.076 $\times10^{13}$**	39.757
	nn-pmrs*	4	0.58428	67.916	**0.001628**	**0.35847**	2.343 $\times10^{13}$	40.159
	nn-pmrs**	4	0.51360	65.321	0.001759	0.35890	2.283 $\times10^{13}$	35.400
USD BRL	PMRS	2	0.99988	56.000	0.001336	**2.8258**	**0.5106**	17.143
	NN (bp)	-	**0.99994**	46.552	**0.000943**	2.8445	0.5700	33.333
	nn-pmrs*	2	0.99989	**64.423**	0.001690	4.7359	0.8820	**39.423**
	nn-pmrs**	2	0.99989	**64.423**	0.001656	4.7359	0.8581	**39.423**
USD CAD	PMRS	2	**0.99998**	47.44	**0.000324**	1.9101	**0.2729**	27.645
	NN (bp)	-	**0.99998**	**54.983**	0.000328	1.9211	0.3115	40.893
	nn-pmrs*	2	**0.99998**	54.027	0.000352	**1.8900**	0.3411	**42.953**
	nn-pmrs**	2	**0.99998**	54.362	0.000354	**1.8900**	0.3425	42.617
USD CHF	PMRS	2	**0.99991**	**48.805**	**0.000817**	1.9384	**0.6265**	27.645
	NN (bp)	-	0.99856	44.674	0.003165	1.9493	2.8534	**43.299**
	nn-pmrs*	2	0.9979	41.275	0.003481	1.8639	3.2294	40.604
	nn-pmrs**	2	0.9976	42.282	0.003727	**1.8638**	3.4605	40.940
USD DEM	PMRS	2	**0.99993**	**50.853**	**0.000898**	1.6618	**0.5617**	26.621
	NN (bp)	-	0.99730	40.550	0.005284	1.6771	3.7971	40.550
	nn-pmrs*	2	0.99622	40.604	0.005600	1.6064	4.1274	40.604
	nn-pmrs**	2	0.99641	41.611	0.005539	**1.6062**	3.9973	**41.611**

Dataset	Model	k	R^2	% Dir. Suc	RMSE	GMRAE ($\times 10^{-3}$)	MAPE	BRW
USD GBP	PMRS	2	0.99995	**52.218**	0.000229	2.9284	0.4277	29.693
	NN (bp)	-	**0.99997**	51.890	**0.000189**	2.9496	**0.3805**	**30.584**
	nn-pmrs*	2	0.99996	45.973	0.000199	**2.8721**	0.3839	29.866
	nn-pmrs**	2	0.99995	46.309	0.000206	**2.8721**	0.3934	29.866
USD JPY	PMRS	2	0.99988	47.440	0.060916	3.7303	0.6552	24.915
	NN (bp)	-	**0.99993**	51.890	**0.047101**	**3.6057**	**0.5727**	31.271
	nn-pmrs*	2	0.99987	55.034	0.058962	3.7064	0.7538	37.919
	nn-pmrs**	2	0.99985	**56.376**	0.063695	3.7870	0.8195	**38.255**

Table 2. Table of results for each time series.

The results are on the performance of the chosen PMRS model, NN model and the two fitted NN models using PMRS inputs on the test set. Results in bold indicate the best result for that error term.

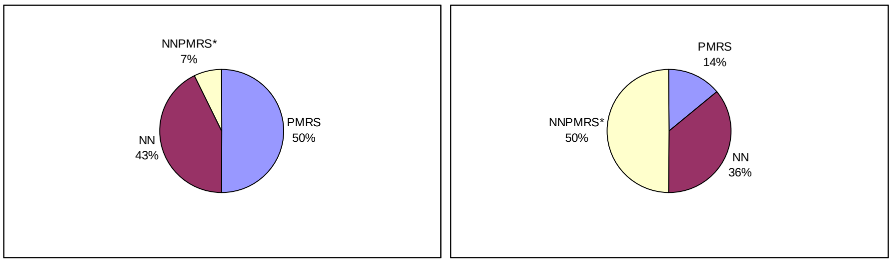


1 (a) Highest R-Squared value

1 (b) Highest direction success

1 (c) Lowest RMSE

1 (d) Lowest GMRAE

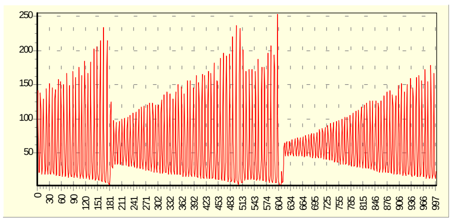


1 (e) Lowest MAPE

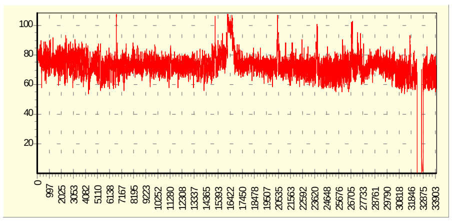
1 (f) Highest BRW

Figure 1 (a)-(f) Proportion of models with the best performance on the error measures averaged across all data tested.

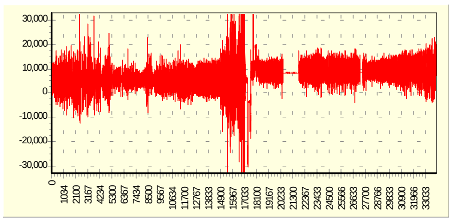
APPENDIX: DATA SETS



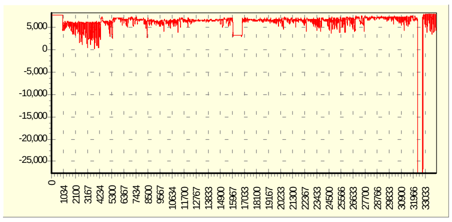
Data Set A



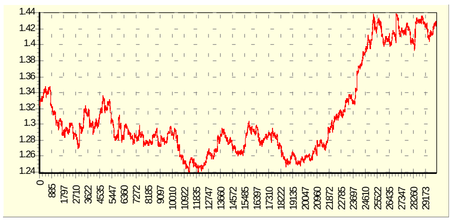
Data Set B1



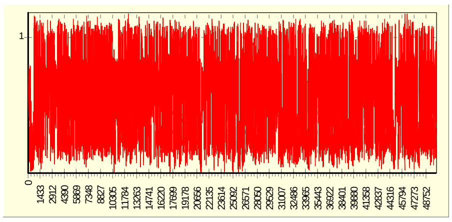
Data Set B2



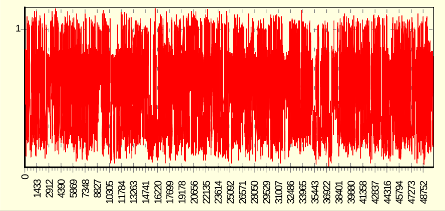
Data Set B3



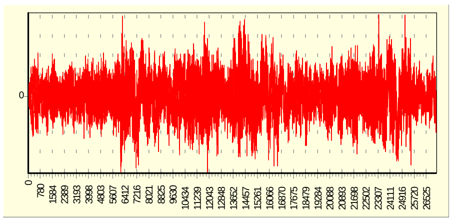
Data Set C



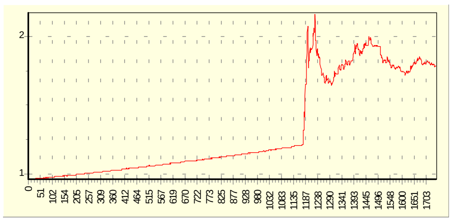
Data Set D1



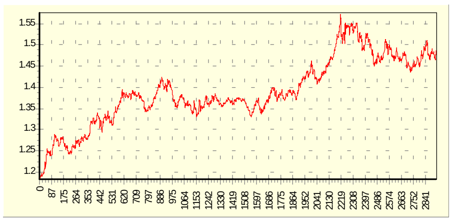
Data Set D2



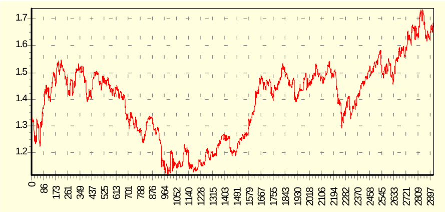
Data Set E



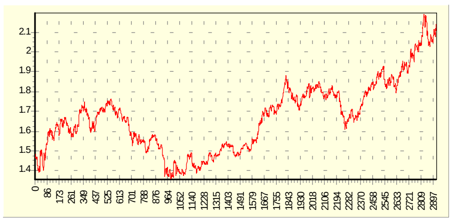
Data Set USDBRL



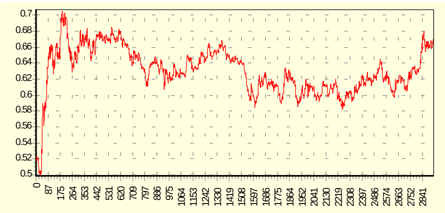
Data Set USDCAD



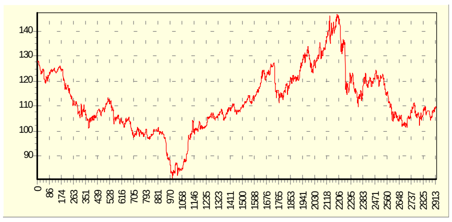
Data Set USDCHF



Data Set USDDEM



Data Set USDGBP



Data Set USDJPY