Parametric Validation of the Reservoir Computing–Based Machine Learning Algorithm Applied to Lorenz System Reconstructed Dynamics

Samuele Mazzi, David Zarzoso

Parametric Validation of the Reservoir-Computing-Based Machine Learning Algorithm Applied to Lorenz System Reconstructed Dynamics

Samuele Mazzi
Aix-Marseille Université, CNRS, PIIM, UMR 7345 Marseille, France
CEA, IRFM, F-13108 Saint-Paul-lez-Durance, France

David Zarzoso
Aix-Marseille Université, CNRS, PIIM, UMR 7345 Marseille, France
Aix-Marseille Université, CNRS, Centrale Marseille, M2P2 UMR 7340, Marseille, France

A detailed parametric analysis is presented, where the recent method based on the Reservoir Computing paradigm, including its statistical robustness, is studied. It is observed that the prediction capabilities of the Reservoir Computing approach strongly depend on the random initialisation of both the input and the reservoir layers. Special emphasis is put on finding the region in the hyper-parameter space where the ensemble-averaged training and generalization errors together with their variance are minimized. The statistical analysis presented here is based on the Projection on Proper Elements method.

Keywords: Reservoir Computing; Lorenz system; Hyperparameters; Error quantification; Machine Learning

1. Introduction

The numerical modelling of multi-scale and complex systems has been continuously growing in the last decade, reaching very high levels of fidelity and accuracy. Some examples can be found in the modelling of turbulence in nuclear fusion plasmas [1], in the prediction of weather and climate in the atmosphere [2], in the multi-scale methods in systems biology [3] and in the analysis of the financial market [4]. This growing phase has inevitably led to the requirement of extremely efficient numerical resources, with the current development of exa-scale high-performance-computing systems, in order to run such very demanding simulations. In some practical applications, however, the elapsed computational time for the exploitation of these numerical modellings represent the bottleneck of the aforementioned validation and/or prediction analyses. Furthermore, in parallel with the im-

provement of the employed numerical techniques, the incorporation of newly discovered physical processes or improved complex multi spatio-temporal scale interactions into the models makes the development of classical numerical methods even more challenging. For these reasons, the application of the rapidly advancing Machine Learning (ML) algorithms has gained a lot of attraction in the framework of the data-driven modelling of complex dynamical systems. The advantages of using such an approach goes from the acceleration of the simulations due to the highly reduced computational cost, to the quality improvement of the physical assessment of the numerical outcomes.

Within the several possible ML techniques applied to the forecasting of complex systems, the Reservoir Computing (RC) has shown appealing features, such as the simplicity and the speed of the training process, and revealed to be able to achieve state-of-the-art results, compared to other Artificial Intelligence Systems (AISs). The RC method takes inspiration from the paradigm of the reservoir computers, which were developed in two independent studies of the early current century [5, 6]. Due to the difficulty of designing efficient recurrent neural network (RNN) architectures and the subsequent inaccuracy of simplistic RNNs [7, 8], Jaeger and Maass proposed the approach of the reservoir computing, which differs from the standard RNN in being essentially split into two levels. The internal layer is composed by the so-called *reservoir*, which is a randomly initialised RNN. The output level, usually called *readout*, is a feed-forward layer which realizes an optimized output function in order to obtain the predicted vector. In our case, as in the case of the RC proposed by Jaeger (the Echo State Network (ESN) [5, 9, 10, 11, 12, 13, 14]), the output function is obtained from a linear regression. The weights of the connected nodes of the *reservoir* are fixed randomly, whereas the *readout* layer is trained to obtain its weights. For more details on the application of the RC technique on the forecasting of dynamical systems one can refer to recent contributions such as, e.g., Refs. [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]. Furthermore, theoretical efforts have been spent throughout the last decade to study the underlying mechanism of the RC algorithm [27, 28, 29, 30, 31, 32, 33, 34]. Among those results, detailed insights prove that a configuration of the weights of the *readout* layer W_{out} exists for which the RC-ESN output dynamics is topologically connected to the input chaotic system [32, 34]. However, the capability of the RC algorithm to find those weights, for which the difference with the input data in the training phase is minimized, still remains elusive.

The present study is motivated by recent contributions [18, 22] in which RC paradigm is used to predict the dynamics of the well-known Lorenz chaotic system [35]. The Lorenz system is given by the equations:

$$\begin{cases} \dot{x} = F_x(x, y, z; a) = a(y - x) & \text{(1a)} \\ \dot{y} = F_y(x, y, z; b) = x(b - z) - y & \text{(1b)} \\ \dot{z} = F_z(x, y, z; c) = xy - cz & \text{(1c)} \end{cases}$$

These equations were derived for the first time by Lorenz, after simplifying a set of partial differential equations describing the motion of a fluid in between two layers of the atmosphere. Therefore, the parameters a, b and c have a real physical meaning, especially b, which correspond to the convection rate. The purpose of this study is to validate and evaluate the accuracy of the RC method applied for reconstructing the dynamics of the Lorenz system. The reported statistical analyses, based on the Projection of Proper elements method [36, 37, 38], demonstrate that the large randomness of the RC approach is strongly related to the random initialisation of the reservoir network components and is highly affected by the chosen configuration parameters. Large variation of the error in the prediction phase is obtained for a broad set of parameters of the reservoir, leading to strong inaccuracy in predicting the time evolution of the Lorenz trajectories for a significant sample of realisations. The results presented in recent contributions [18, 22] are shown to fall within a narrow region of the scanned parameter series. Nevertheless, the prediction error of such a narrow favorable region is affected by large statistical variation as well, making the reconstructed dynamics by this specific ML technique frequently not reliable.

The remainder of the paper is organized as follows: the specific RC approach that has been applied on the Lorenz system [35] and the evaluation techniques, including the Projection on Proper elements method [36, 37, 38], are described in Section 2; Section 3 is divided into five subsections, illustrating the principal results that are achieved. Different hyper-parameters of the reservoir network and of the model are considered and deeply analyzed, with statistical insight of the probability distribution function of the errors; dedicated analysis on the time evolution of the Lorenz coefficients is carried out; the main conclusions are summarized in Section 4.

2. Methods

In this paper we produce the data set by solving the Lorenz system given by Equations 1 using a 4^{th} order Runge-Kutta method (RK4). This integration will produce the data set $\{x_i, y_i, z_i\}$, with N the maximum number of time steps. We will use $\Delta t = 0.02$ and $N = 6250$. Such data set will be divided into a training set and a prediction set, with respective lengths $n_{\text{train}} = 5000$ and $n_{\text{pred}} = 1250$, such that $n_{tot} = n_{\text{train}} + n_{\text{pred}}$. The training set will be used to compute the

parameters of the AIS and the prediction set will be used to test the predictive capabilities of the AIS.

2.1 Reservoir Computing Setup

Figure 1: Schematic representation of the reservoir computing paradigm.

Here we follow exactly the methodology employed by Pathak et al. in Refs. [18, 22] with the same set of parameters they used to build the RC. The architecture of the AIS is shown in figure 1 and will be briefly explained in the following. The input vector is called $\mathbf{u}(t) \in \mathbb{R}^3$ and contains the three components of the Lorenz system, x, y and z. It is fed into the *reservoir*, represented by a square matrix whose element (i, j) indicates the connection between the neuron i and the neuron j. Such matrix is called $\mathbf{A} \in \mathbb{R}^{D_r \times D_r}$, with D_r the number of neurons. If the element $A_{i,j} = 0$, the neurons i and j are not connected. Between the input and the reservoir, one needs an input layer $\mathbf{W}_{in} \in \mathbb{R}^{3 \times D_r}$, mathematically representing an application from $\mathbb{R}^3$ to $\mathbb{R}^{D_r}$. Both the *reservoir* matrix and the input layer are randomly initialised. The elements of both $\mathbf{W}_{in}$ and $\mathbf{A}$ are drawn from a uniform distribution function in [-1,1], with an additional multiplying scaling factor $\gamma_{scal} = 0.1$ applied only to the input layer elements. The elements of the reservoir are re-scaled so that the largest magnitude of the eigenvalues (the so-called spectral radius) is equal to the desired value ρ. In the present study, based on the example of Refs. [18, 22], the *reservoir* matrix $\mathbf{A}$ is given by a sparse Erdős-Rènyi network [39, 40], with an average degree of $d = 6$. It is to be noted that, since the degree of the Erdős-Rènyi network is fixed to $d = 6$, the rewiring probability $p = d/D_r$ is decreased (resp. increased) when D_r is increased (resp. decreased) and therefore the connectivity matrix of the neural network becomes sparser (resp. less sparse). The results can be reproduced also with less weakly connected network [32], and therefore the choice of the sparse Erdős-Rènyi network is made only to be consistent with the aforementioned publications. As it will be presented in the following, the number of reservoir neurons and the spectral radius of the adjacency matrix are widely scanned and the impact of each couple of input parameters on the measured errors is then evaluated. It must be stressed that the Echo State Property [5], which was introduced as a necessary feature for the effectiveness of the RC technique, was originally linked to the rule of thumb $\rho < 1$ for the *reservoir* network. However, this latter consideration has been proven not to be a necessary criterion for the ESP to hold [41, 42]. Hence,

detailed studies in this framework [43, 44, 45] proposed less restrictive conditions, thereby allowing to explore the hyper-parameter space also for the spectral radius $\rho > 1$ of the adjacency matrix A. Justified by this, the analyzed range of ρ values is extended also beyond the unity, as it will be presented in the following.

Each neuron of the reservoir is characterised by a state $\mathbf{r}(t)$, which represents the activation of the neuron (-1: de-activated, $+1$: activated), computed at each time step by a *tanh* activation function:

$$\mathbf{r}(t + \Delta t) = \tanh(\mathbf{A} \cdot \mathbf{r}(t) + \mathbf{W}_{in} \cdot \mathbf{u}(t)) \tag{2}$$

One can note that the recurrence in the RC paradigm is essentially considered in Equation 2, where the neuron states at the i^{th} time step r_i are linked to the neuron states at the previous time steps (r_{i-1}, r_{i-2}, r_{i-3}, and so on). This is also illustrated in the schematic view of the RC architecture of Fig. 1. The particular choice of using the activate function of Equation 2 is made in order to follow the example of Refs. [18, 19, 22], nonetheless diverse RC architectures have been already exploited successfully [46, 47, 48]. The underlying idea of the reservoir computing is to predict the dynamics of the system at the time step $n + 1$, i.e. $\mathbf{u}_{n+1}$ by means of the relation:

$$\mathbf{u}_{n+1} = \mathbf{W}_{out} \cdot \mathbf{r}_n + \mathbf{c}_{out} \tag{3}$$

The weights of the *readout* layer $\mathbf{W}_{out} \in \mathbb{R}^{3\times n}$ can be subsequently computed by minimizing the difference between the actual $\mathbf{u}(t)$ and the predicted $\mathbf{v}(t)$ Lorenz trajectories for each time step of the training phase. Therefore, in order to train the reservoir computing, one has to find $\mathbf{W}_{out}$ and $\mathbf{c} \in \mathbb{R}^3$ minimizing the following quadratic form:

$$||\mathbf{W}_{out} \cdot \tilde{\mathbf{r}} + \mathbf{c} - \mathbf{v}||^2 \tag{4}$$

for all the time steps of the training phase, where $||\mathbf{q}||^2 = \mathbf{q}^T \cdot \mathbf{q}$. The so-called "ridge regression parameter" is set to zero, i.e. $\beta = 0$, since the over-fitting is already avoided [18]. Moreover, it is to be noted that the assumption of symmetry in computing the first two components $x \to -x$ and $y \to -y$ of the Lorenz equations [19, 18, 25] in the predicted vector $\mathbf{v}$ is retained, in order to be consistent with the assumption made in Refs. [18, 22, 19]. For this reason, Equation 4 includes $\tilde{\mathbf{r}}$ instead of $\mathbf{r}$. The modified neuron state matrix $\tilde{\mathbf{r}}$ presents the second half of the third row given by the nonlinear combination $\tilde{r}_i = r_i^2$ (for $i > D_r/2$). However, the same analysis has also been carried out relaxing such symmetry assumption with very similar results.

Then, the minimization problem described in Equation 4 reduces to a linear regression [5], whose solution in the training phase is:

$$\mathbf{W}^*_{out} = (\mathbf{U} \cdot \mathbf{R}^T) \cdot (\mathbf{R} \cdot \mathbf{R}^T)^{-1} \tag{5}$$

where $\mathbf{U} \in \mathbb{R}^{3\times n}$ is the array containing the actual dynamics of the Lorenz system, $\mathbf{R} \in \mathbb{R}^{3\times n}$ the array containing the neuron states (with the nonlinear combination as described above and in Ref. [18]) and $\mathbf{W}^*_{out}$ is the particular solution of the minimization problem. As a result, the *readout* layer $\mathbf{W}_{out}$ is trained and thereby fixed. Hence, the same procedure can be applied in the prediction phase. Indeed, the neuron states are now computed using the predicted vector by:

$$\mathbf{r}(t + \Delta t) = \tanh(\mathbf{A} \cdot \mathbf{r}(t) + \mathbf{W}_{in} \cdot \mathbf{v}(t)) \tag{6}$$

and subsequently applied as:

$$\mathbf{v}_{n+1} = \mathbf{W}_{out} \cdot \mathbf{v}_n \tag{7}$$

for the entire prediction phase. The vector $\mathbf{v}_{n+1} = [\tilde{x}, \tilde{y}, \tilde{z}]_{n+1}$ contains the components of the Lorenz dynamics predicted by the RC approach at the time step $n + 1$.

With respect to traditional RNNs, the RC architecture benefits from the possibility of training the *readout* layer at once, which reduces significantly the elapsed time for the training phase [49], and from the absence of the vanishing gradient issue [50, 25].

Table 1 summarizes the principal parameters of the analyzed RC configurations.

Table 1: Reservoir parameters that has been used for the standard simulations, if not otherwise clearly noted. As explained in the main body of the Letter, the parameters are: β the ridge regression parameter, γ_{scal} the scaling factor applied to the input layer $\mathbf{W}_{in}$, D_r the reservoir number of neurons, ρ the spectral radius of the reservoir network, d the average degree of the Erdős-Rènyi network, Δ_t the time step of the considered Lorenz dynamics, n_{train} and n_{pred} the number of time step in the training and prediction phase respectively, and δ the period of the moving time window for the error calculation. Very similar parameters have been used also in Ref. [18].

Parameter	Value
β	0
γ_{scal}	0.1
D_r	$[50, 610]$
ρ	$[0.5, 2.5]$
d	6
Δ_t	0.02
n_{train}	5000
n_{pred}	1250
δ	1.25

2.2 Error Quantification through the PoPe Method

The AIS predictive capabilities were evaluated in Refs. [18, 22] based on the ability to predict the *climate*, i.e. the ergodic properties of the chaotic system quantified by the Lyapunov exponents. In the present Letter, we employed a different criterion, based on the Projection on Proper elements (PoPe) using the Euclidean distance as the measure to quantify the error of the system to reproduce the Lorenz dynamics. The PoPe method is described in detail in Refs. [36, 37, 38] and here we provide only a brief explanation for our purposes.

Let us assume that we have a set of data $d = \{x_i, y_i, z_i\}_{1 \leq i \leq n_{\text{pred}}}$ provided by the RC. The PoPe method is employed to determine to what extent the set of data d, produced by the the RC technique, has been generated by the Lorenz system given by Equation 1. This is done by computing the coefficients (a, b, c) that better fit d.

For this purpose, let us note that the Lorenz system can be rewritten as follows

$$\begin{cases} \dot{x} = a\,(y - x) + \alpha_x & \text{(8a)} \\ xz + y + \dot{y} = bx + \alpha_y & \text{(8b)} \\ -xy + \dot{z} = -cz + \alpha_z & \text{(8c)} \end{cases}$$

where we have included $(\alpha_x, \alpha_y, \alpha_z)$ for the sake of generality. Such expression highlights the linear dependence on the parameters $(a, b, c, \alpha_x, \alpha_y, \alpha_z)$. Therefore, the calculation of the coefficients which better fit the RC predictions is reduced to a linear regression. To do this, the time derivatives on the LHS of Equations 8 are computed from d using an arbitrarily high precision method (in the present work we use a 4^{th} order finite difference scheme). These derivatives are noted $\left(\tilde{\dot{x}}, \tilde{\dot{y}}, \tilde{\dot{z}}\right)$.

The Lorenz equations can then be rewritten in a matrix form, combining the RC-predicted trajectories d_i with the expression of the computed time derivatives

$$M_i = N_i \cdot \Gamma \tag{9}$$

with $M_i = (\tilde{\dot{x}}_i, \tilde{\dot{y}}_i + x_i z_i + y_i, \tilde{\dot{z}}_i - x_i y_i)$, $N_i = [(y_i - x_i, x_i, -z_i), (1, 1, 1)]$ and $\Gamma = [(a, b, c), (\alpha_x, \alpha_y, \alpha_z)]$. The linear regression consists in calculating $\Gamma^\star = [(a^\star, b^\star, c^\star), (\alpha_x^\star, \alpha_y^\star, \alpha_z^\star)]$ which minimizes

$$\varepsilon = \sum_{i=1}^{n} (M_i - N_i \cdot \Gamma^\star)^2 \tag{10}$$

Hence, it translates into the condition

$$\partial\varepsilon / \partial\Gamma^\star = -\sum_{i=1}^{n} 2N_i^T \cdot (M_i - N_i \cdot \Gamma^\star) = \vec{0} \tag{11}$$

Further mathematical derivations lead to:

$$\Gamma^\star = (N^T \cdot N)^{-1} \cdot (N^T \cdot M) \tag{12}$$

where M and N are the matrices containing the considered data points, i.e. $M = \{M_1, M_2, ..., M_n\}$ and $N = \{N_1, N_2, ..., N_n\}$. Therefore, the coefficients $(a^\star, b^\star, c^\star, \alpha_x^\star, \alpha_y^\star, \alpha_z^\star)$ better fitting the RC-predicted trajectories can be compared to the ones that were used for the numerical integration of the Lorenz system, namely $(a, b, c, 0, 0, 0)$. Such comparison provides the error ϵ_{pred} between the numerical integration of 1 (what we call the *exact* solution) and the prediction made by the AIS. The error is finally expressed as the euclidean distance between the two set of coefficients, i.e.:

$$\epsilon_{\text{pred}} = \sqrt{(a - a^\star)^2 + (b - b^\star)^2 + (c - c^\star)^2 + \alpha_x^{\star 2} + \alpha_y^{\star 2} + \alpha_z^{\star 2}} \tag{13}$$

Note also that another way to measure the prediction error can be used, where the distance between the exact and the predicted trajectory are computed. However, for chaotic time series this is not a useful quantity, since any small error get amplified. Therefore, even though the systems are physically close to each other, their time series can be very disparate. For this reason, we better quantify the prediction error through Relation 13. Let us stress also that the prediction error is computed based on the coefficients $(a^\star, b^\star, c^\star, \alpha_x^\star, \alpha_y^\star, \alpha_z^\star)$ *via* a linear regression using $3n_{\text{pred}}$ data points. Therefore, the prediction error may depend on the length of the prediction phase n_{pred}.

It is worth to highlight that the PoPe method requires the prior knowledge of the system governing equations. For this reason, using the PoPe method for evaluating the prediction accuracy of the model-free RC technique is not generally allowed. Nevertheless, the purpose of the present study is the parametric validation of the RC algorithm in reconstructing the Lorenz dynamics, whose equations are fully available. For this reason, the use of the PoPe method to evaluate the quality of the RC prediction is justified.

In this paper, we quantify the accuracy of the RC approach to predict Lorenz dynamics through a scan on the number of neurons D_r and the spectral radius ρ of the Erdős-Rènyi network. In addition, we explore the impact of the length of the prediction phase on the prediction error, Equation 13. As already stated, the RC approach is based on the fact that all the parameters of the system are initialised randomly, except those of the output layer. This implies that the training phase is reduced to a simple linear regression. But this also implies that the results may depend on the random initialisation of the parameters. To overcome this difficulty, for each point in our scan (D_r, ρ), we perform $N = 500$ realisations. Each of them will be different, since the initialisation of the network and of the input layer is random. For

each realisation s we calculate $\epsilon_{\text{pred}}^{(s)}$ and we use the ensemble averaged quantity, $\langle \epsilon_{\text{pred}} \rangle_{\text{rls}} = \frac{1}{N} \sum_s \epsilon_{\text{pred}}^{(s)}$, where the subscript rls stands for *realisation*.

The prediction error can be calculated in two different ways: (1) for time windows whose left endpoint is set to 0 and the right endpoint is upper-bounded by $t_{\text{pred}} = 25$ and increasing for each time window as $\kappa\delta$, with $\delta = 1.25$ and $\kappa \in [1, 2, ..., n]$ denoting the number of the time window; (2) for moving time windows of the same length and whose endpoints are increasing as $[\kappa\delta, (\kappa+1)\delta]$, with $\delta = 1.25$ and $\kappa \in [0, 1, 2, ..., n]$. In other words, in the first way the error is computed over time windows whose lengths are increasing, considering therefore the dynamics from $t_{\text{pred}} = 0$ up to $\kappa\delta$; and in the second way, the prediction error ϵ_{pred} is calculated over moving time windows of fixed length $\delta = 1.25$. In the following, we use the first method to compute the errors. We call this method increasing-time-window procedure. In this way the dynamics of the entire considered prediction phase is taken into account in calculating the deviation from the actual Lorenz trajectories. Nonetheless, it could be useful to compare these results with the moving average error, i.e. the second procedure described here above, as it is done for instance in section 3.3. The choice of the time window length of $\delta = 1.25$ yields a sufficient number of points to calculate the linear regression and the subsequent predicted coefficients $(a^\star, b^\star, c^\star, \alpha_x^\star, \alpha_y^\star, \alpha_z^\star)$ in each time window of the moving error procedure. However, if not otherwise noted, the standard procedure for computing the errors is the increasing-time-window procedure.

Finally, it is also instructive to quantify the training error ϵ_{train}, which is computed using the same Euclidean distance as in equation 13, but applied only to the training phase.

3. Results

(b)

Figure 2: A scan of the error 13, in logarithmic scale, over the input parameters D_r and ρ of the reservoir network is displayed for the training and the prediction phase, in panel (a) and (b) respectively. The errors are averaged over the whole set of realisations ($N = 500$) for each couple (D_r, ρ). Highlighted with red markers are the configuration employed in Ref. [18].

The training and prediction errors averaged over all the realisations are plotted in figures 2(a) and (b), respectively, fixing the length of the

prediction phase to $t_{\text{pred}} = 25$, i.e. $n_{\text{pred}} = 1250$. The isocontours of the averaged training and prediction errors are represented in logarithmic scale as a function of the number of neurons D_r and the spectral radius ρ of the reservoir network. The error in the prediction phase is calculated for the entire prediction time, i.e. $t_{\text{pred}} = [0, 25]$. It is observed that the training error is minimum in a rather narrow region in the (D_r, ρ) plane. Moreover, mild variations of (D_r, ρ) may lead to an abrupt increase of the training error. This means that there is a strong restriction in the allowed values of D_r and ρ. It is particularly important to realise that increasing the number of neurons does not necessarily reduce the averaged training error. Moreover, we can observe that the averaged prediction error is significant, increased by several orders of magnitude with respect to the training error. The values selected in [18] were $D_r = 300$ and $\rho \in \{1.2, 1.45\}$, which fall into the beneficial narrow region where the training error is minimal, as the markers in figure 2(a) show. Yet, it is observed that the averaged prediction error is far from being negligible, considering the entire prediction phase $t_{\text{pred}} = 25$. Within the set of scanned parameters in figure 2, a region where $\langle \epsilon_{\text{pred}} \rangle_{\text{rls}}$ is minimized is noted for $D_r > 400$ and $\rho > 1.5$. In the remainder of the paper, such a region, and the configuration included therein, will be called *favorable*. It is stressed again that, although the ESP originally required the spectral radius to be $\rho < 1$, the range of values of the spectral radius leading to optimal forecasting is obtained for $\rho > 1$, consistently with the results of Ref. [51]. Furthermore, the region where $D_r > 300$ and $\rho < 1.2$ presents the largest measured averaged errors of the analyzed range of parameters in the prediction phase. For this reason, it will be termed as the *unfavorable* region of the (D_r, ρ) plane. It must be noted that, although in a different framework from the present parametric validation study, very similar dependencies of the errors on the spectral radius have also been reported in a recent publication [45].

3.1 Large Statistical Variation of the Errors due to Random Initialisation

(b)

Figure 3: The standard deviation σ of the logarithm in base 10 of the errors, for both training (a) and prediction (b) phases, within the set of $N = 500$ realisations is displayed for the same scans reported in figure 2. The large variation within the whole set of realisations can be thus appreciated.

It is also observed that the averaged errors in both training and

prediction phases for each couple (D_r, ρ) may be accompanied by a large variation within the set of performed realisations. In figure 3, the standard deviation σ of the averaged errors within the set of $N = 500$ realisations is plotted as a function of the number of neurons D_r and the spectral radius ρ for the training and the prediction phases in (a) and (b), respectively. The set of realisations is the same displayed in figure 2, and the errors shown in panel (b) for the prediction phase are calculated for $t_{\text{pred}} = [0, 25]$, consistently with figure 2. The large dispersion that can be appreciated by measuring σ demonstrates the strong impact of the randomness on the prediction of Lorenz chaotic trajectories by the RC technique. The plot in figure 3(a) provides additional evidence that the possible efficient configurations of this AIS based on the RC approach for the prediction of chaotic trajectories are limited to a narrow region in the (D_r, ρ) plane. Furthermore, similarly to what has been observed in figure 2, the difference of the standard deviation for the set of realisations between the training phase and the prediction phase is significant, especially in the *favorable* region of the scan. Indeed, this proves the more frequent occurrence of large error events in the prediction phase, which are then responsible also for the increase of the averaged prediction error within the set of the realisations with respect to the training phase.

Figure 4: The dynamics of the three components of the Lorenz system, predicted by the RC, is shown for the two configurations studied in Ref. [18], i.e. $(D_r, \rho) = (300, 1.2)$ (left column) and $(D_r, \rho) = (300, 1.45)$ (right column). The black dotted curves are the actual chaotic trajectories obtained through numerical integration of the Lorenz equations with the RK4 method (labelled as Ground truth in the legend). Red and blue curves show respectively the best and a non-favorable solution obtained through RC technique over the whole set of $N = 500$ realisations.

Insightful information on the randomness of the RC prediction can be found by further analysing the two specific study-cases described in Ref. [18]. From the statistically-relevant averaged results illustrated in figure 2, one can infer that the case with $(D_r, \rho) = (300, 1.45)$ performs better than the case with $(D_r, \rho) = (300, 1.2)$ (respectively, red diamond and red star in figure 2), as the $\langle \epsilon_{\text{pred}} \rangle_{\text{rls}}$ is minor for the former than for the latter case. This result might not be in full agreement with what is reported in Ref. [18], where the case with $(D_r, \rho) = (300, 1.45)$ is shown to be less performing to predict chaotic trajectories, with respect to the case with $(D_r, \rho) = (300, 1.2)$. This might be due to the random initialization of the input layer and the reservoir array. Indeed, for one single realization, the case with $(D_r, \rho) = (300, 1.2)$ can perform

better than that with $(D_r, \rho) = (300, 1.45)$, consistently with Ref. [18]. This is illustrated in figure 4, where the time traces of x, y and z are plotted for $D_r = 300$ and the two values $\rho = 1.2$ (left column panels) and $\rho = 1.45$ (right column panels). In each panel, the realisation with minimum error is plotted by a red curve. Another randomly chosen realization is plotted by the blue curve. As a reference, the exact solution is given by the dashed black curve. It is observed that depending on the realization, the case $\rho = 1.2$ can perform better or worse than the case $\rho = 1.45$.

3.2 Statistical Analysis on the Error Distribution

Given the large statistical variation exhibited by the measured errors, we analyse the relevance of using the mean value as an indicator of the error for each single (D_r, ρ) configuration within the set of realisations. The histograms of $log_{10}(\epsilon_{\text{pred}})$ for four couples (D_r, ρ) configurations

(b)

(d)

Figure 5: The error in logarithmic scale in the prediction phase, calculated at $t_{\text{pred}} = 25$, of an incremented set of $N = 5000$ realisations is shown in histogram plots for four different RC configurations. The distribution is binned in $n_{\text{bins}} = 100$ samples. The black dashed curves represent the unimodal normal PDF, whose mean value and standard deviation are the mean value and the standard deviation of the error distribution functions for the corresponding configuration.

are shown in figure 5. The same analysis on the error in the training phase $log_{10}(\epsilon_{\text{train}})$ revealed analogous results. These four different cases are chosen to represent: (a) a *favorable* case, (b) and (c) the same cases studied by Pathak et al. in Ref. [18] and (d) a case in the *unfavorable* region of the (D_r, ρ) plane. In this particular analysis, the number of realisations has been increased up to $N = 5000$ to improve the statistical relevance of the data set, and the histograms are captured in $n_{\text{bins}} = 100$ bins evenly spaced. As can be seen, the histograms are shifted to large error values going from panel (a) to panel (d). In the four panels, the Gaussian probability density function (PDF), whose mean and standard deviation are respectively equal to the statistical mean and standard deviation of the corresponding set of realisations, is over-plotted with black dashed lines. In this way, it is straightforward to compare the histograms and the Gaussian PDFs. Whereas for panels (b) and (c) the distribution of the error over the set of realisations follows a Gaussian, for panels (a) and (d) the his-

tograms exhibit a departure from the normal PDF. In particular for the configuration $(D_r, \rho) = (550, 0.9)$, an inset with the rightmost tail of the distribution function is displayed in a log-log plot. In this way, it is possible to appreciate the deviation for the large-error events of such configuration from the tail of the Gaussian distribution. This deviation is present in all the analyzed configurations of the *unfavorable* region of the (D_r, ρ) plane. Therefore, it is demonstrated the accuracy of using the statistical mean, i.e. the first moment of the distribution function, to quantify the error in both training and prediction phases for a significant range of analyzed configurations. Yet, the *unfavorable* region of the (D_r, ρ) plane presents a non-negligible deviation from the Gaussian distribution, especially concerning the large error events.

Additionally, it could be observed that the slight deviation of the histograms from the normal distribution function, especially in panel (d), can be explained by studying the third and fourth standardized moments of the PDF, i.e. the skewness μ_3 and the kurtosis μ_4 respectively. The skewness measures the loss of asymmetry of a distribution function. For a symmetric PDF the skewness is null ($\mu_3 = 0$), hence a positive (negative) value indicates that the data-set is skewed to the right (left) with respect to the principal mode - the mean value for a unimodal Gaussian PDF. On the other hand, the kurtosis property measures the flatness of the PDF. Generally, the kurtosis is compared to the kurtosis of a Gaussian distribution function, which is equal to 3 ($\mu_4 = 3$). Therefore, it is commonly used the excess kurtosis $\tilde{\mu}_4$, which is the kurtosis re-scaled to 3 ($\tilde{\mu}_4 = \mu_4 - 3$). A large excess kurtosis thus indicates that the distribution has strongly populated tails. Therefore, by inspecting the loss of asymmetry (skewness) and strength of the tails (excess kurtosis) together, it is possible to evaluate the behaviour of the measured PDFs. The measured distribution function in figure 5(a) and (d) reveal a positive skewness, denoting a major broadening towards large error values, as can be also inferred from the plots. Whereas for the configuration $(D_r, \rho) = (470, 2.2)$ the skewness is mildly above zero, the RC with $(D_r, \rho) = (550, 0.9)$ shows a four-time larger value of the skewness, which evidences the more frequent occurring of large error events. For this latter configuration the re-scaled fourth moment, the excess kurtosis, is positive and large indicating that the distribution of the data-set is strongly peaked around the mean value. In conclusion, the statistical properties of the cases in panels (a), (b) and (c) indicate that the distribution functions could be well approximated with a Gaussian around the corresponding mean values. On the other hand, for the unfavorable case in panel (d), the statistical mean value is strongly affected by a more frequent occurring of large-error events.

3.3 Dependence of the Error on the Prediction Phase Length

To shed some light on the validity of the RC to reconstruct the Lorenz dynamics, we additionally explore the dependence of the prediction error on the maximum length of the prediction phase. As illustrated in Ref. [18], the AIS based on the RC approach correctly predicts the short-term trajectories, while significantly deviating from the actual Lorenz trajectories in the long-term phase. The considered length of the prediction phase has thereby a strong impact on the computed error. Therefore, we analyse the dependence of the averaged prediction error ϵ_{pred} on the length of the prediction phase in the two different procedures that have been described at the end of section 2.2. The

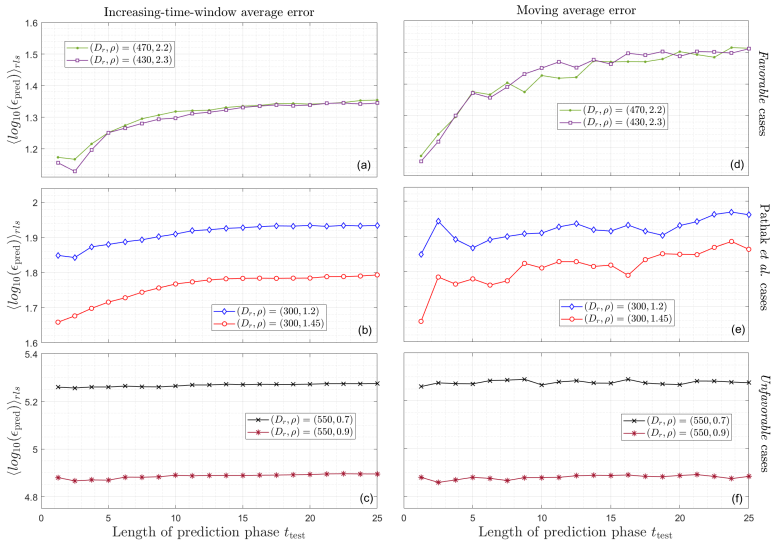

Figure 6: The error in the prediction phase is displayed as a function of the length of the prediction phase for different configurations. The prediction error is calculated at increasing width of the time windows in panels (a)-(c), whereas with moving time window of fixed period $\delta = 1.25$ in panels (d)-(f). The configurations reported are picked from the whole scanned cases, with the purpose of representing: (a) and (d) favorable cases, (b) and (e) the same cases reported in Ref. [18] and eventually (c) and (f) cases from the unfavorable region of the (D_r, ρ) plane explored in figure 2.

results are illustrated in figure 6. Hence, in panels (a)-(c) the prediction error is calculated for time windows whose period is $[0, \kappa\delta]$, with $\delta = 1.25$ and $\kappa = [1, 2, ..., n]$, up to $t_{\text{pred}} = 25$. Indeed, panels (d)-(f) represent the prediction error calculated in the second fashion, for

which the period of the time windows is $[\kappa\delta, (\kappa+1)\delta]$, with $\delta = 1.25$ and $\kappa = [0, 1, 2, ..., n]$. The two procedures are labelled in the figure as increasing-time-window average error and moving average error, respectively.

As it has already been noted, even for the favorable cases in panel 6(a) in previous sections, the $\langle log_{10}(\epsilon_{\text{pred}})\rangle_{\text{rls}}$ indicator is very large compared to the error in the corresponding training phase. It is also observed that there is a strong dependence on the length of the prediction phase in the short-term prediction. Focusing on the increasing-time-window average procedure, after the prompt increase in the short-term prediction, the error remains constant (after an elbow) in the long-term prediction, consistently with Ref. [18]. Such a behaviour is actually present for the *favorable* configurations also when the error is calculated with the moving average procedure, as panels (d) and (e) show.

Furthermore, smoother curves in the time evolution of the calculated error appear when the increasing-time-window average procedure is employed. This is to be understood as, in this latter procedure, the calculation has a kind of 'memory' from previous time windows, since the analyzed time period increases by including also the dynamics of the previous time windows. On the other hand, there is no such 'memory effect' in the moving average procedure, as the solution coefficients are calculated for each time window independently from the previous ones.

Another striking observation comes from the value of the error at the end of the first time period of the prediction phase ($t_{\text{pred}} \in [0, 1.25]$). Indeed, the average prediction error $\langle \epsilon_{\text{pred}}\rangle_{\text{rls}}$ already presents value much larger than the error evaluated for training phase for the same configuration. The difference between these two values can also be inferred from the plots in figure 2. Especially for the *favorable* configurations, the difference could be also more than 4 orders of magnitude. Therefore, the RC approach could show a strong disparity from the expected dynamics also for $t_{\text{pred}} < 2.5$, i.e. for only $3 \sim 4$ oscillations of the Lorenz system. This could essentially be due to two reasons: either the reservoir layers are failed to be trained and the error is thereby propagated in the prediction phase inevitably, or the RC technique predicts strongly different dynamics in the short-term phase. The former condition, for which the training phase already produces a large error and thus the reservoir layers are not well-trained, is less frequent, as the lower mean values of the error in figure 2(a) and the standard deviations in figure 3(a) illustrate. Nevertheless, if a large error event occur in the training phase, this inevitably produces a large error event also in the prediction phase, since the RC is ill-trained and so unable to recover the correct Lorenz dynamics. The latter condition, for which the RC network is well-trained but fails in predicting the short-term

dynamics, is indeed the most frequent. As a result, the average error in the first time window of the prediction phase is much higher than the one measured in the training phase.

3.4 Dependence of the Error on the Train Phase Length

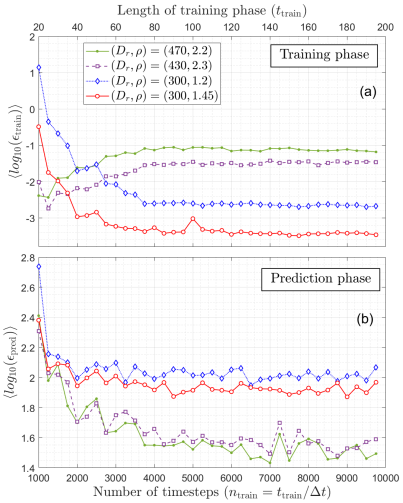

Figure 7: The errors in the training (a) and in the prediction phase (b) are shown as a function of the length of the training phase for four different input configurations of the reservoir network. The configurations are the same analyzed also in figures 5 and 6. The abscissas is expressed both in length of the training phase in normalized time t_{train} and in number of time steps n_{train} in the training phase.

It is now considered the effect of the length of the training phase on the error in predicting the Lorenz trajectories by means of the RC technique. In figure 7, the dependence of both $\langle \epsilon_{\text{train}} \rangle_{\text{rls}}$ and $\langle \epsilon_{\text{pred}} \rangle_{\text{rls}}$ on the length of the training phase is displayed for various chosen reservoir configurations. The plots illustrate the result for $n_{\text{train}} > 1000$, as for smaller number of time steps the RC definitely fails in being trained and subsequently in predicting the chaotic dynamic of the system. The parameters of the various configurations are the same analyzed in panels (a) and (b) (and also (d) and (e)) of figure 6. The error in the prediction phase is considered for the entire prediction phase $t_{\text{pred}} = [0, 25]$. A striking observation about the training phase for the configurations with $(D_r, \rho) = (430, 2.3)$ and $(D_r, \rho) = (470, 2.2)$ is the non-monotonic behaviour of the function. Indeed, for a narrow range of the training phase length, the $\langle \epsilon_{\text{train}} \rangle_{\text{rls}}$ curves presents a min-

imum. Such a behaviour has been found also in other configurations within the *favorable* region of the (D_r, ρ) plane. Instead, for the configuration setups with $(D_r, \rho) = (300, 1.2)$ and $(D_r, \rho) = (300, 1.45)$, the same analyzed by Pathak et al. in [18], the error in the training phase is basically decreasing with the increasing number of considered time steps for training the system until a plateau is reached. This plateau could be expected due to the quasi-periodicity of the chaotic oscillations of the Lorenz system. Regarding the prediction phase, the behaviours of all the detailed configurations resemble. After a sharp decrease for $n_{\text{train}} > 1000$, the curves mildly continue to decrease until $n_{\text{train}} = 4000 \sim 5000$, where they become averagely constant. Therefore, the increase of the number of time steps in the training phase beyond $n_{\text{train}} = 10000$ leads to enhanced elapse time and complexity of the system, without improving the prediction of the chaotic trajectories significantly. Consistent results have been recovered also in Ref. [25], although a modified version of the Lorenz 96 chaotic model [52] was employed as test-bed system instead of the original Lorenz system here studied. Furthermore, it is to be noted the jagged behaviour of $\langle \epsilon_{\text{pred}} \rangle_{\text{rls}}$ in function of the number of time steps n_{train}. Consecutive points of this particular scan could lead to very different results. Thus, increasing n_{train} does not automatically lead to an improvement of the results, but it could be indeed detrimental in average.

3.5 Studies on the Time Evolution of the Predicted Lorenz Coefficients

Additional analyses have been carried out on the evolution in time of the coefficients $(\tilde{a}, \tilde{b}, \tilde{c}, \tilde{\alpha}_x, \tilde{\alpha}_y, \tilde{\alpha}_z)$ which better fit the prediction obtained by means of the RC technique. In the Lorenz system, such coefficients are constant throughout the whole time trajectory. This assumption, together with the constraint of having $(\tilde{\alpha}_x, \tilde{\alpha}_y, \tilde{\alpha}_z) = (0, 0, 0)$, is crucial for the validity of the Lorenz system. Therefore, in order to ensure the reliability of the predicted dynamics of the chaotic system, the fitted coefficients have been plotted for two different RC configurations, which minimizes the prediction error at the end of the prediction phase $\epsilon_{\text{pred}}(t_{\text{pred}} = [0, 25])$, in figure 8. These two configurations are the same analyzed by Pathak et al. in [18], i.e. $(D_r, \rho) = (300, 1.2)$ and $(D_r, \rho) = (300, 1.45)$. As it has already been explained, $(\tilde{a}, \tilde{b}, \tilde{c}, \tilde{\alpha}_x, \tilde{\alpha}_y, \tilde{\alpha}_z)$ are the coefficients which better fit the Lorenz equations for the RC predicted dynamics in the prediction phase. In this particular analysis, the coefficients are calculated for consecutive moving time windows of the prediction phase with a period of $\delta = 1.25$. The same procedure has already been described in section 3.3 when the calculation of moving error has been applied to produce the figures 6(d)-(f). We repeat that this choice of $\delta = 1.25$ enables to have sufficient elements in each time window for the calculation of such predicted coefficients by linear regression tech-

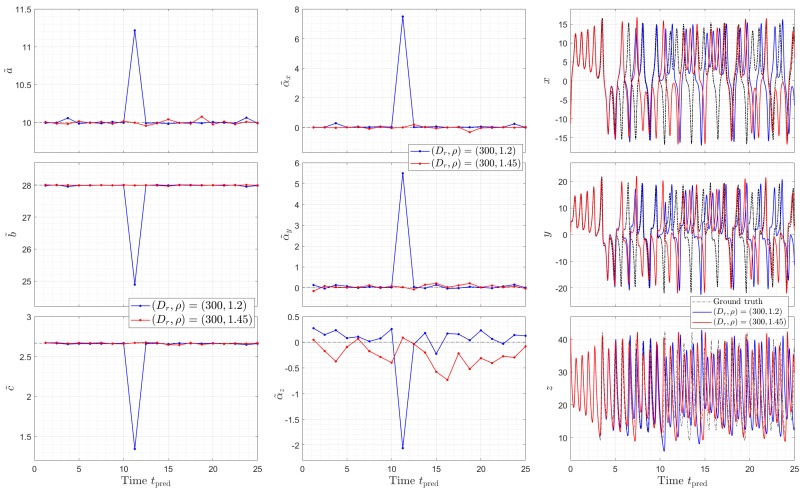

Figure 8: The coefficients $(\tilde{a}, \tilde{b}, \tilde{c}, \tilde{\alpha}_x, \tilde{\alpha}_y, \tilde{\alpha}_z)$ corresponding to the best solution of the RC application for the configurations also explored in Ref. [18] are shown in the first two columns. The horizontal dashed lines represent the values of the relative coefficients in the actual Lorenz system. In the third column, the predicted dynamics of these two configurations are compared to the actual Lorenz dynamics (labeled 'Ground truth').

niques and to grasp, thereby, their time dependence in the prediction phase. Thus, in the first two columns of figure 8, the coefficients of the solution are plotted against the time of the prediction phase t_{pred}, while in the rightmost column the actual Lorenz dynamics are compared to each component of the AIS predicted systems. However, it is observed that in this optimized case the setup with $(D_r, \rho) = (300, 1.45)$ performs better than $(D_r, \rho) = (300, 1.2)$, consistently with what is displayed in figures 6(b) and 6(b) and (e). Indeed, the three predicted coefficients for $(D_r, \rho) = (300, 1.45)$ oscillates around the actual initial Lorenz coefficients (a, b, c) (highlighted in black horizontal dashed lines in the panels of the first column) throughout the whole prediction phase. The deviations from the exact coefficients of the studied Lorenz system appear almost negligible ($\pm 5\%$). Regarding indeed the case with $(D_r, \rho) = (300, 1.2)$, a significant deviation is noted in the time window around $t_{\text{pred}} = 11$. Nevertheless, this deviation does not impair the whole prediction phase, as the actual coefficients are promptly recovered after the deviation occurring. If the third column of figure 8 is inspected for the configuration $(D_r, \rho) = (300, 1.2)$ (blue curves), it is

possible to see that around $t_{\text{pred}} = 11$ there is a pattern loss, which determines the high spike in the calculation of the predicted coefficients. Yet, such an instantaneous deviation from the exact Lorenz trajectory does not prejudice the overall dynamical properties of the predicted system.

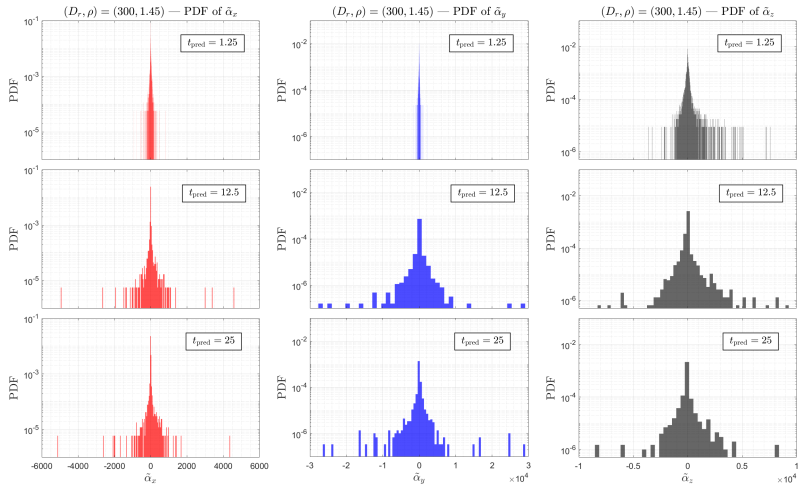

Figure 9: The probability distribution function of the coefficients $(\tilde{\alpha}_x, \tilde{\alpha}_y, \tilde{\alpha}_z)$ for the RC configuration with $(D_r, \rho) = (300, 1.45)$ is shown in histograms for three different time windows of an incremented set of $N = 5000$ realisations. The distribution is binned in $n_{\text{bins}} = 500$ samples. The standard deviation of the PDFs is shown in figure 10 for the entire prediction phase.

However, as it has already been noted, the solution plotted in figure 8 are the predictions that minimizes $\epsilon_{\text{pred}}(t_{\text{pred}} = 25)$ within a statistically relevant set of $N = 5000$ realisations for each RC configuration. Therefore, the reader must be cautious on evaluating it, since in the same sample of realisations cases with large measured errors were frequently observed, as the average prediction error in figure 2 and the PDF of the prediction error in figure 5 demonstrate. The optimized cases reported in this section, indeed, are located in the tail of the corresponding PDF, and therefore the probability of their occurrence is quite low.

A more detailed analysis on the α coefficients is performed and the results are illustrated in figure 9. Here, the PDF of the α coefficients $(\tilde{\alpha}_x, \tilde{\alpha}_y, \tilde{\alpha}_z)$ for the configuration with $(D_r, \rho) = (300, 1.45)$ is plotted for three different time windows in the prediction phase,

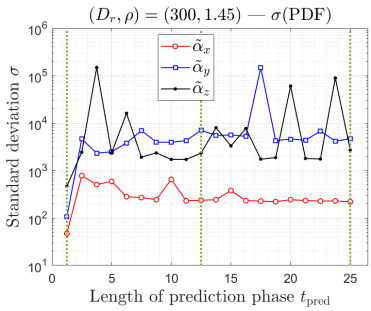

Figure 10: The evolution in time of the standard deviation σ of the α coefficient PDFs diplayed in figure 9 is shown in logarithmic scale for the prediction phase up to $t_{\text{pred}} = 25$. The green dashed vertical lines represent the time windows displayed in figure 9.

i.e. $t_{\text{pred}} = ([0, 1.25], [11, 12.5], [23.5, 25])$. The PDFs are binned in $n_{bins} = 500$. The same analysis has been carried out also for other configurations in the favorable region of the (D_r, ρ) plane, yielding similar results. What can be inferred here is that the distribution functions of the $\tilde{\alpha}$ coefficients are centered around zero, strongly broadening with increasing time in the prediction phase. Measuring the width of the PDFs, one can notice that it is significantly enhanced going from $t_{\text{pred}} = 1.25$ to $t_{\text{pred}} = 12.5$ for all the three coefficients, whereas no such a difference is measured from $t_{\text{pred}} = 12.5$ to $t_{\text{pred}} = 25$. To corroborate this result, the standard deviation σ of the $\tilde{\alpha}$ coefficients within the set of $N = 5000$ realisations is plotted in logarithmic scale against t_{pred}. The green vertical dashed lines represent the upper boundary of the analyzed time windows. It is observed that already after the first time window, σ increases of more than two orders of magnitude. Then, it reaches a quasi-stationary phase with jagged behaviours. Therefore, as already proved by figure 9, the frequency of of large value events occurring is higher already after the first time window, but it does not increase additionally in the rest of the prediction phase. However, the standard deviation presents some spikes (especially for $\tilde{\alpha}_y$ and $\tilde{\alpha}_x$), which are symptomatic of a more frequent occurring of the large value events. These large value events, indeed, are correlated with the large error events in the prediction phase, as the relation 13 proves. To conclude, the analysis here performed shows that in the predicted Lorenz system the $\tilde{\alpha}$ coefficients, measured by the PoPe method, are centered around zero. However, quite frequently they differ from the null value, indicating that the actual Lorenz dynamics is definitely impaired. Moreover, the solutions displayed in figure 8 are only representative of the tail of the probability distribution, as the frequency of

their occurrence is very low. Hence, even for favorable configurations, the accuracy of such a RC approach must be evaluated cautiously.

4. Conclusions

In this study, we have analyzed the validity of the reservoir computing ML technique applied to reconstruct the dynamical coefficients of the predicted trajectories of the Lorenz system, a well-known chaotic system, by the Projection on Proper elements method. Such a method is briefly explained in section 2 (more details can be found in Refs. [37, 38]). The validity of the RC technique is statistically measured by running $N = 500$ realisations for a large set of configuration parameters. Thus, it has been shown that a good accuracy is achieved only for a small range of (D_r, ρ) configurations of the reservoir. Yet, the error in the prediction phase is significantly increased with respect to the error in the training phase, with a significantly large variation of the error, as figure 3 shows. This implies that quite frequently the RC approach produces predictions strongly affected by large errors of the Lorenz system. It is shown that this is essentially due to the intrinsic randomness of the RC technique, which inevitably leads to have large error events frequently also for the *favorable* configurations of the network. Additional scans on the relevant parameters of the ML technique such as the length of both training and prediction phases are carried out, showing that the range of validity of this RC approach is even further narrowed. Indeed, a minimum number of time steps in the training phase is required to achieve acceptable results, as figure 7 illustrates, but only for predicting the short-term dynamics of the Lorenz system. The deviation from the exact Lorenz dynamics in the long-term prediction phase is shown to become very large. Such results are also analyzed deeper by means of detailed studies on the time evolution of the Lorenz coefficients computed by the AIS. It is shown that the large variation of the Lorenz coefficients computed from the predicted solution can be significant. Therefore, the solution predicted by the AIS based on the RC approach is not necessarily representative of a Lorenz system.

This paper, in the context of the RC approach to reconstruction of complex dynamical series, is helpful to establish the range of validity of this AIS technique. It also suggests that further developments of the RC paradigm, as recently proposed in, e.g. , Refs. [53, 54], are required in order to robustly achieve a good accuracy in predicting chaotic time series.

Acknowledgements

This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014–2018 and 2019–2020 under Grant agreement No 633053. The views and opinions express herein do not necessarily reflect those of the European Commission.

Conflict of interest

The authors have no conflicts to disclose.

Data Availability Statement

The data and the algorithms that support the findings of this study are available from the corresponding author upon request.

References

[1] Fasoli, A., Brunner, S., Cooper, W. A., Graves, J. P., Ricci, P., Sauter, O., and Villard, L. Computational challenges in magnetic-confinement fusion physics. *Nature Physics*, 12(5):411–423, 2016.

[2] Lynch, P. The origins of computer weather prediction and climate modeling. *Journal of Computational Physics*, 227(7):3431–3444, 2008.

[3] Dada, J. O. and Mendes, P. Multi-scale modelling and simulation in systems biology. *Integrative Biology*, 3(2):86–96, 2011.

[4] Xiao, Y., Xiao, J., Liu, J., and Wang, S. A multiscale modeling approach incorporating ARIMA and ANNs for financial market volatility forecasting. *Journal of Systems Science and Complexity*, 27(1):225–236, 2014.

[5] Jaeger, H. The "echo state" approach to analysing and training recurrent neural networks. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.

[6] Maass, W., Natschläger, T., and Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

[7] Xue, Y., Yang, L., and Haykin, S. Decoupled echo state networks with lateral inhibition. *Neural Networks*, 20(3):365–376, 2007.

[8] Ma, Q.-L., Zheng, Q.-L., Peng, H., Zhong, T.-W., and Qin, J.-W. Multi-step-prediction of chaotic time series based on co-evolutionary recurrent neural network. *Chinese Physics B*, 17(2):536, 2008.

[9] Jaeger, H. and Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

[10] Schrauwen, B., Verstraeten, D., and Van Campenhout, J. An overview of reservoir computing: theory, applications and implementations. In *Proceedings of the 15th european symposium on artificial neural networks*, pages 471–482, 2007.

[11] Lukoševičius, M. and Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

[12] Lukoševičius, M. A practical guide to applying echo state networks. In *Neural networks: Tricks of the trade*, pages 659–686. Springer, 2012.

[13] Gauthier, D. J. Reservoir computing: Harnessing a universal dynamical system. *Physical Review Letters*, 120(2018):024102, 2018.

[14] Verstraeten, D., Schrauwen, B., d'Haene, M., and Stroobandt, D. An experimental unification of reservoir computing methods. *Neural Networks*, 20(3):391–403, 2007.

[15] Xi, J., Shi, Z., and Han, M. Analyzing the state space property of echo state networks for chaotic system prediction. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 3, pages 1412–1417. IEEE, 2005.

[16] Song, Y., Li, Y., Wang, Q., and Li, C. Multi-steps prediction of chaotic time series based on echo state network. In *2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, pages 669–672. IEEE, 2010.

[17] Bianchi, F. M., Maiorino, E., Kampffmeyer, M. C., Rizzi, A., and Jenssen, R. An overview and comparative analysis of recurrent neural networks for short term load forecasting. *arXiv preprint arXiv:1705.04378*, 2017.

[18] Pathak, J., Lu, Z., Hunt, B. R., Girvan, M., and Ott, E. Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(12):121102, 2017.

[19] Lu, Z., Pathak, J., Hunt, B., Girvan, M., Brockett, R., and Ott, E. Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(4):041102, 2017.

[20] McDermott, P. L. and Wikle, C. K. An ensemble quadratic echo state network for non-linear spatio-temporal forecasting. *Stat*, 6(1):315–330, 2017.

[21] Lu, Z., Hunt, B. R., and Ott, E. Attractor reconstruction by machine learning. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(6):061104, 2018.

[22] Pathak, J., Hunt, B., Girvan, M., Lu, Z., and Ott, E. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Physical Review Letters*, 120(2):024102, 2018.

[23] Zimmermann, R. S. and Parlitz, U. Observing spatio-temporal dynamics of excitable media using reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(4):043118, 2018.

[24] Tanaka, G., Yamane, T., Héroux, J. B., Nakane, R., Kanazawa, N., Takeda, S., Numata, H., Nakano, D., and Hirose, A. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123, 2019.

[25] Chattopadhyay, A., Hassanzadeh, P., and Subramanian, D. Data-driven predictions of a multiscale lorenz 96 chaotic system using machine-learning methods: Reservoir computing, artificial neural network, and long short-term memory network. *Nonlinear Processes in Geophysics*, 27(3):373–389, 2020.

[26] Bianchi, F. M., Scardapane, S., Løkse, S., and Jenssen, R. Reservoir computing approaches for representation and classification of multivariate time series. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[27] Verstraeten, D., Dambre, J., Dutoit, X., and Schrauwen, B. Memory versus non-linearity in reservoirs. In *The 2010 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2010.

[28] Marzen, S. Difference between memory and prediction in linear recurrent networks. *Physical Review E*, 96(3):032308, 2017.

[29] Massar, M. and Massar, S. Mean-field theory of echo state networks. *Physical Review E*, 87(4):042809, 2013.

[30] Goudarzi, A., Marzen, S., Banda, P., Feldman, G., Teuscher, C., and Stefanovic, D. Memory and information processing in recurrent neural networks. *arXiv preprint arXiv:1604.06929*, 2016.

[31] Grigoryeva, L. and Ortega, J.-P. Echo state networks are universal. *Neural Networks*, 108:495–508, 2018.

[32] Hart, A., Hook, J., and Dawes, J. Embedding and approximation theorems for echo state networks. *Neural Networks*, 128:234–247, 2020.

[33] Tino, P. Dynamical systems as temporal feature spaces. *Journal of Machine Learning Research*, 21(44):1–42, 2020.

[34] Hart, A. G., Hook, J. L., and Dawes, J. H. P. Echo state networks trained by Tikhonov least squares are l2 (μ) approximators of ergodic dynamical systems. *Physica D: Nonlinear Phenomena*, page 132882, 2021.

[35] Lorenz, E. N. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20(2):130–141, 1963.

[36] Cartier-Michaud, T. *Code verification and model reduction: Application to transport in turbulent plasmas*. PhD thesis, Aix-Marseille University, 2015.

[37] Cartier-Michaud, T., Ghendrih, Ph., Sarazin, Y., Abiteboul, J., Bufferand, H., Dif-Pradalier, G., Garbet, X., Grandgirard, V., Latu, G., Norscini, C., et al. Projection on proper elements for code control: Verification, numerical convergence, and reduced models. Application to plasma turbulence simulations. *Physics of Plasmas*, 23(2):020702, 2016.

[38] Cartier-Michaud, T., Ghendrih, Ph., Dif-Pradalier, G., Garbet, X., Grandgirard, V., Latu, G., Sarazin, Y., Schwander, F., and Serre, E. Verification of turbulent simulations using PoPe: quantifying model precision and numerical error with data mining of simulation output. *Journal of Physics: Conf. Series*, 1125:012005, 2018.

[39] Erdős, P. and Rènyi, A. On random graphs I. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[40] Gilbert, E. N. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

[41] Yildiz, I. B., Jaeger, H., and Kiebel, S. J. Re-visiting the echo state property. *Neural Networks*, 35:1–9, 2012.

[42] Basterrech, S. Empirical analysis of the necessary and sufficient conditions of the echo state property. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 888–896. IEEE, 2017.

[43] Manjunath, G. and Jaeger, H. Echo state property linked to an input: Exploring a fundamental characteristic of recurrent neural networks. *Neural Computation*, 25(3):671–696, 2013.

[44] Ceni, A., Ashwin, P., Livi, L., and Postlethwaite, C. The echo index and multistability in input-driven recurrent neural networks. *Physica D: Nonlinear Phenomena*, 412:132609, 2020.

[45] Verzelli, P., Alippi, C., and Livi, L. Learn to synchronize, synchronize to learn. *arXiv preprint arXiv:2010.02860*, 2020.

[46] Haynes, N. D., Soriano, M. C., Rosin, D. P., Fischer, I., and Gauthier, D. J. Reservoir computing with a single time-delay autonomous boolean node. *Physical Review E*, 91(2):020801, 2015.

[47] Antonik, P., Haelterman, M., and Massar, S. Brain-inspired photonic signal processor for generating periodic patterns and emulating chaotic systems. *Physical Review Applied*, 7(5):054014, 2017.

[48] Larger, L., Baylón-Fuentes, A., Martinenghi, R., Udaltsov, V. S., Chembo, Y. K, and Jacquot, M. High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification. *Physical Review X*, 7(1):011015, 2017.

[49] Vlachas, P. R., Pathak, J., Hunt, B. R., Sapsis, T. P., Girvan, M., Ott, E., and Koumoutsakos, P. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 126:191–217, 2020.

[50] Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR, 2013.

[51] Jiang, J. and Lai, Y.-C. Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks: Role of network spectral radius. *Physical Review Research*, 1(3):033056, 2019.

[52] Thornes, T., Düben, P., and Palmer, T. On the use of scale-dependent precision in earth system modelling. *Quarterly Journal of the Royal Meteorological Society*, 143(703):897–908, 2017.

[53] Løkse, S., Bianchi, F. M., and Jenssen, R. Training echo state networks with regularization through dimensionality reduction. *Cognitive Computation*, 9(3):364–378, 2017.

[54] Yeo, K. Data-driven reconstruction of nonlinear dynamics from sparse observation. *Journal of Computational Physics*, 395:671–689, 2019.