



The Usage of New Reservoir Computing Approach for the Detection of Chaos

Ali Rida Ismail, Slavisa Jovanovic, S. Petit-Watelot, Hassan Rabah

► To cite this version:

Ali Rida Ismail, Slavisa Jovanovic, S. Petit-Watelot, Hassan Rabah. The Usage of New Reservoir Computing Approach for the Detection of Chaos. Colloque du GDR SoC2, GDR SoC2, Jun 2021, Rennes, France. hal-03981866

HAL Id: hal-03981866

<https://hal.science/hal-03981866>

Submitted on 13 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Usage of New Reservoir Computing Approach for the Detection of Chaos

Ali Rida ISMAIL, Slavisa JOVANOVIĆ, Sébastien PETIT-WATELOT, Hassan RABAH
 Universit  de Lorraine, CNRS, IJL, F-54000 Nancy, France
 {ali-rida.ismail, slavisa.jovanovic, sebastien.petit, hassan.rabah}@univ-lorraine.fr

Abstract—This paper introduces a new reservoir computing approach, called Conceptor-Driven Network (ConDN), for the detection of chaos in time-series. Two known parametrizable systems exhibiting chaotic and non-chaotic behaviours have been analysed. By only observing the selected output results (trace matrix) after training the ConDN, the systems can be classified as chaotic or non-chaotic. In addition, the robustness of this method against white noise is also investigated.

Index Terms—reservoir computing (RC), Conceptor-Driven RC, complex dynamics, detection of chaos, noise robustness.

I. INTRODUCTION

Chaos was first introduced by Poincar  in the later 19th century. A chaotic system is a nonlinear system whose behaviour is unpredictable due to its high sensitivity to initial conditions. Indeed, a chaotic system with nearby initial conditions will end up with extensive divergence of their trajectories [1]. The rate of divergence is specified by the *largest Lyapunov exponent*, whose positiveness probably signifies the presence of chaos. Another method for chaos detection is the 0-1 Test [2], that is more decisive (binary outcome), performs well on short data lengths, and works directly on time-series without the need to reconstruct the original dynamics. However, both methods merely quantifies the degree of "sensitivity to initial conditions" (i.e. local instability in a state space), that can arise from measurement noise, system noise, or other reasons.

Reservoir Computing (RC) is a relatively novel paradigm for training Recurrent Neural Networks (RNNs), first introduced with Liquid State Machines (LSMs) and Echo State Networks (ESNs) [3]. These RC approaches have been used in different scientific domains such as: robust image recognition, modeling of spintronic devices [4], and non-linear time series analysis. In this paper, we propose a Reservoir Computing approach for the detection of chaos in time-series. By only introducing an input time-series to the network, the conclusions about its chaoticity can be given. Moreover, the robustness of the proposed method against stochastic noise is also considered.

The rest of the paper is organized as follows: in Section II, the Conceptor-driven Network (ConDN) approach is demonstrated in details. In Section III, the approach is applied on two conventional systems exhibiting chaotic and non-chaotic behaviours. In this section, the results of the conceptors' analysis for the chaos detection as well as the robustness of such analysis against noise are demonstrated. Finally, Section IV shortly summarizes the whole content.

II. CONCEPTORS-DRIVEN NETWORK

The Conceptor-Driven network (ConDN), is a relatively new Reservoir Computing Approach [5], that has a Long-term memory (LTM) [6] capable of storing large number of time series and re-extracting them. Consequently, a ConDN undergoes two phases sequentially: *Loading phase* in which the time series $p^j(n)$ are uploaded into the network, and *Recall phase* in which any time-series is regained using its key, called *Conceptor* C^j (j denotes the index of an input). Each input pattern p^j is assigned by a conceceptor matrix C^j . Thus, by

studying the characteristics of matrices C^j , the natural behaviour of the corresponding patterns p^j can be revealed.

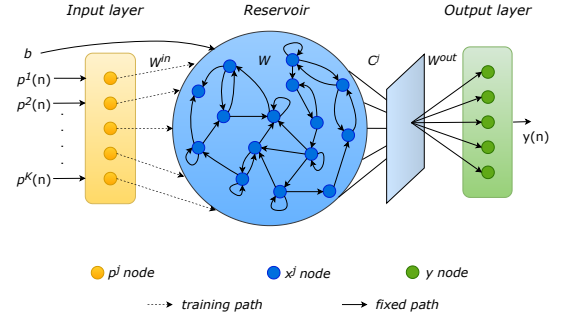


Figure 1: General structure of Conceptor-driven Network.

To begin with, an N -neurons RNN is built by randomly creating an $N \times N$ internal connection matrix W^* , an $N \times K$ input weights matrix W^{in} , and an N random bias vector b . The readout layer is represented by the readout weight matrix W^{out} which is trained using any random signal. Taking a white noise input $\nu(n)$, the states are collected (Eq. 1a) and the optimization problem (Eq. 1b) is solved:

$$x^\nu(n+1) = f(W^*x^\nu(n) + W^{in}\nu(n) + b) \quad (1a)$$

$$\min_{W^{out}} \sum_n (W^{out}x^\nu(n) - \nu(n))^2 \quad (1b)$$

In the loading phase, the scalar patterns $p^j(n)$ always drive the network by the state-update equation shown in 2a, where p^j enter the reservoir separately to compute their corresponding x^j . After a washout period n_0 , the quadratic loss of Eq. 2b is minimized:

$$x^j(n+1) = \tanh(W^*x^j(n) + W^{in}p^j(n) + b) \quad (2a)$$

$$\sum_{j=1, \dots, K} \sum_{n=n_0, \dots, L} \|W^*x^j(n) + W^{in}p^j(n) - Wx^j(n)\|^2 \quad (2b)$$

Accordingly, the characteristics of the input patterns are transmitted to the input internalization weights matrix W . In recall phase, the network is driven by the "conceptors" (Eq. 3a). At this stage, the ConDN is ready for use. It is able to predict any of the registered patterns, by running the state-update equation (Eq. 3a) followed by the output equation shown in 3b:

$$x^j(n+1) = C^j f(Wx^j(n) + b) \quad (3a)$$

$$y(n) = W^{out}x(n) \approx p(n) \quad (3b)$$

A. Conceptors Analysis

In the recall phase, the input patterns $p^j(n)$ are no more in use. However, the ConDN is driven by the "conceptors" to avoid any unpredictable dynamics at the output. Each registered pattern $p^j(n)$ is represented in the recall phase by an $N \times N$ conceceptor matrix C^j as shown in Eq. 3a. C^j can be seen as a filter, that allows the passage

System	Ch.Rossler	Ch.Chen	NonCh.Rossler	NonCh.Chen
$\Delta(\%)$	27.97	28.69	-0.84	1.48

of the state dynamics x^j related to the pattern $p^j(n)$ (identity matrix), while trying to ban all other dynamics associated to different patterns (null matrix). Accordingly, the following quadratic loss is set:

$$L(C^j) = E\|C^j x^j(n) - x^j(n)\|^2 + (\gamma^j)^{-2} \|C^j\|_{fro}^2 \quad (4)$$

where the expectation $E\|\cdot\|^2$ covers all the states $x^j(n)$ collected in the testing phase (Eq. 2a), and $\|\cdot\|_{fro}^2$ is the squared Frobenius matrix norm. The first term of Eq. 4 converges to its minimum if C^j is an identity matrix, whereas the second term takes its minimal value if C^j is a null matrix. The aperture parameters γ^j should be carefully hand tuned as they affect the convergence speed of C^j . After minimizing $L(C^j)$, the following solution of C^j is obtained:

$$C^j = R^j(R^j + (\gamma^j)^{-2}I)^{-1} \quad (5)$$

where $R^j = E[x^j(n)x^j(n)^T]$ are the correlation matrices.

To study a given time-series, we firstly sample it by factor 2. Then we introduce both, the raw and the sampled signals, into one ConDN. Knowing C^1 and C^2 , we apply the Singular-Value Decomposition (SVD) to compute the diagonal matrices S^1 and S^2 :

$$C^j = U^j S^j U^{jT}, \quad (6)$$

After that, we calculate the quantities $Trace(S^1)$ and $Trace(S^2)$. $Trace(S^j)$ represents the summation of all the eigenvalues of C^j , being arranged by convention on the diagonal of S^j in the descending order. Thus, the trace value varies inversely proportional to the level of stability of the studied signal. Moreover by taking into consideration the effect of sampling on stability [7], the values of $Trace(S^1)$ and $Trace(S^2)$ are expected to be distinct if the input pattern p^1 is chaotic, otherwise $Trace(S^1) \approx Trace(S^2)$. For a better vision, we compute the Variation Percentage (Δ) as follows:

$$\Delta(\%) = 100 \times \left\| \frac{Trace(S^2) - Trace(S^1)}{Trace(S^1)} \right\| \quad (7)$$

III. RESULTS AND DISCUSSION

In this part, we consider two systems for study: Rossler and Chen. For each system, two different behaviours corresponding to different input parameters are generated: chaotic and non-chaotic. Figure 2 shows the variation of $Trace(S^1)$ and $Trace(S^2)$ corresponding respectively to the raw and sampled inputs, versus Data-Length T , for the two studied systems. As predicted, the values of $Trace(S^j)$ in the chaotic case are relatively high compared to those of the non-chaotic case, for any value of T . This is mainly because chaotic systems have rich dynamics compared to non-chaotic systems. However, it is still difficult to generalize this finding by setting a fixed border between chaotic and non-chaotic systems.

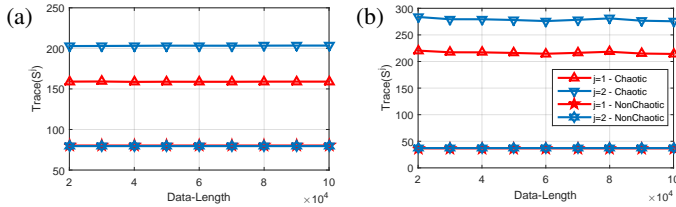


Figure 2: $Trace(S^j)$ versus data length, for chaotic (Δ, ∇) and regular ($\star, *$) behaviours of (a) Rossler and (b) Chen systems.

Another important difference that can be noticed between the two behaviours is the effect of SF . Two distinct graphs of $Trace(S^1)$ and $Trace(S^2)$ are obtained in the chaotic case, separated by a noticeable gap. On the other side, varying SF does not make significant changes

Table I: Percentage difference (Δ in %) for chaotic and non-chaotic behaviours of Rossler and Chen systems.

between the two plots of $Trace(S^j)$. The values of the variation percentage (Δ) are all summarized in Table I. As noticed, the percentage difference is large between the sampled and non-sampled chaotic plots, while it is much smaller between the non-chaotic plots (around 0). We attribute this disparity to the nature of the two behaviours and their vulnerability to the sampling, where sampling a time-varying signal could help to better prominent its natural behaviour. Consequently, by combining the two features discussed above, it becomes more applicable to use the ConDNs to determine the nature of input time-varying signals.

We also investigate the robustness of the proposed method against white noise and its ability to differentiate between chaos and noise. A white noise of a percentage up to 30% is gradually added to all raw time-varying signals, before introducing them to the ConDNs. The corresponding results are all shown in Figure 3. In the chaotic case of Rossler system, the trace method can handle up only to 5% of noise where $Trace(S^1)_{0\%} > Trace(S^1)_{30\%}$. On the other hand, Chen system is more robust and can handle up to 10% of noise. These findings confirms the capability of using the ConDN approach to detect chaos in noisy systems to a given limit.

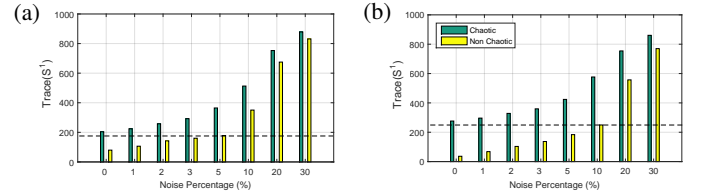


Figure 3: $Trace(S^j)$ versus noise percentage (%) for chaotic (green) and regular (yellow) behaviours of (a) Rossler and (b) Chen systems.

IV. CONCLUSION

In this work, a RC method based on ConDN is proposed for the detection of chaos in time-varying signals. Two systems, Rossler and Chen, are considered to illustrate the findings of such method. The robustness of the proposed approach to white noise is also addressed. This study can be considered as a first milestone in this research, to be further elaborated.

REFERENCES

- [1] G. Benettin, L. Galgani, and J.-M. Strelcyn, "Kolmogorov entropy and numerical experiments," *Physical Review A*, vol. 14, pp. 2338–2345, dec 1976.
- [2] G. A. Gottwald and I. Melbourne, "On the implementation of the 0–1 test for chaos," *SIAM Journal on Applied Dynamical Systems*, vol. 8, pp. 129–145, jan 2009.
- [3] H. Jaeger, "The echo state approach to analysing and training recurrent neural networks," *GMD-Report 148*, German National Research Institute for Computer Science, 01 2001.
- [4] A. R. Ismail, S. Jovanovic, S. Petit-Watelot, and H. Rabah, "Application of reservoir computing for the modeling of nano-contact vortex oscillator," *Electronics*, vol. 8, p. 1315, nov 2019.
- [5] H. Jaeger, "Using conceptors to manage neural long-term memories for temporal patterns," *Journal of Machine Learning Research*, vol. 18, no. 13, pp. 1–43, 2017.
- [6] T. Furlanello, J. Zhao, A. Saxe, L. Itti, and B. Tjan, "Active long term memory networks," 06 2016.
- [7] S. TERACHI and S. OGATA, "Effects of sampling conditions on chaotic characteristics in time series," in *Proceedings of the ISCIE International Symposium on Stochastic Systems Theory and its Applications*, vol. 1997, pp. 83–88, The ISCIE Symposium on Stochastic Systems Theory and Its Applications, 1997.