Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 53-2 (2020) 1255–1260

An empirical assessment of the universality of ANNs to predict oscillatory time series

F. Dercole * **M. Sangiorgio** * **Y. Schmirander** *

* *Department of Electronics, Information, and Bioengineering Politecnico di Milano, I-20133 Italy (e-mail: fabio.dercole, matteo.sangiorgio @polimi.it, yunus.schmirander@mail.polimi.it)*

Abstract: Artificial neural networks (ANNs) are universal function approximators, therefore suitable to be trained as predictors of oscillatory time series. Though several ANN architectures have been tested to predict both synthetic and real-world time series, the universality of their predictive power remained unexplored. Here we empirically test this universality across five well-known chaotic oscillators, limiting the analysis to the simplest architecture, namely multi-layer feed-forward ANN trained to predict one sampling step ahead. To compare different predictors, data are sampled according to their frequency content and the ANN structure scales with the characteristic dimensions of the oscillator. Moreover, the quality of recursive multi-step-ahead predictions are compared in terms of the system's (largest) Lyapunov exponent (LLE), i.e., the predictive power is measured in terms of the number of Lyapunov times (LT, the LLE inverse) predicted within a prescribed (relative) error. The results confirm the rather uniform predictive power of the proposed ANN architecture.

Keywords: Machine learning; Time series modelling; Nonlinear system identification.

1. INTRODUCTION

The paradigm of deterministic chaos provides an effective explanation for irregular oscillatory behaviours in real systems which we do not consider inherently stochastic (Kantz and Schreiber, 2003; Bradley and Kantz, 2015). The boundary between short- and long-term predictability and unpredictability is not clearly defined and depends on the performances of the models used for the forecasting. In the last few decades, many attempts to push this limit as far as possible have been made, adopting a wide range of predictive models. Early attempts were performed in the 1980s and 1990s (Farmer and Sidorowich, 1987; Casdagli, 1989; Verdes et al., 1998), but the topic became more prominent in recent years, due to the development of machine learning techniques, the most widely used are artificial neural networks (ANN), in the field of time series analysis and prediction.

The main distinction within ANN is between feed-forward, i.e. memory-free networks and recurrent ones. Examples of the first category are traditional multi-layer perceptrons (Woolley et al., 2010; Covas and Benetos, 2019), radial basis function networks (Leung et al., 2001; Van Truc and Anh, 2018), and fuzzy neural networks (Maguire et al., 1998). Recurrent neurons are endowed with an internal state and are therefore more suited to process temporal sequences. They have been used as predictors in Han et al. (2004); Ma et al. (2007); Yu et al. (2017); Wan et al. (2018). In the very last years, a particular class of recurrent networks, reservoir computers, have been demonstrated to be extremely effective in the prediction of chaotic dynamics (Butcher et al., 2013; Pathak et al., 2017; Lu et al., 2017, 2018; Pathak et al., 2018a; Antonik et al., 2018; Weng et al., 2019). Some authors developed hybrid forecasting schemes combining machine learning tools and knowledge-based models (Pathak et al., 2018b; Vlachas et al., 2018; Doan et al., 2019) or combining multiple predictive models (Inoue et al., 2001; Okuno et al., 2019).

While pursuing the best predictive performance, none of the previous studies specifically addressed the universality of the predictive power of ANN-based predictors. This is the aim of this preliminary work. For simplicity, we limit our analyses to feed-forward ANN, fully connected, trained to predict synthetic chaotic time series one sampling step ahead; and we recursively use the obtained predictor over multiple-steps. We scale the network structure to the complexity of the time series and define an appropriate metric to compare predictions of different datasets that is readily applicable to real-world cases.

2. METHODS

2.1 Chaotic oscillators

To generate synthetic time series, we consider five well-known Chaotic oscillators: the Logistic and Hénon maps, the classical prototypes of chaos in non-reversible and reversible discrete-time systems, respectively; the generalized Hénon map, to include a case of hyperchaos; the Lorenz system, the chaos prototype in continuous-time; and the Liley model of neocortical activity (Liley et al., 1999, 2002), to include a high-dimensional system with close-to-real (EEG) dynamics.

The continuous-time systems are sampled at a constant rate not excessively larger than twice the highest frequency significantly present in the state variables' power spectra

10.1016/j.ifacol.2020.12.1850

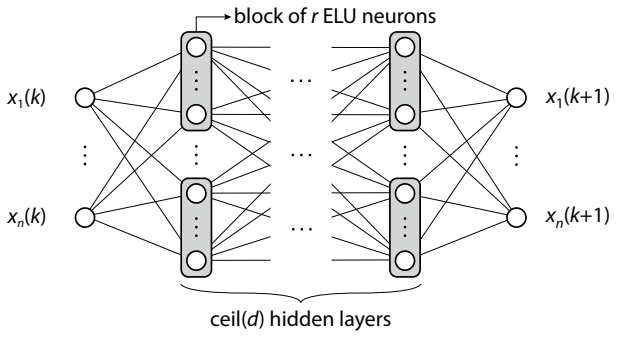

Fig. 1. ANN architecture. The resolution parameter r is set to 10 in all experiments; n is the dimension of the oscillator used as data generator; d the dimension of the corresponding attractor.

(computed numerically on oversampled signals; Welch's method). The same has been checked for discrete-time systems, for which the original time-unit turned out to be the correct sampling time. Table 1 summarizes the relevant features of the chaotic attractors, including sampling time, the (largest) Lyapunov exponent (LLE) and the Lyapunov time (LT, the LLE inverse), and the fractal dimension d (approximated with the Kaplan-Yorke formula, except for the Logistic for which box-counting and correlations dimensions are around 0.5). The systems' equations and parameter values are reported in Appendix A.

System	dim.	sampling	attractor		
	n	τ	LLE	LT=1/LLE	$d \simeq$
Logistic	1	1	0.354	2.825	0.50
Hénon	2	1	0.432	2.315	1.26
generalized Hénon	3	1	0.277	3.610	2.13
Lorenz	3	0.1	0.905	1.105	2.06
Liley	10	2ms	0.047ms^{-1}	21.28ms	2.10

Table 1. Chaotic attractors

2.2 Neural architecture and training

As anticipated in the Introduction, we limit our analyses to fully connected, feed-forward ANN, trained to predict one sampling step ahead. To predict with a consistent quality across different data generators, the proposed ANN is structured as follows:

- The number of input and output neurons correspond to the dimension n of the data generator. When dealing with real-world time series, n can be replaced by the embedding dimension of the reconstructed dynamics in the space of delayed samples (Kantz and Schreiber, 2003; Bradley and Kantz, 2015).
- A basic building block of r neurons (with exponential-linear unit ELU activation) is defined, where r (set to 10 in all experiments) is a resolution parameter.
- Each network's layers contains n blocks of r neurons, because n are the functions to be identified, each giving one output, and r neurons per-function are used to resolve each of the independent dimensions in state space over which the function need to be identified.
- Because we only use regime data, the number of these independent dimensions corresponds to the attractor's dimension, a priori known in case of synthetic data (see Tab. 1) or estimated on the reconstructed attractor for real-world time series. For each dimension, up-rounding to the closest integer for fractal dimensions, we place a layer.

The ANN architecture is illustrated in Fig. 1. Note that it is technically *deep* when the attractor's dimension is larger than one.

We build the *training* dataset by means of a noise-free numerical simulation of the oscillator's model, starting from an initial condition on (i.e., sufficiently close to) the oscillator's attractor. Let $x_i(k)$, $k = 1, \dots, N$, be the ground truth samples of the oscillator's i-th state variable, $i = 1, \dots, n$, $x(k)$ be the oscillator's state vector, and $\hat{x}_i^{(h)}(k)$ be the h-step-ahead prediction of $x_i(k)$ based on the true state $x(k-h)$, $x(1-h)$ being the initial condition, at time $t-h$, used to generate the dataset. Let also $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i^{(h)}$ pack the ground truth samples and predictions into N-dimensional vectors. The number N of ground truth samples to be used for each oscillator must be selected to grant a consistent quality across different data generators. We therefore extend the dataset until we reach a one-step-ahead ($h=1$) training loss-value within a prescribed threshold. To this end, however, we cannot use the standard mean-squared-error (MSE) loss function

$$\text{MSE}^{(h)} = \frac{1}{n}\sum_{i=1}^{n} \text{MSE}(\mathbf{x}_i, \hat{\mathbf{x}}_i^{(h)}),$$
$$\text{MSE}(\mathbf{x}_i, \hat{\mathbf{x}}_i^{(h)}) = \frac{1}{N}\sum_{k=1}^{N} \big(x_i(k) - \hat{x}_i^{(h)}(k)\big)^2, \quad (1)$$

because it assumes absolute, rather than relative values, making impossible to set a meaningful threshold across different systems.

For this reason, we introduce a relative loss function, the R2-score (sometimes known as Nash-Sutcliffe model efficiency coefficient and not to be confused with the determination coefficient), defined by

$$\text{R2}^{(h)} = \frac{1}{n}\sum_{i=1}^{n} \text{R2}\big(\mathbf{x}_i, \hat{\mathbf{x}}_i^{(h)}\big),$$
$$\text{R2}\big(\mathbf{x}_i, \hat{\mathbf{x}}_i^{(h)}\big) = 1 - \frac{\text{MSE}\big(\mathbf{x}_i, \hat{\mathbf{x}}_i^{(h)}\big)}{\text{MSE}\big(\mathbf{x}_i, \bar{x}_i\big)}$$
$$= 1 - \frac{\sum_{k=1}^{N} \big(x_i(k) - \hat{x}_i^{(h)}(k)\big)^2}{\sum_{k=1}^{N} \big(x_i(k) - \bar{x}_i\big)^2}, \quad (2)$$

where $\bar{x}_i$ is the mean value of $\mathbf{x}_i$.

The R2-score measures the quality of the predictions $\hat{x}_i^{(h)}(k)$ with respect to the trivial forecasting of the mean value of the observed data (R2 = 0 for the trivial prediction). This metric varies in the range $(-\infty, 1]$: the upper bound is obtained in case of a perfect forecasting. The R2-score is widely used because it can be seen as a normalized MSE, as $\text{MSE}(\mathbf{x}_i, \bar{x}_i)$ at denominator in (2) is the data variance.

As tradition in machine learning, we train the networks by optimizing the MSE loss function (one-step ahead, using Adam optimization algorithm), but we set the size N of the dataset to get an R2-score (one-step ahead on the dataset itself) of at least 0.9999 (almost perfect fitting of

the training dataset; we start with $N = 10^3$ and ten-fold N until we meet the threshold; 80% of the data are used for training and 20% for validation). Following this criterion, 10^4 samples are enough for the three discrete maps and for the Lorenz system; 10^6 samples are necessary for the Liley system. We also tune the optimal combination of the hyper-parameters (the learning rate and batch size of the gradient descent), by adopting a traditional grid search approach. The selected learning rates are 10^{-3} for the discrete maps, and 10^{-4} for the Lorenz and Liley systems. A batch size of 128 ensures a good compromise between forecasting performance and training time.

2.3 Performance metrics

To quantify the predictive power of the trained ANN, we build a *test* dataset by simulating the oscillator's model from a different initial condition with respect to the one used for training. As discussed in the previous section, the MSE metric is not designed to compare the quality of predictions on different systems, so we compute the R2-score (defined in (2)) of multiple-step-ahead predictions ($h>1$) over the test dataset (multiple-step-ahead predictions are obtained recursively, by feeding the network outputs back as inputs). However, even when measured by the R2-score, the quality of predictions can still significantly vary across different systems, and even within the attractor of a system, because the quality is not normalized with respect to the difficulty of the forecasting task. The universal way to measure how challenging is the prediction of an oscillating time series is the (largest) Lyapunov exponent (LLE) (Pathak et al., 2018a,b), the average exponential rate of expansion/contraction (if positive/negative) of nearby trajectories. We therefore define the predictive power of our ANN architecture as the number of Lyapunov times (LT, the LLE inverse) within which the forecasting horizon h gives an R2-score above a prescribed threshold.

To analyze the distribution of the ANN predictive power within the system's attractor, we introduce a local metric based on the absolute error. For each ground truth value $x_i(k)$ of the system's state variables x_i in the test dataset, we normalize the absolute prediction error $|x_i(k+h) - \hat{x}_i^{(h)}(k+h)|$ by 3-times the data standard deviation $\sigma_i = \sqrt{\mathrm{MSE}(\mathbf{x}_i, \bar{x}_i)}$ and we consider the largest h for which the so-obtained relative error stays within 10%. We then express the horizon length in unit of LTs and we average the obtained metric over the state's components, $i = 1, \dots, n$.

3. RESULTS

To assess the universality of the predictive power of the proposed ANN architecture, Fig. 2 reports the R2-score (2) for increasing length h of the forecasting horizon for the five considered chaotic oscillators, separately for each state variable x_i. The forecasting horizon in the bottom horizontal axis is expressed in the system's physical time $t = \tau\, h$, where τ is the sampling time reported in Tab. 1, while the unit of the top horizontal axis is the LT. For each $h \geq 1$ (multiple-step-ahead predictions, obtained recursively), the R2-score is computed for a test dataset of $N = 10^5$ points on the oscillator's attractor.

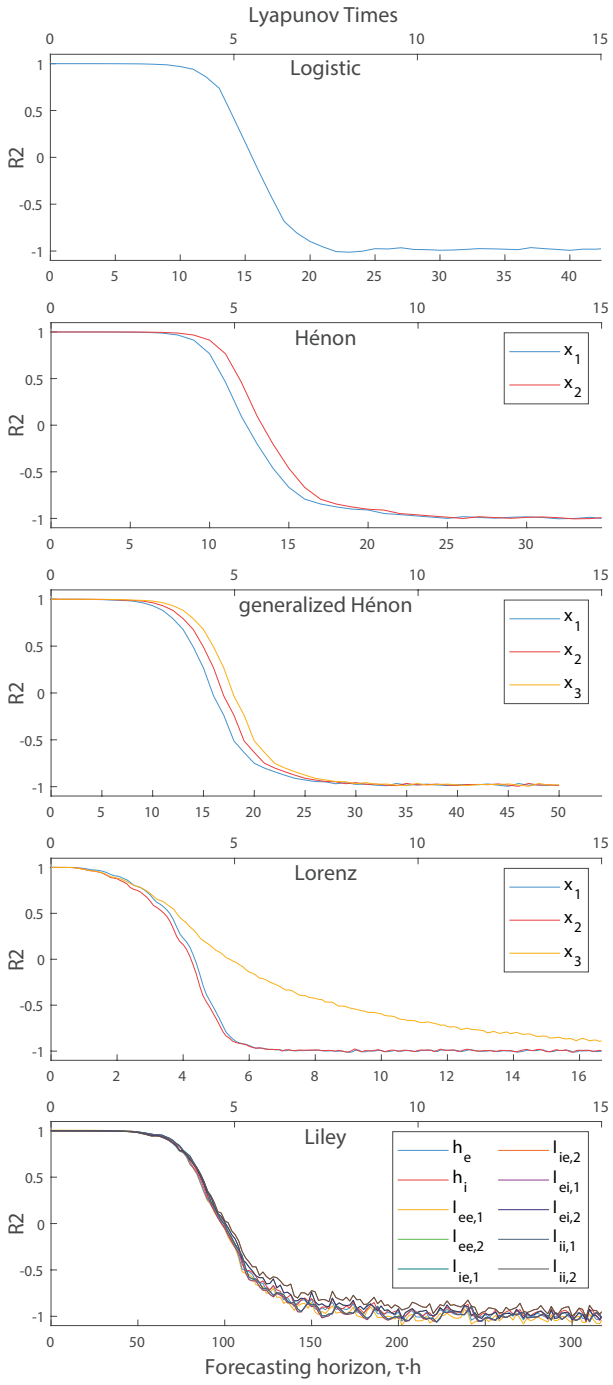

Fig. 2. R2-score $\mathrm{R2}\big(\mathbf{x}_i, \hat{\mathbf{x}}_i^{(h)}\big)$ for increasing length h of the forecasting horizon, separately for each state variable.

The figure confirms the universal predictive power of our ANN architecture. Though predictors for different systems do show different (average) forecasting quality, differently degrading for increasing h, the patterns uniformize if read on the LT-time-scale (top horizontal axis). E.g., averaging over the systems' state variables, the R2-score drops below 0 after 5.41, 5.44, 4.65, 4.37, 4.66 LT for the five considered chaotic oscillators (top-to-bottom).

Note that when the forecasting horizon is too long, the R2-score approaches -1 for all systems. For chaotic oscillators, this is a good feature of the predictor. Essentially, when h is large, the data $x_i(k)$ and its best possible prediction

$\hat{x}_i^{(h)}(k)$ becomes two uncorrelated samples over the system's attractor. The expected squared error then becomes twice the data variance, so the R2-score approaches -1 (see (2); see Appendix B for a formal proof).

To assess the universality of the predictive power within the attractor of each system, Fig. 3 shows the scatter-plots of our local metric over the points of the test datasets (the same datasets used in Fig. 2). Again, we can observe the rather uniform predictive power.

4. CONCLUSIONS

We have successfully tested the universality of the predictive power of the ANN architecture proposed in Fig. 1 on five well-known chaotic oscillators. For the robustness of the test, the oscillators range from simple low-dimensional to complex high-dimensional systems, including non-reversible and hyper-chaotic ones (see Tab. 1). The universality has been assessed in three ways. 1) Across the different oscillators, the R2-scores computed for increasing length of the forecasting horizon on test datasets, show common deterioration patterns if the forecasting length is measured in units of the oscillator Lyapunov time (LT, the inverse of the largest Lyapunov exponent of the chaotic attractor). 2) For all predictors, the R2-score approaches -1 for long forecasting horizons, a feature that we realized to be an interesting quality check. 3) Within the attractor of each system, the universality of the predictive power is shown by a local metric scaled by the system's LT. Though we only used synthetic data, our approach is readily applicable to noisy and real-world time series.

Despite the focus of this work is not on performance, the proposed architecture—feed-forward one-step-ahead predictors used recursively—excels in forecasting short ranges, making it suitable for real-time applications which require fast computations. Further enhancement could be obtained with a careful hyper-parameters tuning or using deeper networks. These are however marginal improvements. For mid-to-long range prediction of chaotic systems, architectures trained to predict multiple-steps ahead might be preferable, since they should be able to contrast the error propagation inherent with the recursive approach. Furthermore, more advanced recurrent architectures, such as reservoir computers (Pathak et al., 2018a) and LSTM networks (Vlachas et al., 2018), should be employed to further confirm both performance and universality.

Note, however, that even disposing the perfect predictor, any initialization error in the predictor's input (due to unavoidable measurement errors and noise in real-world applications) is tenfold every each $\log(10) \approx 2.3$ LTs, so that forecasting noise-free time series beyond 3-4 LTs becomes a theoretical exercise. Longer forecasting horizons can however obtained in the parts of the system's attractor in which the dynamics is mildly chaotic or contracting. A related issue, to be addressed in future work, is indeed the normalization of our local metric by the local LT, the inverse of the local exponential rate of divergence.

Specifically for control applications (Box et al., 2015), while predictive control is usually model-based or employing observers such as a Kalman filter, robust forecasting can provide a model-free approach (Piche et al., 2000; Kumar et al., 2018).

REFERENCES

Antonik, P., Gulina, M., Pauwels, J., and Massar, S. (2018). Using a reservoir computer to learn chaotic attractors, with applications to chaos synchronization and cryptography. *Phys Rev E*, 98(1), 012215.

Box, G.E., Jenkins, G.M., Reinsel, G.C., and Ljung, G.M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons, 5 edition.

Bradley, E. and Kantz, H. (2015). Nonlinear time-series analysis revisited. *Chaos*, 25(9), 097610.

Butcher, J.B., Verstraeten, D., Schrauwen, B., Day, C., and Haycock, P. (2013). Reservoir computing and extreme learning machines for non-linear time-series data analysis. *Neural networks*, 38, 76–89.

Casdagli, M. (1989). Nonlinear prediction of chaotic time series. *Physica D*, 35(3), 335–356.

Covas, E. and Benetos, E. (2019). Optimal neural network feature selection for spatial-temporal forecasting. *Chaos*, 29(6), 063111.

Doan, N.A.K., Polifke, W., and Magri, L. (2019). Physics-informed echo state networks for chaotic systems forecasting. In *International Conference on Computational Science*, 192–198. Springer.

Farmer, J.D. and Sidorowich, J.J. (1987). Predicting chaotic time series. *Phys rev lett*, 59(8), 845.

Han, M., Xi, J., Xu, S., and Yin, F.L. (2004). Prediction of chaotic time series based on the recurrent predictor neural network. *IEEE T Signal Proces*, 52(12), 3409–3416.

Inoue, H., Fukunaga, Y., and Narihisa, H. (2001). Efficient hybrid neural network for chaotic time series prediction. In *International Conference on Artificial Neural Networks*, 712–718. Springer.

Kantz, H. and Schreiber, T. (2003). *Nonlinear Time Series Analysis*. Cambridge University Press, 2 edition.

Kumar, S.S.P., Tulsyan, A., Gopaluni, B., and Loewen, P. (2018). A deep learning architecture for predictive control. *IFAC-PapersOnLine*, 51(18), 512–517.

Leung, H., Lo, T., and Wang, S. (2001). Prediction of noisy chaotic time series using an optimal radial basis function neural network. *IEEE T Neural Networ*, 12(5), 1163–1172.

Liley, D.T., Cadusch, P.J., and Dafilis, M.P. (2002). A spatially continuous mean field theory of electrocortical activity. *Network-Comp Neural*, 13(1), 67–113.

Liley, D.T., Cadusch, P.J., and Wright, J.J. (1999). A continuum theory of electro-cortical activity. *Neurocomputing*, 26, 795–800.

Lu, Z., Hunt, B.R., and Ott, E. (2018). Attractor reconstruction by machine learning. *Chaos*, 28(6), 061104.

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

Ma, Q.L., Zheng, Q.L., Peng, H., Zhong, T.W., and Xu, L.Q. (2007). Chaotic time series prediction based on evolving recurrent neural networks. In *Int Conf Mach Learn*, volume 6, 3496–3500. IEEE.

Maguire, L.P., Roche, B., McGinnity, T.M., and McDaid, L. (1998). Predicting a chaotic time series using a fuzzy

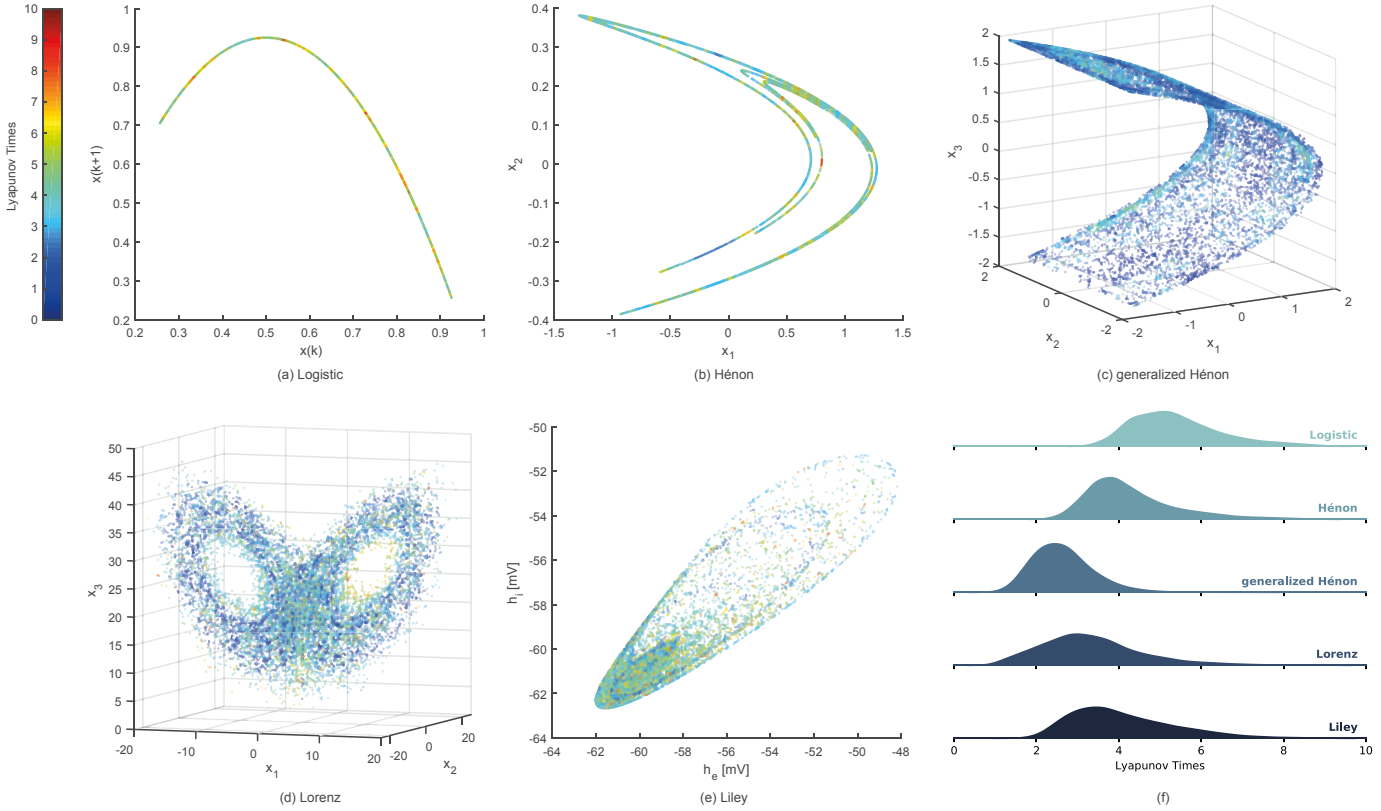

Fig. 3. Prediction performance and universality within the oscillators' attractor. Each point of the test dataset is color-coded with the local metric defined in Sect. 2.3. The metric distributions are reported in panel f.

neural network. *Inform Sciences*, 112(1-4), 125–136.
Okuno, S., Aihara, K., and Hirata, Y. (2019). Combining multiple forecasts for multivariate time series via state-dependent weighting. *Chaos*, 29(3), 033128.
Pathak, J., Hunt, B., Girvan, M., Lu, Z., and Ott, E. (2018a). Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys Rev*, 120(2), 024102.
Pathak, J., Lu, Z., Hunt, B.R., Girvan, M., and Ott, E. (2017). Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. *Chaos*, 27(12), 121102.
Pathak, J., Wikner, A., Fussell, R., Chandra, S., Hunt, B.R., Girvan, M., and Ott, E. (2018b). Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model. *Chaos*, 28(4), 041101.
Piche, S., Keeler, J.D., Martin, G., Boe, G., Johnson, D., and Gerules, M. (2000). Neural network based model predictive control. In *Adv Neur Inf*, 1029–1035.
Van Truc, N. and Anh, D.T. (2018). Chaotic time series prediction using radial basis function networks. In *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*, 753–758. IEEE.
Verdes, P., Granitto, P., Navone, H., and Ceccatto, H. (1998). Forecasting chaotic time series: Global vs. local methods. *Novel Intelligent Automation and Control Systems*, 1, 129–145.
Vlachas, P.R., Byeon, W., Wan, Z.Y., Sapsis, T.P., and Koumoutsakos, P. (2018). Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *P Roy Soc A-Math Phy*, 474(2213), 20170844.
Wan, Z.Y., Vlachas, P., Koumoutsakos, P., and Sapsis, T. (2018). Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PloS one*, 13(5), e0197704.
Weng, T., Yang, H., Gu, C., Zhang, J., and Small, M. (2019). Synchronization of chaotic systems and their machine-learning models. *Phys Rev E*, 99(4), 042203.
Woolley, J.W., Agarwal, P., and Baker, J. (2010). Modeling and prediction of chaotic systems with artificial neural networks. *Int J Numer Meth Fl*, 63(8), 989–1004.
Yu, R., Zheng, S., and Liu, Y. (2017). Learning chaotic dynamics using tensor recurrent neural networks. In *Proceedings of the ICML*, volume 17.

Appendix A. CHAOTIC OSCILLATORS

The state vector of the oscillator is denoted by x and its dimension by n, the single components are x_i, $i = 1, \ldots, n$; the discrete and continuous times by k and t, so that $x(k)$ and $x(t)$ are vectors in $\mathbf{R}^n$ and $x(0)$ is the initial condition.

A.1 Logistic map

The logistic map is a one-dimensional discrete-time model introduced by Pierre Francois Verhulst to model the demographic growth a population assuming a limited carrying capacity of the environment. The state equation is

$$x(k+1) = px(k)(1 - x(k)) \qquad \text{(A.1)}$$

with $x(k)$ representing the ratio between the current and maximum population density and parameter $p \in (0, 4)$ the

per-unit growth-rate at low density. We use $p = 3.7$, for which the observed behavior is chaotic (see Tab. 1).

A.2 Hénon map

The Hénon map is a two-dimensional discrete-time system introduced by Michel Hénon as a simplified model of the Poincaré section of the Lorenz model. The equations are

$$x_1(k+1) = 1 - a x_1(k)^2 + x_2(k) \tag{A.2a}$$
$$x_2(k+1) = b x_1(k) \tag{A.2b}$$

We use the standard parameter values $a = 1.4$ and $b = 0.3$, for which the LEs are $\lambda_1 = 0.432$ and $\lambda_2 = -1.636$.

A.3 Generalized Hénon map

Baier and Klein generalized the Hénon map to a system of any dimension. It is suitable for the purposes of this work to employ the three-dimensional implementation given by

$$x_1(k+1) = a - x_2(k)^2 + b x_3(k) \tag{A.3a}$$
$$x_2(k+1) = x_1(k) \tag{A.3b}$$
$$x_3(k+1) = x_2(k) \tag{A.3c}$$

where $a = 1.9$ and $b = 0.03$. The system is hyperchaotic, with $\lambda_1 = 0.277$, $\lambda_2 = 0.262$, and $\lambda_3 = -4.046$.

A.4 Lorenz system

The Lorenz system models the convective motion of a two-dimensional fluid cell warmed from above and cooled from below. The state equations are

$$\dot{x}_1 = \sigma(x_2 - x_1) \tag{A.4a}$$
$$\dot{x}_2 = x_1(\rho - x_3) - x_2 \tag{A.4b}$$
$$\dot{x}_3 = x_1 x_2 - \beta x_3 \tag{A.4c}$$

where x_1, x_2, and x_3 describe the rate of convective overturning, horizontal temperature variation and vertical temperature variation respectively. We use the standard parameter setting $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$, for which the LEs are $\lambda_1 = 0.900$, $\lambda_2 = 0.004$, and $\lambda_3 = -14.340$.

A.5 Liley system

The Liley system models the neocortical activity of a column of 40,000 to 100,000 neurons, measurable during noninvasive EEG (Liley et al., 1999, 2002). The state equations are

$$\tau_e \dot{h}_e = \Big[h_{er} - h_e + \frac{h_{eeq} - h_e}{|h_{eeq} - h_{er}|} I_{ee,1} + \frac{h_{ieq} - h_e}{|h_{ieq} - h_{er}|} I_{ie,1}\Big] \tag{A.5a}$$
$$\tau_i \dot{h}_i = \Big[h_{ir} - h_i + \frac{h_{eeq} - h_i}{|h_{eeq} - h_{ir}|} I_{ei,1} + \frac{h_{ieq} - h_i}{|h_{ieq} - h_{ir}|} I_{ii,1}\Big] \tag{A.5b}$$
$$\dot{I}_{ee,1} = I_{ee,2} \tag{A.5c}$$
$$\dot{I}_{ee,2} = -2a I_{ee,2} - a^2 I_{ee,1} + Aae[N_{ee} S_e(h_e) + p_{ee}] \tag{A.5d}$$
$$\dot{I}_{ie,1} = I_{ie,2} \tag{A.5e}$$
$$\dot{I}_{ie,2} = -2b I_{ie,2} - b^2 I_{ie,1} + BbeN_{ie} S_i(h_i) \tag{A.5f}$$
$$\dot{I}_{ei,1} = I_{ei,2} \tag{A.5g}$$
$$\dot{I}_{ei,2} = -2a I_{ei,2} - a^2 I_{ei,1} + Aae[N_{ei} S_e(h_e) + p_{ei}] \tag{A.5h}$$
$$\dot{I}_{ii,1} = I_{ii,2} \tag{A.5i}$$
$$\dot{I}_{ii,2} = -2b I_{ii,2} - b^2 I_{ii,1} + BbeN_{ii} S_i(h_i), \tag{A.5j}$$

The state variables, h_e and h_i, describe by the mean soma membrane potentials of the excitatory and inhibitory neuron population, respectively. The variables $I_{pq,1}$ with $p, q \in \{e, i\}$ model the synaptic activity from neuron population p to population q, while their temporal derivations are the variables $I_{pq,2}$. The function $S_q(h_q)$ converts the mean soma membrane potentials h_q into an equivalent mean firing rate

$$S_q(h_q) = m_q(1 + \exp(-\sqrt{2}(h_q - \theta_q)/s_q))^{-1} \tag{A.6}$$

with firing thresholds $\theta_e = \theta_i = -50\,\mathrm{mV}$, corresponding standard deviations $s_e = s_i = 5\,\mathrm{mV}$ and mean maximal firing rates $m_e = m_i = 0.5\,\mathrm{ms}^{-1}$. Other parameters include the resting potentials $h_{er} = h_{ir} = -70\,\mathrm{mV}$, the equilibrium potentials $h_{eeq} = 45\,\mathrm{mV}$ and $h_{ieq} = -90\,\mathrm{mV}$, population membrane time constants $\tau_e = 9\,\mathrm{ms}$ and $\tau_i = 39\,\mathrm{ms}$, excitatory and inhibitory population postsynaptic potential peak amplitudes $A = 0.81\,\mathrm{mV}$ and $B = 4.85\,\mathrm{mV}$, synaptic rate constants $a = 0.49\,\mathrm{ms}^{-1}$ and $b = 0.592\,\mathrm{ms}^{-1}$, the numbers of synapses received by excitatory and inhibitory neurons from nearby excitatory and inhibitory neurons $N_{ee} = N_{ei} = 3034$ and $N_{ie} = N_{ii} = 536$, and the inputs from distant excitatory cortical and subcortical neurons $p_{ee} = 9.3\,\mathrm{ms}^{-1}$ and $p_{ei} = 4.0\,\mathrm{ms}^{-1}$. The computed LEs are $\lambda_1 = 0.047\,\mathrm{ms}^{-1}$, $\lambda_2 = -0.003\,\mathrm{ms}^{-1}$, and $\lambda_3 = -0.462\,\mathrm{ms}^{-1}$.

Appendix B. THE PREDICTOR'S ATTRACTOR

Imagine to have the perfect predictor, i.e., a copy of the data generator. If the data are chaotic, it is well known that the prediction $\hat{x}_i^{(h)}(k)$ will anyway diverge, for increasing forecasting horizon h, from the true data $x_i(k)$, because the local divergence that typifies chaos (the LLE is the average exponential rate of local divergence) amplifies any small initialization difference $|x_i(k-h) - \hat{x}_i^{(h)}(k-h)|$ between the data generator and the predictor. Because the predictor is perfect, for large enough h, we can imagine $x_i(k)$ and $\hat{x}_i^{(h)}(k)$ as two uncorrelated trajectories within the chaotic attractor, so that, for large N as well, we have

$$\frac{1}{N}\sum_{k=1}^{N} x_i(k) = \frac{1}{N}\sum_{k=1}^{N} \hat{x}_i^{(h)}(k) = \mathrm{E}[x_i(k)] = \mathrm{E}[\hat{x}_i^{(h)}(k)] \tag{B.1a}$$

$$\frac{1}{N}\sum_{k=1}^{N} x_i(k)\hat{x}_i^{(h)}(k) = \mathrm{E}[x_i(k)\,\hat{x}_i^{(h)}(k)] = \mathrm{E}[x_i(k)]\,\mathrm{E}[\hat{x}_i^{(h)}(k)]. \tag{B.1b}$$

Exploiting (B.1), we can then write

$$\begin{aligned}
\mathrm{MSE}\big(\mathbf{x}_i, \hat{\mathbf{x}}_i^{(h)}\big) &= \mathrm{E}\big[\big(x_i(k) - \hat{x}_i^{(h)}(k)\big)^2\big] \\
&= \mathrm{E}[x_i(k)^2] - \mathrm{E}[2x_i(k)\,\hat{x}_i^{(h)}(k)] + \mathrm{E}[\hat{x}_i^{(h)}(k)^2] \\
&= \mathrm{E}[x_i(k)^2] - 2\mathrm{E}[x_i(k)]\,\mathrm{E}[\hat{x}_i^{(h)}(k)] + \mathrm{E}[\hat{x}_i^{(h)}(k)^2] \\
&= 2\mathrm{E}[x_i(k)^2] - 2\mathrm{E}[x_i(k)]^2 \\
&= 2(\mathrm{E}[x_i(k)^2] - \bar{x}_1^2),
\end{aligned}$$

that substituted into (2) gives $\mathrm{R2}\big(\mathbf{x}_i, \hat{\mathbf{x}}_i^{(h)}\big) = -1$.

The fact that the R2-score approaches -1 for large h is hence a quality check for the predictor. It means that the predictor, used recursively as a dynamical system itself, is able to generate an attractor with the same characteristics of the data generator.