Ensemble Reservoir Computing for Dynamical Systems: Prediction of Phase-Space Stable Region for Hadron Storage Rings

Maxime Casanova[1,2*], Barbara Dalena[1*†], Luca Bonaventura[2†] and Massimo Giovannozzi[3†]

[1*]DRF/Irfu/DACM, CEA Paris Saclay and Paris Saclay University, Gif-Sur-Yvette, 91191, France.
[2]Dipartimento di Matematica, Politecnico di Milano, Via Bonardi 9, Milano, 20132, Italy.
[3]Beams Department, CERN, Esplanade des Particules 1, Geneva, 1211, Geneva, Switzerland.

*Corresponding author(s). E-mail(s): casanovamaxime@outlook.com; barbara.dalena@cea.fr;
Contributing authors: luca.bonaventura@polimi.it; massimo.giovannozzi@cern.ch;
†These authors contributed equally to this work.

Abstract

We investigate the ability of an ensemble reservoir computing approach to predict the long-term behaviour of the phase-space region in which the motion of charged particles in hadron storage rings is bounded, the so-called dynamic aperture. Currently, the calculation of the phase-space stability region of hadron storage rings is performed through direct computer simulations, which are resource- and time-intensive processes. Echo State Networks (ESN) are a class of recurrent neural networks that are computationally effective, since they avoid backpropagation and require only cross-validation. Furthermore, they have been proven to be universal approximants of dynamical systems. In this paper, we present the performance reached by ESN based on an ensemble approach for the prediction of the phase-space stability region and compare it

with analytical scaling laws based on the stability-time estimate of the Nekhoroshev theorem for Hamiltonian systems. We observe that the proposed ESN approach is capable of effectively predicting the time evolution of the extent of the dynamic aperture, improving the predictions by analytical scaling laws, thus providing an efficient surrogate model.

Keywords: Non-linear beam dynamics, Echo State Network, Colliders and storage rings, Dynamical systems

1 Introduction

The advent of superconducting, high-energy hadron storage rings and colliders elevated non-linear beam dynamics to the forefront of accelerator design and operation. When studying phenomena in the field of single-particle beam dynamics, the concept of dynamic aperture (DA), that is, the extent of the phase space region where bounded motion occurs, has been a key observable to guide the design of several past (see, e.g. [1, 2, 3, 4, 5, 6]), present, e.g. the CERN Large Hadron Collider (LHC) [7], and future hadron machines (see e.g. [8, 9, 10, 11, 12, 13, 14, 15]).

DA prediction involves many challenging aspects, including understanding the mechanisms that determine its behaviour and addressing several computational problems. An important issue is the possibility of modelling the evolution of DA as a function of the number of turns, which has been studied since the end of the 90s [16, 17]. Indeed, determining how to describe and efficiently predict the value of the DA might solve some fundamental problems in accelerator physics, linked to performance optimisation of storage rings and colliders. The high computational cost of direct numerical simulations would be significantly reduced if a reliable model for the time evolution of the DA were available. In fact, the numerical simulations required to assess the performance of a circular accelerator cannot cover a time span comparable with operational intervals. For the LHC case, simulations up to 10^6 turns are at the limit of the CPU-time capabilities, although this represents only about 89 s of storage time, knowing that a typical fill time is of the order of several hours. Eventually, a model for the evolution of DA over time would also open the possibility of studying observables that are more directly related to machine performance, such as beam losses and lifetime [18] and luminosity evolution in colliders [19, 20].

A successful solution to this problem has been found by building models for DA scaling with time based on fundamental results of dynamical system theory, such as the Nekhoroshev theorem [21, 22, 23]. In fact, models with two or three parameters can be derived that can be fitted to numerical data that represent the evolution of the DA and used to predict the DA value for times beyond the current computational capabilities [24].

In the last decade, the use of neural networks has increased significantly in a large number of diverse research areas, and this observation has suggested their application to the prediction of the evolution of DA. For example, neural networks are used for speech recognition [25] or to forecast wind power [26]. Among neural network techniques, the most common architectures are feedforward [27], convolutional [28], and recurrent [29] neural networks. Feedforward neural networks are made up of neurons connected to other neurons, only. They provide only input-output relationships and can approximate very large classes of functions. On the other hand, recurrent neural networks are made up of neurons connected to themselves and other neurons. They preserve an internal state that is a non-linear transformation of the input signal and can therefore be considered as dynamical systems.

Echo State Networks (ESN) are one of the classes of recurrent neural networks that use the reservoir computing approach [30]. This approach has the main advantage of significantly reducing the computational time required by the training process, which is performed to find the optimal parameters (called weights) of a neural network. In fact, the peculiarity of the ESN is that training is performed, usually using linear regression [31], to calculate the weights used to project the reservoir state onto the output state. Therefore, no backpropagation is needed. Backpropagation [32] refers to the numerical procedure, usually based on the stochastic gradient method, used for the training of feedforward networks, which is responsible for a large share of its computational cost. ESN have also been proven to be universal approximants of dynamical systems [33]. Thus, ESN seem to be natural candidates for performing the prediction of DA for a large number of turns, and hence challenge the performance of the deterministic models developed so far.

This paper is organised as follows: In Section 2, we introduce the concept of DA and the approach used to provide numerical estimates of its value. Analytical scaling laws, based on the Nekhoroshev theorem and used to predict the time evolution of DA, are also presented. Section 3 introduces the continuous-time leaky ESN framework that is used for the prediction of DA. The Echo State Property (ESP), and a sufficient condition that can be applied in practice to satisfy it, are discussed in the Appendix A. Section 4 describes the ensemble procedure used in the cross-validation of the ESN and in the prediction of DA. The results are presented and discussed in Section 5, while conclusions are drawn in Section 6.

2 Dynamic Aperture

2.1 Generalities

We consider a Hamiltonian system in $\mathbb{R}^{2n}$, with a stable fixed point at the origin, whose dynamics is generated by a polynomial map $\mathcal{M}$, and such that the linear part of $\mathcal{M}$ is described by the direct product of rotations. Under these conditions, the DA of the system under consideration is the extent of the region of phase space in which bounded motion occurs.

Following [34] and restricting the analysis to the case of Hamiltonian systems in $\mathbb{R}^4$, which are relevant for accelerator physics, we consider the phase space volume of the initial conditions that are bounded after N iterations, namely

$$\int\int\int\int \chi(x_1, p_{x_1}, x_2, p_{x_2})\, dx_1\, dp_{x_1}\, dx_2\, dp_{x_2}\,, \tag{1}$$

where $\chi(x_1, p_{x_1}, x_2, p_{x_2})$ is the characteristic function defined as equal to one if the orbit starting at $(x_1, p_{x_1}, x_2, p_{x_2})$ is bounded and zero if it is not.

To exclude the disconnected part of the stability domain in the integral (1), we have to choose a suitable coordinate transformation. As linear motion is given by the direct product of constant rotations, the natural choice is to use the polar variables (r_i, ϑ_i), where r_1 and r_2 are the linear invariants of dynamics. The non-linear part of the equations of motion adds a coupling between the two planes, the perturbative parameter being the distance from the origin. Therefore, it is natural to replace r_1 and r_2 by the polar variables $r\cos\alpha$ and $r\sin\alpha$, respectively:

$$\begin{cases} x_1 &= r\cos\alpha\cos\vartheta_1 \\ p_{x_1} &= r\cos\alpha\sin\vartheta_1 \\ x_2 &= r\sin\alpha\cos\vartheta_2 \\ p_{x_2} &= r\sin\alpha\sin\vartheta_2\,. \end{cases} \qquad \begin{array}{l} r \in [0, +\infty[\\ \alpha \in [0, \pi/2] \\ \theta_i \in [0, 2\pi[\qquad i = 1, 2 \end{array} \tag{2}$$

Substituting in Eq. (1) we obtain

$$\int_0^{2\pi}\int_0^{2\pi}\int_0^{\pi/2}\int_0^{\infty} \chi(r, \alpha, \vartheta_1, \vartheta_2)\, r^3 \sin\alpha\cos\alpha\; d\Omega_4\,, \tag{3}$$

where $d\Omega_4$ represents the volume element

$$d\Omega_4 = dr\, d\alpha\, d\vartheta_1\, d\vartheta_2\,. \tag{4}$$

Having fixed α and $\boldsymbol{\vartheta} = (\vartheta_1, \vartheta_2)$, let $r(\alpha, \boldsymbol{\vartheta}, N)$ be the last value of r whose orbit is bounded after N iterations. Then, the volume of a connected domain in which the motion is bounded is given by

$$A_{\alpha,\boldsymbol{\vartheta},N} = \frac{1}{8}\int_0^{2\pi}\int_0^{2\pi}\int_0^{\pi/2} [r(\alpha, \boldsymbol{\vartheta}, N)]^4 \sin 2\alpha\; d\Omega_3\,, \tag{5}$$

where

$$d\Omega_3 = d\alpha\, d\vartheta_1\, d\vartheta_2\,. \tag{6}$$

In this way, we exclude stable islands that are not connected to the main stable domain. Note that, in principle, this method might also lead to excluding connected parts. We then define the DA as the radius of the hypersphere that

has the same volume as the stability domain

$$r_{\alpha,\boldsymbol{\vartheta},N} = \left(\frac{2A_{\alpha,\boldsymbol{\vartheta},N}}{\pi^2}\right)^{1/4} . \tag{7}$$

When Eq. (5) is implemented in a computer code, one considers K steps in the angle α and L steps in the angles ϑ_i, and the dynamic aperture reads

$$r_{\alpha,\boldsymbol{\vartheta},N} = \left[\frac{\pi}{2\,KL^2}\sum_{k=1}^{K}\sum_{l_1,l_2=1}^{L}[r(\alpha_k,\boldsymbol{\vartheta}_\ell,N)]^4\sin 2\alpha_k\right]^{1/4} ,$$

where $\ell = (l_1, l_2)$.

The numerical error is given by the discretization in angles ϑ_i, α, and radius r, which gives a relative error proportional to L^{-1}, K^{-1}, and J^{-1}, respectively. This numerical error can be optimised by choosing integration steps that produce comparable errors, i.e. $J \propto K \propto L$. In this way, neglecting the constants in front of the error estimates, one can obtain a relative error of $1/(4J)$ by evaluating the J^4 orbits, i.e. NJ^4 iterates. The fourth power in the number of orbits comes from the dimensionality of phase space and makes a precise estimate of the dynamic aperture very CPU time consuming.

It is possible to reduce the size of the scanning procedure, and hence the CPU time needed, by setting the angles $\boldsymbol{\theta}$ to a constant value, e.g. zero, thus performing only a 2D scan over r and α. This is what is generally done in SixTrack simulations [35, 36]. In this case, the transformation (2) reads

$$\begin{cases} x_1 &= r\cos\alpha \\ p_{x_1} &= 0 \\ x_2 &= r\sin\alpha \\ p_{x_2} &= 0\,, \end{cases} \qquad \begin{array}{l} \\ r \in [0,+\infty[\\ \alpha \in [0,\pi/2] \\ \\ \end{array} \tag{8}$$

and the original integral is transformed to

$$\int_0^{\pi/2}\int_0^{\infty} r\; dr\, d\alpha\,. \tag{9}$$

Having fixed α, let $r(\alpha, N)$ be the last value of r whose orbit is bounded after N iterations. Then, the volume of a connected stability domain is given by

$$A_{\alpha,N} = \frac{1}{2}\int_0^{\pi/2}[r(\alpha,N)]^2\; d\alpha\,. \tag{10}$$

We define the dynamic aperture as the radius of the sphere that has the same volume as the stability domain[1]

$$r_{\alpha,N} = \left(\frac{4A_{\alpha,N}}{\pi}\right)^{1/2} . \tag{11}$$

When Eq. (10) is implemented in a computer code, one considers K steps in the angle α, and the dynamic aperture reads

$$r_{\alpha,N} = \left[\frac{1}{K}\sum_{k=1}^{K}[r(\alpha_k, N)]^2\right]^{1/2} , \tag{12}$$

so that the numerical error is given by discretising the angle α and the radius r, which yields a relative error proportional to K^{-1} and J^{-1}, respectively. In this case, the integration steps should also be selected to produce comparable errors, i.e. $J \propto K$. In this way, neglecting the constants which are in front of the error estimates, one can obtain a relative error of $1/(2J)$ by evaluating J^2 orbits, i.e. NJ^2 iterates[2]. Note that Eq. (10) can be evaluated using higher-order numerical integration rules as implemented in the post-processing tools linked with SixTrack [36].

It is worth noting that, in some applications, the simplified formula

$$r_{\alpha,N} = \frac{1}{K}\sum_{k=1}^{K}[r(\alpha_k, N)] , \tag{13}$$

which corresponds to computing the average of $r(\alpha_k, N)$ over the angle α_k, could be used [17].

2.2 DA Scaling Law

All the definitions of DA estimates presented in the previous section are functions of N, the turn number used to estimate the orbit stability from the results of numerical simulations. It is evident that the definition of DA itself implies that it is a non-increasing function of N. The key point is whether it is possible to find the functional form of this time dependence, and several studies have shown that this is indeed the case [17, 24]. In fact, such a functional form can be built by considering the estimate of the stability time provided by the Nekhoroshev theorem [21, 22, 23], which is a key and very general theorem in the theory of Hamiltonian dynamical systems.

[1] Note that the region providing the stability domain is confined to a surface that is 1/4 of a circle and this has been considered in Eq. (11).

[2] The factor 2 in the error estimate is due to the dimensionality of the phase space

The first models were described in [17] and then reviewed in [24] and the two that we retained after the review read

$$\textbf{Model 2} \qquad \Rightarrow \qquad D(N) = \rho_* \left(\frac{\kappa}{2\mathrm{e}}\right)^{\kappa} \frac{1}{\ln^{\kappa} \frac{N}{N_0}}\,, \tag{14}$$

where the free parameters are ρ_*, κ, N_0, but it is customary to set $N_0 = 1$, and

$$\textbf{Model 4} \qquad \Rightarrow \qquad D(N) = \rho_* \times \\ \times \frac{1}{\left[-2\,\mathrm{e}\,\lambda\,\mathcal{W}_{-1}\!\left(-\frac{1}{2\,\mathrm{e}\,\lambda}\left(\frac{\rho_*}{6}\right)^{1/\kappa}\left(\frac{8}{7}N\right)^{-1/(\lambda\,\kappa)}\right)\right]^{\kappa}}\,, \tag{15}$$

where the free parameters are ρ_*, κ, and possibly λ, unless it is fixed to the value of $1/2$ according to the analytic Nekhoroshev estimate. $\mathcal{W}_{-1}$ stands for the negative branch of the Lambert-$\mathcal{W}$ function, a multi-valued special function (see, e.g. [37] for a review of the properties and applications of the Lambert function). Note that $D(N)$ stands for $r_{\alpha,\vartheta,N}$ or $r_{\alpha,N}$, depending on the numerical approach used to estimate the DA. The nomenclature of the models presented in Eqs. (14) and (15) reflects the historical development of these models and the nomenclature used in [24].

An example of the numerical calculation of the DA for a realistic model of the luminosity upgrade of the CERN LHC, HL-LHC [13], and the corresponding fitted scaling law using all available DA data are shown in Fig. 1, where the excellent agreement between the numerical data and the fit model is clearly visible. We denote by SL-ALL the fitting of **Model 2** using all available DA data.

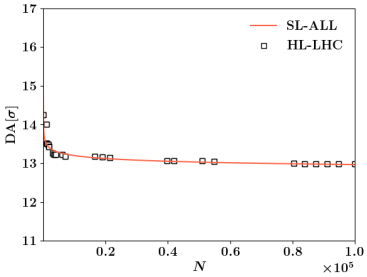


Fig. 1: Example of DA numerical computation for a realistic model of the HL-LHC with the corresponding fitted scaling law. The excellent agreement between the numerical data and the fit model is clearly visible.

Note that in the rest of the paper **Model 2** is the only scaling law model used.

2.3 DA data organisation

In this section, we present the data sets used to test the predictive model introduced in Section 4. The first data set is obtained from a realistic model of the HL-LHC, whereas the second one is obtained from the 4D Hénon map.

2.3.1 The HL-LHC case

The HL-LHC data set, presented in Fig. 2, is composed of 60 realisations (also called seeds due to the underlying random generator used for the generation of the realisations) of the magnetic field errors of the magnetic lattice of the HL-LHC, for the collision optics with β^*=15 cm and proton energy of 7 TeV. The 60 realisations are supposed to accurately represent the actual lattice of the HL-LHC; for this reason, the DA computation is customarily performed using the complete set of realisations to provide an accurate estimate of the DA of the actual accelerator. Magnetic field errors are assigned to all magnets that make up the ring. Initial conditions (also called particles) are distributed in physical space to probe the orbit stability and thus determine the DA. Different amplitudes and angles in the $x - y$ plane are used to sample the phase space. In the cases considered here, 11 angles, uniformly distributed in the interval $]0, \pi/2[$, are used, while the amplitudes are uniformly distributed in the interval $]0, 28\sigma[$, with 30 initial conditions defined in each 2σ interval. Note that 30 particles are evenly distributed in each amplitude interval of 2σ, and σ represents the root mean square (rms) beam size, which is used as a natural unit in these studies. All initial conditions are tracked for 10^5 turns. The numerical estimates of DA as a function of N are calculated according to Eq. (10) and are shown in Fig. 2 (left).

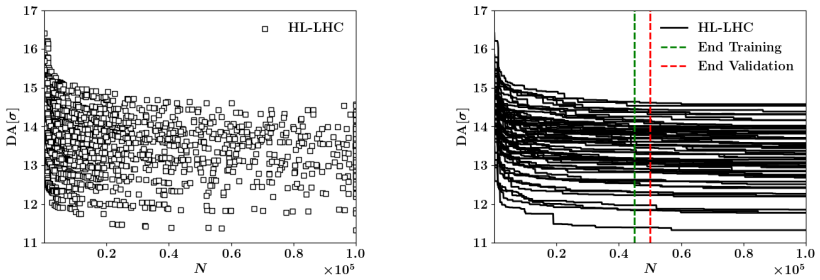


Fig. 2: Left: Evolution of DA as a function of time for the 60 realisations of the HL-LHC magnetic lattice. Right: Splitting of the HL-LHC data set into training, validation, and test sets.

We build piecewise constant functions so that each DA estimate now contains 10^3 data points, with the aim of obtaining DA estimates in constant time steps. These 10^3 data points are then divided into *training set*, *validation set*, and *test set*. The first $k_{\text{train}} = 450$ data are used for training, the next $k_{\text{val}} = 50$

data for validation, and the remaining $k_{\text{test}} = 500$ data for testing. Note that the end of the training and validation sets corresponds to $N = 5.10^4$, and the end of the testing to $N = 10^5$ turns. A graph of the 60 piecewise constant functions split into *training set*, *validation set* and *test set* is shown in Fig. 2 (right). Note that each of the 60 realisations corresponds to a different DA on which we will train, validate, and test our ESN model.

2.3.2 The 4D Hénon map case

The 4D Hénon map is a well-known dynamical system that displays a rich dynamical behaviour as presented in, e.g. [38]. The model used to generate DA estimates is defined as:

$$\begin{pmatrix} x_{n+1} \\ p_{x,n+1} \\ y_{n+1} \\ p_{y,n+1} \end{pmatrix} = \widetilde{R} \begin{pmatrix} x_n \\ p_{x,n} + x_n^2 - y_n^2 + \mu \left(x_n^3 - 3y_n^2 x_n \right) \\ y_n \\ p_{y,n} - 2x_n y_n + \mu \left(y_n^3 - 3x_n^2 y_n \right) \end{pmatrix} \tag{16}$$

where the subscript n denotes the discrete time and $\widetilde{R}$ is a 4×4 matrix given by the direct product of two 2×2 rotation matrices R:

$$\widetilde{R} = \begin{pmatrix} R(\omega_{x,n}) & 0 \\ 0 & R(\omega_{y,n}) \end{pmatrix}, \tag{17}$$

where the linear frequencies vary with the discrete time n according to

$$\omega_{x,n} = \omega_{x,0} \left(1 + \varepsilon \sum_{k=1}^{m} \varepsilon_k cos(\Omega_k n) \right) \tag{18}$$

$$\omega_{y,n} = \omega_{y,0} \left(1 + \varepsilon \sum_{k=1}^{m} \varepsilon_k cos(\Omega_k n) \right), \tag{19}$$

where ε denotes the amplitude of the frequency modulation and ε_k and Ω_k are fixed parameters, which are taken from previous studies [24][3].

The 4D Hénon map is a simplified model of a circular accelerator. In particular, it describes the effects of a sextupole and octupole magnet on the transverse particle motion through the quadratic, due to the sextupole, and cubic, due to the octupole, non-linear terms. Being a simplified accelerator model, it allows one to track particles up to a much larger number of turns, and for more amplitudes and angles, namely 100 amplitudes and angles uniformly distributed in the interval $]0, 0.25[$ and $]0, \pi/2[$ respectively. The 4D Hénon

[3]Note that all ε_k are of order 10^{-4}. Therefore, even if ε is large, the effective modulation of the frequencies shown in Eqs. (18) and (19) is very small.

map data set is composed of 60 cases, for 20 different values of ε uniformly distributed in the interval $[0, 20[$ and $\mu \in \{-0.2, 0, 0.2\}$, covering up to 10^8 turns. Similarly to the HL-LHC data set, we build piecewise-constant functions so that each case yields 1000 data points. The first $k_{\text{train}} = 450$ data are used for training, the next $k_{\text{val}} = 50$ data for validation, and the last $k_{\text{test}} = 500$ data for testing. Note that we used the same number of training, validation, and test data for the HL-LHC case. The 60 piecewise constant functions divided into *training set*, *validation set*, and *test set* are shown in Fig. 3.

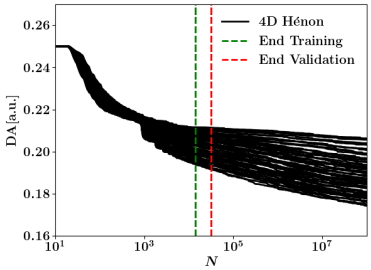


Fig. 3: Splitting of the 4D Hénon map data set into training, validation, and test sets. The sudden drop in DA visible for $N \approx 10^3$ occurs when $\varepsilon > 15$.

Note that because of the larger number of amplitudes and angles considered, the DA data are smoother than those of the HL-LHC case. Furthermore, each of the 60 cases generated in this data set corresponds to a different dynamics for which we will train, validate, and test our ESN model.

3 Echo State Networks

In this section, we present some general concepts about ESN. More specifically, we introduce the mathematical framework of continuous-time leaky ESN applied to supervised learning tasks.

3.1 Shallow ESN

Shallow ESN are a class of Recurrent Neural Networks using the Reservoir Computing approach [30]. In this type of neural network, the data input is fed into a single, random, and non-trainable network, called the reservoir. The reservoir is eventually connected by trainable weights to the ESN output. The use of ESN for time series prediction has become widespread due to its inexpensive training process and its remarkable performance in the modelling of dynamical systems [39].

Contrary to feedforward neural networks, ESN do not suffer from vanishing or divergent gradients (caused by the fact that the parameters of neural networks remain almost constant or lead to numerical instabilities), which induces poor performance of the training algorithm [40].

ESN can be defined for discrete- or continuous-time systems. The reservoir dynamics can be defined with or without the leaking rate parameter, which can be considered as the speed of the reservoir update dynamics. We introduce the definition of a shallow leaky ESN in continuous time as in [41]. We consider the case of networks with continuous-time t, K inputs, N_r reservoir neurons, and M outputs. Note that we will use small letters to indicate vectors and capital letters to indicate matrices. We define by $u = u(t) \in \mathbb{R}^K$ the input data and $x^{\mathrm{train}} = x^{\mathrm{train}}(t) \in \mathbb{R}^M$ the training data that we want to learn with the ESN model. The ESN output is denoted by $x^{\mathrm{out}} = x^{\mathrm{out}}(t) \in \mathbb{R}^M$, while the internal reservoir activation state is given by $x = x(t) \in \mathbb{R}^{N_\mathrm{r}}$. Furthermore, we define the input weight matrix $W^{\mathrm{in}} \in \mathcal{M}_{N_\mathrm{r} \times K}(\mathbb{R})$, the reservoir weight matrix $W \in \mathcal{M}_{N_\mathrm{r} \times N_\mathrm{r}}(\mathbb{R})$, and the output weight matrix $W^{\mathrm{out}} \in \mathcal{M}_{M \times (N_\mathrm{r}+K)}(\mathbb{R})$. The discretised (by the Euler method) time dynamics of a leaky ESN is given by:

$$x_k = F(x_{k-1}, u_k) = (1 - a\Delta t)\, x_{k-1} + \Delta t f(W^{\mathrm{in}} u_k + W x_{k-1}) \tag{20}$$

$$x_k^{\mathrm{out}} = g(W^{\mathrm{out}}[x_k;\ u_k]) \tag{21}$$

where $\Delta t = \delta / c$ with δ the size of the Euler discretization step and c a global time constant, a the leaking rate, f a sigmoid function, g the output activation function, $[.;\ .]$ denotes vector concatenation, x_k the update of the reservoir activation state at discrete time k and x_k^{out} the ESN output at the same time k.

In the case of a linear readout, i.e. when g is the identity function, we can rewrite Eq. (21) in matrix notation as:

$$X^{\mathrm{out}} = W^{\mathrm{out}} X \tag{22}$$

where $X^{\mathrm{out}} \in \mathcal{M}_{M \times (k_{\mathrm{train}} - BI)}(\mathbb{R})$ contains the M ESN outputs x^{out} at every time step $k = BI, \ldots, k_{\mathrm{train}}$ and where $X \in \mathcal{M}_{(N_\mathrm{r}+K) \times (k_{\mathrm{train}} - BI)}(\mathbb{R})$ contains the concatenation of the input u and the activation state of the reservoir x at every $k = BI, \ldots, k_{\mathrm{train}}$, namely

$$X = \begin{pmatrix} u_{BI} & \ldots & u_{k_{\mathrm{train}}} \\ x_{BI} & \ldots & x_{k_{\mathrm{train}}+1} \end{pmatrix}, \tag{23}$$

where BI denotes the *Burn-In* data, i.e. the number of input data we want to discard at the beginning of the training phase.

The optimal output weight matrix W^{out} can be found by solving the following minimisation problem:

$$\begin{aligned} W^{\text{out}} &= \operatorname{argmin} J(W^{\text{out}}) \\ &= \operatorname{argmin} \frac{1}{M} \sum_{i=1}^{M} \Big(\sum_{k=BI}^{T} (x_{ik}^{\text{out}} - x_{ik}^{\text{train}})^2 + \beta \|w_i^{\text{out}}\|^2 \Big), \end{aligned} \tag{24}$$

where J denotes the cost function we want to minimise and $\|w_i^{\text{out}}\|$ is the Euclidean norm of the ith row of W^{out}.

The solution of the minimisation problem stated in Eq. (24) can be found efficiently using linear regression with Tikhonov (Ridge) regularisation [42]:

$$W^{\text{out}} = X^{\text{train}} X^T (XX^T + \beta I)^{-1} \tag{25}$$

where the superscript T denotes the transpose, $I \in \mathcal{M}_{(N_r+K)\times(N_r+K)}(\mathbb{R})$ is the identity matrix, and $X^{\text{train}} \in \mathcal{M}_{M\times(k_{\text{train}}-BI)}(\mathbb{R})$ is the training data matrix, which contains the M training data x^{train} at time step $k = BI, \ldots, k_{\text{train}}$.

The learning phase is carried out on the so-called *training set*, which contains the k_{train} training data x^{train}. A sketch of the training phase of the ESN is provided in Fig. 4.

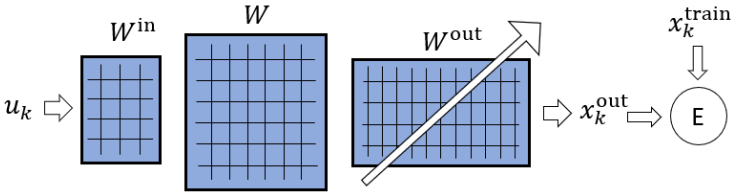


Fig. 4: Sketch of the training procedure for a shallow leaky ESN. The size of the matrices has been arbitrarily selected. E denotes the square of the Euclidean norm error between the ESN output x_k^{out} and the training data x_k^{train}, $k = BI, \ldots, k_{\text{train}}$.

After training, the ESN hyperparameters, defined in Section 4, are tuned using k_{val} validation data. Finally, the ESN is tested using the k_{test} data to check the ability of the ESN to predict new data. The validation and test procedures are detailed in Section 4. As stated in Eq. (24), only the output weight matrix W^{out} is trained, while the input and reservoir matrices W^{in} and W are randomly generated, as explained in detail in Section 4.

3.2 Deep ESN

A deep ESN is an ESN composed of L stacked reservoirs, as shown in the sketch of the deep ESN training phase in Fig. 5.

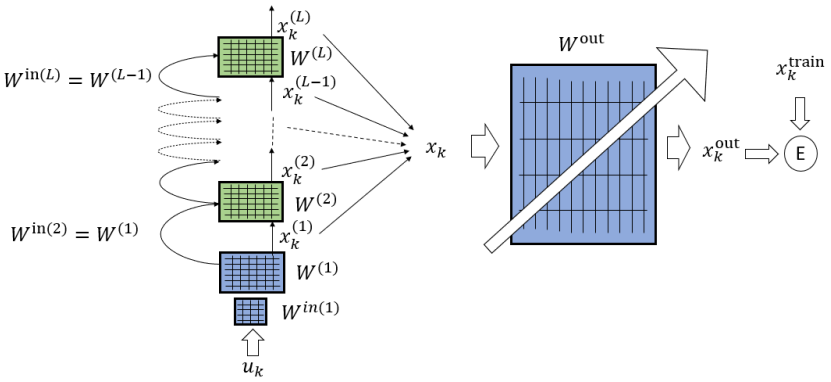


Fig. 5: Sketch of the training procedure for deep ESN with L reservoirs.

In this case, $W^{(l)}$ denotes the lth reservoir weight matrix, $W^{\text{in}(l)}$ the lth input weight matrix, $x_k^{(l)}$ the local internal reservoir state vector, and x_k the global internal reservoir state vector. Equations (20) and (21) for a shallow ESN read now

$$\begin{aligned} x_k^{(l)} &= (1 - a\Delta t)\, x_{k-1}^{(l)} + \Delta t f(W^{(l-1)} x_k^{(l-1)} + W^{(l)} x_{k-1}) \qquad l > 1 \\ x_k^{\text{out}} &= g(W^{\text{out}}[x_k;\ u_k])\,, \end{aligned} \tag{26}$$

where x_k is the concatenation of all $x_k^{(l)}$.

4 ESN predictive model for DA evolution

In the previous section, we have introduced the definition of a shallow leaky ESN and its extension as a deep ESN. In Eqs. (20) and (26) we can already identify some parameters (called hyperparameters) of the ESN predictive model. These are the leaking rate a, the number of stacked reservoirs L, the dimension N_r of the reservoir matrix W and the activation function f usually set as the hyperbolic tangent function tanh. In Appendix A, we give a sufficient condition on the spectral radius ρ of the reservoir matrix W, which can also be considered as a hyperparameter, that guarantees the Echo State Property (ESP).

Other hyperparameters are often introduced in the implementation of ESN equations. Specifically, the sparsity ratio s of the reservoir matrix W, i.e. the

fraction of 0 elements in the reservoir matrix W and BI (as in [43]), which corresponds to the number of time steps of the input data that are discarded. Furthermore, the regularisation parameter β in Eq. (25) also needs to be optimised and is also considered a hyperparameter of the ESN model. Setting large values for β is generally used to avoid overfitting and may improve prediction in the *test set*. To complete the definition of the predictive model of the ESN, we must assign a value to all hyperparameters, knowing that the performance of the model strongly depends on the choice of their values.

It is a common procedure in ESN training to perform an optimisation of these hyperparameters, which is usually done by grid search methods [44], in the *validation set*. The validation procedure considered here is based on an ensemble approach to deal with the randomness of the reservoirs. Eventually, once the predictive model has been trained and validated, we can test it in the *test set* with unseen data.

4.1 ESN ensemble validation approach

The ensemble validation approach used in our studies is based on the principle of minimising the average of the Relative Root Mean Square Error (RRMSE) of N_{d} dynamics predicted (i.e, 60 seeds for the HL-LHC dataset and 60 cases for the 4D Henon map) for N_{W} different randomly generated reservoirs and various hyperparameters values on the *validation set*. Note that for each of the N_{d} dynamics, we predict a mean over the N_{W} reservoirs. Additionally, each of the N_{d} dynamics contains different input/training/validation/test data, so that each prediction is performed independently of the others. We define this RRMSE on the *validation set* $\mathrm{RRMSE}^{\mathrm{val}}$ as:

$$\mathrm{RRMSE}^{\mathrm{val}} = \frac{1}{N_{\mathrm{d}}} \sum_{i=1}^{N_{\mathrm{d}}} \left(100 \sqrt{\frac{\sum_{k=1}^{k_{\mathrm{val}}} (x_{\mathrm{mean},k}^{\mathrm{out}-i} - x_k^{\mathrm{val}-i})^2}{\sum_{k=1}^{k_{\mathrm{val}}} (x_k^{\mathrm{val}-i})^2}} \right) \tag{27}$$

where k_{val} is the number of validation data, $x_{\mathrm{mean}}^{\mathrm{out}-i}$ is the mean over the N_{W} reservoirs for the ith dynamics at time k, and $x_k^{\mathrm{val}-i}$ is the validation data at the same time k for the same ith dynamics.

This procedure aims to build a robust predictive model in which all hyperparameters are fixed. The search of the hyperparameters values minimizing the $\mathrm{RRMSE}^{\mathrm{val}}$ is done over a domain S_h. Each of the hyperparameters is updated one by one using the value in S_h, which minimises $\mathrm{RRMSE}^{\mathrm{val}}$. Furthermore, as mentioned above, this ensemble validation method requires the generation of different random matrices W and W^{in}. This is done by sampling their elements from a uniform pseudorandom distribution in $(0, 1)$ and scaling them to the interval (-0.5,0.5) so that they also have negative elements. The procedure for generating W^{in} and W is detailed in Algorithm 1, while a pseudocode of the general ensemble validation procedure is presented in Algorithm 2.

Note that the functions Training() and Prediction() implement the equations presented in Section 3.

Algorithm 1 Generation of the random matrix W_{in} and W.

Require: ρ spectral radius we want to set to $\tilde{W} = \Delta t\|W\| + (1\text{-}a\Delta t)I$, K input size, N_{r} reservoir size

Ensure: W^{in} input weight matrix, W reservoir weight matrix

1: Random initialisation of $W^{\text{in}}_{i,j} \sim \mathcal{U}$ (0,1) - 0.5 , $i = 1, \ldots, N_{\text{r}}$, $j = 1, \ldots, K$, $W_{i,j} \sim \mathcal{U}$ (0,1) - 0.5 , $i, j = 1, \ldots, N_{\text{r}}$
2: Compute spectral radius ρ_{rand} of $\tilde{W}$
3: Scale W = ρ/ρ_{rand} W

Algorithm 2 Validation.

Require: S_h domain of search of an hyperparameter h, N_{W} random different pairs of (W^{in},W), N_{d} dynamics we want to validate with the associated inputs data u, training data x^{train} and validation data x^{val}.

Ensure: H set of all the fixed hyperparameters h (N_{r}, L, BI ρ, β,Δt) which minimise in average RRMSE$^{\text{val}}$ for the N_{d} dynamics and N_{W} reservoirs.

1: **for** $h \in H$ **do**
2: **for** $h_{val} \in S_h$ **do**
3: **for** $i = 1$ to N_d **do**
4: $x^{\text{out}-i}_{mean} = 0$
5: **for** $j = 1$ to N_{W} **do**
6: $W^{\text{out}} = \text{Training}(u^{-i}, x^{\text{train}-i}, h_{val}, (W^{\text{in}}, W)^j)$
7: $x^{\text{out}-i} = \text{Prediction}(h_{val}, (W^{\text{in}}, W)^j, W^{\text{out}})$
8: $x^{\text{out}-i}_{\text{mean}} += x^{\text{out}-i}$
9: **end for**
10: $x^{\text{out}-i}_{\text{mean}} = x^{\text{out}-i}_{mean}$ / N_{W}
11: $\text{RRMSE}^{\text{val}}_{h_{val}} \mathrel{+}= 100\sqrt{\frac{\sum_{k=1}^{k_{\text{val}}}(x^{\text{out}-i}_{\text{mean},k} - x^{\text{val}-i}_k)^2}{\sum_{k=1}^{k_{\text{val}}}(x^{\text{val}-i}_k)^2}}$
12: **end for**
13: $\text{RRMSE}^{\text{val}}_{h_{val}} = \text{RRMSE}^{\text{val}}_{h_{val}}/N_d$
14: **end for**
15: $h_{val} = \arg\min(\text{RRMSE}^{\text{val}}_{h_{val}})$
16: Set $h := h_{val}$ and update H
17: **end for**

4.2 ESN ensemble test approach

Once the parameters and hyperparameters of the ESN predictive model have been tuned using *training set* and *validation set*, we can test our ESN model for the prediction of not previously used data, i.e. DA values at a larger time. We denote by k_{test} the number of data in the *test set* we try to predict.

The algorithm 3 describes the test procedure for a single dynamics, i.e. a single realisation of the HL-LHC magnetic lattice or a single case for the 4D Hénon map data set. We can loop the procedure to perform the prediction in

Algorithm 3 Test

Require: N_{W} number of different pairs of (W^{in},W), x^{test} test data of size k_{test}, $H = \{N_{\mathrm{r}}, \Delta t, \rho, a, BI, L, \beta\}$ have been tuned using the k_{val} data in the *validation set* with the corresponding output weight matrix W^{out}.

Ensure: $x^{\mathrm{out}}_{\mathrm{mean}}$ mean of the predicted outputs over the N_{W} reservoirs, $\mathrm{RRMSE}^{\mathrm{test}}$ between $x^{\mathrm{out}}_{\mathrm{mean}}$ and x^{test}

1: $x^{\mathrm{out}}_{\mathrm{mean}} = 0$
2: **for** $j = 1$ *to* N_{W} **do**
3: $\quad x^{\mathrm{out}} = \mathrm{Prediction}(H, (W^{\mathrm{in}},W)^j, W^{\mathrm{out}})$
4: $\quad x^{\mathrm{out}}_{\mathrm{mean}} \mathrel{+}= x^{\mathrm{out}}$
5: **end for**
6: $x^{\mathrm{out}}_{\mathrm{mean}} \mathrel{/}= N_{\mathrm{W}}$
7: $\mathrm{RRMSE}^{\mathrm{test}} = 100\sqrt{\frac{\sum_{k=1}^{k_{\mathrm{test}}}(x^{\mathrm{out}}_{\mathrm{mean},k} - x^{\mathrm{test}}_k)^2}{\sum_{k=1}^{k_{\mathrm{test}}}(x^{\mathrm{test}}_k)^2}}$

the *test set* for the N_{d} dynamics. Note that, contrary to the validation, here the prediction is performed in the *test set* for data not previously used.

5 Results and Discussion

In this section, we present the DA predictions obtained with our ESN-based predictive model. In particular, we compare these predictions with those of the fitted scaling law presented in Eq. (14) and used in [24]. We recall that the ESN output $x^{\mathrm{out}}_{\mathrm{mean}}$ is the mean prediction over $N_{\mathrm{W}} = 100$ random reservoirs. The validation and testing methods are those introduced in Section 4. We tested the proposed approaches with the HL-LHC data sets and the 4D Hénon map presented in Section 2.

5.1 DA Predictions for the HL-LHC data set

5.1.1 Validation of the ESN

In this stage, we search for the set of hyperparameters H that minimises, on average over the $N_{\mathrm{d}} = 60$ seeds and $N_{\mathrm{W}} = 100$ randomly generated reservoirs, the RRMSE in the *validation set*. Here, the number of predicted dynamics is equal to the number of seeds. We also recall that the number of validation data is $k_{\mathrm{val}} = 50$ and the definition of $\mathrm{RRMSE}^{\mathrm{val}}$ is presented in Algorithm 2. The optimal hyperparameters are determined one by one by a grid search over a wide range of possible parameter values, and the search domains S_h of the hyperparameters are listed in Table 1.

Figure 6 shows $\mathrm{RRMSE}^{\mathrm{val}}$ as a function of the various hyperparameters in S_h. The values of the hyperparamters are updated one-by-one with those that minimise RMMSE^{val}. As we can see, a shallow ESN with a small number of neurons N_{r} provides the best results. Stacking more reservoirs does not improve the predictions. In fact, adding reservoirs or increasing the number of neurons

Table 1: Search domains S_h of the various hyperparameters h. Note that the hyperparameters h are tuned one by one while the others are kept constant.

h	S_h
N_r	$\{20, 40, 60, \ldots, 200\}$
L	$\{1, 2, 3\}$
ρ	$\{0.1, 0.2, \ldots, 0.9, 0.99\}$
BI	$\{0, 25, 50, \ldots, 200\}$
β	$\{2 \times 10^{-10}, 2 \times 10^{-8}, \ldots, 2 \times 10^{-2}, 2 \times 10^{-1}\}$
Δt	$\{9 \times 10^{-7}, 9 \times 10^{-6}, \ldots, 9 \times 10^{-2}, 9 \times 10^{-1}\}$

makes the model overfit, so it cannot predict correctly in the *validation set*. This can be explained by the small number of features that the ESN must learn and by the characteristics of the DA data, which are not enough.

Regarding the other hyperparameters, the optimum spectral radius value initially set to 0.1 is updated to 0.99 and satisfies the ESP. Furthermore, since the optimal value of N_r is smaller than 100, it can be considered small, which justifies setting the sparsity ratio $s = 0$ so that all elements of W are non-zero. Then, we decided to choose the activation function $f = \tanh$, since it is the most used in ESN, and the leaking rate $a = 1$ to simplify the equations described in (26). Eventually, the values of β and Δt initially set to 2.10^{-1} and 9.10^{-2} are updated to 2.10^{-2} and 9.10^{-3} respectively. The values of the hyperparameters updated after validation and used for the prediction stage in the *test set* are summarised in Table 2.

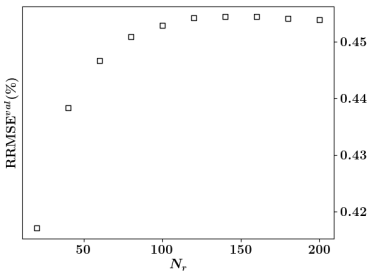
Table 2: Set H of the hyperparameters tuned after validation using HL-LHC DA data.

N_r	β	ρ	a	BI	L	Δt	f	s
20	2.10^{-2}	0.99	1	0	1	9.10^{-3}	tanh	0

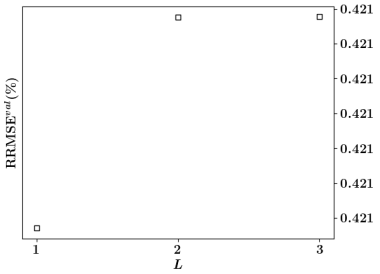
5.1.2 The ESN model

Once the ESN has been trained and validated, we can test it with the *test set* for data not previously used using the hyperparameters reported in Table 2. We recall that the number of test data is $k_\mathrm{test} = 500$, i.e. half of the total number of data used. In Fig. 7, we show the mean prediction $x^\mathrm{out}_\mathrm{mean}$ in the *test set* together with the envelope (i.e. minimum and maximum) of the predictions x^out that are associated with the $N_\mathrm{W} = 100$ randomly generated reservoirs for an arbitrary seed (number 1). We also plot the distribution of the prediction of DA at $N = 10^5$ turns (end of the *test set*).

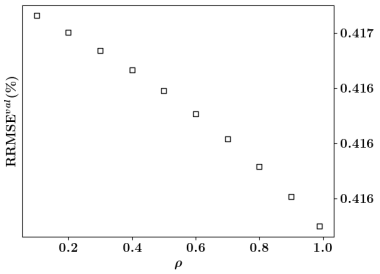
As mentioned above, we will denote by $x^\mathrm{out}_\mathrm{mean}$ the ESN mean prediction and only plot this mean value to avoid overloading the graphs with the values generated by N_W random reservoirs. To have a complete view, Fig. 8 shows the predictions of $N_\mathrm{d} = 60$ seeds in the *train set*, *validation set* and *test set*. Vertical dashed lines indicate the end of the *train set* and *validation set* for



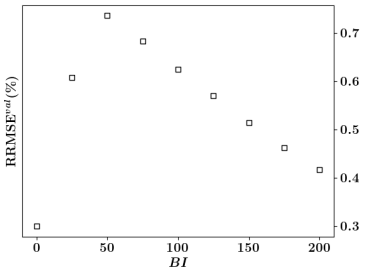
Values of constant hyperparameters: $L = 1$, $\rho = 0.1$, $BI = 2 \times 10^2$, $\beta = 2 \times 10^{-1}$, $\Delta t = 9 \times 10^{-2}$.



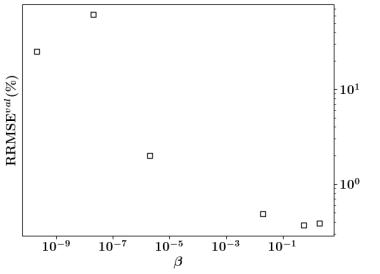
Values of constant hyperparameters: $N_\mathrm{r} = 20$ $\rho = 0.1$, $BI = 2 \times 10^2$, $\beta = 2 \times 10^{-1}$, $\Delta t = 9 \times 10^{-2}$.



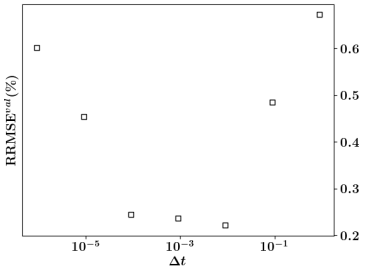
Values of constant hyperparameters: $N_\mathrm{r} = 20$, $L = 1$, $BI = 2 \times 10^2$, $\beta = 2 \times 10^{-1}$, $\Delta t = 9 \times 10^{-2}$.



Values of constant hyperparameters: $N_\mathrm{r} = 20$, $L = 1$, $\rho = 0.99$, $\beta = 2 \times 10^{-1}$, $\Delta t = 9 \times 10^{-2}$.



Values of constant hyperparameters: $N_\mathrm{r} = 20$, $L = 1$, $\rho = 0.99$, $BI = 0$, $\Delta t = 9 \times 10^{-2}$.



Values of constant hyperparameters: $N_\mathrm{r} = 20$, $L = 1$, $\rho = 0.99$, $BI = 0$, $\beta = 2 \times 10^{-2}$.

Fig. 6: RRMSE$^{\mathrm{val}}$ as a function of the various hyperparameters in S_h.

ESN (left graph) and SL (right graph). The scaling law fit is performed using the first $k_{\mathrm{fit}} = k_{\mathrm{train}} + k_{\mathrm{val}} = 500$ DA data. Note that ESN and SL share the same *test set*. Figure 9 shows the distribution of the RRMSE$^{\mathrm{test}}$ values defined in Algorithm 3, for both the ESN model and SL.

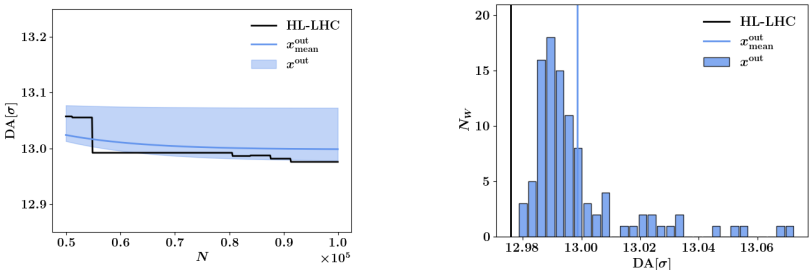


Fig. 7: Left: Numerical DA data, prediction of DA $x^{\text{out}}_{\text{mean}}$, average, minimum, and maximum over the $N_{\text{W}} = 100$ randomly generated reservoirs as a function of time. Right: distribution for the $N_{\text{W}} = 100$ randomly generated reservoirs at $N = 10^5$. The seed used for both plots is number 1.

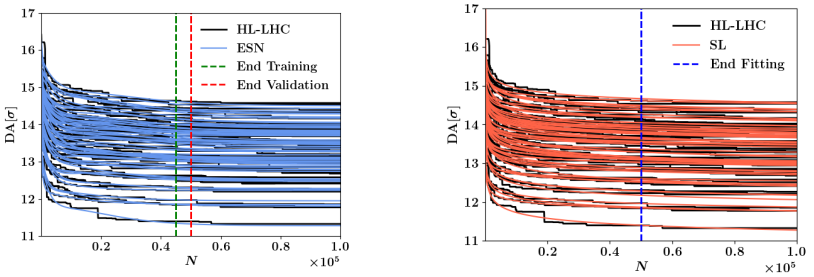


Fig. 8: DA predictions for ESN (left) and SL (right) for $N_{\text{d}} = 60$ seeds.

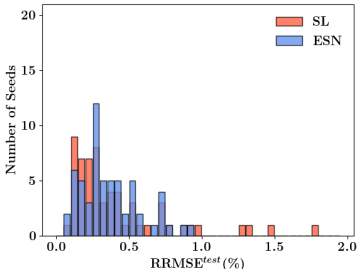


Fig. 9: Distribution of RRMSE$^{\text{test}}$ for N_{d}=60 seeds for ESN and SL.

We report in Table 3 the mean, maximum, minimum, and standard deviation of RRMSE^{test} for the predictions of ESN and SL over $N_{\text{d}} = 60$ seeds.

The ESN model and SL generate predictions whose distributions have essentially the same mean and minimum values. However, some outliers appear

Table 3: Mean, maximum, minimum, and standard deviation of the $\text{RRMSE}^{\text{test}}$ distribution.

	Mean	Max	Min	Std
ESN	0.37	0.94	0.06	0.20
SL	0.42	1.78	0.07	0.35

in the SL distribution, which affect the maximum and standard deviation values. This contributes to the generation of more stable predictions by ESN, i.e. without outliers, and significantly lower values of the standard deviation and maxima.

5.1.3 The SL-ESN model

In this section, we consider whether ESN predictions can possibly be used to replace the tracking simulations that generated the data in the *test set*. In this sense, we fit the SL to the k_{fit} data plus the ESN predictions in the *test set*. We denote this fit procedure by SL-ESN and compare it with the results of SL-ALL, which represents the best results that can be achieved with the SL approach[4]. The idea is to check the quality of the approximation of SL-ESN in the *test set*, in view of further prediction beyond this set. The predictions provided by SL-ESN and SL-ALL for the $N_{\text{d}} = 60$ seeds can be seen in Fig. 10 and the distribution of $\text{RRMSE}^{\text{test}}$ is shown in Fig. 11, while the mean, maximum, minimum, and standard deviation of $\text{RRMSE}^{\text{test}}$ in Table 4.

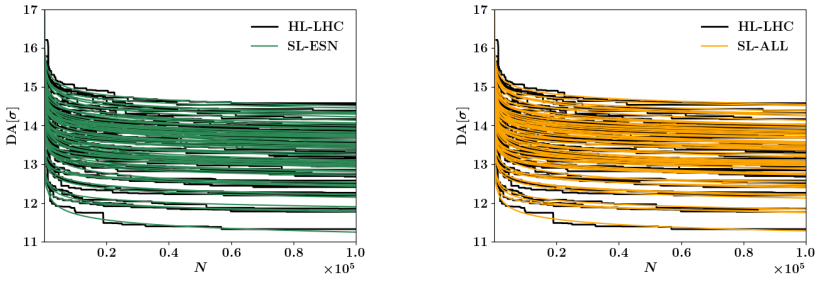


Fig. 10: Predictions for SL-ESN (left) and SL-ALL (right) for $N_{\text{d}} = 60$ seeds.

As it might be expected, all indicators of the distribution of $\text{RRMSE}^{\text{test}}$ for SL-ESN are significantly larger than those for SL-ALL, as the first approach fits the prediction of ESN, not the real DA data. In fact, SL-ESN is essentially equivalent to ESN alone and hence more stable than SL alone as far as outliers are concerned. In other words, the SL-ESN seems to be an effective surrogate model that improves the predictions given by the SL only.

[4]We recall that SL-ALL denotes the fit obtained by using all the available DA data, namely $k_{\text{fit}} + k_{\text{test}}$.

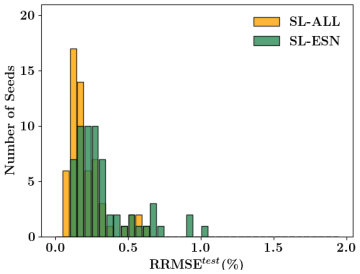


Fig. 11: Distribution of $\mathrm{RRMSE}^{\mathrm{test}}$ for $N_{\mathrm{d}} = 60$ seeds for SL-ESN and SL-ALL.

Table 4: Mean, maximum, minimum, and standard deviation of the $\mathrm{RRMSE}^{\mathrm{test}}$ distribution.

	Mean	Max	Min	Std
SL-ESN	0.33	1.01	0.10	0.21
SL-ALL	0.21	0.64	0.06	0.13

After having evaluated the accuracy of the SL-ESN model in the *test set*, we can check if it can replace the tracking simulations in this set. To do so, we compute predictions beyond the *test set* and up to $N = 10^8$ turns. Since we do not have real DA data in this time interval, we cannot compute any metrics, and we use the envelope, i.e. minimum and maximum, of the predictions given by SL-ESN and SL-ALL to check whether SL-ESN approximates well the predictions given by SL-ALL beyond the *test set*. We plot the envelope of the predictions given by SL-ESN and SL-ALL beyond the *test set* in Fig. 12 (left), and we also show the relative error ϵ_{r} defined as $\epsilon_{\mathrm{r}}^{i} = (DA^{i}_{\mathrm{SL-ALL}} - DA^{i}_{\mathrm{SL-ESN}})/DA^{i}_{\mathrm{SL-ALL}}$ where i is max or min (right).

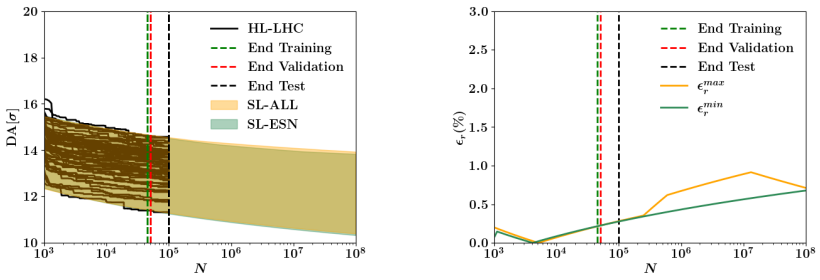


Fig. 12: Left: Envelope, i.e. minimum and maximum values of the SL-ESN and SL-ALL predictions extrapolated beyond the *test set*. Right: Relative error ϵ_{r} of the minimum and maximum DA predictions up to $N = 10^8$ turns.

The two envelopes almost overlap until $N = 10^8$ turns, with ϵ_r^{max} and ϵ_r^{min} that are below 1%. From this observation we conclude that we may only need to perform the tracking simulation until the end of the validation so that the tracking in the *test set* could be spared. In fact, the predictions provided by SL-ESN are very similar to those of SL-ALL. In this way, we could use the ESN predictions to replace the tracking in the *test set*. This result is in line with what was found in [45], i.e. that the addition of synthetic points obtained by using Gaussian Processes improved the quality of the fitted SL model.

Running the SixTrack code [35, 36] and the ESN model on the same CPU architecture, we have a speed-up of a factor 20 by replacing the tracking simulations on 5×10 4 turns, representing the *test set*, with the prediction of the DA values by ESN. This evaluation of CPU time reduction can be easily improved by a trivial parallelisation of the ESN over the 100 reservoirs. Of course, the actual gain depends on several details, such as the model under consideration and the definition of the times that define the validation and test sets. It is worth stressing that whenever an actual accelerator lattice is used for the numerical DA computations, the CPU time needed depends not only on the number of turns used for the tracking, but also on the size of the accelerator, which corresponds approximately to the number of magnets comprised in the lattice, and on the characteristics of the magnetic field errors included in the accelerator model. In this respect, the computational gain implied by the proposed approach is even more relevant for the case of large future colliders, such as the Future Circular Hadron Collider (FCC-hh) under study at CERN [46, 47].

5.2 DA Predictions for the Hénon map data set

To check the robustness of the current strategy, we apply it to a new system, which is the 4D Hénon map introduced in Section 2.

5.2.1 The ESN model

Hyperparameters have been determined using the same approach as for the HL-LHC data and are reported in Table 5. In this case, we also use $N_d = 60$, but we have to stress that the various dynamics differ between them much more than the dynamics of the HL-LHC case. In fact, changes in the values of ε and μ lead to radically different dynamical behaviours, whereas the HL-LHC realisations are much closer to each other, representing minor variations of the same dynamical behaviour.

Only the values of Δt and β are different from those of the HL-LHC case. Note that the value of β found is much lower than that of HL-LHC. This means that the model is less overfitting than with the HL-LHC data, especially because the Hénon DA data are much smoother.

In Fig. 13, we plot the $N_d = 60$ DA predictions given by ESN and SL. For ESN, we recall that we used $k_{train} = 450$ and $k_{val} = 50$ data, and for SL we used the $k_{fit} = 500$ data. Furthermore, *test set* is the same for both ESN

Table 5: Set H of the hyperparameters tuned after validation using Hénon map DA data.

N_r	β	ρ	a	BI	L	Δt	f	s
20	9.10^{-6}	0.99	1	0	1	0.004	tanh	0

and SL. As we can see, the SL predictions in the *test set* do not perform well, whereas those provided by the ESN fit the training/validation/test data much better.

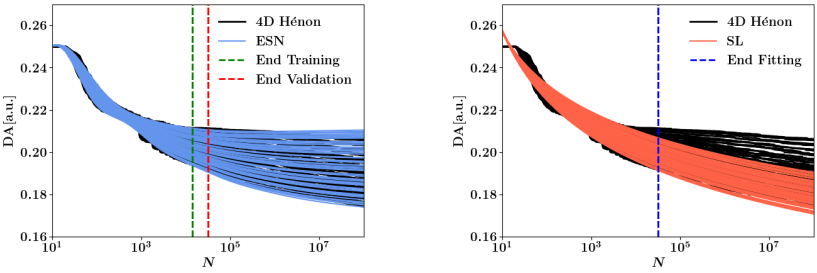


Fig. 13: DA predictions for ESN (left) and SL (right) for $N_d = 60$ seeds.

In Fig. 14 we compare the distributions of $RRMSE^{test}$ for ESN and SL, and the first is clearly much narrower and closer to zero than the latter. This behaviour is easily explained by considering the fact that the scaling law is an asymptotic law that aims to describe the long-term behaviour of the DA (using very few model parameters). Therefore, it is not effective in reproducing the detailed behaviour of the DA for low numbers of turns. Our ESN model is able to fit both the short-term and long-term behaviour simultaneously, thus explaining the observed better performance.

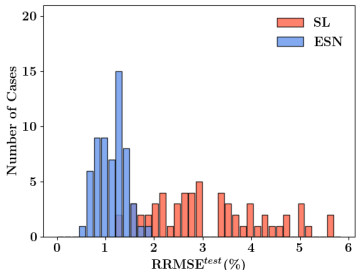


Fig. 14: Distribution of $RRMSE^{test}$ for N_d=60 seeds for ESN and SL.

The mean, maximum, minimum, and standard deviation of $\text{RRMSE}^{\text{test}}$ for the two approaches are reported in Table 6.

Table 6: Mean, maximum, minimum, and standard deviation of the $\text{RRMSE}^{\text{test}}$ distribution.

	Mean	Max	Min	Std
ESN	1.13	1.89	0.59	0.28
SL	3.17	5.85	1.25	1.18

The table shows, in a quantitative way, the differences observed in the histogram of the distributions. In fact, the RRMSE of the ESN is on average about 3 times lower than that of the SL, which is a significant improvement compared to the case of HL-LHC. Several reasons can explain this behaviour. First, the DA data for the Hénon map are much smoother than those of the HL-LHC data set, which improves training and limits overfitting of the ESN. Second, as already mentioned, the behaviour of the N_{d} dynamics is very diverse, and the SL, with only two free parameters, is clearly disadvantaged with respect to the ESN. Moreover, since the SL is an asymptotic law, its performance has been downgraded by including low-turn DA data.

5.2.2 The SL-ESN model

We repeat the procedure to check if the ESN predictions can replace the tracking simulation in the *test set*. As previously, we compare SL-ESN with SL-ALL. The predictions given by SL-ESN and SL-ALL for the 60 cases can be seen in Fig. 15, the distribution of $\text{RRMSE}^{\text{test}}$ is shown in Fig. 16, and the mean, maximum, minimum, and standard deviation of $\text{RRMSE}^{\text{test}}$ are reported in Table 7.

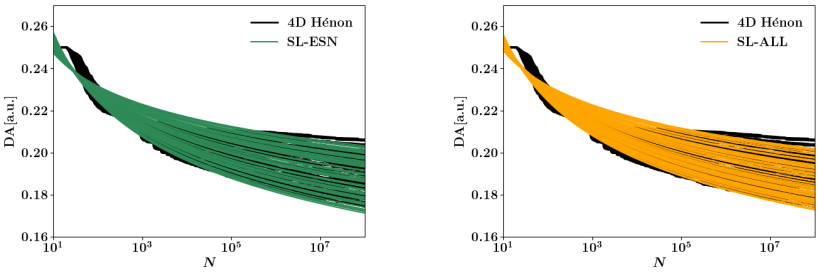


Fig. 15: Predictions for SL-ESN (left) and SL-ALL (right) for $N_{\text{d}} = 60$ seeds.

In this case, the SL-ESN performs equally well as the SL-ALL. In fact, the mean of $\text{RRMSE}^{\text{test}}$ is the same. Furthermore, fitting the SL to the predictions of the ESN allows us to improve the accuracy of the ESN and the SL. Taking

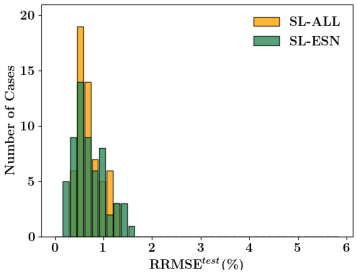


Fig. 16: Distribution of $\mathrm{RRMSE}^{\mathrm{test}}$ for $N_{\mathrm{d}} = 60$ seeds for SL-ESN and SL-ALL.

Table 7: Mean, maximum, minimum, and standard deviation of the $\mathrm{RRMSE}^{\mathrm{test}}$ distribution.

	Mean	Max	Min	Std
SL-ESN	0.71	1.57	0.26	0.33
SL-ALL	0.72	1.29	0.35	0.24

into account the average, SL-ESN is almost 2 times and 4 times more accurate than ESN and SL, respectively. Similarly to the HL-LHC case, the standard deviation and maximum $\mathrm{RRMSE}^{\mathrm{test}}$ of SL-ESN are much lower than those of SL, which shows a certain robustness of the conclusions that SL-ESN helps improve SL.

To further check whether the ESN predictions can replace the tracking simulation in the *test set*, we perform the prediction beyond the *test set* up to $N = 10^{11}$ turns. As previously, we do not have the real DA data in this range, so we cannot compute any metrics. We plot the envelope of the predictions given by SL-ESN and SL-ALL in Fig. 17.

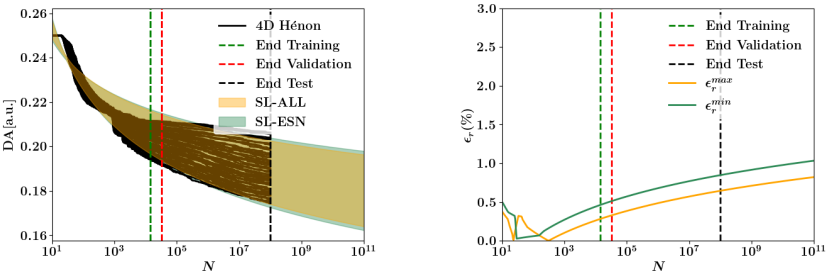


Fig. 17: Left: Envelope, i.e. minimum and maximum values of the SL-ESN and SL-ALL predictions extrapolated beyond the *test set*. Right: Relative error ϵ_{r} of the minimum and maximum DA predictions up to $N = 10^{11}$ turns.

The two envelopes of the predictions almost overlap until $N = 10^{11}$, and the relative errors ϵ_r^{max} and ϵ_r^{min} are below 1.5%, as for the case HL-LHC. This indicates, once again, that the tracking simulation in the *test set* could be replaced by the ESN predictions.

6 Conclusions

In this article, we have presented the results obtained with an ensemble approach to ESN reservoir computing for the prediction of the dynamic aperture of a circular hadron accelerator. In particular, we have compared the performance of ESN with that of a scaling law based on the Nekhoroshev theorem to predict the evolution of the dynamic aperture over time. This analysis has been carried out on two data sets that have been generated using numerical simulations performed on realistic models of the transverse beam dynamics in the HL-LHC and on a modulated 4D Hénon map with quadratic and cubic non-linearities.

We have shown that the average accuracy in the *test set* of the scaling law used to fit the ESN predictions was better than that of the scaling law alone. In particular, we have observed that the standard deviation of the RRMSE of the scaling law combined with the ESN is much lower than that of the scaling law alone. This leads to more reliable predictions. The fact that this observation is confirmed for both data sets gives us confidence that the combination of the scaling law and the ESN is the best approach.

A consequence of this result is that the tracking performed in the *test set* can be avoided by replacing it with the predictions of the ESN. In fact, for both the HL-LHC and Hénon map data sets, the predictions of the scaling law combined with the ESN and of the scaling law fitted to the entire data set are close to the percent level, even for numbers of turns three orders of magnitude beyond that of the *test set*. The gain in CPU time depends on the size of the accelerator and the complexity of its model. However, it is clear that the proposed approach is particularly appealing for hadron colliders of the post-LHC era that are currently being studied.

The study presented here represents only the beginning of a research area that could be further developed in the future given the promising results obtained. The partition of available data into training, validation, and test data sets should be studied in more detail to assess whether such a partition could be obtained using an appropriate algorithm. The established link between dynamic aperture and models for the evolution of intensity in hadron rings and the evolution of luminosity in hadron colliders could be further developed by using the promising results discussed in this paper. Investigations on the possibility of using ESN to improve the modelling of beam lifetime and luminosity evolution should be seriously considered and pursued. Finally, the predictive power of ESN could be applied to indicators of chaos, which are dynamical observables computed over the orbit of an initial condition to establish whether the motion is regular or chaotic, to improve their performance.

This would be another important topic that could bring important insight to the field of non-linear beam dynamics.

Acknowledgements

We would like to express our gratitude to C.E. Montanari for providing the software to perform dynamic aperture simulations for the 4D Hénon map.

Appendix A The Echo State Property

An important prerequisite for the output-only training is the so-called Echo State Property (ESP), which guarantees that initial conditions have an effect that vanishes over time. We use the results presented in [48] to recall the definition of ESP and a new sufficient condition that can be used in practice. In fact, satisfying the ESP allows us to guarantee that the reservoir activation state x_{k-1} is uniquely determined by any left-infinite input sequence $\ldots, u_{k-2}, u_{k-1}$.

To define ESP, we require the *compactness condition*, that is, we assume that the states and inputs belong to compact sets $\mathcal{X} \subset \mathbb{R}^{N_r}, \mathcal{U} \subset \mathbb{R}^K$ and that $F(x_{k-1}, u_k) \in \mathcal{X}$ and $u_{k-1} \in \mathcal{U}, \forall k \in \mathbb{Z}$. In practice, the ESN inputs will always be bounded, so the compactness of $\mathcal{U}$ is guaranteed. Furthermore, for bounded sigmoid functions f, such as tanh, the state space $\mathcal{X}$ is also compact. We define $\mathcal{U}^{-\infty} = \{u^{-\infty} = (\ldots, u_{-1}, u_0),\ u_k \in \mathcal{U}\, \forall k \in \mathbb{Z}\}$ and $\mathcal{X}^{-\infty} = \{x^{-\infty} = (\ldots, x_{-1}, x_0),\ x_k \in X\, \forall k \in \mathbb{Z}\}$, which are the sets of infinite left input and reservoir activation state sequences.

Definition 3.1 (ESP). A network $F : \mathcal{X} \times \mathcal{U} \to \mathcal{X}$ with the *compactness condition* has the ESP with respect to $\mathcal{U}$ if for any left input sequence $u^{-\infty} \in \mathcal{U}^{-\infty}$ and any two state sequences $x^{-\infty}, y^{-\infty} \in \mathcal{X}^{-\infty}$ compatible with $u^{-\infty}$ (i.e. $x_k = F(x_{k-1}, u_k), \forall k \leq 0$), then for all $k \geq 0, \|x_k - y_k\| \leq \delta_k$, where δ_k denotes a small value.

Definition A is not easily applicable in practice. Thus, we introduce the following Theorem A that should be used in practise as it provides a sufficient condition to satisfy the ESP in the case of a leaky ESN:

Theorem 3.1 (Sufficient condition of the ESP). If the spectral radius of the matrix

$$\tilde{W} = \frac{\Delta t}{c} W + \left(1 - a\frac{\Delta t}{c}\right) I$$

is smaller than 1, then the leaky ESN with $f = \tanh$ satisfies the ESP for all inputs. However, this condition is only sufficient, but not necessary. In other words, setting $\rho(\tilde{W}) \geq 1$ does not necessarily lead to poor performance of leaky ESN.

References

[1] Visnjic, V.: Dynamic Aperture of Low Beta Lattices at Tevatron Collider. In: Proc. PAC'91, pp. 1701–1704. JACoW Publishing, Geneva, Switzerland

[2] V. Visnjic: Dynamic aperture of the future Tevatron Collider. Nonlinear Problems in Future Particle Accelerators. World Scientific, ??? (1991)

[3] Brinkmann, R., Willeke, F.: Persistent Current Field Errors and Dynamic Aperture of HERA-P. In: Proc. EPAC'88, pp. 911–914. JACoW Publishing, Geneva, Switzerland

[4] Zimmermann, F., Willeke, F.: Long term stability and dynamic aperture of the HERA proton ring (1991)

[5] Zimmermann, F.: Dynamic aperture and transverse proton diffusion in hera. AIP Conference Proceedings **326**(1), 98–166 (1995) https://aip.scitation.org/doi/pdf/10.1063/1.47320. https://doi.org/10.1063/1.47320

[6] Luo, Y., et al.: Dynamic Aperture Evaluation at the Current Working Point for RHIC Polarized Proton Operation. In: Proc. PAC'07, pp. 4363–4365. JACoW Publishing, Geneva, Switzerland. https://jacow.org/p07/papers/FRPMS111.pdf

[7] Brüning, O.S., Collier, P., Lebrun, P., Myers, S., Ostojic, R., Poole, J., Proudlock, P.: LHC Design Report. CERN Yellow Rep. Monogr. CERN, Geneva (2004). https://doi.org/10.5170/CERN-2004-003-V-1

[8] Appleby, R., et al.: Dynamic Aperture Studies of the nuSTORM FFAG Ring. In: Proc. IPAC'14, pp. 1574–1577. JACoW Publishing, Geneva, Switzerland. https://doi.org/10.18429/JACoW-IPAC2014-TUPRI013. https://jacow.org/IPAC2014/papers/TUPRI013.pdf

[9] Jing, Y.C., Litvinenko, V., Trbojevic, D.: Optimization of Dynamic Aperture for Hadron Lattices in eRHIC. In: Proc. IPAC'15, pp. 757–759. JACoW Publishing, Geneva, Switzerland. https://doi.org/10.18429/JACoW-IPAC2015-MOPMN027. https://jacow.org/IPAC2015/papers/MOPMN027.pdf

[10] Dalena, B., et al.: First Evaluation of Dynamic Aperture at Injection for FCC-hh. In: Proc. IPAC'16, pp. 1466–1469. JACoW Publishing, Geneva, Switzerland. https://doi.org/10.18429/JACoW-IPAC2016-TUPMW019. https://jacow.org/ipac2016/papers/TUPMW019.pdf

[11] Dalena, B., Boutin, D., Chance, A., Holzer, B.J., Schulte, D.: Advance on Dynamic Aperture at Injection for FCC-hh. In: Proc. IPAC'17, pp. 2027–2030. JACoW Publishing, Geneva, Switzerland. https://doi.org/10.18429/JACoW-IPAC2017-TUPVA003. https://jacow.org/ipac2017/papers/TUPVA003.pdf

[12] Alaniz, E.C., Seryi, A., Maclean, E.H., Martin, R., Tomas, R.: Non Linear Field Correction Effects on the Dynamic Aperture of the FCC-hh. In: Proc. IPAC'17, pp. 2143–2146. JACoW Publishing, Geneva, Switzerland. https://doi.org/10.18429/JACoW-IPAC2017-TUPVA038. https://jacow.org/ipac2017/papers/TUPVA038.pdf

[13] Apollinari, G., Béjar Alonso, I., Brüning, O., Fessia, P., Lamont, M., Rossi, L., Tavian, L.: High-Luminosity Large Hadron Collider (HL-LHC). CERN Yellow Rep. Monogr., vol. 4. CERN, Geneva (2017). https://doi.org/10.23731/CYRM-2017-004

[14] Dalena, B., et al.: Dipole Field Quality and Dynamic Aperture for FCC-hh. In: Proc. IPAC'18, pp. 137–140. JACoW Publishing, Geneva, Switzerland. https://doi.org/10.18429/JACoW-IPAC2018-MOPMF024. http://accelconf.web.cern.ch/ipac2018/papers/MOPMF024.pdf

[15] Alaniz, E.C., Abelleira, J.L., Seryi, A., van Riesen-Haupt, L., Martin, R., Tomas, R.: Methods to Increase the Dynamic Aperture of the FCC-hh Lattice. In: Proc. IPAC'18, pp. 3593–3596. JACoW Publishing, Geneva, Switzerland. https://doi.org/10.18429/JACoW-IPAC2018-THPAK145. http://accelconf.web.cern.ch/ipac2018/papers/THPAK145.pdf

[16] Giovannozzi, M., Scandale, W., Todesco, E.: Prediction of long-term stability in large hadron colliders. Part. Accel. **56**, 195 (1997)

[17] Giovannozzi, M., Scandale, W., Todesco, E.: Dynamic aperture extrapolation in the presence of tune modulation. Phys. Rev. E **57**, 3432–3443 (1998). https://doi.org/10.1103/PhysRevE.57.3432

[18] Giovannozzi, M.: Proposed scaling law for intensity evolution in hadron storage rings based on dynamic aperture variation with time. Phys. Rev. ST Accel. Beams **15**, 024001 (2012). https://doi.org/10.1103/PhysRevSTAB.15.024001

[19] Giovannozzi, M., Van der Veken, F.F.: Description of the luminosity evolution for the CERN LHC including dynamic aperture effects. Part I: the model. Nucl. Instrum. Methods Phys. Res. **A905**, 171–179 (2018) arXiv:1806.03058 [physics.acc-ph]. https://doi.org/10.1016/j.nima.2019.01.072. [Erratum: Nucl. Instrum. Methods Phys. Res. A927,471(2019)]

[20] Giovannozzi, M., Van der Veken, F.F.: Description of the luminosity evolution for the CERN LHC including dynamic aperture effects. Part II: application to Run 1 data. Nucl. Instrum. Methods Phys. Res. **A908**, 1–9 (2018) arXiv:1806.03059 [physics.acc-ph]. https://doi.org/10.1016/j.nima.2018.08.019

[21] Nekhoroshev, N.: An exponential estimate of the time of stability of nearly-integrable Hamiltonian systems. Russ. Math. Surv. **32**, 1 (1977)

[22] Bazzani, A., Marmi, S., Turchetti, G.: Nekhoroshev estimate for isochronous non resonant symplectic maps. Cel. Mech. **47**, 333 (1990)

[23] Turchetti, G.: Nekhoroshev stability estimates for symplectic maps and physical applications. In: Proc. of the Winter School. Springer Proceedings in Physics, vol. 47, p. 223. Les Houches, France ('89) (1990)

[24] Bazzani, A., Giovannozzi, M., Maclean, E.H., Montanari, C.E., Van der Veken, F.F., Van Goethem, W.: Advances on the modeling of the time evolution of dynamic aperture of hadron circular accelerators. Phys. Rev. Accel. Beams **22**, 104003 (2019)

[25] Gevaert, W., Tsenov, G., Mladenov, M.: Neural networks used for speech

recognition. Journal of Automatic Control **20**, 1–7 (2010)

[26] Huang, H., Castruccio, S., Genton, M.G.: Forecasting high-frequency spatio-temporal wind power with dimensionally reduced Echo State Networks. Journal of the Royal Statistical Society **71**, 449–466 (2019)

[27] Svozil, D., Kvasnicka, V., Pospichal, J.: Introduction to multi-layer feed-forward neural networks. Chemometrics and Intelligent Laboratory Systems **39**, 43–62 (1997)

[28] O'Shea, K., Nash, R.: An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458 (2015)

[29] Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. arXiv preprint arXiv:1409.2329 (2014)

[30] Lukosevicius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. Computer Science Review **3**(3), 127–149 (2009). https://doi.org/10.1016/j.cosrev.2009.03.005

[31] Rodan, A., Tino, P.: Minimum complexity Echo State Network. IEEE Transactions on Neural Networks **22**, 131–144 (2011)

[32] Hecht-Nielsen, R.: Theory of the backpropagation neural network. In: Neural Networks for Perception, pp. 65–93. Elsevier, ??? (1992)

[33] Grigoryeva, L., Ortega, J.P.: Echo State Networks are universal. Neural Networks **108**, 495–508 (2018)

[34] Todesco, E., Giovannozzi, M.: Dynamic aperture estimates and phase-space distorsions in nonlinear betatron motion. Physical review E **53**, 4067–4076 (1996)

[35] De Maria, R., et al.: SixTrack – 6D Tracking Code. Available at http://sixtrack.web.cern.ch/SixTrack/

[36] De Maria, R., Andersson, J., Berglyd Olsen, V.K., Field, L., Giovannozzi, M., Hermes, P.D., Høimyr, N., Kostoglou, S., Iadarola, G., Mcintosh, E., Mereghetti, A., Molson, J., Pellegrini, D., Persson, T., Schwinzerl, M., Maclean, E.H., Sjobak, K.N., Zacharov, I., Singh, S.: SixTrack V and runtime environment. Int. J. Mod. Phys. A **34**, 1942035–17 (2020). https://doi.org/10.1142/S0217751X19420351

[37] Corless, R.M., Gonnet, G.H., Hare, D.E.G., Jeffrey, D.J., Knuth, D.E.: On the lambertw function. Advances in Computational Mathematics **5**(1), 329–359 (1996). https://doi.org/10.1007/BF02124750

[38] Bazzani, A., Servizi, G., Todesco, E., Turchetti, G.: A Normal Form Approach to the Theory of Nonlinear Betatronic Motion. CERN Yellow Reports: Monographs. CERN, Geneva (1994). https://doi.org/10.5170/CERN-1994-002

[39] Li, D., Han, M., Wang, J.: Chaotic time series prediction based on a novel robust Echo State Network. IEEE Transactions on Neural Networks and Learning Systems **23**(5), 787–799 (2012)

[40] Hanin, B.: Which neural net architectures give rise to exploding and vanishing gradients? In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 31. Curran Associates, Inc., ??? (2018)

[41] Jaeger, H., Lukoševičius, M., Popovici, D., Siewert, U.: Optimization and applications of echo state networks with leaky-integrator neurons. Neural Networks **20**(3), 335–352 (2007)

[42] Johansen, T.A.: On Tikhonov regularization, bias and variance in nonlinear system identification. Automatica **33**, 441–446 (1997)

[43] Kawai, Y., Park, J., Asada, M.: A small-world topology enhances the echo state property and signal propagation in reservoir computing. Neural Networks **112**, 15–23 (2019)

[44] Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B.: Algorithms for hyper-parameter optimization. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 24. Curran Associates, Inc., ??? (2011)

[45] Giovannozzi, M., Maclean, E., Montanari, C.E., Valentino, G., Van der Veken, F.F.: Machine learning applied to the analysis of nonlinear beam dynamics simulations for the cern large hadron collider and its luminosity upgrade. Information **12**(2) (2021). https://doi.org/10.3390/info12020053

[46] Abada, A., Abbrescia, M., AbdusSalam, S.S., Abdyukhanov, I., Abelleira Fernandez, J., Abramov, A., Aburaia, M., Acar, A.O., Adzic, P.R., P., A., et al.: FCC–hh: The Hadron Collider: Future Circular Collider Conceptual Design Report Volume 3. Future Circular Collider. Eur. Phys. J. Spec. Top. **228**, 755–1107 (2019). https://doi.org/10.1140/epjst/e2019-900087-0

[47] Benedikt, M., Chance, A., Dalena, B., Denisov, D., Giovannozzi, M., Gutleber, J., Losito, R., Mangano, M.L., Raubenheimer, T., Riegler, W., Risselada, T., Schulte, D., Zimmermann, F.: Status and challenges of the Future Circular Hadron Collider FCC-hh. In: Proceedings of 41st International Conference on High Energy Physics — PoS(ICHEP2022), vol. 414, p. 058 (2022). https://doi.org/10.22323/1.414.0058

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