

# Empirical Evaluation of the Cycle Reservoir with Regular Jumps for Time Series Forecasting: A Comparison Study

Mais Haj Qasem<sup>1</sup>(✉), Hossam Faris<sup>1</sup>, Ali Rodan<sup>1</sup>, and Alaa Sheta<sup>2</sup>

<sup>1</sup> King Abdullah II School for Information Technology,  
The University of Jordan, Amman, Jordan  
[mais\\_hajqasem@hotmail.com](mailto:mais_hajqasem@hotmail.com)

<sup>2</sup> Computing Science Department, Texas A&M University-Corpus Christi,  
Corpus Christi, TX, USA

**Abstract.** The cycle reservoir with regular jumps (CRJ) is a recent deterministic reservoir model with a very simple structure and highly constrained weight values. CRJ was proposed as an alternative to the randomized Echo State Network (ESN) reservoir. In this work, we empirically evaluate the performance of CRJ for time series forecasting problems, and compare it to ESN and Auto-Regressive with eXogenous inputs (NARX) models. The comparison is conducted based on seven time series datasets that represent different real world cases. Simulation results show that CRJ outperforms ESN and NARX models. The results also demonstrate the effectiveness of CRJ when applied for different time series forecasting problems

**Keywords:** Echo State Network · Recurrent neural network · Reservoir Computing · Times series data · Forecasting

## 1 Introduction

Time series data are observations of well-defined data items that represent repeated measurements over a period of time, such as a month, quarter, or year [1]. A time series shows the actual movements in the data over time caused by cyclical, seasonal, and irregular events on the data item being measured. Time series data are used in different areas such as statistics, signal processing, pattern recognition, earthquake prediction, weather forecasting, trajectory forecasting, control engineering and communications engineering.

The development of a time series forecasting for nonlinear behavior represents a challenge for both engineers and mathematicians. Typically, nonlinear time series modeling involves two major steps: the selection of a model structure with a certain set of parameters and the selection of an algorithm which estimate these parameters. The later issue usually biases the former one. There are still many unsolved problems related to the implementation and design of time series models.

In literature, there is a wide range of machine learning algorithms proposed and applied for the task of time series forecasting. One of the common types of these algorithms is the Echo State Networks (ESN) [2]. ESN is a supervised learning recurrent neural network (RNN) with a fixed random hidden (reservoir) layer and a memoryless readout. The aim of ESN is to drive large, random and fixed RNN for the input signal, where the neurons (units) are promoted in the reservoir network through a nonlinear response signal [3]. The desired output signal is merged with a trainable linear combination of all response signals. The basic idea of ESN is similar to that of the liquid state machine (LSM), which was developed independently by Mass et al. [4].

However, standard ESN possesses several drawbacks, which affect its acceptability. First, the fixed random reservoir is difficult to understand. Second, the reservoir specification and input connections require many trials. Third, imposing a constraint on the spectral radius of the reservoir matrix is useless when setting the reservoir parameter [5]. Lastly, the reservoir's connectivity and weight structure are not optimal, and the reservoir's dynamic organization is still unclear.

In attempt to overcome these problems, Rodan and Tino [6] proposed the deterministic Cycle Reservoir with regular Jumps (CRJ). CRJ is considered as a new class of state-space reservoir models where it possesses a fixed state transition structure (the "reservoir") and an adjustable readout from the state space as in all Reservoir Computing (RC) models.

CRJ has highly-constrained weight values while the nodes in the reservoir are connected into a unidirectional cycle with a fixed value  $r_c$ , similar to that of the Simple Cycle Reservoir (SCR) [7]. In addition to that, a bidirectional jump weight  $r_j$  is found which serves as a shortcut for the CRJ network. Previous works have shown that the addition of these regular jumps can improve the performance of the model [6]. Recently, CRJ has shown very promising performance in different types of applications [8,9].

In this work, we investigate the application of CRJ [6] for different time series forecasting problems. For this purpose, seven time series datasets of different real world applications are utilized. The performance of the developed CRJ model is evaluated and compared with ESN [2,10], and the NARX model. The ultimate goal of this study is to reveal the efficiency of CRJ when used for times series forecasting in different applications.

This paper is organized as follows: all methods used in this work for the task of time series forecasting are described in Sect. 2. The selected time series datasets for the purpose of evaluating and benchmarking are presented in Sect. 3. The details of the conducted experiments and the discussion are given in Sect. 4. Finally, the findings of this work are summarized in Sect. 5.

## 2 Methods

### 2.1 Echo State Network (ESN)

ESN is a discrete time recurrent neural network with  $\{A\}$  input units,  $\{M\}$  internal units and  $\{S\}$  output units over discrete time slots  $n = \{1, 2, 3, \dots\}$ .

The activation of the ESN is expressed using a vector for each layer, as given in Eq. 1.

$$a(n) = (a_1(n), \dots, a_A(n))^T, b(n) = (b_1(n), \dots, b_M(n))^T, c(n) = (c_1(n), \dots, c_S(n))^T \quad (1)$$

The linking weights between the neurons are gathered in  $M \times A$  size matrix, for the input, which is referred to by  $W^{in} = (W_{ji}^{in}), M \times M$  size matrix for the internal weight, which is referred to by  $W = (w_{ij}), S \times (A + M)$  size matrix for the output, which is referred to by  $W^{out} = w_{ij}^{out}$ , and  $M \times S$  size matrix for the connection that projects back from output to the internal unit, which is referred to by  $w_{back} = w_{ij}^{back}$ .

Unlike RNN, where all the weights for the inputs, internals and output are adaptable, in ESN the reservoir connection weights as well as the input weights are randomly generated and fixed (non trainable). The only trainable part is the output weights. The fixed random weights for the input and reservoir layers are then scaled with a chosen values,  $v$  for the input and  $\lambda$  for the reservoir, where  $v, \lambda \in (0, 1)$ . Moreover, the spectral radius of the reservoir matrix should be less than 1 to ensure a sufficient condition for the ‘‘echo state property’’ (ESP). By doing so, ESN ensures that the reservoir state is an ‘‘echo’’ for all input history. The internal units are updated, when moving from time slot  $n$  to time slot  $n + 1$ , according to Eq. 2.

$$b(n + 1) = f(W^{in}a(n + 1) + Wb(n) + W^{back}c(n) + z(n + 1)) \quad (2)$$

Where  $f$  is the reservoir activation function (usually tangent hyper function (tanh)) and  $z$  is optional small white noise that might be needed in some cases for solving the overfitting problem. The linear output is computed using Eq. 3.

$$c(n + 1) = f^{out}(W^{out}x(n + 1)) \quad (3)$$

Where  $f_{out} = (f_{out}^1, \dots, f_{out}^S)$  are the output units function and  $x(n + 1) = [b(n + 1); a(n + 1)]$  are a concatenation for the internals and the input activation vectors.

## 2.2 Cycle Reservoir with Regular Jump (CRJ)

CRJ is a simple deterministic reservoir model with highly constrained weight values [6]. CRJ deterministically generates reservoir that could have the potential of a better performance than standard ESN and other models previously proposed in the literature [7].

To implement CRJ, you need to optimize several parameters including the cycle weight  $r_c$ , jump weight  $r_j$ , input weight  $v$ . Then, the reservoir size  $N$ , similar to ESN, is determined. Moreover, the number of input and output units with the added bias value to input units are also determined based on the nature of the task and the target output.

Unlike ESN, CRJ has a simple regular topology with full connectivity between the input and reservoir, there is no need to specify different weight

value for each connection between two nodes, where all the reservoir nodes connected via unidirectional cycle with the same value  $r_c$ . The value of  $r_c$  should be on the range of  $[0,1]$ .

CRJ also has a bidirectional shortcut (jumps) between the reservoir units  $r_j$ . These jumps increase the density of the connections in the internal units, which in term facilitates a good training. Unlike ESN, which generates the input weight randomly, CRJ required to set its input weight sign values in a complete deterministic manner. The deterministic input signs are generated based on the  $\pi$  digits where each digit is thresholded at 4; if the value of the digit is between  $0 \leq \text{digit} \leq 4$  then the connection sign will be minus (-), and if the value of the digit is between  $5 \leq \text{digit} \leq 9$  the connection sign will be positive (+).

### 2.3 Auto-regressive with eXogenous Inputs (NARX)

NARX model was first presented in 1985 by Leontaritis and Billings [11, 12] as a means of describing the input-output relationship of a nonlinear system [13]. Time Series prediction using the NARX model was explored in many articles [14, 15]. The general NARX model structure can be represented using the following nonlinear differential equation:

$$y(t) = f(y(t-1), \dots, y(t-n), u(t-1), \dots, u(t-m)) \quad (4)$$

$f$  represents a nonlinear mapping between the system input  $u(t)$  and the past outputs  $y(t-1), y(t-2), \dots$ . The order of the model input and output is assumed to be  $n$  and  $m$ , respectively. The NARX model can be represented as given in Eq. 4. The model parameters can be estimated using LSE.

$$\begin{aligned} y(t) = & a_0 + \sum_{k_1=1}^n f_{k_1}(x_{k_1}) \\ & + \sum_{k_1=1}^n \sum_{k_2=k_1}^n f_{k_1 k_2}(x_{k_1}(t), x_{k_2}(t)) + \dots \\ & + \sum_{k_1=1}^n \dots \sum_{k_l=k_{l-1}}^n f_{k_1 k_2 \dots k_l}(x_{k_1}(t), \dots, x_{k_l}(t)) \end{aligned} \quad (5)$$

Given that:

$$\sum_{k_1=1}^n \dots \sum_{k_l=k_{l-1}}^n f_{k_1 k_2 \dots k_z}(x_{k_1}(t), \dots, x_{k_z}(t)) = a_{k_1 k_2 \dots k_z} \prod_{i=1}^z x_{k_i}(t) \quad (6)$$

$z$  is in the interval of  $[1, l]$ .  $a_{k_1 k_2 \dots k_z}$  are the model parameters to be estimated.

$$x_k(t) = \begin{cases} y(t-k) & \text{if } 1 \leq k \leq n \\ u(t-(k-n)) & \text{if } n+1 \leq k \leq n+m \end{cases}$$

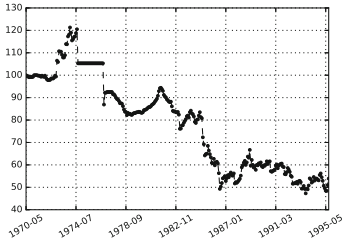
Identifying a NARX model requires two steps: (1) pick the best model structure [16], (2) estimating the model parameters. NARX model was used to solve many time series analysis, modeling and identification of nonlinear systems [17, 18].

### 3 Datasets

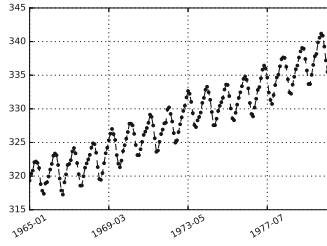
Seven time-series datasets are drawn from DataMarket Repository for experimenting and benchmarking the described models in the previous section. The DataMarket Repository is sponsored by Qliktech. DataMarket delivers intuitive platform solutions for self-service data visualization, guided analytic applications, embedded analytics, and reporting to approximately 40,000 customers worldwide [19]. The selected datasets for our experiments represent real world data including: financial forecasting problems, unemployment rates, environmental modeling and pollutants concentrations. The datasets describe a variety of problems over different time periods and have different levels of complexity. All datasets are described in Table 1 and depicted in Fig. 1.

**Table 1.** Dataset

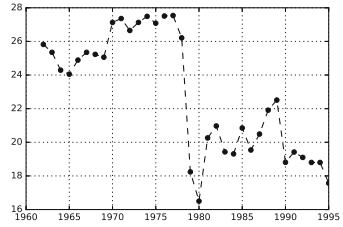
	Dataset name	Feature
1	Exchange Rate TWI	304 fact values in 1 monthly time series
2	CO2 (ppm) mauna loa	192 fact values in 1 yearly time series
3	High and low water levels of the Amazon at Iquitos	34 fact values in 1 yearly time series
4	Annual common stock price, U.S.	112 fact values in 1 yearly time series
5	Weekly closing price of AT&T common share	56 fact values in 1 Weekly time series
6	Women unemployed (1000's) U.K.	80 fact values in 1 monthly time series
7	Highest mean monthly level, Lake Michigan	109 fact values in 1 monthly time series



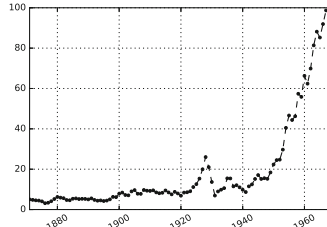
(a) Exchange Rate TWI dataset



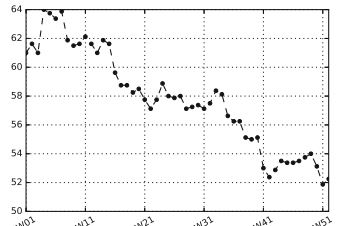
(b) CO2 (ppm) mauna loa



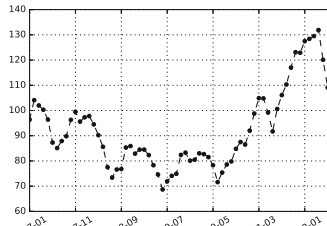
(c) High and low water levels of the Amazon at Iquitos



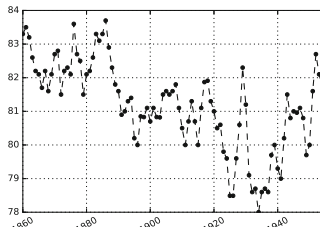
(d) Annual common stock price, U. S



(e) Weekly closing price of AT&T common share



(f) Women unemployed (1000's) U.K.



(g) Highest mean monthly level, Lake Michigan

**Fig. 1.** Seven representative time series.

## 4 Experiments and Result

### 4.1 Parameters Settings

All datasets are divided into two sets for training and testing; 70% was used for training, and the rest is used for testing. In order to obtain the best performance of each model in terms of lowest Error, the parameters of the models should be optimized. Therefore, multiple values for each parameter are tested, and the best value is used to obtain the final output.

- **Echo State Network (ESN):** For spectral radius ( $\lambda$ ), 20 different values in the range [0.05–1] were tested. For connectivity ( $con$ ), 10 different values in the range [0.05–0.5] were tested. The model was also tested under different internal unit sizes ( $N$ ) in the range [50–500]. The final and best parameter’s values for the seven datasets used are shown in Table 2.
- **Cycle Reservoir with Regular Jump (CRJ):** For internal unit weights ( $r_c$ ) and ( $r_j$ ), 20 different values for each parameter in the range [0.05–1] were tested. For input unit scaler ( $v$ ), 20 different values in the range [0.05–1] were tested. Also as in ESN, the model was also tested under different internal unit sizes ( $N$ ) in the range [50–500]. For the jump size,  $N/2$  different jump values were tested. The final and best parameter’s values for the seven datasets are provided in Table 3.
- **NARX:** The Levenberg-Marquardt training algorithm is utilized to train the model for all the datasets. Different number of hidden nodes are tested for each dataset starting with 5 neurons up to 50 with a step of 5. The results are reported for this model by averaging the obtained RMSE values over 10 independent runs. Evaluation results along with the best parameters are shown in Table 4 for the seven datasets.

Evaluation of the models performance will be done via Normalized Mean Square Error (NMSE), as given in Eq. 7.

$$RMSE = \sqrt{\frac{\sum_{n=1}^T (\hat{c}(n) - c(n))^2}{T}} \quad (7)$$

Where  $\hat{c}(n)$  is a predicted output,  $c(n)$  is a desired output.

### 4.2 Comparison Results

The final evaluation results of the CRJ, ESN and NARX models are summarized and listed in Table 5. The results are presented in terms of RMSE and the standard deviation of the 10 independent runs of each model (denoted as  $RMSE \pm STD$ ). Note that CRJ models don’t have a standard deviation since they are deterministic models. On the other had, ESN and NARX yielded different RMSE results over different datasets due to their random weight generation.

As it can be noticed in the evaluation results, the CRJ model showed the lowest RMSE values for all datasets. Examining the results of the other two

models, we can't see a dominant model between NARX and ESN as a second best model for all datasets. ESN model achieved the second best evaluation results in four datasets (3, 4, 5, 6), while NARIX model was the second best in three datasets (1, 2, 7). Moreover, comparing NARIX to ESN in terms of standard deviation, we can notice that NARX model is more robust since it showed lower values in most of datasets, while ESN showed noticeably high values of standard deviation in Datasets 4 and 6. Overall, we can conclude that the CRJ model is very efficient when applied for complex time series forecasting problems.

**Table 2.** ESN result

Dataset	$\lambda$	$con$	$N$	RMSE
1	0.05	0.4	125	2.83
2	0.1	0.35	325	2.57
3	0.95	0.2	50	1.80
4	0.05	0.05	175	1.40
5	0.95	0.35	150	1.48
6	0.8	0.05	300	2.02
7	0.05	0.15	150	2.57

**Table 3.** CRJ result

Dataset	$r_c$	$v$	$r_j$	Step size	$N$	RMSE
1	0.05	0.3	0.05	1	450	1.72
2	0.1	0.05	0.1	10	100	1.23
3	0.95	0.6	0.05	20	235	1.38
4	0.05	0.05	0.6	1	50	1.02
5	0.05	0.1	0.05	1	400	1.25
6	0.1	0.95	0.05	10	75	1.05
7	0.05	0.2	0.05	1	500	0.84

**Table 4.** NARX result

Dataset	$N$	Delay	RMSE
1	10	2	2.59
2	50	2	1.99
3	10	2	5.71
4	20	2	2.79
5	10	2	1.56
6	10	2	4.70
7	50	2	1.45



**Table 5.** Comparison result.

Dataset	ESN	CRJ	NARIX
	RMSE $\pm$ STD	RMSE	RMSE $\pm$ STD
1	2.83 $\pm$ 1.81	1.72	2.59 $\pm$ 1.53
2	2.57 $\pm$ 2.22	1.23	1.99 $\pm$ 2.25
3	1.80 $\pm$ 2.68	1.38	5.71 $\pm$ 1.86
4	1.40 $\pm$ 4.41	1.02	2.79 $\pm$ 1.94
5	1.48 $\pm$ 1.53	1.25	1.56 $\pm$ 1.60
6	2.02 $\pm$ 5.76	1.05	4.70 $\pm$ 1.49
7	2.57 $\pm$ 2.39	0.84	1.45 $\pm$ 1.32

The CRJ model has the advantage of simplicity and robustness when compared to other well known models like ESN and NARX.

## 5 Conclusion

In this work, the application of cycle reservoir with jumps (CRJ) model is evaluated for time series forecasting. For the purpose of benchmarking and evaluation, seven time series datasets that describe different real world applications are utilized. The evaluation results of CRJ are compared with those obtained for two well regarded models which are the Echo State Network (ESN) and Auto-Regressive with eXogenous inputs (NARX). Evaluation results showed that CRJ achieved the lowest RMSE in all datasets. Subsequently, we argue that CRJ has the potential to outperforms ESN and NARIX in terms of accuracy and robustness when applied to time series forecasting problems.

## References

1. Thakur, G.S., Thakur, R.S., Thakur, R.S.: Design of 2-level clustering framework for time series data sets. In: Deep, K., Nagar, A., Pant, M., Bansal, J. (eds.) Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011). Advances in Intelligent and Soft Computing, vol. 131, pp. 205–212. Springer, New Delhi (2012)
2. Jaeger, H.: The “echo state” approach to analysing and training recurrent neural networks-with an Erratum note. Bonn, Ger.: Ger. Natl. Res. Cent. Inf. Technol. GMD Tech. Rep. **148**(34), 13 (2001)
3. Jaeger, H., Lukoševičius, M., Popovici, D., Siewert, U.: Optimization and applications of echo state networks with leaky-integrator neurons. Neural Netw. **20**(3), 335–352 (2007)
4. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Comput. **14**(11), 2531–2560 (2002)

5. Schrauwen, B., Defour, J., Verstraeten, D., Campenhout, J.: The introduction of time-scales in reservoir computing, applied to isolated digits recognition. In: Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D. (eds.) ICANN 2007. LNCS, vol. 4668, pp. 471–479. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-74690-4\\_48](https://doi.org/10.1007/978-3-540-74690-4_48)
6. Rodan, A., Tiño, P.: Simple deterministically constructed cycle reservoirs with regular jumps. *Neural Comput.* **24**(7), 1822–1852 (2012)
7. Rodan, A., Tino, P.: Minimum complexity echo state network. *IEEE Trans. Neural Netw.* **22**(1), 131–144 (2011)
8. Qasem, M.H., Al Assaf, M.M., Rodan, A.: Data mining approach for commercial data classification and migration in hybrid storage systems. *World Acad. Sci. Eng. Technol. Int. J. Comput. Electr. Autom. Control Inform. Eng.* **10**(3), 481–484 (2016)
9. Rodan, A., Faris, H.: Credit risk evaluation using cycle reservoir neural networks with support vector machines readout. In: Nguyen, N.T., Trawiński, B., Fujita, H., Hong, T.-P. (eds.) ACIIDS 2016. LNCS (LNAI), vol. 9621, pp. 595–604. Springer, Heidelberg (2016). doi:[10.1007/978-3-662-49381-6\\_57](https://doi.org/10.1007/978-3-662-49381-6_57)
10. Jaeger, H.: Adaptive nonlinear system identification with echo state networks. *Networks* **8**(9), 17 (2003)
11. Leontaritis, I.J., Billings, S.A.: Input-output parametric models for non-linear systems. Part I: deterministic non-linear systems. *Int. J. Control* **41**(2), 303–328 (1985)
12. Leontaritis, I.J., Billings, S.A.: Input-output parametric models for non-linear systems. Part II: stochastic non-linear systems. *Int. J. Control* **41**(2), 329–344 (1985)
13. Mohamed Vall, O.M., M’hiri, R.: An approach to polynomial NARX/NARMAX systems identification in a closed-loop with variable structure control. *Int. J. Autom. Comput.* **5**(3), 313–318 (2008)
14. Menezes Jr., J.M.P., Barreto, G.A.: Long-term time series prediction with the narx network: an empirical evaluation. *Neurocomputing* **71**(16–18), 3335–3343 (2008)
15. Pisoni, E., Farina, M., Carnevale, C., Piroddi, L.: Forecasting peak air pollution levels using NARX models. *Eng. Appl. Artif. Intell.* **22**(4–5), 593–602 (2009)
16. Ho, C.K.S., French, I.G., Cox, C.S., Fletcher, I.: Genetic algorithms in structure identification for NARX models. In: Smith, G.D., Steele, N.C., Albrecht, R.F. (eds.) *Artificial Neural Nets and Genetic Algorithms*, pp. 597–600. Springer, Vienna (1998)
17. Menezes Jr, J.M.P., Barreto, G.A.: A new look at nonlinear time series prediction with NARX recurrent neural network. In: *The 2006 Ninth Brazilian Symposium on Neural Networks*, pp. 160–165, October 2006
18. Diaconescu, E.: Prediction of chaotic time series with NARX recurrent dynamic neural networks. In: *Proceedings of the 9th WSEAS International Conference on International Conference on Automation and Information. ICAI 2008*, pp. 248–253. World Scientific and Engineering Academy and Society (WSEAS) (2008)
19. Qliktech: DataMarket Repository (2017). Accessed 12 Feb 2017