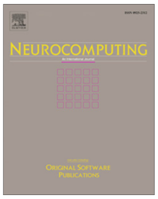
Neurocomputing 522 (2023) 214–228

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Survey paper

Parameterizing echo state networks for multi-step time series prediction

Johannes Viehweg [a,*], Karl Worthmann [b], Patrick Mäder [a,c,*]

[a] Johannes Viehweg and Patrick Mäder are with the Data-intensive Systems and Visualization Group (dAI.SY), Technische Universität Ilmenau, Helmholtzplatz 5, 98693 Ilmenau, Germany
[b] Karl Worthmann is with the Optimization-based Control group, Technische Universität Ilmenau, Weimarer Straße 25, 98693 Ilmenau, Germany
[c] Faculty of Biological Sciences, Friedrich Schiller University, Philosophenweg 16, 07743 Jena, Germany

ARTICLE INFO

Article history:
Received 16 September 2021
Revised 18 August 2022
Accepted 15 November 2022
Available online 28 November 2022

Keywords:
reservoir computing
design guidelines
hyper-parameter
dynamical system
chaotic time series
auto-regressive prediction
echo state network

ABSTRACT

Prediction of multi-dimensional time-series data, which may represent such diverse phenomena as climate changes or financial markets, remains a challenging task in view of inherent nonlinearities and non-periodic behavior. In contrast to other recurrent neural networks, echo state networks (ESNs) are attractive for (online) learning due to lower requirements w.r.t. training data and computational power. However, the randomly-generated reservoir renders the choice of suitable hyper-parameters as an open research topic. We systematically derive and exemplarily demonstrate design guidelines for the hyper-parameter optimization of ESNs. For the evaluation, we focus on the prediction of chaotic time series, an especially challenging problem in machine learning. Our findings demonstrate the power of a hyper-parameter-tuned ESN when auto-regressively predicting time series over several hundred steps. We found that ESNs' performance improved by $85.1\% - 99.8\%$ over an already wisely chosen default parameter initialization. In addition, the fluctuation range is considerably reduced such that significantly worse performance becomes very unlikely across random reservoir seeds. Moreover, we report individual findings per hyper-parameter partly contradicting common knowledge to further, help researchers when training new models.

1. Introduction

Reservoir computing [87] aggregates two machine learning methods independently and nearly contemporaneously introduced as echo state network (ESN) by Jaeger [48] and as liquid state machine (LSM) by Maass et al. [63]. Among recurrent neural networks (RNN), reservoir computing constitutes an especially lightweight approach. An inexpensive training with comparatively low data requirements allows for an efficient inference making reservoir computing specifically suited for online learning. ESNs have successfully proven their applicability in the fields of robotic navigation [8], anomaly detection [23], time-series prediction [53,103], and speech recognition [88].

The general idea of reservoir computing, also called the reservoir computing trick [18], is a neural network topology consisting of three layers: input, reservoir, and output. The input layer maps a, potentially multi-dimensional, input per time step into a high-dimensional reservoir state space. The recurrently connected reservoir layer combines this new input with the current state of the reservoir thereby utilizing a randomly-initialized weight matrix (the adjacency matrix of the reservoir's internal processing graph) and a nonlinear activation function. A purely linear output layer transforms the reservoir state into the output. Since only the weights of the output layer are adapted, training poses a linear regression problem that can be solved with substantially less computations than, e.g., a gradient descent optimization.

The ESN model comes with a number of hyper-parameters, whose selection is a non-trivial task. On the one hand, we provide a thorough overview of previous studies to substantiate this claim. On the other hand, the randomness entering reservoir computing through, e.g., the majority of the weights, further complicates the derivation of meaningful guidelines. Guidance on choosing these parameters is rarely available [28], especially in a way suitable to reduce the influence of the randomness on the results shown by Prokhorov [75]. Often, an additional optimization, e.g., an evolutionary algorithm [47,62,32] or a Bayesian optimization [101,64], is conducted. However, in our experience, this approach provides hardly any (deeper) insight into the influence of hyper-parameters and often fails in view of the highly-complex optimization problem to be solved.

In this paper, we propose a systematical procedure for searching ESNs' hyper-parameters that specifically takes the influence

[*] Corresponding author.
E-mail addresses: johannes.viehweg@tu-ilmenau.de (J. Viehweg), patrick.maeder@tu-ilmenau.de (P. Mäder).

https://doi.org/10.1016/j.neucom.2022.11.044
0925-2312/© 2022 Elsevier B.V. All rights reserved.

of randomly generated weight parameters into account. We focus on the ESN as proposed by Jaeger and his group [48,50,62] including the leakage concept, a promising ESN evolution. We apply the proposed optimization procedure on the challenging task of predicting chaotic time series, a benchmark problem in machine learning, being representative for and of interest in such diverse fields as financial analyses, weather forecasting, and climate change modeling. Forecasting of time series in general is applied for even more tasks, such as controlling [8,91], fault and signal detection [23,68], and reservoir computing even in picture restoration [27] and pattern recognition [28]. Based on our experimental results, we derive a guideline for choosing initial hyper-parameters to be used in training or to speed-up the hyper-parameter search, depending on the characteristic features of the time-series in consideration.

The contribution of our paper is threefold: Firstly, we provide an overview on previous studies applying ESNs for predicting chaotic time series with a particular focus on the choice of hyper-parameters. Secondly, we propose a systematic hyper-parameter grid-search algorithm tailored to the stochastic nature of ESNs. Thereby, we propose a method for deriving reservoir weight matrices of varying size and density from a static initially-seeded set to ensure comparability of the conducted experiments despite the stochastic model nature. Furthermore, we propose a sequential search order that minimizes interactions among the optimized parameters as much as possible. Repeating the optimization loop several times stabilizes the obtained optimization results. Thirdly, we conduct a systematic study with baseline problems in chaotic time-series forecasting to derive guidelines for practical use of the reservoir computing model. Overall, we study seven hyper-parameters across four datasets and 100 randomly-initialized ESNs. While many earlier studies solely evaluate single-step predictions, we investigate auto-regressive prediction scenarios with up to several hundred time steps to make the task at hand even more challenging. Fourth, we extensively analyze and discuss the outcome of our study. On the one hand, we highlight clear trends. On the other hand, we comment on the interplay of the hyper-parameters and their dependence on the problem in consideration. Doing so allows us to derive guidelines providing orientation on the choice of hyper-parameters for other researchers when training new models. Hereby, we fill the gap in understanding ESNs by not only optimizing the hyper-parameters but also showing the reasoning of selecting their actual values in dependence to the dataset. While not explicitly studied, our concepts may be transferable to other randomly initialized neural networks, e.g., stochastic configuration networks in the future [21,20,19].

This paper is structured as follows. In Section 2, we introduce the fundamentals of echo state networks and their training. In Section 3, we briefly overview previous work on applying ESNs for time series prediction including the choice of hyper-parameters. Then, we briefly introduce four time series datasets to be utilized in our systematic study. In Section 5, we discuss our experimental setup and the grid search procedure applied in this study and report its results in Section 6. In Section 7, we derive and discuss design guidelines for ESNs based on our findings and eventually conclude the paper in Section 8.

2. The Echo State Network

Reservoir Computing refers to the concept of mapping a model's input into a higher dimensional space, the reservoir [48]. This mapping differs from conventional RNNs, such as long short-term memory (LSTM, see [44]) and gated recurrent unit (GRU, see [25]), by training only the weights W^{out} connecting the reservoir and the output. The weights W^{in} mapping the input into the state space of the reservoir as well as those of W^r (connections edges in within the reservoir) are set fixed for each realization according to the current set hyper-parameter selection Nevertheless, using the validation data similarly to other neural network structures, an iterative training process – corresponding to a highly nonlinear and, in general, non-convex optimization – needs to be conducted. One major contribution of our work is to provide easy-to-implement guidelines for this hyper-parameter tuning and an excellent educated guess to drastically speed up the otherwise very time-consuming optimization – supposing that the time-series data of interest exhibits comparable key characteristics to one of our four representative ones. Solely the weights mapping the high-dimensional reservoir back to the lower-dimensional output layer have to be determined using methods with low computational costs [90], e.g., linear regression, support-vector machines, and multi-layer perceptrons [62,86].

Primarily, two flavors of reservoir computing have been studied in the last two decades: ESNs and LSMs. While the biologically-inspired LSMs use so-called spiking neurons and are mainly used for modeling in neuroscience, we focus on ESNs as the reservoir-computing prototype used in machine learning for time-series prediction.

2.1. Reservoir dynamics and input layer

An ESN exhibits a recurrent neural network topology consisting of three layers: input, hidden, and output, cp. Fig. 1.

The hidden state $h^{(k)}$ at time instant $k, k \in \mathbb{N}_0$ (non-negative integers), is determined by

$$h^{(k)} = \tanh\left(W^{in}x^{(k)} + W^r s^{(k-1)}\right), \qquad (1)$$

where $x^{(k)}$ represents the data vector feed into the network augmented by an additional entry, i.e. $x_1^{(k)} = 1$ holds to include a bias term. Hence, in a slight abuse of notation (N^{in} denotes the incremented input dimension to reflect this artificial augmentation), the input $x^{(k)}$ is combined with the internal state $s^{(k-1)}$ of the network using the matrices $W^{in} \in \mathbb{R}^{N^r \times N^{in}}$ and $W^r \in \mathbb{R}^{N^r \times N^r}$. The input weights W^{in} can, in general, be drawn from any random distribution, while almost consistently a uniform distribution is used [80], i.e. $w_{ij}^{(in)} \sim \mathcal{U}(a,b)$ with $a < 0 < b, (i,j) \in \{1,\ldots,N^{(res)}\} \times \{1,\ldots,N^{(in)}\}$. Furthermore, we sparsely initialize W^r, where the non-zero entries satisfy $w_{ij}^r \sim \mathcal{U}(-1,1), (i,j) \in \{1,\ldots,N^r\}^2$. The result $W^{in}x^{(k)} + W^r s^{(k-1)}$ becomes the input to the activation function, i.e.

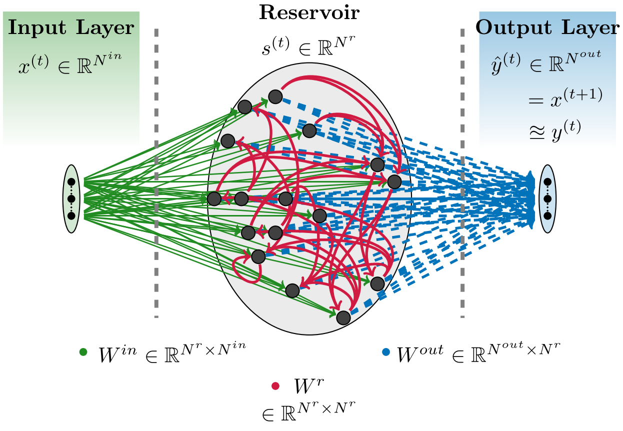

Input Layer
$x^{(t)} \in \mathbb{R}^{N^{in}}$
Reservoir
$s^{(t)} \in \mathbb{R}^{N^r}$
Output Layer
$\hat{y}^{(t)} \in \mathbb{R}^{N^{out}}$
$= x^{(t+1)}$
$\approxeq y^{(t)}$
● $W^{in} \in \mathbb{R}^{N^r \times N^{in}}$
● $W^{out} \in \mathbb{R}^{N^{out} \times N^r}$
● W^r
$\in \mathbb{R}^{N^r \times N^r}$

Fig. 1. Building blocks of an ESN: layered structure, core parameters, and their dimensionality.

an element-wise hyperbolic tangent. tanh is a typical choice and applied by the majority of related work, cp. Table 1 below. Linear activation functions have also been considered [89], mainly to facilitate the mathematical analysis. However, nonlinear activation functions consistently yield superior performance, see, e.g., Bollt [15].

The reservoir state s is updated within the hidden layer

$$s^{(t)} = \gamma s^{(t-1)} + (1-\gamma)h^{(t)}, \qquad s^{(0)} = \vec{0}, \tag{2}$$

which represents the dynamical reservoir. We consider a leakage as proposed by Jaeger et al. [50] to allow a portion of the ESN's last state to directly determine its successor state with leakage rate γ.

The matrix $W^r \in \mathbb{R}^{N^r \times N^r}$ can be interpreted as a graph structure connecting N^r nodes in graph $\mathscr{G}$. The sparse initialization is inspired by the following two ideas: (1) capturing different dynamics with a less densely connected graph [48], and (2) reducing the computational costs [60]. We measure the density d of W^r by

$$d(W^r) = \left\{ \frac{\sum_{i,j=1}^{N^r} \chi(w_{ij}^r)}{(N^r)^2} \right\} \text{with} \chi(w_{ij}^r) = \begin{cases} 1 & \text{if } w_{ij}^r \neq 0 \\ 0 & \text{otherwise} \end{cases},$$

Table 1
Hyper-parameter selection in previous ESN studies.

					ESNs' hyper-parameters				
Study	$X \sim F$	N^r	$d(W^r)$	ρ	γ	β	S_I	S_T	S_P
	Lorenz '63: ODE time-series (three dimensions)								
[57]	$\mathscr{U}$	2,000	0.02	0.9	–	10^{-6}	5,000	3,000	2,000
[58] + feedback	$\mathscr{U}$	400	0.05	1	0	–	–	2,600	1
[72]	$\mathscr{U}$	500	0.006	0.4	0	–	–	1,000	2,500
[6]	$\mathscr{U}$	800	0.0075	1.05	0	0.1	–	100	100
[81]	–	500	–	0.95	0	–	1,000	6,000	–
[55]	$\mathscr{U}$	900	0.1	0.95	0.9	10^{-10}	300	3,000	–
	Lorenz '63 (modified): ODE time-series (three dimensions)								
[40]	$\mathscr{U}$	300	0.02	0.1	0	10^{-2}	100	4,900	10,000
[39]	$\mathscr{U}$	200	0.05	0.17	0	$1.9 \cdot 10^{-11}$	5,000	5,000	–
	Mackey–Glass: DDE time-series (one dimension)								
[97]	$\mathscr{N}$	3,300	0.80	0.98	0	–	1,000	10,000	1,000
[14]	$\mathscr{U}$	100	0.25	$[0.01, 2]$	0	10^{-1}	100	147.890	1
[85]	–	50	–	0.8	–	–	5	1,000	200
[76]	–	250	–	1	0	–	1,000	2,000	–
[32]	$\mathscr{U}$	82	$[0.1, 1]$	1.59	$[0, 0.9]$	$[10^{-8}, 10^{-1}]$	–	–	1
[83]	$\mathscr{U}$	100	0.1	–	0	$2 \cdot 10^{-3}$	1,000	2,000	168
[100]	–	20	–	–	–	–	1,000	7,000	–
[55]	$\mathscr{U}$	900	0.1	1	0.9	10^{-10}	300	6,400	–
[54]	$\mathscr{U}$	200	–	0.8	0	–	–	–	–
[67]	$\mathscr{U}$	300	0.1	0.95	0	10^{-7}	–	4,000	300
[81]	–	500	–	0.85	0	–	1,000	3,000	–
	NARMA: nonlinear autoregressive moving-average time series (one dimension)								
[80]	$\mathscr{B}(\frac{1}{2}), \mathscr{U}$	100	–	0.95	0	–	100	100,000	–
[77]	$\mathscr{U}$	200	0.18	0.85	0	–	200	1,800	1
[32]	$\mathscr{U}$	174	$[0.1, 1]$	1.15	$[0, 0.9]$	$[10^{-8}, 10^{-1}]$	–	–	1
[104]	$\mathscr{N}$	300	1	0.9	0	–	–	1,000	1
[76]	–	250	–	1	0	–	5,000	5,000	–
[87]	$\mathscr{N}$	180	–	1.4	0.8	–	–	1,000	1
[46]	$\mathscr{N}$	100	–	–	0	–	–	–	1
[83]	$\mathscr{U}$	100	0.10	–	0	$2 \cdot 10^{-3}$	–	–	1
[26]	–	400	–	–	–	–	100	2,000	–
[69]	–	200	–	–	–	–	–	–	–
[35]	–	100	–	–	–	–	–	2,000	–
[78]	–	–	–	–	–	–	200	2,000	–
[100]	–	15	–	–	–	–	1,000	5,600	–
[29]	–	50	–	–	–	–	100	1,000	–
[6]	$\mathscr{U}$	800	0.0075	0.8	0	0.1	–	900	100
[30]	$\mathscr{U}$	1,000	0.01	0.9	0	–	–	2,800	–
[55]	$\mathscr{U}$	400	0.1	0.8	0.9	10^{-10}	120	2,560	–
[33] cp. appendix	$\mathscr{U}$	300	0.89	1.1	0	$[10^{-13}, 10^{2}]$	–	100,000	–
[12]	$\mathscr{U}$	200	–	–	0	10^{-5}	200	1,400	–
[54]	$\mathscr{U}$	200	–	0.8	0	–	–	–	–
	Santa Fe Laser: empirical time series (one dimension)								
[77]	$\mathscr{U}$	200	–	–	0	–	200	1,800	1
[46]	$\mathscr{N}$	100	–	–	0	–	–	–	1
[83]	$\mathscr{U}$	100	0.1	–	0	$2 \cdot 10^{-3}$	100	1,000	1
[26]	–	200	–	–	–	–	100	2,500	–
[78]	–	–	–	–	–	–	200	2,000	–
[12]	$\mathscr{U}$	200	–	–	0	10^{-3}	10	499	–

We show hyper-parameters as intervals if a publication does not explicitly state an optimum. —$X \sim F$ = universal distribution function [93].

that is, the number of edges with non-zero weight in relation to the size of the reservoir N^r squared (possible number of edges).

2.2. The echo state property

A necessary condition for a well-performing ESN is the echo state property, meaning that the current state of the reservoir sufficiently reflects the inputs up to this point in time. Jaeger [48] argues that a sufficient condition to achieve the echo state property is scaling W^r so that its highest singular value is strictly less than one. However, Jaeger also discusses the spectral radius ρ of the matrix W^r, i.e., the eigenvalue with the maximal absolute value, as a less strict condition for scaling that he found effective in extensive experimentation. We observe that scaling W^r based on the spectral radius is far more popular across previous studies applying ESNs (cp. Table 1), albeit Lukoševičius and Jaeger [62] discuss that $\rho < 1$ is neither necessary nor sufficient for achieving the echo state property.

We apply spectral radius scaling due to its almost consistent application in previous work. We determine the spectral radius ρ_o of W^r and use it to scale W^r in relation to a desired spectral radius ρ. In conclusion, this yields the scaled reservoir matrix $\rho\rho_o^{-1} \cdot W^r$ instead of W^r.

2.3. Training of an ESN: W^{out}

We consider a supervised training with data tuples $(x^{(k)}, y^{(k)})$ of input and (expected) output values $x^{(k)}$ and $y^{(k)}$, respectively. For predicting time series we can, w.l.o.g., further assume that $x^{(k+1)} = y^{(k)}$ holds for $k \in \mathbb{N}_0$. We divide our data into initialization, training, and prediction data such that we have S_I, S_T, and S_P data tuples, respectively. E.g., the data tuples $(x^{(k)}, y^{(k)}), k = 0, \ldots, S_I - 1$, are assigned to the initialization phase. The initialization phase is used to *washout* a reservoir's initial values adapting it to the given dataset. Hereby, the necessary length of this washout depends on the reservoir's memory capacity, i.e., how long previous inputs influence the reservoir's current state.

The network's prediction $\hat{y}^{(k)}$ at time k is the result of the matrix–vector multiplication $W^{out}s^{(k)}$, where $W^{out} \in \mathbb{R}^{(1+N^r)\times N^{out}}$ holds and the reservoir's state $s^{(k)}$ is augmented with a bias; again – in a slight abuse of notation – $s^{(k)}$ denotes this augmented state with $s_1^{(k)} = 1$. The goal is to determine the weights of the matrix W^{out} using the training data $(k = S_I, \ldots, S_I + S_T - 1)$ such that.

$$y^{(k)} \approx \hat{y}^{(k)} = W^{out}s^{(k)}, \quad k = S_I + S_T, \ldots, S_I + S_T + S_P - 1, \tag{3}$$

holds for the reservoir's prediction $\hat{y}^{(k)}$. We measure the model's prediction quality, i.e., the similarity between predicted and actual values, with a loss function $\mathscr{L}$. Multiple different evaluation metrics are possible depending on a model's task. Within this paper, we restrict our focus to regression problems and measure mean-squared-error (MSE) loss for the S_P predicted values by

$$\frac{1}{S_P}\sum_{k=S_I+S_T+1}^{S_I+S_T+S_P} \left\|W^{out}s^{(k)} - y^{(k)}\right\|_2^2. \tag{4}$$

Note that only the output layer's weights, i.e. the matrix W^{out}, are subject to an optimization. To this end, we employ the initialization and training data to compose the matrices

$$S = \left[s^{(S_I)}\ s^{(S_I+1)}\ \ldots\ s^{(S_I+S_T-1)}\right],$$
$$Y = \left[y^{(S_I)}\ y^{(S_I+1)}\ \ldots\ y^{(S_I+S_T-1)}\right]$$

and, then, assign the minimizer of the following optimization problem to the matrix W^{out*}, i.e.

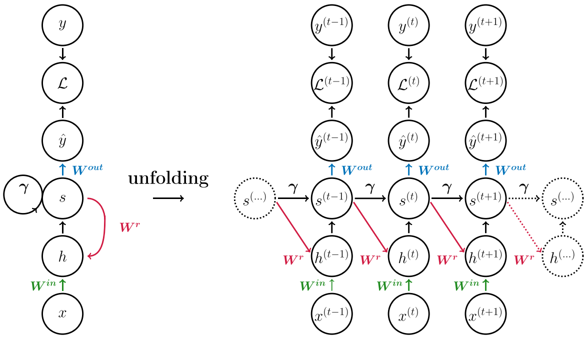


Fig. 2. ESN's computational graph: folded (left) and unfolded over time (right).

$$W^{out*} = argmin_{W^{out}}\ \frac{1}{S_T}\left\|W^{out}S - Y\right\|_2^2. \tag{5}$$

The procedure is illustrated in Fig. 2.

Eq. 5 poses a linear regression problem that can, e.g., be solved by multiplication with the inverse or pseudo-inverse. Lukoševičius et al. discuss the instability of the inverse and possible over-fitting of the pseudo-inverse as challenges in the optimization [59]. Aiming for a more robust and better generalizing solution, we include a Tikhonov regularization (aka ridge regression) to compute W^{out*} as

$$W^{out} = YS^T\left(SS^T + \beta I_{N^r}\right)^{-1}, \tag{6}$$

with β being the regularization coefficient and I_{N^r} being a unity matrix of size $N^r \times N^r$. The additional regularization term is used to penalize large values of W^{out} and, thus, to attain smaller, more balanced values in the solution of the optimization problem.

ESNs are especially suitable for sequential tasks, such as control [18], sequential classification [86], and time series analysis and prediction [40]. We emphasize that we measure the prediction performance in an auto-regressive manner, i.e. $\hat{x}^{(k+1)} = \hat{y}^{(k)}, k = S_I + S_T, \ldots, S_I + S_T + S_P - 1$, is fed back into the reservoir.

3. ESNs' Hyper-parameters

In this section, we review related work on time series forecasting and related tasks using ESN models. Thereby, we specifically focus on the hyper-parameter selection. Table 1 shows the hyper-parameters introduced in the previous section as columns and lists their selection across previous studies in rows grouped by dataset. We restrict our focus to five prominently studied data sets, which will be outlined in detail in Section 4.

Weight initialization. Given that the reservoir matrix is interpreted as an adjacency matrix of a graph $\mathscr{G}$, authors study whether an explicitly constructed graph topology w.r.t. a given dataset (aka weight initialization) may yield better prediction results. Rodriguez et al. [79] study memory capacity in relation to explicitly constructed reservoir modularity and do not observe significant difference to randomly initialized reservoirs for non-linear activated ESNs. Cernansky and Makula [22] observe decreasing performance while explicitly modelling a feed-forward neural network topology within an ESN's reservoir. Similarly, Rodan and Tiňo [77] did not observe a consistent effect when comparing a variety of deterministic constructed reservoirs to randomly initialized reservoirs. Consequently, we observe that a vast majority of previous studies initializes W^{in} and W^r randomly, mainly using a uniform distribution. Exceptions are four studies that draw from a normal distribution and argue that this yields even more nonlinear reservoir dynamics [97] as well as one study drawing W^{in} from a Bernoulli distribution with weights 0.1 and −0.1. We observe that studies typically use a zero-centered uniform distribution, e.g., $\mathscr{U}(-1,1)$, that is sometimes scaled to adapt to non-normalized datasets. Given that studies apply a fairly consistent initialization of W^{in} and W^{r}, we derived our guidelines for choosing hyper-parameters assuming no prior knowledge about the meta statistics of the time series under consideration. However, hyper-parameter optimization based on prior knowledge, e.g., on the meta statistics, is a very interesting avenue for future research. Here, Cao et al.'s [20] results might be of great help for conducting this task w.r.t. ESNs.

Size of the reservoir N^r.

All studies report the utilized size of the reservoir N^r, but exhibit a large variation w.r.t. its selection. Furthermore, while in theory the low computational cost of ESNs allows for a large reservoir, the majority of studies applies comparably small reservoirs. This observation contradicts Lukoševičius' [59] proposition for using the largest value for N^r, which is feasible from a computational point of view. A reason might be that a small reservoir yields satisfactory results on the trained benchmark problems. However, a small reservoir size may also indicate that a large size did in fact not yield better performance. This hypothesis is supported by a study observing that for classification tasks the largest reservoir did not yield best results [82]. Koryakin et al. [52] observe similar results when choosing a reservoir size below the computable maximum. Partly contradicting this observation, Haluszczynski et al. [39] observe that ESNs' predictive capabilities decreased in case of a shrinking reservoir size. We argue that theses observations motivate a further study of this hyper-parameter.

Weight density $d(W^r)$. Roughly half of the studies report the selected density $d(W^r)$ of the reservoir. We observe a wide range of values between 0.006 and 1 with a trend to sparser reservoirs. Several studies choose density as a number of average connections per node in the reservoir when interpreted as adjacency matrix [40,39,58,72,6]. Song and Feng [81] propose choosing density from $[0.01, 0.2]$, albeit without further justification. In conclusion, we found no discussion on how this parameter should be selected depending on the characteristics of a dataset or in relation to the size of the reservoir. In contrast, we did not further explore density $d(W^{in})$ since former studies consistently observed matrices of full rank to yield better performing models [19] which is represented in this work by the use of dense random matrices. Other approaches to generate the weight matrices were studied in [74] for the task of classification and compressed sensing.

Spectral radius ρ. Spectral radius ρ is reported by three quarter of the studies. We observe a large variety of selected values, even for the same problem. However, studies tend to favor higher values between 0.8 and 1. Three studies even chose a spectral radius beyond 1. We question the optimality of the chosen parameter given its wide variety for training the same problem. The influence of the spectral radius is also demonstrated in [74]. The authors argue its necessity as a scaling parameter for W^r for their tested cases of classification and compressed sensing.

Leakage rate γ. A surprising observation is the rare application of the leakage concept. Merely three studies consider the concept. However, Ferreira et al. [32] propose the concept in a different form making the leakage an additional input to the non-linear activation (cp. Eq. 1) rather than the traditional design of letting it bypass the non-linearity (cp. Eq. 2). Only five studies [58,87,26,55,32] considers a leakage as introduced in Eq. 2 and discuss it with mentioning of the used value, though, in case of [58] only for one of their experiments with the Kuramoto–Sivashinsky system not shown in Table 1. We argue that the leakage concept deserves more attention and study. Other studies used the concept of a leakage rate but did not discuss the used value, e.g. in the case of proposing a genetic algorithm for optimization as shown in [26].

Regularization coefficient β. Eleven studies apply L2-regularization (aka ridge regression) with a regularization coefficient β influencing its strength chosen between zero and one. All studied benchmark problems are synthetic and approximated from mathematical models except for the experimental Santa Fe laser dataset. This means that training as well as testing data stem from the exact same distribution and are not influenced noise or context shifts being a potential explanation for the overall "mild" regularization applied by previous studies. Exceptions are studies adding artificial noise [49,58,39,46,26,33,12] and those using the Santa Fe laser dataset containing natural measurement noise. Other studies as [97] added noise to the state of the reservoir.

Initialization length (aka washout) S_I. Seventeen previous studies report the length S_I of the utilized washout. Among those, we observe a large variety even when training for the same task. An optimization based on the evolutionary algorithm Differential Evolution by Otte et al. [71] investigated the optimal length of S_I showing an influence on the generalizability of the ESN.

Training length S_T. The length of the training interval substantially varies across studies even when training on the same dataset. We do not observe a correlation between a chosen reservoir size N^r and training length as proposed by Lukoševičius [59], who argues that S_T should be chosen significantly larger than N^r.

Prediction length S_P. Mostly, single-step predictions have been chosen for the evaluation of trained models. We argue that training and evaluating a model that performs well in predicting one single step succeeding an initialization sequence is almost meaningless when the actual task to solve is predicting multi-step time series. However, seven studies evaluate their methods in multi-step, i.e., auto-regressive, evaluations up to the training length S_T. Three of these studies further qualitatively discuss their achieved prediction performance. Thereby, studies found that they could predict $\geqslant 110$ [72], $\leqslant 400$ [40], and $\leqslant 500$ [57] time steps with an "acceptable" accuracy. We argue that only the evaluation of multi-step predictions can truly assess the performance of an ESN trained for predicting time series with single-step predictions being an implicitly included use case.

Prediction performance. We decided not to include models' prediction performance in Table 1 since we found experiments hardly comparable. More specifically, even studies that aim to predict the same system or dataset vary considerably in parameters of the system, such as predicted dimensions and time step length, in hyper-parameters of the training process and model, and in reported evaluation metrics.

Conclusions for our study. Based on this survey of previous work studying ESNs for time series analysis and prediction, we argue that a systematic evaluation of ESNs' hyper-parameters is required to get a better understanding of their influence on prediction performance, their relationship on the trained task and their interplay. We found the hyper-parameters weight initialization and activation function almost consistently used across previous studies and decided to not explicitly explore them in our study.

4. Chaotic Time-series data

We study the influence of hyper-parameter choice on an ESN's prediction accuracy with a variety of benchmark problems in time series prediction (Fig. 3). Specifically, we implicitly assume that the time series is generated by a dynamical system iteratively defined by the autonomous system dynamics

$$x^{(k+1)} = f(x^{(k)}, x^{(k-1)}, \dots, x^{(k-\bar{k})}), \qquad x(0) = x^{(0)} \tag{7}$$

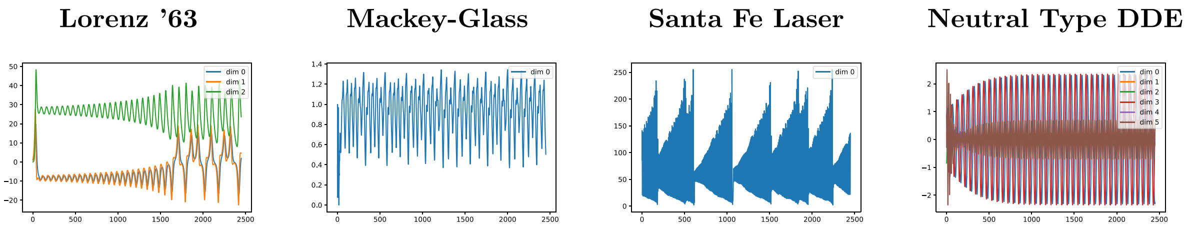


Fig. 3. Overview of the four studied time series datasets with 2,450 time steps depicted per dataset.

without an additional driving signal from an external source [38]. Here, $x^{(k)}$ and $x^{(0)}$ stand for the state at time $k, k \in \mathbb{N}$, and the initial time $k = 0$, resp. For systems without delay, $\bar{k} = 0$ holds. Otherwise, $\bar{k}$ depends on the time delay. Overall, we consider multiple chaotic systems to cover a wide range of potential, and quite different application areas. The time-series data of the first three examples are obtained from numerical approximations of continuous-time systems governed by an ordinary differential equations (ODE), i.e. the Lorenz system [56], a delayed differential equations (DDE), i.e., the Mackey–Glass equation, and a neutral type DDE, while the Santa Fe laser dataset stems from experimental data. Throughout this section, each parameter value is denoted by $\bar{\cdot}$ in order to indicate that these are no hyper-parameters used in the reservoir computing model.

Since the first three examples are continuous-time systems, we conduct a numerical approximation in order to generate the respective time-series data. To this end, we approximate the Lorenz system with the Livermore solver for ODEs with automatic mode selection (LSODA), see [1] for details, and generate training as well as test data with a time step of width $\Delta t = 0.01$. For the Mackey–Glass and the neutral DDE, we use a time steps of width $\Delta t = 1$ and $\Delta t = 0.1$, respectively, and use Ansmann's DDE solver [7]. In conclusion, we obtain a time-series dataset $x \in \mathbb{R}^{(S_I+S_T+S_P)\times(N^{in}-1)}$ where the k-th row contains the data at time step $(k-1)\Delta t$ for each system. Furthermore, we conduct a data preprocessing for the reservoir computing model, i.e. the i-th component of each dataset are normalized as

$$x_i = \frac{(x_i - \text{mean}(x))}{\text{std}(x)} \qquad \forall i \in [0 : S_I + S_T + S_P], \tag{8}$$

where $[0 : S_I + S_T + S_P]$ denotes the non-negative integers less or equal $S_I + S_T + S_P$, $\text{mean}(\cdot)$ and $\text{std}(\cdot)$ being the mean value and the standard deviation over all time steps.

4.1. The Lorenz system

Lorenz is a well-known three-dimensional system described by the ODE [56]

$$\begin{aligned}
\dot{x}_1(t) &= \bar{\sigma}(x_3(t) - x_1(t)) \\
\dot{x}_2(t) &= x_1(t)(\bar{\rho} - x_3(t)) - x_2(t) \\
\dot{x}_3(t) &= x_1(t)x_2(t) - \bar{\beta}x_3(t)
\end{aligned} \tag{9}$$

with $\bar{\sigma} = 10, \bar{\rho} = 28$, and $\bar{\beta} = \frac{8}{3}$ and initial value $x^{(0)} = (1\ 1\ 0)^\top$.

4.2. The Mackey–Glass equation

We argue that the recurrent nature of an ESN should well match the behavior of DDEs and therefore enclose the Mackey–Glass equation. From the two equations described by Mackey and Glass [65], we consider the one exposing chaotic behavior, i.e.

$$\dot{x}(t) = \frac{\bar{\beta}\bar{\theta}^{\bar{n}} \cdot x(t - \bar{\tau})}{\bar{\theta}^{\bar{n}} + x(t - \bar{\tau})^{\bar{n}}} - \bar{\gamma}x(t), \tag{10}$$

for $\bar{\beta} = 0.2, \bar{\theta} = 1, \bar{n} = 10, \bar{\gamma} = 0.1$, and $\bar{\tau} = 17$ according to [37]. While $x(t)$ is the state at the current time step, $x(t - \bar{\tau})$ represents the (delayed) state at time $t - \bar{\tau}$. For the initialization, $\bar{\tau}$ randomly drawn values $\{x(0), \ldots, x(\bar{\tau} - 1)\} \sim \mathcal{U}(0, 1)$ are used. This was done to achieve a trajectory of the attractor analogous to the results in [37].

4.3. A neutral type DDE

Auvray et al. [9] propose a mathematical model describing nonlinear coupled oscillators being an instance of a neutral type DDE. The dimensionless system is denoted as.

$$\ddot{x}_n(t) + \bar{\epsilon}_n\bar{\nu}_n\dot{x}_n(t) + \bar{\nu}_n^2 x_n(t) = \bar{\mu}_n \frac{\mathrm{d}^2 \tanh(x(t - \bar{\tau}) - \bar{x}_0)}{\mathrm{d}t^2} - \bar{\zeta}_n \frac{\mathrm{d}|x(t)|^2}{\mathrm{d}t} \tag{11}$$

with $x(t) = \sum_n x_n(t)$. We use the neutral dataset available at [3] within our experiments and refer to this reference for the initialization and configuration of the neutral type DDE consisting of 14 parameter values in total.

4.4. The Santa Fe laser dataset

We also include the Santa Fe laser dataset, an experimental dataset, which is a well-known benchmark for time series prediction. The dataset consists of measured data from a NH_3 laser experiment without noise. Weiss et al. [95] discuss such data as being comparable to the chaotic behavior of the Lorenz system and explicitly demonstrate its chaoticity. We retrieved and used the dataset as available at [2].

5. Methodology

Our goal is to systematically investigate the impact of the seven hyper-parameters θ defined by

$$\theta := \{S_I, N^r, d(W^r), \rho, \gamma, \beta, S_T\} \tag{12}$$

introduced in Section 3. To this end, we train a reservoir-computing model to assess its performance w.r.t. prediction based on the four datasets discussed in Section 4. Contrary to many previous studies, that often solely evaluate their approaches in a single-step prediction scenario (cp. Section 3), we study the more realistic autoregressive, i.e. multi-step, prediction case. We schematically depicted the procedure in 4.

5.1. Model inference and performance measures

After training, we use each model to predict a number of S_P time steps, where the value of S_P depends on the dataset. For the Lorenz system we predict four Lyapunov times, i.e., SP = 444 [41], while for the Mackey–Glass equation we predict two Lyapunov times, i.e., Sp = 286 [31]. Lyapunov times are dataset specific and refer to the timescale on which the underlying dynamical system is

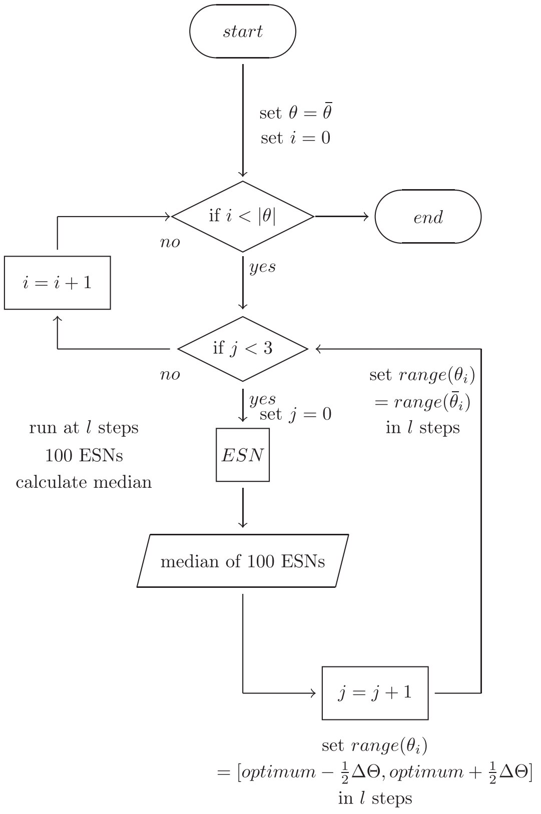


Fig. 4. Schematic overview of the optimization procedure.

chaotic. Hence, this should allow for more comparable results. For the Santa Fe laser dataset, we set $S_P = 100$ time steps being the prediction horizon of the original competition while we choose $S_P = 500$ for the neutral type equation (cp. Table 2). The interval S_P starts at $S_T + 1$. All sequences are completely unseen and have not been part of any training or optimization procedure. (See 5).

We measure the prediction performance of a trained model using the normalized root mean square error (NRMSE)

$$NRMSE = \frac{\sqrt{\frac{1}{S_P}\sum_{i=1}^{S_P}\|y^{(i)} - \hat{y}^{(i)}\|^2}}{\|\text{mean}(y)\|^2}, \tag{13}$$

which quantitatively assesses the fit between the predicted and actual trajectory while penalizing outliers more heavily; $\text{mean}(y) = \text{mean}_{S_P}(y)$ stands for $S_P^{-1}\sum_{i=1}^{S_P} y_j^{(i)}$, i.e. the mean value calculated over the test interval of S_P time steps, and $\|\cdot\|$ for the Euclidean norm. Moreover, we compute the R^2-score

$$R^2 = 1 - \frac{\sum_{i=1}^{S_P}(\hat{y}^{(i)} - y^{(i)})^2}{\sum_{i=0}^{S_P}\|(y^{(i)} - \|\text{mean}(y)\|^1)\|^2} \tag{14}$$

Table 2
Overview of length of S_P for all tested systems.

System	S_P
Lorenz'63	444
Mackey–Glass	286
SantaFe Laser	100
Neutral DDE	500

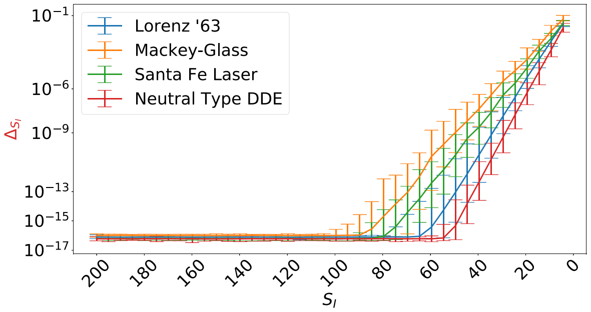


Fig. 5. Difference Δ_{S_I}, cp. Eq. 16, between reservoir state at first training step $s^{(0)}$ when initializing with a growing $S_I \in [0; 1,000]$ and a baseline of $S_I = 1,000$ with median over 100 reservoirs per dataset.

to additionally measure the correlation between the target and predicted values in S_P, cp. [5,4].

5.2. Parameter search procedure and model initialization

We conduct each experiment (training process) with 100 randomly drawn initializations of input and reservoir matrices W^{in} and W^r (realizations) and analyse the results by reporting the median to increase reliability of the inferred guidelines for the design of reservoir-computing models, i.e. w.r.t. the choice of suitable hyper-parameters. We reuse the same 100 initializations per experiment to gain maximum comparability across experimental runs. In particular, we propose the following procedure to deal with different reservoir sizes N^r: Firstly, we draw our 100 samples of the randomly initialized reservoir matrix W^r at a size of $N^r = 16,394$ from a uniform distribution in the interval $[0,1]$. Then, we iteratively apply average pooling with a window size of two to construct matrices of smaller size N^r to be applied across experiments in order to increase comparability along the conducted hyper-parameter study. After pooling, we set $(1 - d(W^r)) \cdot N^{2\times r}$ randomly chosen weights of the reservoir matrix W^r to zero. It is important to perform this step only after pooling to the desired reservoir size, otherwise both operations would interfere making a reservoir's weights deviating from the initially seeded distribution. We then randomly set 50% of the values per reservoir negative aiming for a zero-centered input distribution to the ESN's activation function tanh. Eventually, we scale W^r with the given spectral radius ρ.

Searching for an ESN's best performing hyper-parameters poses a highly non-linear and non-convex optimization problem and we are aware of the fact that our proposed search procedure will likely not terminate in a global optimum. However, we argue and demonstrate that we indeed find increasingly well-performing configurations and justify each step of the procedure. More specifically, we propose an incremental and iterative search procedure consisting of 34,300 trained ESNs (resulting from 100 repetitions over seven parameters analyzed in three search iterations) for exploring each hyper-parameter of θ individually with initial value

$$\begin{aligned}\bar{\theta} &= \{S_I, N^r, d(W^r), \rho, \gamma, \beta, S_T\} \\ &= \{300, 256, 1, 0.995, 0, 10^{-6}, 2000\}.\end{aligned} \tag{15}$$

To this end, we proceed in a sequential manner in the order defined in θ, (cp. Eq. 12). The only exception is that we consider the two hyper-parameters N^r and $d(W^r)$ together. The initial values given by Eq. (15) are inspired by the parametrization of previous studies and are expected to yield satisfying performance.

Let us briefly outline our approach in more detail: In an initial experiment, we study the length of the initialization interval S_I.

Next, since we expect larger and sparser reservoirs to yield better performing models, we study N^r and $d(W^r)$ in combination motivated by the fact that a reservoir, which is large and sparser at the same time, requires an equal amount of multiplications as a more dense and smaller one. An initial spectral radius of $\rho = 0.995$ follows Jaeger's own choice and his claimed requirement of selecting a value close to one [50]. Hence, a satisfactory performance is expected for this choice. We argue that the reservoir is the ESN's core concept and therefore optimize its size, density and weight scaling at the initial step of our (sequential) search procedure. In contrast, we consider the leakage concept a kind of residual connection [42] or attention mechanism [11,84] that gives the output layer direct access to the ESN's previous state while bypassing the reservoir's non-linear and possibly saturating activation. In order to stimulate maximum utilization of the reservoir, we only activate and optimize the concept once we have determined the best performing reservoir. Eventually, based on the observation that previous studies often consider small regularization coefficients, we apply it throughout the search but only optimize it towards the end of the search procedure, cp. Table 1. Based on a pre-study, we train each model for $S_T = 2,000$ time steps being a compromise between computational cost and model performance and study this choice at the end of the experimental series.

We iteratively repeat the grid search three times per parameter in the order described above, thereby, adjusting the search space in two directions: (1) when finding that the best performing parameter value lies within the search interval, we increase search resolution, i.e., centering the new search interval around this best performing value from the previous iteration and reducing interval width to a search step size $\Delta\Theta$ of the previous iteration with l values to test dependent on the searched hyper-parameter θ_i; and (2) when finding that the best performing parameter value lies at the boundary of the search range, we increase the search range in this direction if plausible. Within the grid search we explore three parameters $d(W^r)$, ρ, and γ on a linear scale spanning a fixed interval and two parameters N^r and β on a logarithmic scale aiming to cover a larger search interval, i.e., towards large reservoir sizes and small regularization coefficients. Once we have determined an optimal value for one hyper-parameter and dataset combination, we use this across all succeeding experiments.

6. Results and Discussion

In this section, we report our results per hyper-parameter and discuss their implications in the order defined by θ. In an initial experiment, we study meaningful washout lengths S_I per dataset. Next, we systematically explore $N^r, d(W^r), \rho, \gamma$, and β and show the progress of the optimization process in Table 3 where the order of columns from left to right corresponds to the order in which the hyper-parameters were optimized. Gray-shaded cells refer to the start value while green-shaded cells refer to the optimized value per parameter and dataset. Additionally, Table 4 shows plots of results across the relevant parts of the search interval of each combination of hyper-parameter and dataset. Both tables report the median across all 100 trained ESNs on the independent test set in terms of NRMSE and R^2-score, see Eqs. (13) and (14) resp. Finally, we study the length of the training interval S_T in an additional study.

6.1. Washout S_I

We explore the influence of the washout S_I in an initial experiment where we aim to answer the question: When does a previous input $x^{(-S_I)}$ become insignificant for the current state $s^{(0)}$ of an ESN's reservoir? We argue that this point determines a maximum meaningful washout since any earlier inputs would be "forgotten" at this point. We explore washouts up to $S_I = 1,000$ and initialize ESNs per dataset with an increasing $S_I \in [0;1,000]$. Similarly to the other experiments, each datapoint is the result of 100 ESNs initialized with default hyper-parameters $\bar{\theta}$ and using the same reservoir seeds (cp. Section 5). Based on how previous studies chose S_I, we use $s^{(0)}_{S_I=1,000}$, i.e., the reservoir state gained by a washout of 1,000 inputs, as a baseline and compare it to $s^{(0)}_{S_I=0}, \ldots, s^{(0)}_{S_I=999}$ measured as L_1-norm of their differences:

$$\Delta_{S_I} = \|s^{(0)}_{S_I=1,000} - s^{(0)}_{S_I=j}\|_1; \quad j = 0\ldots999.$$

We observe that beyond a washout of $S_I = 59$ (Lorenz system), $S_I = 82$ (Mackey–Glass equation), $S_I = 74$ (Santa Fe laser), and $S_I = 49$ (neutral type DDE) initializing inputs, differences of the initialized ESN's reservoir state $s^{(0)}$ at the time when the actual training would start become insignificant, i.e., smaller than 10^{-15}. We argue that a longer washout would not be relevant for the behavior of the trained ESN. This is an interesting observation since we found that the majority of previous studies uses substantially longer washouts potentially wasting valuable training data (cp. Section 2).

6.2. Size of the reservoir N^r

We explore $N^r \in \{256, 512, 1024, 2048, 4096\}$ (cp. Section 5.2). For the Lorenz system, the start configuration $\bar{\theta}$ yields a NRMSE of 0.66. This NRMSE decreases for an enlarged and dense reservoir of $N^r = 512$ to 0.45 being an improvement of 32.1%. Increasing the reservoir size further does not yield a better NRMSE when remaining at a dense reservoir. Since we consider reservoir size and density together, we found that the actual best performing configuration lies at $N^r = 2,048$ given that density is reduced. For the Mackey–Glass equation, the start configuration yields a NRMSE of 15.71. We observe a NRMSE of 0.44 being a decrease by 97.2% when enlarging the reservoir to $N^r = 2,048$ and a further enlarged reservoir with $N^r = 4,096$ would not yield better performance in despite the chosen density. For the Santa Fe laser dataset, we observe a NRMSE of 12.45 for the start $N^r = 256$, which can be decreased to 2.03 by 83.7% when increasing the reservoir to $N^r = 1,024$. For the neutral type DDE, the default configuration yields a NRMSE of 0.37 which we also identify as the best performing size for this dataset. A reservoir enlarged to $N^r = 512$ already results in a NRMSE of 0.60, being an increase of 60.2%.

We explore reservoir size on a coarse grained five step logarithmic scale, being the result of aiming for comparable reservoirs across all experiments. Therefore, we have to assume the best performing reservoir size laying at or in between the sizes that we evaluated. We observe $32 - 100\%$ performance gain by choosing a more appropriate reservoir size. This size is dataset-specific, suggesting that characteristics of a dataset impact optimal size. Contrary to previous studies [59,77], we found larger reservoirs not to continuously perform better. Our findings suggest that each system has an optimal reservoir size, where a further enlarged reservoir worsens results. Koryakin et al. [52] observe similar results for smaller than maximal possible reservoirs and thereby justify our observations.

6.3. Weight density $d(W^r)$

We explore reservoirs' weight density over the entire possible interval of $d(W^r) \in [0,1]$. For the Lorenz system, the start density $d(W^r) = 1$ at best performing reservoir size $N^r = 2,048$ yields a NRMSE of 0.57. This NRMSE shrinks to 0.06, a reduction of

Table 3
Individual results of the parameter search procedure. Grey-shaded cells refer to initial default parameter, while green-shaded cells show the discovered best performing parameter.

Search progress	Best parameter configuration					Model performance	
	N^r	$d(W^r)$	ρ	γ	β	$NRMSE\downarrow$	$R^2\uparrow$
	Time-series: Lorenz '63 ODE						
–	256	1	0.995	0	10^{-6}	0.666	0.162 482 66
$[N^r, d(W^r)]$	2048	0	0.995	0	10^{-6}	0.060	0.993 479 97
–	2048	0	0.995	0	10^{-6}	0.060	0.993 479 97
γ	2048	0	–	0.88	10^{-6}	0.006	0.999 930 23
β	2048	0	–	0.88	$1.1\cdot10^{-6}$	0.006	0.999 931 70
	Time-series: Mackey-Glass DDE						
–	256	1	0.995	0	10^{-6}	15.710	0.093 977 12
$[N^r, d(W^r)]$	2048	0.44	0.995	0	10^{-6}	0.341	0.999 573 07
ρ	2048	0.44	1.406	0	10^{-6}	0.190	0.999 868 29
γ	2048	0.44	1.406	0.68	10^{-6}	0.026	0.999 997 48
β	2048	0.44	1.406	0.68	$6.0\cdot10^{-7}$	0.025	0.999 997 64
	Time-series: Santa Fe Laser experimental dataset						
–	256	1	0.995	0	10^{-6}	12.453	0.794 244 16
$[N^r, d(W^r)]$	1024	0.84	0.995	0	10^{-6}	1.756	0.995 907 75
ρ	1024	0.84	0.906	0	10^{-6}	1.488	0.997 060 77
γ	1024	0.84	0.906	0.41	10^{-6}	1.178	0.998 158 02
β	1024	0.84	0.906	0.41	$8.1\cdot10^{-7}$	1.166	0.998 198 06
	Time-series: Neutral Type DDE						
–	256	1	0.995	0	10^{-6}	0.374	0.999 997 89
$[N^r, d(W^r)]$	256	0.91	0.995	0	10^{-6}	0.339	0.999 998 54
ρ	256	0.91	0.995	0	10^{-6}	0.339	0.999 998 54
γ	256	0.91	0.995	0.84	10^{-6}	0.063	0.999 999 78
β	256	0.91	0.995	0.84	$8.3\cdot10^{-7}$	0.056	0.999 999 79

89.5%, by choosing a density of $d(W^r) = 0$. This leads to the reservoir state being a mapping of the input in the higher dimensional state space without influence of any former states. For the Mackey–Glass equation, we observe a decrease of 22.1% in NRMSE, from 0.44 to 0.34, by choosing a density of $d(W^r) = 0.44$ at $N^r = 2,048$. For the Santa Fe laser dataset, NRMSE can be reduced by 13.3%, from 2.03 to 1.76, when choosing a density of $d(W^r) = 0.84$ at $N^r = 1,024$. For the neutral type DDE, the default density yields a NRMSE of 0.37 for the best performing reservoir size $N^r = 256$. We observe the lowest NRMSE at a density of $d(W^r) = 0.91$ yielding a NRMSE of 0.34 and being a reduction of 9.41%.

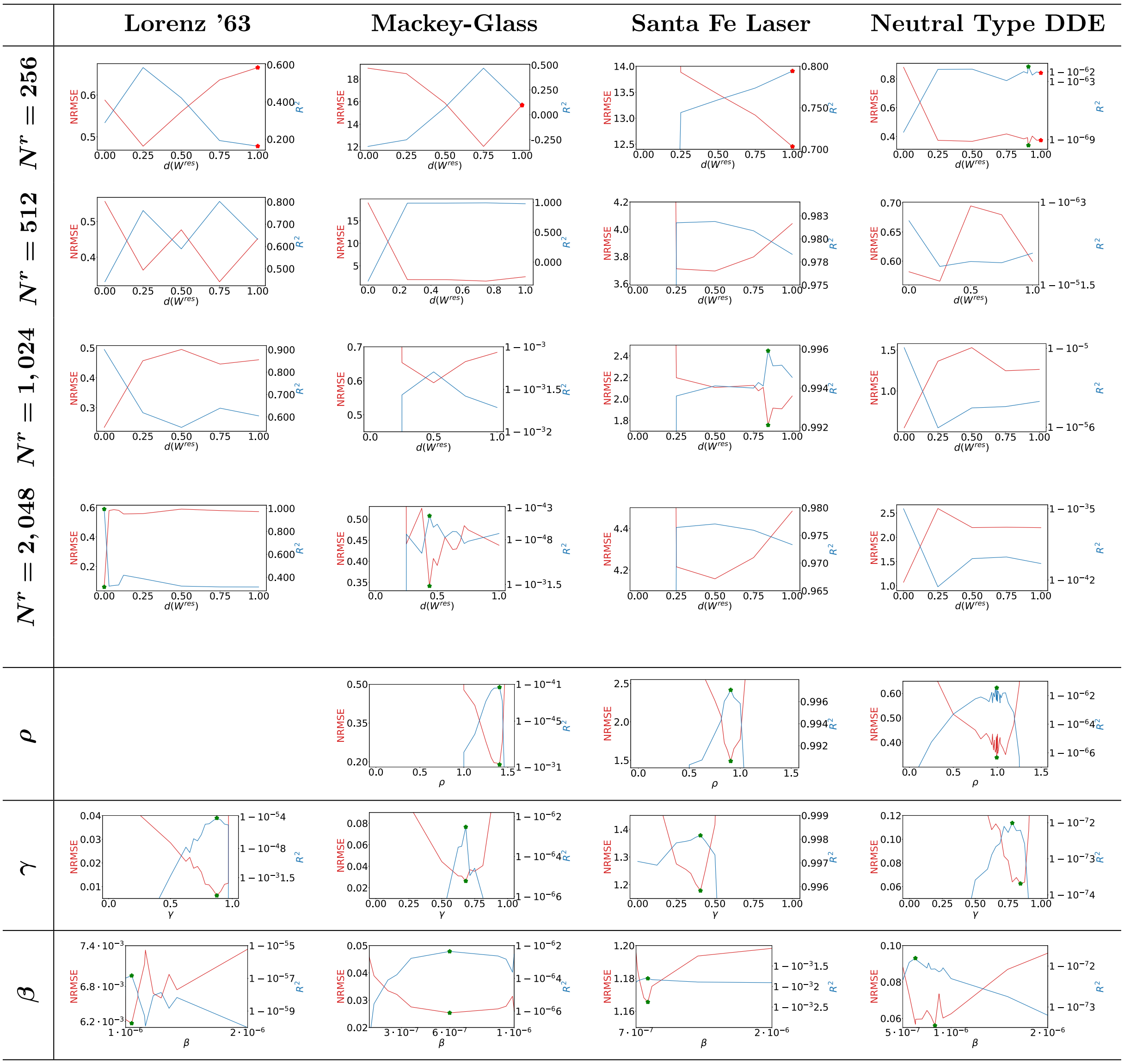
We observe $9 - 90\%$ performance gain by choosing an appropriate weight density. All identified density optima are less than 100%, with the Lorenz system and the neutral type DDE requiring the lowest and highest density respectively. We observe weight density to be indeed interwoven with reservoir size as suspected when designing our parameter search, e.g., we would have selected different optimal reservoir size and density for the Lorenz system when optimizing both sequentially. Our results contradict earlier assumptions, generally suggesting sparse matrices with a density of $d(W^r) = \frac{10}{N^r}$ [59]. Surprisingly, the Lorenz system benefits most from an empty reservoir, meaning that an input value is solely projected into a high-dimensional space of size N^r, while the recurrent memory of old reservoir states becomes entirely "deactivated". This observation has not been reported before, but several previous studies parametrize their ESNs with very sparse densities when predicting the Lorenz system, e.g. [72,99,40]. Though often used in ESN evaluations (cp. Table 1), our findings suggest that the Lorenz system is not an ideal benchmark for evaluating recurrent ESNs. An alternative would be using so-called *delay embeddings*, see, e.g. Brunton et al. [16], which we have not pursued due to comparability to the majority of previously conducted studies based on the Lorenz system. Another interesting observation is the rather high density discovered for the neutral type DDE. While for the other three datasets, we find that model performance improves when moving from a smaller very dense reservoir weight matrix to a larger more sparse matrix, we seem not to observe this behavior for the neutral type DDE. We argue that for the neutral type DDE, as the least complex dataset in our study, the increase from $N^r = 256$ to $N^r = 512$ might be too coarse to observe a similar behavior as for the other datasets and that the true best performing reservoir size may lay in between. Our findings are somewhat contrary to previous guidelines, e.g., [59], that suggest selecting a density based on the size of the reservoir as $d(W^r) = \frac{w}{N^r}$ where w is a fixed number. This guideline implies a linear relationship between the decrease of density and the increase of the size of the reservoir. Our study shows that an upper boundary on the optimal reservoir size exists and it suggest that the combination of optimal reservoir size and density are dataset-specific.

6.4. Spectral radius ρ

We explore weight scaling via the spectral radius $\rho \in [0.0, 1.5]$. For the Lorenz system, weight scaling is not applicable since we discovered the best performing weight density at $d(W^r) = 0$, i.e., a weight matrix solely consisting of zeros. For the Mackey–Glass equation, we observe a decrease of 44.5% in NRMSE from 0.34 to 0.19 by choosing a spectral radius of $\rho = 1.41$ rather than the initial value $\rho\prime = 0.995$. For the Santa Fe laser dataset, the NRMSE shrinks by 15.3% from 1.76 to 1.49 by choosing a spectral radius of $\rho = 0.906$. For the neutral type DDE, the default spectral radius

Table 4

Results of the parameter search across the four studied datasets and aggregated as median over 100 trained models per hyper-parametrization and evaluated in terms of test set NRMSE (red) and R^2 (blue). Green stars refer to the best performing value per hyper-parameter and dataset while red stars refer to the performance of the start parametrization $\bar{\theta}$.



of $\rho\prime = 0.995$ yields a NRMSE of 0.34 being also the best performing spectral radius.

We observe $0 - 45\%$ performance gain by varying the spectral radius by choosing a dataset-specific one in comparison to the initial value given by $\bar{\theta}$, which had chosen based on Jaeger's suggestion [48]. Jaeger pointed out that $\rho < 1$ is neither necessary nor sufficient, but his own configuration $\rho = 0.995$ yields satisfactory results in practice – at least for a lazy figure eight time series prediction and Japanese vowel time series classification [50]. The discovered best spectral radii for the Santa Fe laser dataset and the neutral type DDE almost perfectly underline this rule. However, we also found that the Mackey–Glass equation benefits from a spectral radius substantially beyond one, i.e., $\rho = 1.41$, thereby contradicting Jaeger's rule of thumb. We hypothesize that the importance of earlier states for the prediction of this DDE explains the more emphasized use of the reservoir. This observation seems to be in contrast to Goodfellow et al. [38] that formally derive $\rho = 1$ as an upper bound. However, they restrict their focus to linear activation functions that are of little practical relevance. We argue that the enclosing nonlinearity, i.e., tanh($\cdot$), prevents an exponential growing of the reservoir state and that the zero-centered weight seeding prevents a saturation of the nonlinearity. Our finding is supported by Ferreira et al.'s study [32] that also reports $\rho > 1$ as beneficial for predicting the Mackey–Glass equation. Since we found in the previous step that an ESN predicts the Lorenz system best when neglecting the reservoir at all, a spectral radius has no effect for this system. Haluszczynski and Räth [40] report that their lowest tested $\rho = 0.1$ performed best for the Lorenz system, which has a similar effect as using an empty reservoir. This dependence of the selection of ρ on the input data is further encouraged by the findings in [14].

6.5. Leakage rate γ

We explore leakage rate $\gamma \in [0, 1.0]$. For the Lorenz system, the start leakage rate of $\gamma\prime = 0$ yields a NRMSE of 0.06. This NRMSE shrinks to $6.24 \cdot 10^{-3}$, a decrease of 89.7%, by adapting the leakage rate of $\gamma = 0.88$. For the Mackey–Glass equation, we observe a decrease of 86.2% in NRMSE, from 0.19 to 0.03, by choosing a leakage rate of $\gamma = 0.68$. For the Santa Fe laser dataset, we observe the NRMSE being reduced by 20.8%, from 1.49 to 1.18, by choosing a leakage rate of $\gamma = 0.41$. For the neutral type DDE, the start leakage rate yields a NRMSE of 0.34. We found the lowest NRMSE at a leakage rate of $\gamma = 0.84$ yielding a NRMSE of 0.06 and being a reduction of 81.5%.

We observe 21–90% performance gain by selecting a dataset-specific leakage between $41 - 88\%$ by-passing the non-linear activation and constituting a residual connection for the reservoir's previous state. We observe the highest leakage for the Lorenz system and argue that the explanation is that ODE's succeeding state solely depends on its previous state with only gradual changes in between states. In contrast, the succeeding state of both DDEs, the Mackey–Glass equation as well as the neutral type DDE, depends on n former states putting more relevance on the activated reservoir state. We observe the lowest optimal leakage rate for the Santa Fe laser dataset. This dataset is characterized by the highest NRMSE and the lowest R^2 in our evaluation indicating it to be the hardest to predict. We hypothesize that the non-linear activation acts as a filter reducing the propagation of error that inherits the empirically measured data and a higher leakage contradicts this effect. In conclusion, these results confirm Jaeger's [50] suggestions towards the concept being advantageous for model performance but are in contradiction to ESNs state-of-the-art that almost entirely neglects the concept (cp. Table 1). Lu et al.'s study [58] is a rare exception but only considers leakage for one experiment predicting the Kuramoto–Sivashinsky equations.

6.6. Regularization coefficient β

We explored potential regularization coefficients with a logarithmic step size in $\beta \in [10^{-10}, 1]$. For the Lorenz system, the start regularization coefficient $\beta\prime = 1 \cdot 10^{-6}$ yields a NRMSE of $6.24 \cdot 10^{-3}$. This NRMSE shrinks to $6.17 \cdot 10^{-3}$, a reduction of 1.05%, by choosing a regularization coefficient of $\beta = 1.05 \cdot 10^{-6}$. For the Mackey–Glass equation, we observe a decrease of 3.37% in NRMSE, from $2.62 \cdot 10^{-2}$ to $2.54 \cdot 10^{-2}$, by selecting a regularization coefficient of $5.99 \cdot 10^{-7}$. For the Santa Fe laser dataset, the NRMSE decreases by 1.08%, from 1.18 to 1.17, when choosing a regularization coefficient of $\beta = 8.13 \cdot 10^{-7}$. For the neutral type DDE, the start regularization coefficient yields a NRMSE of $6.25 \cdot 10^{-2}$. We observe the lowest NRMSE at $\beta = 8.33 \cdot 10^{-7}$ yielding a NRMSE of $5.61 \cdot 10^{-2}$ and being a reduction of 10.3%.

We observe a dataset-specific gain between $1 - 10\%$ by choosing an appropriate regularization coefficient. That means all datasets benefit from a regularization, which might be at first surprising given that three of them represent differential equations and only the Santa Fe laser dataset represents empirical measurements. However, also the differential equations are somewhat noisy due to their approximation at discrete time steps. In conclusion, we find that a weak regularization seems to be universally beneficial.

6.7. Training length S_T

We explore training length $S_T \in \{100; 200; 500; 1,000; 1,500; 2,000; 2,500; 3,000; 4,000; 5,000\}$ (Fig. 6). We train 100 ESNs per dataset and training length with the best performing parametrization determined before. When lowering training length below a dataset-specific threshold, we observe a drastically increasing NRMSE, i.e., $S_T < 2,000$ (Lorenz system), $S_T < 1,000$ (Mackey–Glass equation), $S_T < 1,000$ (Santa Fe laser), and $S_T < 100$ (neutral type DDE) respectively. Beyond this training length, model performance is not substantially improving anymore. Our results are partly contradicting a rule of thumb that suggests to choose $S_T > N^r$ [59] and underline that our start training length of $S_T = 2,000$ was a good choice.

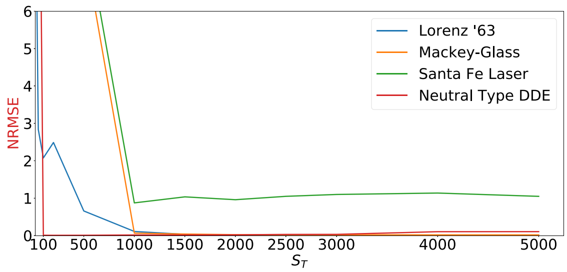


Fig. 6. Test error (NRMSE) over increasing training length S_T measured as median over 100 trained ESNs per dataset.

6.8. Variability across reservoir seeds

Since each datapoint of our hyper-parameter optimization is the result of 100 trained ESNs being derived from the same 100 randomly seeded reservoir weight matrices (cp. Section 5), we cannot only analyze their average performance but also their performance variation across this set. We are specifically interested in whether standard deviation of model performance across the trained reservoirs changes over the course of the hyper-parameter optimization. For the Lorenz system, we observe for the start configuration a standard deviation of 232.9% relative to the median, which decreases to 50.8% relative to the median once all hyper-parameters are optimized. For the Mackey–Glass equation, we observe for the start configuration a standard deviation of 438.0% relative to the median, which decreases to 88.9% relative to the median once the optimization is completed. For the Santa Fe laser dataset we observe for the start configuration a standard deviation of 1701.5% relative to the median, which decreases to 54.4% relative to the median once the optimization is completed. For the neutral type DDE we observe for the start configuration a standard deviation of 3,592.0% relative to the median, which decreases to 69.8% relative to the median once the optimization is completed.

In conclusion, over the course of the hyper-parameter optimization performance variance caused by the randomly seeded reservoir matrices substantially reduces. We argue that this is a very impressive result illustrating the importance of well tuned hyper-parameters but also indicates the influence of the reservoir seed on the eventual performance.

6.9. Limitations of our study

Our observations are restricted to the common form of the ESN as originally proposed by [48]. That means that we solely consider a $\tanh(\cdot)$ non-linearity as activation function, while individual studies, e.g., also considered linear activation functions [89]. Furthermore, alternative leakage concepts have been proposed using the leakage rate inside the activation function [32] or calculating the leakage rate from time step width Δt [18,87]. Other studies proposed a dropout concept [10] and local plasticity [92] for ESNs, propose and study ESNs as dynamical systems rather than pure predictors of those [45] or propose input representation in quaternions $\mathbb{H}$ rather than real numbers [98]. Those and other existing concepts may have impacted our results and deserve additional experimentation. Aiming for generalizable findings, we employ four representative datasets with different characteristics for our study. While a direct application to fundamentally different time-series data of our findings might be difficult, the proposed methodology of systematically conducting a grid search to identify suitable hyperparameters is. In particular, we have shown the need to take sufficiently many samples (realizations) into account when deriving design guidelines based on key characteristics. Otherwise, a transferability to other similar data sets cannot be expected. However, this is still a restricted set and further studies would be necessary to substantiate our results. In this regard, further tests evaluating properties of the reservoir beyond evaluation metrics, e.g., as shown in [74] or discussed in the context of infinite neural networks [43] would be necessary. Neural networks typically suffer from learning problems like catastrophic forgetting and concept drift. Catastrophic forgetting is defined as "when a model initially trained on task A is later trained on task B, its performance on task A can decline calamitously" [13]. We argue that this problem cannot occur for the ESNs that we study since they learn non-iteratively from a single batch of training data. Would the same ESN, learned with the inputs of task A, be used to learn with the inputs of task B, instead of only calculating an alternate W^{out}, the dynamics of the reservoir would be overwritten in the course of the second washout and learning steps loosing the predictive capabilities for the first problem. Concept drift refers to a change in the distribution of the data [94] and is clearly a problem for ESNs. We argue that their computationally lightweight nature allows a continuous retraining of the model given that the dataset is prone to concept drift. Furthermore, we compared the variances of the results over 100 seeds and used the median to reduce them. An alternative approach, based on a bidirectional learning scheme was proposed in [21], which we plan to include in future work and adapt it to ESNs.

7. Design Guide

In this section, we unite our findings into a guideline for the efficient training of ESNs. As a first step, we recommend drawing a number of input and reservoirs matrices from a uniform distribution at a comparably large size N^r. These large reservoirs can easily be scaled to smaller N^r by average pooling and ensure more comparable performance results. We suggest parameter optimization with a set of seeds in parallel to gain more predictable results given the stochasticity of reservoir and input weights.

The density of the reservoir $d(W^r)$ should be chosen in relation to its size with a larger reservoir benefitting more from sparsity. A spectral radius close to one, e.g., $\rho = 0.995$, is supposed to yield acceptable performance. If the system to predict is characterized by a high dependence on former states, using a spectral radius beyond one may further improve performance. If on the contrary no dependence on the history of the time-series is of importance, a high spectral radius can be a negative influence and can arguably be seen as noise on the data, accordingly we suggest a spectral radius of zero in cases of ODEs as in benchmarking problems like the Lorenz system.

We found that leakage largely improves prediction performance with leakage rates γ between $40 - 90\%$. Analogous to our study, we suggest to only activate and optimize it once a satisfying reservoir in terms of size, density, and scaling has been determined. Thereby, we argue based on the Lorenz system to choose a higher leakage rate with a short term history dependence and a smaller leakage rate in case of an already large deviation from the ground truth to impede an accelerated increase of the training error. Alternatively, one may employ delay coordinates as, e.g., explored in [51]. A mild regularization with $\beta \approx 10^{-7}$ has shown to be universally beneficial.

We argue that it is essential to evaluate prediction performance in a multi-step auto-regressive setting if this shall be the later application scenario. Finally, given that we suggest a parallel training of multiple ESNs, the whole set or a best performing selection thereof could be used in a prediction ensemble given the low computational requirements of the ESN concept.

8. Conclusion and future work

In this paper, we studied echo state networks' (ESN) performance for predicting time series. More specifically, we designed a systematic study of their hyper-parameters. For our experimentation, we used four different datasets representing one approximated ODE system, two approximated DDE systems, and one empirically measured dataset allowing us to generalize our findings to a wider population of (chaotic) dynamical systems. We argue for the applicability of our results for further time series provided that the key characteristics are similar to the ones used within this work. Furthermore, we expect to strengthen our hypothesis as well to further extend the proposed design guidelines for other types of time series in future work. Hyper-parameter optimization of ESNs is not trivial due to their stochastic nature arising from randomly seeded input and reservoir matrices. We overcome this challenge by reusing a set of 100 seeded reservoirs and proposing an algorithm to derive smaller and sparser instances of these matrices. Our findings impressively demonstrate the power of a hyper-parameter-tuned ESN when auto-regressively predicting time series over several hundred steps into the future with minimal error. We found that ESNs' performance improved by $85.1\% - 99.8\%$ over an already wisely chosen default parameter initialization while performance variability arising from a reservoir's seed is dramatically reduced. These results emphasize the benefit of a careful hyper-parameter optimization. Apart from these general observations we report individual findings per hyper-parameter that partly contradict earlier findings and will help researchers as a guideline when training new models.

In the future, we plan to extend our study to other problem types and datasets as well as studying less prominent variations of the ESN model focusing on more advanced topologies, e.g., neural hierarchies [61], sub-goal divided reservoirs [33], deep ESN [73], next generation reservoir computing [36], long-short term ESN [105], ϕ-ESN [34] or Autoreservoirs [24]. We also aim to evaluate the influence of other metrics such as isometric properties and separation between states as proposed in [74] in the field of time-series prediction. Similar to the analysis of the currently very popular extended Dynamic Mode Decomposition (eDMD, see [96,17,66]), we want to consider reservoir computing as a (data-driven) lifting technique for time-series prediction to derive performance certificates using mass concentration inequalities, see, e.g, the recent papers [102,70] on eDMD for ordinary and stochastic differential equations, respectively. To this end, the original highly-nonlinear dynamics is lifted into an infinite-dimensional spaces of observables, which is – then – approximated by a high-dimensional, but linear dynamics. For ESNs, the reservoir is predestine to play the role of the high-dimensional, but easy to evaluate surrogate model.

Data availability

The Implementation was forked from: [107]. Our data is available at doi:10.7910/DVN/I7TPJD [106].

Funding

We are funded by the Carl Zeiss Foundation's grants: DeepTurb and E4SM. Moreover, Karl Worthmann gratefully acknowledges funding be the German Research Foundation (DFG; grants WO 2056/6-1, WO 2056/14-1).

CRediT authorship contribution statement

Johannes Viehweg: Conceptualization, Methodology, Software, Data curation, Investigation, Visualization, Writing - original draft, Writing - review & editing. **Karl Worthmann:** Conceptualization, Writing - review & editing, Project administration, Funding acquisition. **Patrick Mäder:** Conceptualization, Methodology, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We thank Herbert Jaeger for his constructive criticism to an earlier version of this manuscript as well as the editor and the anonymous reviewers for their insightful comments.

References

[1] Documentation scipy integrate odeint. https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint.html.
[2] Dynaml. https://github.com/transcendent-ai-labs/DynaML/tree/master/data.
[3] jitcddeneutral. https://github.com/neurophysik/jitcdde/blob/master/examples/neutral.py.
[4] Linear correlation. https://condor.depaul.edu/sjost/it223/documents/correlation.htm.
[5] User guide r2. https://scikit-learn.org/stable/modules/model_evaluation.html#r2-score.
[6] T. Akiyama, G. Tanaka, Analysis on characteristics of multi-step learning echo state networks for nonlinear time series prediction, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.
[7] G. Ansmann, Efficiently and easily integrating differential equations with jitcode, jitcdde, and jitcsde. Chaos: An Interdisciplinary, Journal of Nonlinear Science 28 (2018).
[8] E.A. Antonelo, B. Schrauwen, On learning navigation behaviors for small mobile robots with reservoir computing architectures, IEEE transactions on neural networks and learning systems 26 (2014) 763–780.
[9] R. Auvray, B. Fabre, P.Y. Lagrée, Regime change and oscillation thresholds in recorder-like instruments, The Journal of the Acoustical Society of America 131 (2012) 1574–1585.
[10] D. Bacciu, F. Crecchi, Augmenting recurrent neural networks resilience by dropout, IEEE transactions on neural networks and learning systems 31 (2019) 345–351.
[11] Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
[12] Basterrech, S., 2015. An empirical study of the l2-boost technique with echo state networks. arXiv preprint arXiv:1501.00503.
[13] Benyahia, Y., Yu, K., Smires, K.B., Jaggi, M., Davison, A.C., Salzmann, M., Musat, C., 2019. Overcoming multi-model forgetting, in: International Conference on Machine Learning, PMLR. pp. 594–603.
[14] F.M. Bianchi, L. Livi, C. Alippi, Investigating echo-state networks dynamics by means of recurrence analysis, IEEE transactions on neural networks and learning systems 29 (2016) 427–439.
[15] Bollt, E., 2020. On explaining the surprising success of reservoir computing forecaster of chaos? the universal machine learning dynamical system with contrasts to var and dmd. arXiv preprint arXiv:2008.06530.
[16] S.L. Brunton, B.W. Brunton, J.L. Proctor, E. Kaiser, J.N. Kutz, Chaos as an intermittently forced linear system, Nature communications 8 (2017) 1–9.
[17] S.L. Brunton, J.N. Kutz, Data-driven science and engineering: Machine learning, dynamical systems, and control, Cambridge University Press, 2019.
[18] D.M. Canaday, Modeling and Control of Dynamical Systems with Reservoir Computing Ph.D. thesis, The Ohio State University, 2019.
[19] W. Cao, L. Hu, J. Gao, X. Wang, Z. Ming, A study on the relationship between the rank of input data and the performance of random weight neural network, Neural Computing and Applications (2020) 1–12.
[20] W. Cao, M.J. Patwary, P. Yang, X. Wang, Z. Ming, An initial study on the relationship between meta features of dataset and the initialization of nnrw, in: 2019 international joint conference on neural networks (IJCNN), IEEE, 2019, pp. 1–8.
[21] W. Cao, Z. Xie, J. Li, Z. Xu, Z. Ming, X. Wang, Bidirectional stochastic configuration network for regression problems, Neural Networks 140 (2021) 237–246.
[22] Cernansky, M., Makula, M., 2005. Feed-forward echo state networks, in: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005, IEEE. pp. 1479–1482.
[23] M. Chang, A. Terzis, P. Bonnet, Mote-based online anomaly detection using echo state networks, International Conference on Distributed Computing in Sensor Systems, Springer. (2009) 72–86.
[24] P. Chen, R. Liu, K. Aihara, L. Chen, Autoreservoir computing for multistep ahead prediction based on the spatiotemporal information transformation, Nature communications 11 (2020) 1–15.
[25] Cho, K., Van Merriënboer, B., Bahdanau, D., Bengio, Y., 2014. On the properties of neural machine translation: Encoder-decoder approaches. arXiv preprint arXiv:1409.1259.
[26] Dale, M., 2018. Neuroevolution of hierarchical reservoir computers, in: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 410–417.
[27] H. Duan, X. Wang, Echo state networks with orthogonal pigeon-inspired optimization for image restoration, IEEE transactions on neural networks and learning systems 27 (2015) 2413–2425.
[28] Embrechts, M.J., Alexandre, L.A., Linton, J.D., 2009. Reservoir computing for static pattern recognition., in: ESANN, Citeseer.
[29] L. Fan-Jun, L. Ying, Effects of the minimal singular value on the performance of echo state networks, in: 2017 36th Chinese Control Conference (CCC), IEEE, 2017, pp. 3905–3909.
[30] Fanjun, L., Xiaohong, W., Ying, L., Shizeng, L., Shaoli, J., 2019. Design of weight matrices for echo state networks with truncated singular values, in: 2019 Chinese Automation Congress (CAC), IEEE. pp. 3043–3047.
[31] J.D. Farmer, Chaotic attractors of an infinite-dimensional dynamical system, Physica D: Nonlinear Phenomena 4 (1982) 366–393.
[32] A.A. Ferreira, T.B. Ludermir, R.R. De Aquino, An approach to reservoir computing design and training, Expert systems with applications 40 (2013) 4172–4182.
[33] M. Freiberger, P. Bienstman, J. Dambre, A training algorithm for networks of high-variability reservoirs, Scientific reports 10 (2020) 1–11.
[34] C. Gallicchio, A. Micheli, Architectural and markovian factors of echo state networks, Neural Networks 24 (2011) 440–456.
[35] V.M. Gan, Y. Liang, L. Li, L. Liu, Y. Yi, A cost-efficient digital esn architecture on fpga for ofdm symbol detection, ACM Journal on Emerging Technologies in Computing Systems (JETC) 17 (2021) 1–15.
[36] D.J. Gauthier, E. Bollt, A. Griffith, W.A. Barbosa, Next generation reservoir computing. Nature communications 12 (2021) 1–8.
[37] L. Glass, M. Mackey, Mackey-glass equation. Scholarpedia 5 (2010) 6908.
[38] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press. (2016), http://www.deeplearningbook.org.
[39] A. Haluszczynski, J. Aumeier, J. Herteux, C. Räth, Reducing network size and improving prediction stability of reservoir computing. Chaos: An Interdisciplinary, Journal of Nonlinear Science 30 (2020).
[40] A. Haluszczynski, C. Räth, Good and bad predictions: Assessing and improving the replication of chaotic attractors by means of reservoir computing. Chaos: An Interdisciplinary, Journal of Nonlinear Science 29 (2019).
[41] Hartl, M.D., 2003. Lyapunov exponents in constrained and unconstrained ordinary differential equations. arXiv preprint physics/0303077.
[42] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[43] M. Hermans, B. Schrauwen, Recurrent kernel machines: Computing with infinite echo state networks, Neural Computation 24 (2012) 104–133.
[44] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.
[45] W. Hu, Y. Zhang, R. Ma, Q. Dai, J. Yang, Synchronization between two linearly coupled reservoir computers, Chaos, Solitons & Fractals 157 (2022).
[46] M. Inubushi, K. Yoshimura, Reservoir computing beyond memory-nonlinearity trade-off, Scientific reports 7 (2017) 1–10.
[47] Ishu, K., van Der Zant, T., Becanovic, V., Ploger, P., 2004. Identification of motion with echo state network, in: Oceans' 04 MTS/IEEE Techno-Ocean'04 (IEEE Cat. No. 04CH37600), IEEE. pp. 1205–1210.
[48] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks-with an erratum note, Bonn, Germany: German National Research Center for Information Technology GMD Technical Report 148 (2001) 13.
[49] H. Jaeger, Adaptive nonlinear system identification with echo state networks, Advances in neural information processing systems 15 (2002) 609–616.
[50] H. Jaeger, M. Lukoševičius, D. Popovici, U. Siewert, Optimization and applications of echo state networks with leaky-integrator neurons, Neural networks 20 (2007) 335–352.

[51] L. Jaurigue, E. Robertson, J. Wolters, K. Lüdge, Reservoir computing with delayed input for fast and easy optimisation, Entropy 23 (2021).
[52] D. Koryakin, J. Lohmann, M.V. Butz, Balanced echo state networks, Neural Networks 36 (2012) 35–45.
[53] Krušna, A., Lukoševičius, M., 2018. Predicting mozart's next note via echo state networks, in: Symposium for Young Scientists in Technology, Engineering and Mathematics SYSTEM.
[54] Q. Li, Y. Chen, N. Ao, X. Han, Z. Wu, Echo state network-based visibility graph method for nonlinear time series prediction, in: 2018 Chinese Control And Decision Conference (CCDC), IEEE, 2018, pp. 1854–1859.
[55] Z. Li, G. Tanaka, Deep echo state networks with multi-span features for nonlinear time series prediction, in: 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–9.
[56] E.N. Lorenz, Deterministic nonperiodic flow, Journal of the atmospheric sciences 20 (1963) 130–141.
[57] Z. Lu, B.R. Hunt, E. Ott, Attractor reconstruction by machine learning. Chaos: An Interdisciplinary, Journal of Nonlinear Science 28 (2018).
[58] Z. Lu, J. Pathak, B. Hunt, M. Girvan, R. Brockett, E. Ott, Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. Chaos: An Interdisciplinary, Journal of Nonlinear Science 27 (2017).
[59] Lukoševičius, M., 2012. A practical guide to applying echo state networks, in: Neural networks: Tricks of the trade. Springer, pp. 659–686.
[60] M. Lukoševicius, Reservoir computing and self-organized neural hierarchies, Jacobs University, Bremen, 2012.
[61] M. Lukoševicius, Reservoir Computing and Self-Organized Neural Hierarchies Ph.D. thesis, Jacobs University Bremen, 2012.
[62] M. Lukoševičius, H. Jaeger, Reservoir computing approaches to recurrent neural network training, Computer Science Review 3 (2009) 127–149.
[63] W. Maass, T. Natschläger, H. Markram, Real-time computing without stable states: A new framework for neural computation based on perturbations, Neural Computation 14 (2002) 2531–2560.
[64] Maat, J.R., Gianniotis, N., Protopapas, P., 2018. Efficient optimization of echo state networks for time series datasets, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE. pp. 1–7.
[65] M.C. Mackey, L. Glass, Oscillation and chaos in physiological control systems, Science 197 (1977) 287–289.
[66] A. Mauroy, Y. Susuki, I. Mezić, Koopman operator in systems and control, Springer, 2020.
[67] G.B. Morales, C.R. Mirasso, M.C. Soriano, Unveiling the role of plasticity rules in reservoir computing, Neurocomputing (2021).
[68] S.S. Mosleh, L. Liu, C. Sahin, Y.R. Zheng, Y. Yi, Brain-inspired wireless communications: Where reservoir computing meets mimo-ofdm, IEEE transactions on neural networks and learning systems 29 (2017) 4694–4708.
[69] K. Nakajima, H. Hauser, T. Li, R. Pfeifer, Exploiting the dynamics of soft materials for machine learning, Soft robotics 5 (2018) 339–347.
[70] F. Nüske, S. Peitz, F. Philipp, M. Schaller, K. Worthmann, Finite-data error bounds for Koopman-based prediction and control, 33, Springer, 2021, pp. 1–34, https://arxiv.org/abs/2108.07102.
[71] S. Otte, M.V. Butz, D. Koryakin, F. Becker, M. Liwicki, A. Zell, Optimizing recurrent reservoirs with neuro-evolution, Neurocomputing 192 (2016) 128–138.
[72] J. Pathak, A. Wikner, R. Fussell, S. Chandra, B.R. Hunt, M. Girvan, E. Ott, Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model. Chaos: An Interdisciplinary, Journal of Nonlinear Science 28 (2018).
[73] PEDRELLI, L., 2019. Deep reservoir computing: A novel class of deep recurrent neural networks.
[74] A. Prater-Bennette, Randomness and isometries in echo state networks and compressed sensing, Compressive Sensing VII: From Diverse Modalities to Big Data Analytics, SPIE. (2018) 149–158.
[75] Prokhorov, D., 2005. Echo state networks: appeal and challenges, in: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005, IEEE. pp. 1463–1466.
[76] Rad, A.A., 2008. Dynamical networks (miniporject) effect of topology of the reservoir on performance of echo state networks.
[77] A. Rodan, P. Tiňo, Minimum complexity echo state network, IEEE transactions on neural networks 22 (2010) 131–144.
[78] Rodan, A., Tiňo, P., 2011. Negatively correlated echo state networks., in: ESANN, Citeseer.
[79] N. Rodriguez, E. Izquierdo, Y.Y. Ahn, Optimal modularity and memory capacity of neural reservoirs, Network Neuroscience 3 (2019) 551–566.
[80] B. Schrauwen, M. Wardermann, D. Verstraeten, J.J. Steil, D. Stroobandt, Improving reservoirs using intrinsic plasticity, Neurocomputing 71 (2008) 1159–1171.
[81] Q. Song, Z. Feng, Effects of connectivity structure of complex echo state network on its prediction performance for nonlinear time series, Neurocomputing 73 (2010) 2177–2185.
[82] A. Souahlia, A. Belatreche, A. Benyettou, Z. Ahmed-Foitih, E. Benkhelifa, K. Curran, Echo state network-based feature extraction for efficient color image segmentation, Concurrency and Computation: Practice and Experience 32 (2020).
[83] J.J. Steil, Online reservoir adaptation by intrinsic plasticity for backpropagation–decorrelation and echo state learning, Neural Networks 20 (2007) 353–364.
[84] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. arXiv preprint arXiv:1706.03762.
[85] G.K. Venayagamoorthy, B. Shishir, Effects of spectral radius and settling time in the performance of echo state networks, Neural Networks 22 (2009) 861–863.
[86] D. Verstraeten, Reservoir Computing: computation with dynamical systems Ph.D. thesis, Ghent University, 2009.
[87] D. Verstraeten, B. Schrauwen, M. D'Haene, D. Stroobandt, An experimental unification of reservoir computing methods, Neural Networks 20 (2007) 391–403.
[88] Verstraeten, D., Schrauwen, B., Stroobandt, D., 2006. Reservoir-based techniques for speech recognition, in: The 2006 IEEE International Joint Conference on Neural Network Proceedings, IEEE. pp. 1050–1053.
[89] P. Verzelli, C. Alippi, L. Livi, echo state networks with self-normalizing activations on the hyper-sphere, Scientific reports 9 (2019) 1–14.
[90] P.R. Vlachas, J. Pathak, B.R. Hunt, T.P. Sapsis, M. Girvan, E. Ott, P. Koumoutsakos, Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics, Neural Networks (2020).
[91] T. Waegeman, B. Schrauwen, et al., Feedback control by online learning an inverse model, IEEE transactions on neural networks and learning systems 23 (2012) 1637–1648.
[92] X. Wang, Y. Jin, K. Hao, Evolving local plasticity rules for synergistic learning in echo state networks, IEEE transactions on neural networks and learning systems 31 (2019) 1363–1374.
[93] L. Wasserman, All of statistics: a concise course in statistical inference, Springer Science & Business Media, 2013.
[94] G.I. Webb, R. Hyde, H. Cao, H.L. Nguyen, F. Petitjean, Characterizing concept drift, Data Mining and Knowledge Discovery 30 (2016) 964–994.
[95] C.O. Weiss, U. Hübner, N.B. Abraham, D. Tang, Lorenz-like chaos in nh3-fir lasers, Infrared Physics & Technology 36 (1995) 489–512.
[96] M.O. Williams, I.G. Kevrekidis, C.W. Rowley, A data–driven approximation of the Koopman operator: Extending dynamic mode decomposition, Journal of Nonlinear Science 25 (2015) 1307–1346.
[97] F. Wyffels, B. Schrauwen, D. Verstraeten, D. Stroobandt, Band-pass reservoir computing, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 3204–3209.
[98] Y. Xia, C. Jahanchahi, D.P. Mandic, Quaternion-valued echo state networks, IEEE Transactions on Neural Networks and Learning Systems 26 (2014) 663–673.
[99] C. Yang, X. Zhu, Z. Ahmad, L. Wang, J. Qiao, Design of incremental echo state network using leave-one-out cross-validation, IEEE Access 6 (2018) 74874–74884.
[100] X. Yao, Z. Wang, Fractional order echo state network for time series prediction, Neural Processing Letters 52 (2020) 603–614.
[101] Yperman, J., Becker, T., 2016. Bayesian optimization of hyper-parameters in reservoir computing. arXiv preprint arXiv:1611.05193.
[102] C. Zhang, E. Zuazua, A quantitative analysis of Koopman operator methods for system identification and predictions. Hal-03278445, 2021.
[103] J. Zhao, Q. Liu, W. Wang, W. Pedrycz, L. Cong, Hybrid neural prediction and optimized adjustment for coke oven gas system in steel industry, IEEE transactions on neural networks and learning systems 23 (2012) 439–450.
[104] Zhao, Q., Yin, H., Chen, X., Shi, W., 2015. Performance optimization of the echo state network for time series prediction and spoken digit recognition, in: 2015 11th International Conference on Natural Computation (ICNC), IEEE. pp. 502–506.
[105] K. Zheng, B. Qian, S. Li, Y. Xiao, W. Zhuang, Q. Ma, Long-short term echo state network for time series prediction, IEEE Access 8 (2020) 91961–91974.
[106] J. Viehweg, P. Mäder, K. Worthmann, Replication Data for: Parameterizing echo state networks for multi-step time series prediction, Harvard Dataverse, 2022, https://doi.org/10.7910/DVN/I7TPJD.
[107] N. Trouvain, L. Pedrelli, T.T. Dinh, X. Hinaut, ReservoirPy: An Efficient and User-Friendly Library to Design Echo State Networks, in: Artificial Neural Networks and Machine Learning-ICANN 2020, Springer International Publishing, 2020, pp. 494–505, https://doi.org/10.1007/978-3-030-61616-8_40.

Johannes Viehweg received a Masters Degree in Computer Science from the University of Technology Ilmenau, Germany, in 2020 and works as doctorate student in the department of Data-intensive Systems and Visualization at the University of Technology Ilmenau, whereby his area of research lies in the field of Reservoir Computing and Machine Learning for time-series prediction.

Karl Worthmann received the Diploma degree in business mathematics and the Ph.D. degree in mathematics from the University of Bayreuth, Germany. 2014 he was appointed assistant professor for "Differential Equations" at Technische Universität Ilmenau (TU Ilmenau), Germany. 2019 he was promoted to full professor after receiving the Heisenberg-professorship "Optimization-based Control" by the German Research Foundation (DFG) in 2018. Karl Worthmann's current research interests include optimization-based techniques for data- or model-based control of nonlinear dynamical systems.

He was recipient of the Ph.D. Award from the City of Bayreuth, Germany, and stipend of the German National Academic Foundation. 2013 he has been appointed Junior Fellow of the Society of Applied Mathematics and Mechanics (GAMM), where he served as speaker in 2014 and 2015. Currently, Karl Worthmann is chairman "Mathematical Systems Theory" of the interdisciplinary GAMM activity group "Dynamics and Control Theory".

Patrick Mäder is a Professor at the University of Technology Ilmenau, Germany for Data-intensive Systems and Visualization. He received a Diploma degree in Industrial Engineering in 2003, a Ph.D. degree in Computer Science in 2009, both from the University of Technology Ilmenau. He worked postdoctoral Lise Meitner-fellow of the Institute for System Engineering and Automation of the Johannes Kepler University in Linz, Austria, as well as as postdoctoral researcher at Software and Requirements Engineering Centre at the DePaul University in Chicago, USA.

In 2009 he received the Thuringian STIFT-Award for this doctorate thesis as well as 2020 the Thuringian Award for Research in the field of Applied Research.

His research interest lay in the field of Software Engineering with the focus on safety criticality, explainable Machine Learning and Computational Biology and Ecology.