Stepping forward through echoes of the past: forecasting with Echo State Networks

Iulian Ilies, Herbert Jaeger, Olegas Kosuchinas, Monserrat Rincon, Vytenis Šakėnas, Narunas Vaškevičius

***Abstract*— During the Spring term of 2007, the Machine Learning seminar at Jacobs University Bremen tackled the *NN3 Artificial Neural Network and Computational Intelligence Forecasting Competition*. The objective was to forecast 111 monthly, financial timeseries (of unknown origin) by 18 months. We implemented a number of standard textbook prediction methods (exponential smoothing, dampened exponential smoothing) as a baseline; compared them with likewise standard methods from computational intelligence (feedforward NNs, support vector regression (SVR), local methods, wavelet-decomposition based predictors) and found no convincing advantage; and finally opted for recurrent neural networks of the Echo State Network type, which we bundled in large voting collectives which were trained on blocks of time series.**

I. Introduction

Since people become aware of the movement of time, they dreamed on knowing what will happen next. "It is far better to foresee even without certainty than not to foresee at all" were the words of the French mathematician and physicist Henri Poincaré. To us humans, it seems that knowing the future can ameliorate it, or at least can prepare us to receive it.

Across history, there have been diverse philosophical dissertations on the idea that the future could be predicted by the knowledge of the past. Nowadays, that thought is reflected, among others, in the utilization of machine learning methods for forecasting. In a similar way to the human mind, mathematical models can be trained to detect rules in the evolution in time of different variables, and then use such rules to predict future events.

The *NN3 Artificial Neural Network and Computational Intelligence Forecasting Competition* (`www.neural-forecasting-competition.com`), sponsored by the International Institute of Forecasters (`www.forecasters.org`) and the statistical software company SAS (`www.sas.com`), is a stage where this ancient dream is put to a dire and very mundane test. One hundred and eleven time series, of which not more is known than that they come from the world of finance and that they are monthly series, have to be predicted by 18 months.

Such a task seemed a perfect project for our Machine Learning seminar at Jacobs University Bremen. On one hand, it required an substantial amount of study, investigation, and hands-on work, and on the other hand the challenge of producing competitive results was the perfect motivational "kick". An additional spur was to take up the gauntled thrown at our machine learner's feed in [8], who essentially state that current methods of computational intelligence do not consistently outperform the standard, much simpler methods in the tradition of statistical forecasting, and who wonder why this simple lesson seems so hard to accept...

The authors are students resp. faculty at Jacobs University, 28759 Bremen, Germany. Corresponding author: H. Jaeger, h.jaeger@jacobs-university.de

We decided to work on the complete group of 111 time series. We first implemented some of the standard textbook [9] predictors which in [8] were presented as robust and often competitive; here we will report only on (i) dampened exponential smoothing and (ii) Theta prediction [1], based on likewise standard decompositions of the series into a trend, cycle, and residual components. This provided us with a baseline. We then proceeded to try out methods of computational intelligence, where each of the five seminar participants was in charge of one of the following kinds of methods: (iii) local methods (following [10]), (iv) multilayer perceptrons, (v) support vector machines, (vi) wavelet decomposition based predictions (following [12]), and (vii) Fourier decomposition based predictions. It turned out that these methods indeed did not fare better than (dampened or pure) exponential smoothing.

Finally, we adopted Echo State Networks (ESNs), a recurrent neural network (RNN) architecture which in the past has been applied to other time series modelling tasks ([4] [5]). We devised a scheme where a collective of many such networks is trained on an entire subset of the 111 competition series, and the predictions of the individual ESNs are then combined to produce the ultimate predictions for the competition. In this paper we focus on the ESN-based prediction and mention the results which we obtained with the other methods (i) – (vii) only for purposes of comparison.

II. Echo State Networks

Echo State Networks present an RNN architecture which in its basic version is made of two main components:

- a large, randomly created, non-adaptive "reservoir" RNN, and
- a set of readout neurons (one per output signal dimension) connected to the reservoir.

Each readout neuron is connected to all (or a subset) of the reservoir units; the reservoir-to-output connections are the only trainable connections in an ESN. An ESN operates, and is trained, as follows:

- The reservoir functions as a nonlinear excitable medium. It is excited by input signals fed into it through external input neurons and/or feedback connections from the output neurons.
- When the reservoir is fed by input signals, each of the reservoir units generates a nonlinear transform signal of

the driving input. Due to the recurrency of the reservoir, information is integrated over time.
- The output neurons are, typically, simple linear readout devices. Each output neuron computes its output signal by linearly combining the signals obtained at the reservoir units; the linear combination weights are the synaptic connection weights.
- An ESN is trained, in a supervised schema, by first driving the reservoir with the teacher input (and/or the fed-back teacher output); and then secondly by computing the linear regression weights of the desired output signals from the reservoir-internal signals.

A theoretical introduction to ESNs can be found in [2], a practical guide and tutorial in [3]; an overview on current ESN research is provided by a special issue of Neural Networks [6]. The basic working principle of ESNs was simultaneously discovered in computational neuroscience as a biological information processing mechanism. In this domain, the principle is known under the name of *Liquid State Machines* [7].

The neurons used within the reservoir can be of any type (sigmoid additive, leaky integrator, or spiking models of various degrees of biological accuracy). We used leaky integrator neurons for the NN3 competition. We defer a complete formal specification to an eventual long version of this paper.

III. Approach

In other work, H. Jaeger has found that ESNs can classify stochastic (speech) time series very well when a large number of very small ESNs are combined in a voting collective [5]. Thus, one initial design decision was to employ such collectives.

Preliminary investigations showed however no advantage of such an ESN-based voting collective approach over the other methods which we implemented. This motivated a second basic design decision, namely, to combine the time series into "blocks", and train ESN predictors block-wise. This approach was based on the observation that the 111 competition series come in six clearly discernible groups, where each group contains series which are approximately or perfectly co-temporal. Figure 1 illustrates this observation.

At this point we based our design on a bet: namely, that the series within a block had been obtained from somehow causally correlated sources. If this was indeed the case, then in principle it should be possible to improve the prediction of a given series from a block, by utilizing information from the other series in the block. We carried out preparatory studies where we used linear correlation measures to check whether series within a block were systematically related; the findings were mixed (some blocks had highly mutually correlated subsets of series; some blocks hadn't; finally, there were correlations across blocks). However, our eventual results supported the assumption of exploitable information transfer within blocks.

We defined the blocks by visual inspection of figure 1. The memberships of the blocks thus obtained are listed in table I.

Three further series (nrs. 76 88 109) were not sufficiently aligned in time with any of the blocks; these three series were predicted individually using the SVR predictor.

A. Cross-validation scheme

In order to assess the performance of the comparison methods and our ultimately used ESN method, we used a simple cross-validation scheme. The last 12 points from each of the competition series was used as a validation set. All error figures reported in table I refer to mean errors on these 12 points, for models trained on the remaining points.

Since the competition submissions will be evaluated using the SMAPE error measure, we used the same error measure as a basis for optimizing, comparing, and selecting prediction methods based on validation scores on the witheld 12 last points. Table I gives an overview of the block-wise mean SMAPEs for three of our baseline predictors and the ESN predictors.

B. Decomposition

All the time series were decomposed into trend-cycle, seasonal and residual multiplicative components using the X-12-ARIMA seasonal adjustment program developed at the United States Bureau of the Census (`//www.census.gov/srd/www/x12a/`). We used the automatic ARIMA model selection procedure that is implemented in the program to find a suitable model for forecasting and backcasting the time series. A moving average with a window size of 39 for the trend estimation was used to produce smoother trends that we found are better handled by our prediction method.

Other decompositions (additive instead of multiplicative, STL, other smoothing window sizes) were tested for validation set SMAPE with the ESN method and found inferior (although often only by a small margin). The ESN method was applied to these three components individually, and the component predictions reconstituted to the original format by multiplication.

C. Applying the ESN-based method block-wise

For each of the three component versions of each of the six blocks, a collective of 500 ESNs was trained to predict that particular block-component. More specifically, if a block had N members, 500 reservoirs were randomly created, and each of them was trained individually on the task to predict the N-vector of time series one time step ahead. For training, the competition series minus the last 12 points were used. After training, the last 12 points were predicted by each ESN, via iterated 1-step predictions; the trend/season/residual component predictions of each ESN were recombined; these 500 12-step predictions were then averaged; and finally, the mean SMAPE (across the N series of the block) on the resulting mean-voted combination prediction of the 12 last block steps was calculated.

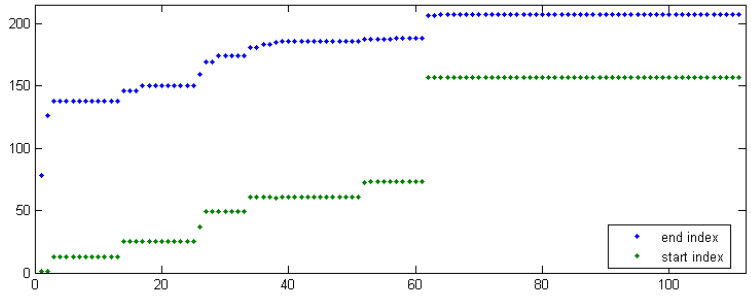

Fig. 1. Time span of time series (sorted by end date). The grouping into 6 "blocks" is clearly discernible.

Block	Members	Theta	Dampen	SVR	ESN
1	65 71 74 81 93 95 96 97 98 110 111	23.4	27.2	26.6	20.7
2	58 62 84 66 78 79 83 85 86 102 103 106	12.8	12.8	14.0	11.5
3	69 70 60 61 72 89 105	15.1	17.7	15.10	13.0
4	51-57 63 67 68 73 75 77 80 87 90 101 107	7.4	7.9	6.4	5.6
5	59 64 82 91 92 94 99 100 104 108	10.9	9.2	9.3	8.7
6	1-50	19.2	18.9	17.6	17.5

TABLE I

BLOCK MEMBERS AND MEAN SMAPE SCORES OF FOUR PREDICTORS ON THE VALIDATION SET OF THE LAST 12 POINTS (MODELS TRAINED ON REMAINING INITIAL POINTS).

Each ESN was set up without external input, and with feedback from the output units into the reservoir (identical to the setup described in [4] for chaotic time series prediction).

In most blocks, some series were shorter or longer than others by a few steps. This was dealt with by trimming all series which had "too early" values to the latest beginning time in the block. To cope with unequal end points, a two-stage learning/prediction process was implemented that first filled the "end-gaps" and then proceeded to generate the requisite further prediction points.

An ESN reservoir is a random excitable medium, whose dynamic response characteristics is crucial for the accuracy in a given modelling task. This characteristics is shaped by a small number of global scaling parameters (global scaling of reservoir weights, output feedback weights, plus a global leaking rate for the leaky integrator neurons which were used), as well as the network size and the Tikhonov regularizing constant which enters the linear regression (these last two parameters affect not the dynamical properties of the ESNs but the model capacity in the sense of statistical learning theory). All in all, there were five global parameters which had to be optimized per block and per decomposition component. This was done by manual experimentation, using the validation set performance as a guide. Automated, stochastic-gradient based optimization methods are an object of current research [5] and were not robustly available at this time.

The reservoir sizes that resulted from this manual tuning ranged between 45 (for block 6 with its short series) and 110 (for the longest blocks). A more detailed survey will be given in an eventual long paper.

IV. RESULTS

Of course, at the time of writing – a few minutes before the submission deadline – this section must remain essentially void. The validation set SMAPEs of the ESN method (see table I) look encouraging, but ... "the future's not ours to see".

REFERENCES

[1] V. Assimakopoulos and K. Nikolopoulos. The theta model: a decomposition approach to forecasting. *International Journal of Forecasting*, 16:521–530, 2000.

[2] H. Jaeger. Short term memory in echo state networks. GMD-Report 152, GMD - German National Research Institute for Computer Science, 2002. http://www.faculty.iu-bremen.de/hjaeger/pubs/STMEchoStatesTechRep.pdf.

[3] H. Jaeger. Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach. GMD Report 159, Fraunhofer Institute AIS, 2002. http://www.faculty.iu-bremen.de/hjaeger/pubs/ESNTutorial.pdf.

[4] H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304:78–80, 2004. http://www.faculty.iu-bremen.de/hjaeger/pubs/ESNScience04.pdf.

[5] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert. Echo state networks with leaky integrator neurons. *Neural Networks*, 2007, to appear.

[6] H. Jaeger, W. Maass, and J. Principe. Special issue on echo state networks and liquid state machines. *Neural Networks*, 20(3), 2007. doi:10.1016/j.neunet.2007.04.001.

[7] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002. http://www.lsm.tugraz.at/papers/lsm-nc-130.pdf.

[8] S. Makridakis and M. Hibon. The M3-competion: results, conclusions and implications. *Int. J. of Forecasting*, 16:451–476, 2000.

[9] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman. *Forecasting: Methods and Applications*. John Wiley and Sons, Hoboken, NJ, 1998.

[10] J McNames. *Innovations in local modeling for time series prediction*. Phd thesis, Dept. of Electrical Engineering, Stanford University, 1999. www.ee.pdx.edu/∼mcnames/Publications/Dissertation.pdf.

[11] Goodwin P. and Lawton R. On the asymmetry of the symmetric MAPE. *International Journal of Forecasting*, 15(4):405–408, 1999.

[12] O. Renaud, J.-L. Starck, and F. Murtagh. Wavelet-based combined signal filtering and prediction. *IEEE Trans. on Systems, Man and Cybernetics B*, 35(6):1241– 1251, 2005.