Leveraging the structure of dynamical systems for data-driven modeling

Michele Alessandro Bucci, Onofrio Semeraro, Alexandre Allauzen, Sergio Chibbaro, Lionel Mathelin

Leveraging the structure of dynamical systems for data-driven modeling

Michele Alessandro Bucci[1], Onofrio Semeraro[2], Alexandre Allauzen[3], Sergio Chibbaro[2,4], and Lionel Mathelin[2]

[1]A&O team, INRIA Saclay, LISN, Université Paris-Saclay, 91400 Orsay, France
[2]LISN, CNRS, Université Paris-Saclay, 91400 Orsay, France
[3]LAMSADE, Université Paris Dauphine, Place du Maréchal de Lattre de Tassigny, 75016 Paris, France
[4]Sorbonne Université, CNRS, Institut Jean Le Rond d'Alembert, 75005 Paris, France

Abstract

The reliable prediction of the temporal behavior of complex systems is required in numerous scientific fields. This strong interest is however hindered by modeling issues: often, the governing equations describing the physics of the system under consideration are not accessible or, when known, their solution might require a computational time incompatible with the prediction time constraints. Nowadays, approximating complex systems at hand in a generic functional format and informing it ex nihilo from available observations has become a common practice, as illustrated by the enormous amount of scientific work appeared in the last years. Numerous successful examples based on deep neural networks are already available, although generalizability of the models and margins of guarantee are often overlooked. Here, we consider Long-Short Term Memory neural networks and thoroughly investigate the impact of the training set and its structure on the quality of the long-term prediction. Leveraging ergodic theory, we analyze the amount of data sufficient for a priori guaranteeing a faithful model of the physical system.

We show how an informed design of the training set, based on invariants of the system and the structure of the underlying attractor, significantly improves the resulting models, opening up avenues for research within the context of active learning. Further, the non-trivial effects of the memory initializations when relying on memory-capable models will be illustrated. Our findings provide evidence-based good-practice on the amount and the choice of data required for an effective data-driven modeling of any complex dynamical system.

Article Info

Correspondence:
M.A. Bucci
michele-alessandro.bucci@inria.fr

Keywords:
Short-term forecasting, chaotic dynamical systems, recurrent models, LSTM

Lead paragraph **The paper discusses the quality of predictions obtained by memory-capable neural-networks with respect of the selection of the amount of data used during the training. In particular:**

- **We suggest that ergodic-based criteria can provide a-priori indications on the sufficient amount of data required for training satisfactory models and avoid overfitting;**
- **We observe that informed design of the training set, based on invariants of the system and the structure of the underlying attractor, significantly improves the resulting models;**
- **We illustrate that memory initializations have non-trivial effect on the prediction abilities of models.**

1 Introduction

In the present era of mathematization of Nature and essentially every aspects of our lives, the need for modeling is ubiquitous. Focusing on engineering as an example, models are invaluable for predicting quantities of interest, designing and optimizing physical systems, understanding their behavior in an environment, allowing to derive control strategies to achieve some goals, etc. Physical models have historically been developed from a combination of observational data and first principles, in varying parts depending on the amount and type of prior knowledge. This approach is still followed today where the mathematical structure of a model is typically provided and observational data are used to inform it, *i.e.*, learn the coefficients associated with the different terms of the retained structure. Countless variants exist but they generally follow this philosophy.

Nonetheless, in many relevant cases, the physics of the phenomena is very complex and the interplay of observation with physical intuition is not sufficient to sort out a formalized model, even roughly accurate, let alone more rigorous general laws. In those cases, it is tempting to resort to a pure data-driven approach, in which some model is built up entirely from data. In fact, this approach has been always tried and already critized in the XIX-th century by Maxwell [8]. The idea has however received some rigorous methodology and approach, even recently, for instance after the work of Takens [28, 6, 27]. Thanks to these efforts, it is possible to derive predictive models from time-series [18], but only for low-dimensional systems.

The data-driven approach has found new motivation following the significant progress in the field of Machine Learning (ML), fostered by the massive augmentation of computer-based activities [14]. Even if a clear framework is still missing, the use of such algorithms to build models from data has been recently attempted in a variety of different physical phenomena, *e.g.*, [5, 3]. However, this recent blooming of deep learning approaches has not necessarily changed the global picture of modeling. Neural networks can provide a highly expressive and generic model structure suitable for a wide class of applications. The potential versatility of the resulting model is however associated with a usually large amount of necessary observational data to suitably inform the associated coefficients of the model, here being the weights and biases of the network nodes. Yet, not every situations enjoy massive amount of observational data and can afford to inform a widely generic model. In practice, it is more likely that data are scarce due the cost and/or technical difficulty in obtaining them and complex high-dimensional systems may need a quantity of data significantly exceeding the observing capability, [1]. In that sense, the lack of data for inferring physical models might have direct consequence on their relevance.

A strategy to alleviate this limitation is to simplify the postulated structure of the model so that it is less data hungry, with simpler models here understood as involving fewer parameters to inform from the available data, hence a weaker constraint on the amount of training samples. Another strategy is to leverage prior knowledge one has on the physical system under consideration to guide the structure and the learning process, hence saving on data. For instance, known symmetries can be enforced by design in the structure or in the representation format of the input data, effectively reducing the need for data. A similar idea applies to invariances one may know the system obeys (conserved quantities such as energy, translational or rotational invariances, etc.) or properties such as stability in a given sense. This physics-aware machine learning approach has now become trendy and is the motivation of numerous recent efforts, see [26, 33, 19, 31] among many more.

A particular relevant play-field where to assess the machine learning approach for modeling is that of chaotic dynamical systems. While these systems are generic and of great importance in different branches of science, they have received attention only recently [24, 2], and the development and understanding of machine learning algorithms for chaotic systems is yet at the beginning. Remarkably, previous works seem to indicate an intriguing capability to give accurate forecasting of chaotic systems, even relatively high-dimensional [23], apparently improving over deterministic embedding techniques [18]. These results therefore deserve to be thoroughly analyzed.

In the present work, we aim to understand the possible limitations of the pure data-driven approach and to explore the idea of leveraging some of the known organization of the dynamics to inform the process of collecting observations on the system. In particular, we explore different strategies for collecting observations from a dynamical system in the aim of learning and modeling its behavior and predict its state over time. These strategies only differ by the region in the phase space where observational samples are taken from, every other aspects being identical (sampling frequency, number of samples, observed quantity, etc.). These

different configurations are chosen to be typical of situations ranging from no prior information besides an estimated upper bound of the dimension of the dynamical system up to approximate knowledge of some invariants of the system. More specifically, dynamical systems are known to be organized around critical features such as fixed points which can be seen as constitutive of a backbone of the whole system. While fixed points do not drive the whole dynamics and only provide information on the local behavior of the flow, they are nonetheless a crucial feature. To obtain a more complete picture of the dynamical structure, higher dimensional invariant sets are also required, [13, 12]. We here explore the idea that fixed points are privileged regions of the state space for learning the dynamics.

To illustrate the role of the sampling strategy, we consider the celebrated Lorenz'63 system. While deterministic and low dimensional, the Lorenz system features many challenges encountered in more complex configurations, such as sensitivity to initial conditions and an attractor dimension $d_{\mathbb{A}}$ significantly lower than the ambient dimension d, here $d_{\mathbb{A}} \simeq 2.06 < d = 3$. It is thus somehow a minimally complex unit and justifies its relevance for illustrating some challenges in learning dynamical systems. Results presented and discussed below show that the different strategies lead to vastly different effectiveness in exploiting the information. The simplicity of the Lorenz'63 model allows us to comprehensively investigate the sampling impact on several observables and to have accurate statistics thanks to a large amount of independent realizations.

From an algorithmic point of view, recurrent structures such as auto-regressive models or recurrent neural networks are popular choices in such situations. In the present work, we focus on the LSTM (Long-Short Term Memory) architecture [15]. We have chosen this architecture for the present study both for its widespread use, including applications in prediction of dynamical systems behaviour [9]. Moreover, it has been recently shown that the performance given by different architectures including reservoir computing is similar [30], although for low dimensional systems like Lorenz'63, LSTMs appear to give the best results. Given this Neural Network approach, we have also investigated the possible impact of initial conditions and memory effects on the results of supervised learning models. This is relevant in the sense of providing best-practice guidance.

The paper is organized as follows. General principles in learning a system from data are briefly discussed in Section 2 to set up the stage and define notations. The general class of models we consider in this work is presented in Section 3. The specific configuration we retain for the present discussion is the Lorenz'63 introduced in Section 4, together with the different sampling strategies. The results are gathered and presented in Section 5 before the discussion in Section 6 closes the paper.

2 Theoretical background

2.1 Learning dynamical systems

The present work lies in the framework of learning a model to approximate the time evolution of a state of the system at hand from observations $\boldsymbol{s}(t) \in \mathbb{R}^{n_s}$. To do so, one needs a representation for the state and a dynamical model for describing its evolution in time. A representation of the state is given by a function g mapping observations over a measurement time horizon Γ, $\boldsymbol{s}_\Gamma(t)$, $t \in \Gamma$, to a set of coordinates $\boldsymbol{v} \in \mathbb{R}^{d_v}$. Several mappings have been proposed in the literature, [4, 11], including delays by time intervals Δt, $\boldsymbol{v}_i = \boldsymbol{s}(t - i\Delta t)$ for $i = 0, 1, \ldots$, and time derivatives of varying orders i, $\boldsymbol{v}_i = \boldsymbol{s}^{(i)}$.

To obtain the size of the coordinate space, we assume that the state of the system $\boldsymbol{u}$ lies on an attractor $\mathbb{A}$ of dimension $d_{\mathbb{A}} < d$ that is bounded boundaryless and a smooth submanifold of $\mathbb{R}^d$. In this case, the Whitney embedding theorem [32] shows that if the number of coordinates d_v is such that $d_v > 2\, d_{\mathbb{A}}$ then the dynamics of the state can be entirely captured by the new coordinate system. The Takens embedding theorem [28, 27] provides a way to make this result practical by asserting that one can take this number of time delays. Recent results have extended this theorem by providing guaranteed recovery properties based on the measurement operator, type of attractor, and other parameters of the problem, [7].

We assume no prior knowledge of the governing equations of the model and rely on a purely data-driven approach. The dynamical model for describing the time evolution of $\boldsymbol{v}(t) \in \mathbb{R}^{d_v}$ can be formulated in various ways, including feed-forward neural networks, owing to their high expressiveness.

As an alternative, for instance in case an upper bound of the attractor dimension is unknown due to process noise or measurement noise, a recurrent model $F_{\boldsymbol{\theta}}$ can be employed for describing the dynamics of

$\boldsymbol{v}$. The model is parameterized by $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ and writes

$$\dot{\boldsymbol{v}}(t) = F_{\boldsymbol{\theta}}\left(\boldsymbol{v}\right), \tag{1}$$

where a dotted quantity denotes its time-derivative.

Learning a model for the system under consideration takes the form of a classical supervised learning problem based on the misfit between observations $\boldsymbol{s}_\Gamma \equiv \boldsymbol{s}\left(\boldsymbol{u}\left(t \in \Gamma\right)\right)$ collected through a measurement operator $h : \boldsymbol{u} \mapsto \boldsymbol{s}$ over a time domain $\Gamma = [t_0, t_T]$ and predictions, in the sense of some norm $\|\cdot\|_\Gamma$. It typically results in an optimization problem in terms of the model parameters, possibly including a regularization term r:

$$\boldsymbol{\theta} \in \underset{\widetilde{\boldsymbol{\theta}} \in \mathbb{R}^{n_\theta}}{\arg\inf} \left\| \boldsymbol{s}(t) - h\left(\boldsymbol{v}\left(t; \widetilde{\boldsymbol{\theta}}\right)\right)\right\|_\Gamma^2 + r\left(\widetilde{\boldsymbol{\theta}}\right), \qquad \text{s.t.} \qquad \dot{\boldsymbol{v}}\left(t; \widetilde{\boldsymbol{\theta}}\right) = F_{\widetilde{\boldsymbol{\theta}}}\left(\boldsymbol{v}\right). \tag{2}$$

2.2 Collecting data

The main issue impacting the predictions based on observations is the quantity of data actually needed to approximate the dynamics in a given sense. The issue is related to the ergodic theory of dynamical systems, whose founding idea is that the long-time statistical properties of a deterministic system can be equivalently described in terms of the invariant (time-independent) probability, μ, such that $\mu(S)$ is the probability of finding the system in any specified region S of its phase space. If the trajectories of a d-dimensional ergodic system evolve in a bounded phase space $\Omega \subset \mathbb{R}^d$, the Poincaré recurrence theorem [25] ensures that analogues exist as it proves that the trajectories exiting from a generic set $S \in \Omega$ will return to such set S infinitely many times. The theorem holds for any points in S, almost surely. It was originally formulated for Hamiltonian systems, but it can be straightforwardly extended to dissipative ergodic systems provided initial conditions are chosen on the attractor. In d-dimensional dissipative systems, the attractor $\mathbb{A}$ typically has a dimension $d_\mathbb{A} < d$. The dimension $d_\mathbb{A}$ describes the small scale (that is, with resolution $\varepsilon \ll 1$) driving the probability $\mu\left(B_{\mathbf{x}}^d(\varepsilon)\right)$ of finding points on the attractor which are in the d-dimensional ball of radius ε around $\mathbf{x} \in \mathbb{A} \subset \mathbb{R}^d$:

$$\mu\left(B_{\mathbf{x}}^d(\varepsilon)\right) = \int_{B_{\mathbf{x}}^d(\varepsilon)} \mathrm{d}\mu(\widetilde{\mathbf{x}}) \sim \varepsilon^{d_\mathbb{A}}. \tag{3}$$

When $d_\mathbb{A}$ is not an integer, the attractor and its probability measure are termed *fractal*. The attractor is typically multifractal [22], but we can assume without lack of generality that it is homogeneous. Given that the Poincaré theorem proves the existence of good analogues, the issue is shifted to finding out how much time is needed for a Poincaré cycle to end.

This key problem was considered by Smolouchoski and resolved by Kac [17], who showed that for an ergodic system, given a set $S \equiv B_{\mathbf{x}^\star}^d(\varepsilon)$, the mean recurrence time $\langle \tau_S(\mathbf{x}^\star) \rangle$ relative to S is inversely proportional to the measure of the set S, *i.e.*,

$$\langle \tau_S(\mathbf{x}^\star) \rangle \propto \frac{1}{\mu(S)} \sim \varepsilon^{-d_\mathbb{A}}, \tag{4}$$

where the average $\langle \cdot \rangle$ is computed over all the points $\mathbf{x} \in S$ according to the invariant measure. Therefore, if we require ε to be small and if $d_\mathbb{A}$ is large, the average recurrence time becomes huge, a symptom of the curse of dimensionality in data-driven methods.

3 Machine-learning approach

3.1 Architecture

As discussed above, a set of d-dimensional observations retained to express a state of the system under consideration might not be a state vector, in the sense that it does not necessarily contain enough information to uniquely define its future evolution as an autonomous deterministic dynamical system. Even though the system were Markovian, it might not be so in the retained representation and may implicitly involve latent variables. This is typically the case in situations of partial observability or when measurements are affected

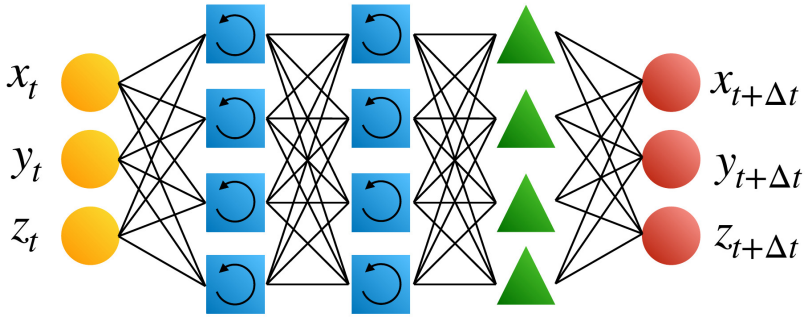


Figure 1: Sketch of the retained architecture. Each hidden layer is made of LSTM cells.

by noise. As discussed in Sec. 2, under conditions rigorously studied in the embeddology line of research, [6, 27], latent state variables can be substituted with observations over a recent past, that is, temporal information is traded for unavailable high-dimensional phase space information.

As said in the Introduction, in the present work, we consider the LSTM (Long-Short Term Memory) architecture. Compared with standard multi-layer perceptrons, LSTMs additionally feature internal variables $\boldsymbol{\zeta}$ carried-over through the recurrence and effectively acting as a "memory". Training an LSTM then allows to derive a discrete-in-time model predicting the future "state" $\boldsymbol{v}_{k+1} \equiv \boldsymbol{v}\left(t_k + \Delta t\right)$ from the current state and memory:

$$\boldsymbol{v}_{k+1} = F_{\boldsymbol{\theta}}\left(\boldsymbol{v}_k; \boldsymbol{\zeta}_k\right). \tag{5}$$

The retained architecture comprises 2 LSTM layers of 50 neurons each and one fully connected layer, with `tanh` activation functions in the LSTM cells and `sigmoids` for the state output, see also a sketch in Fig. 1. A 1-step ahead model is learned based on a whole trajectory, or a set of trajectories, in the L^2-sense; the training is performed with ADAM optimizer and an adaptive learning rate.

3.2 Strategies for sampling

Learning a model here consists in estimating the best set of parameters $\boldsymbol{\theta}$ in the sense of the optimization problem defined in Eq. (2). In the context of neural networks, this acts as the loss function used for training. The regularization term may promote several properties of the optimal solution $\boldsymbol{\theta}$, such as the popular choice of low magnitude (ridge regression), $r\left(\boldsymbol{\theta}\right) \sim \left\|\boldsymbol{\theta}\right\|_2$, or sparsity, $r\left(\boldsymbol{\theta}\right) \sim \left\|\boldsymbol{\theta}\right\|_1$, possibly mimicked with a suitable dropout strategy.

We adopt a discrete-in-time viewpoint, in a formulation consistent with the discrete model (5). The loss function $\mathscr{L}$ associated with the training problem then writes

$$\mathscr{L}\left(\boldsymbol{\theta}\right) = \sum_{k=1}^{K} \left\|\boldsymbol{s}_k - h\left(\boldsymbol{v}_k\left(\boldsymbol{\theta}\right)\right)\right\|_2^2 + r\left(\boldsymbol{\theta}\right), \tag{6}$$

with K such that $T = K\,\Delta t$ and the 1-step ahead dynamical model $\boldsymbol{v}_{k+1} = F_{\boldsymbol{\theta}}\left(\boldsymbol{v}_k; \boldsymbol{\zeta}_k\right)$.

The loss is here simply defined in terms of the Euclidean distance but many alternative definitions could be employed. For instance, optimal transport-based metrics, such as Dynamic Time Warping (DTW), can prove useful when observations come at a poorly known or controlled time pace. While the way the prediction from the model is compared with observational data is pivotal, the focus in this work is instead on the relevance of the training data. The present findings are believed to hold disregarding of the definition of the loss. A standard L^2 distance, implicitly assuming regularly sampled measurements, is hence retained.

The measurement operator is the identity, $h \equiv I$, so that the measurements are associated with the state of the system, $\boldsymbol{s}_k \equiv \boldsymbol{u}_k$, $\forall\, k$. No noise affects the observations, so that one is in a full observability situation. In this context, the observed system is Markovian in the representation space of the observations. However, a recurrent architecture is here considered as a general structure for learning, relevant for a wide class of situations. It further has the benefit of improving upon the robustness of the learning by virtue of some degree of redundant representation.

Without loss of generality, we disregard the regularization term $r(\boldsymbol{\theta})$ in the definition of the loss function which finally writes

$$\mathscr{L}(\boldsymbol{\theta}) = \sum_{k=1}^{K} \|\boldsymbol{s}_k - \boldsymbol{v}_k(\boldsymbol{\theta})\|_2^2, \qquad \text{s.t.} \qquad \boldsymbol{v}_{k+1} = F_{\boldsymbol{\theta}}(\boldsymbol{v}_k), \quad \boldsymbol{v}_0 = \boldsymbol{s}_0. \tag{7}$$

While this context is favorable for learning (low-dimensional deterministic system, noiseless observations, full observability, etc.), it will be seen below that the measurement strategy still significantly affects the learning performance.

4 Learning the Lorenz'63 system

4.1 Presentation

To illustrate the different learning strategies, we consider the celebrated Lorenz'63 system [21], originally introduced to mimic the thermal convection in the atmosphere. It consists of three coupled nonlinear ordinary differential equations (ODEs):

$$\begin{cases} \dot{x} = \sigma(y - x), \\ \dot{y} = \rho x - y - xz, \\ \dot{z} = xy - \beta z, \end{cases} \qquad \{\sigma, \rho, \beta\} \in \mathbb{R}^+.$$

The system of ODEs is integrated in time with a RK4 scheme and a time step $\Delta t = 0.01$. Constants σ, ρ, β are positive parameters, related to dimensionless scales such as the Prandtl and the Rayleigh numbers. For $\rho > \sigma(\sigma + \beta + 3)/(\sigma - \beta - 1)$ and $\sigma > (\beta + 1)$, the solution $\boldsymbol{u} := (x, y, z)$ exhibits a chaotic behavior. It is important for the following to note that the system features three unstable fixed points $\boldsymbol{u}_{\text{FP}}$, associated with a vanishing time-derivative:

$$\begin{aligned} \boldsymbol{u}_{\text{FP}_+} &= \left(\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho - 1\right), \\ \boldsymbol{u}_{\text{FP}_-} &= \left(-\sqrt{\beta(\rho-1)}, -\sqrt{\beta(\rho-1)}, \rho - 1\right), \\ \boldsymbol{u}_{\text{FP}_0} &= (0, 0, 0). \end{aligned}$$

In the following we shall consider the parameters fixed to $\beta = 8/3, \sigma = 10, \rho = 28$ as in the original work, such that the solution is in the chaotic regime. In this regime, the Kaplan-Yorke dimension of the attractor is $d_{\mathbb{A}} \simeq 2.06$.

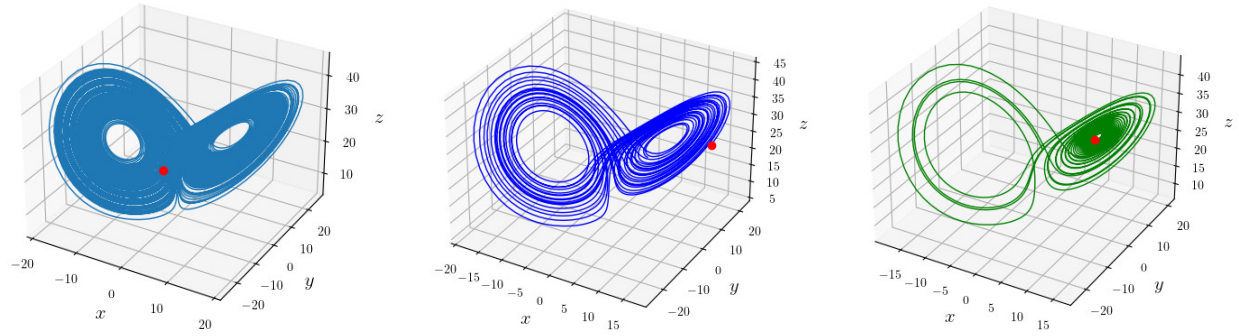
4.2 Set-up of the learning method

We turn our attention to the training set $\mathcal{T} = \{\boldsymbol{s}_k\}_{k=1}^K$ for learning the Lorenz'63 system. It is a crucial ingredient of the learning process and largely conditions the quality of the resulting model. We consider four different training sets ($\mathcal{T}_{\text{ergo}}$, $\mathcal{T}_{\text{split}}$, $\mathcal{T}_{\text{ens}}$, and $\mathcal{T}_{\text{FP}}$) of identical settings (sampling frequency, number of samples, measurement operator, etc.) but following different sampling strategies.

4.2.1 Ergodicity-compliant training set

The $\mathcal{T}_{\text{ergo}}$ dataset is a long trajectory on the attractor thought to act as a reference, providing sufficient statistics for learning a reliable and accurate model. With $\varepsilon = 10^{-2}$ and the Kaplan-Yorke dimension $d_{\mathbb{A}} \simeq 2.06$ of the Lorenz'63 for the retained settings $\{\sigma, \rho, \beta\}$, [20], the required number of samples resulting from a constant sampling period of $\Delta t = 0.01$ is about $27,000$.

Another dataset, $\mathcal{T}_{\text{split}}$, is derived from the previous dataset, and consists of the trajectory on the attractor, yet split in 9 chunks of equal size and reshuffled, such that each chunk is thus composed by 3000 samples. As it will be clear soon, this dataset will allow us to investigate some effects of memory.



(a) Illustration of the $\mathcal{T}_{\text{ergo}}$ dataset. (b) Illustration of the $\mathcal{T}_{\text{ens}}$ dataset. (c) Illustration of the $\mathcal{T}_{\text{FP}}$ dataset.

Figure 2: Illustration of the different training sets.

4.2.2 Fixed points, $\mathcal{T}_{\text{FP}}$

The training set $\mathcal{T}_{\text{FP}}$ exploits prior knowledge on the structure of the system at hand, namely the fixed points. As mentioned in the Introduction, it is well known dynamical systems are organized around critical features, such as fixed points, and that invariant sets provide information on the behavior of the solution, [12]. Besides fixed points, these features include higher-dimensional invariant sets such as slow invariant manifolds. The sampling strategy here consists in acquiring observations from the system originating from its (unstable) fixed points, in contrast with other training sets obtained from the attractor. Specifically, three 3000-sample trajectories are acquired from each of the three fixed points of the Lorenz'63 system (see Sec. 4.1). These trajectories originate from a small deviation from each fixed points along each of the $d = 3$ directions given by the eigenvectors of the local Jacobian matrix. Note that if the Jacobian is not available, a set of orthonormalized deviations can also be used. The resulting training set then consists of $3 \times 3 \times 3000 = 27,000$ samples, consistent in size with the other training sets illustrated in Fig. 2.

4.2.3 Ensemble sampling, $\mathcal{T}_{\text{ens}}$

With this strategy, the dynamical system is sampled along short trajectories on the attractor. Nine trajectories are considered, each originating from a state randomly chosen on the attractor. Each trajectory is 3000 sample long as for the other strategies introduced above so that the total size of the training set is again 27,000 samples, similar to the reference training set $\mathcal{T}_{\text{ergo}}$. This training set is representative of the widespread situation where information about a system comes in several independent glimpses, as resulting from several measurement campaigns. It is illustrated in Fig. 2.

5 Results

We now assess the quality of the models learned from various observation datasets of the Lorenz'63 system.

5.1 How much training information?

The influence of the amount of information to train a model is first considered. Two models with the same architecture, cf. Sec. 3, are trained from two different datasets. These datasets consist of observations of the solution of the Lorenz system along the *same* trajectory on the attractor, but of different length. One dataset, $\mathcal{T}_{\text{ergo}}$, comprises 27,000 samples while the second one $\mathcal{T}_{\text{short}}$ is its restriction to the first 3000 samples, so that $\mathcal{T}_{\text{short}} \subset \mathcal{T}_{\text{ergo}}$. Once trained, the resulting models, hereafter respectively termed F_{ergo} and F_{short}, are evaluated in terms of their long-term prediction capability.

Results are gathered on Fig. 3. The performance of the models is here evaluated when predicting from a known (*i.e.*, contained in their training set) and unknown (not contained) initial condition, to assess the

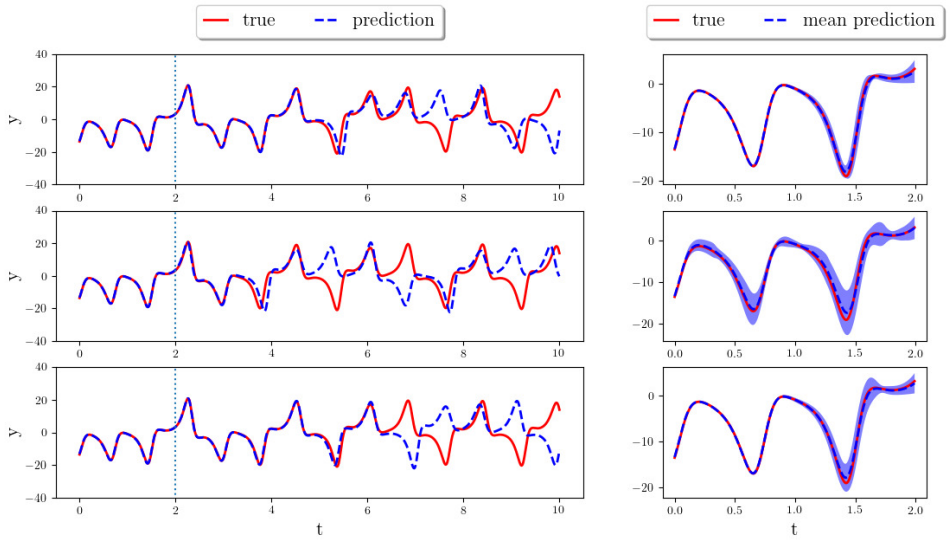


Figure 3: Left column: Time-evolution of the y-component of the Lorenz'63 system, both from the true (`true`, solid red) and the learned model (`prediction`, dashed blue) from different training sets and initial conditions. Right column: zoom on the [0, 2] time units interval, with the mean prediction (dashed blue line) from an ensemble of 100 trained models, and the associated 1 standard deviation.
Top row: The prediction of the best (out of the ensemble set) model trained from $\mathcal{T}_{\text{short}}$ is shown. The initial condition is drawn from the training set. The prediction is seen to remain reliable up to about 4.3 Lyapunov times (5 time units) before drifting significantly from the truth. Middle row: When the same model is evaluated with an unknown initial condition (test set), the performance deteriorates and the accuracy is lost after less than 4 time units. When a unique ergodic-compliant trajectory $\mathcal{T}_{\text{ergo}}$ is used to train a model (bottom row), the prediction from an unknown initial condition improves significantly.

generalization capability of the models. The model is the LSTM architecture described in Sec. 3.1, with a null initial memory.

A known initial condition is first considered. As can be seen from the top row (left column) where it is plotted against the truth (red), the F_{short} model allows a reasonable quality of the prediction (plotted in blue) over a significant time horizon of about 6 Lyapunov times (7 time units). Since training the models is achieved via a stochastic procedure, *e.g.* stochastic gradient descent and random initialization of the parameters, and the landscape of the loss function is not globally convex, the underlying optimization problem ends up in a local optimum and bears an aleatoric contribution. To strengthen the above conclusions, we hence consider an ensemble of 100 identical independent models and evaluate their performance. The right column of Fig. 3 shows the mean and standard deviation of the prediction of the ensemble of models for the same conditions as in the left column. From the top row, it is clear that the prediction of the ensemble of models is very consistent and that the learning is robust.

With the same short training set $\mathcal{T}_{\text{short}}$, the trained model is now evaluated on an initial condition not contained in the training data, see Fig. 3 middle row. Compared with the previous situation (top row), the performance is seen to deteriorate, both for a single model (left column) and in average (right). In particular, the time horizon before the prediction significantly differs from the truth is much shorter, indicating a poor generalizability properties.

When more training data are used, $\mathcal{T}_{\text{ergo}}$, the resulting model remains accurate over a long time horizon even when predicting from an unknown initial condition, as seen from Fig. 3 (bottom row). Again, this is a reliable finding as evidenced from the ensemble performance seen on the right column.

These results illustrate the fact that the training data, relied on to learn a model, have to be informative enough of the system under consideration to allow for generalizable models to be inferred. In particular, the superior performances shown in the top row, when the initial conditions are contained in the learning

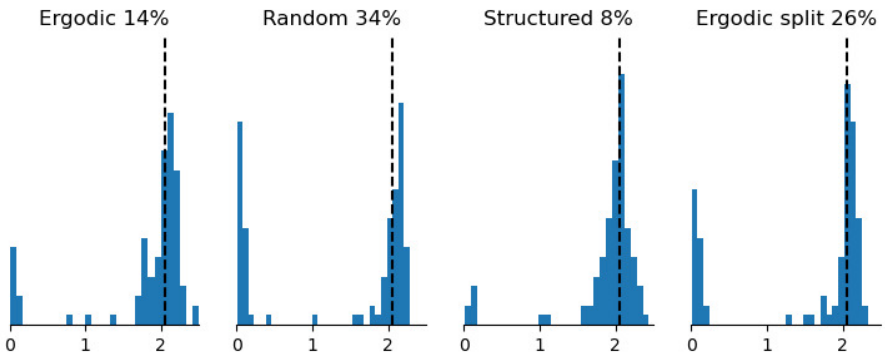


Figure 4: The quality of the 100 models trained with each of the 4 training strategies $\mathcal{T}_{\text{ergo}}$, $\mathcal{T}_{\text{ens}}$, $\mathcal{T}_{\text{FP}}$, and $\mathcal{T}_{\text{split}}$ is here appreciated in terms of the distribution of the resulting d_2 dimension. The dimension is estimated from a trajectory not belonging to the training set. For each training set, the proportion of models associated with an estimated dimension differing by more than 25% from the true dimension $d_{\mathbb{A}} = 2.06$ is also indicated.

dataset, appear to be due to overfitting.

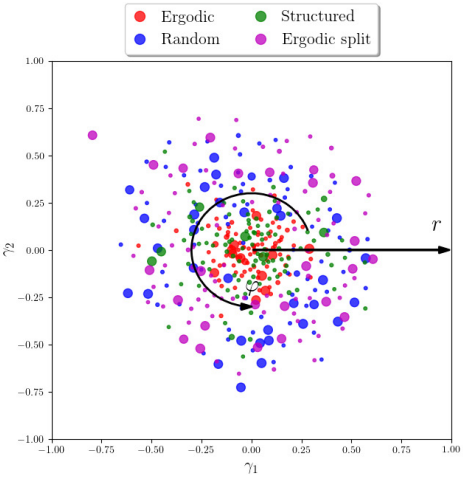
5.2 Relevance of the data

Besides the mere size of the training set, a critical aspect is its *relevance* for informing a model in view of a given objective function. To shed light on this pivotal aspect, the four different training sets introduced in Sec. 3.2, namely $\mathcal{T}_{\text{ergo}}$ ("Ergodic"), $\mathcal{T}_{\text{split}}$ ("Ergodic split"), $\mathcal{T}_{\text{ens}}$ ("Random") and $\mathcal{T}_{\text{FP}}$ ("Structured"), all of the same size of 27,000 samples, are now evaluated in terms of the quality of the associated resulting model. To alleviate any bias due to the initialization of the parameters, an ensemble of 100 models is trained for each situation and the comparison is made in terms of the ensemble performance.

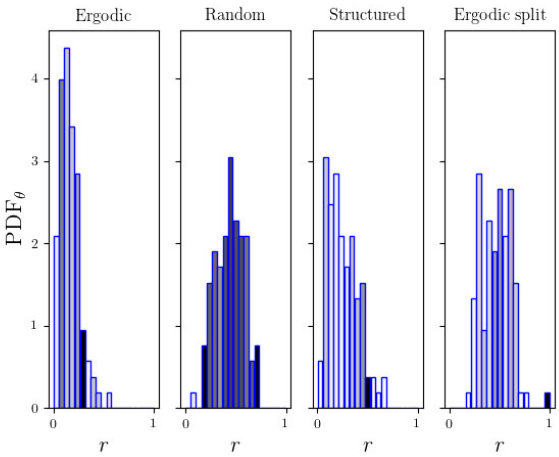
A first metric to quantify the quality of the resulting model is the d_2-dimension of the predicted model. This criterion is not associated with a given initial condition and is affected by the way the trained model consistently reproduces the fractal nature of the Lorenz'63 system. Results are gathered in Fig. 4 in terms of a histogram of the estimated d_2 dimension for the ensemble of 100 models for each of the four training sets. For a more quantitative appreciation, a threshold is defined when the dimension resulting from the trained model differs by more than 25 % from the truth $d_{\mathbb{A}} = 2.06$. As expected, the ergodicity-compliant training set $\mathcal{T}_{\text{ergo}}$ leads to a good performance, with most models associated with a d_2 dimension close to the true value and a few degenerate models such that $d_2 \approx 0$. Only 14 % of the models are beyond the threshold. Models trained from $\mathcal{T}_{\text{FP}}$ are also seen to achieve a similar level of performance. Again, almost all these models are associated with a rather good d_2 dimension, with very few degenerate models. Only 8% of the models are beyond the threshold, an even better performance than models trained from $\mathcal{T}_{\text{ergo}}$.

In contrast, the $\mathcal{T}_{\text{ens}}$ (random) and $\mathcal{T}_{\text{split}}$ (split) dataset leads to a rather poor performance model, with a significant part of the models associated with $d_2 \approx 0$ and respectively 34 and 26 % of the models beyond the threshold. It may be not quite surprising that models trained with $\mathcal{T}_{\text{ens}}$ and $\mathcal{T}_{\text{split}}$ are associated with a similar performance. Since the only difference between the two datasets is that the samples in $\mathcal{T}_{\text{ens}}$ originate from a collection of 9 *independent* segments of trajectory, randomly located on the attractor, while $\mathcal{T}_{\text{split}}$ is made of 9 *contiguous* segments. Yet, such behavior for the $\mathcal{T}_{\text{split}}$ dataset appears to be not trivial, because it contains exactly the same information of the $\mathcal{T}_{\text{ergo}}$ one about the phase space.

Further assessment of these observations can be made using a different technical tool, namely t-distributed stochastic neighbor embedding (t-SNE) of the parameters, shown in Fig. 5a. It is essentially a dimensionality reduction technique convenient for assessing high-dimensional quantities [29]. The model parameters vectors are here mapped onto a 2-dimensional space where each model can hence be represented in terms of its corresponding coefficients γ_1 and γ_2.



(a) t-distributed stochastic neighbor embedding (t-SNE) of the parameters of the trained models analyzed in Fig. 4. Each dot corresponds to a different model and its size visually indicates the accuracy: the larger the dot, the larger the error in respect of $d_{\mathbb{A}}$.



(b) Radial distribution of the t-SNE projected model parameters. r indicates the distance from the barycenter of the cluster in Fig. 5a, while the distribution of the models is reported on the vertical axis using the same number of bins. We order from light cyan to darker blue models performing progressively worse in terms of resulting d_2 accuracy; note that the level set of the blue gradient is normalized for each subplot, such that only a qualitative information on the distribution along r is included in the graphs.

Figure 5: Illustration of the model parameters via a t-SNE representation. The LSTM memory is initialized to zero.

F_{ergo} and F_{FP} are mostly clustered in the center of the points cloud, indicating that, while the models are different, they are however similar to each other in terms of parameter distribution. Instead, the models obtained from the random sampling strategy ($\mathcal{T}_{\text{ens}}$) and the ones obtained stacking chunks from an ergodicity-compliant trajectory ($\mathcal{T}_{\text{split}}$) are distributed in the outer region.

LSTM parameters are initialized as independent realizations of a Gaussian random variable, whose t-SNE representation is typically visually close to a Gaussian as well since it penalizes deviation from a t-Student distribution. Recent works have shown that neural networks associated with parameters weakly evolving during training while yet able to significantly improve the loss function, are often associated with good generalization properties, [16]. In the present study, the t-SNE representation of models trained from the ergodicity-compliant and the structured datasets both appear close to a Gaussian distribution, a further indication that they might enjoy good generalization properties. In contrast, the t-SNE distribution of the models trained from $\mathcal{T}_{\text{ens}}$ or $\mathcal{T}_{\text{split}}$ appear to deviate significantly from their initial distribution, which corroborates well with their poor performance.

This distribution indicates that the best models are robust, in the sense of having similar parameters. Furthermore, the training using the information about fixed points appears equivalent to that based on a long trajectory. This observation is also quantified in terms of a radial distribution, see Fig. 5b. Both the models trained from $\mathcal{T}_{\text{ergo}}$ and $\mathcal{T}_{\text{FP}}$ present a distribution most dense around zero, while $\mathcal{T}_{\text{ens}}$ and $\mathcal{T}_{\text{split}}$ lead to a bias in the parameter distribution, hence being associated with poor models.

5.3 Handling the memory

The role of the memory in the recurrent model is now studied. Instead of the null memory (initialized to zero) considered so far, an initial memory chosen at random, drawn from a multivariate Gaussian distribution $\mathcal{N}(0, \mathbf{I})$ is used. Compared to the null initial memory, the performances are seen to significantly increase for all models, as evidenced in Fig. 6. This is consistent with the fact that a non-zero hidden state improves the

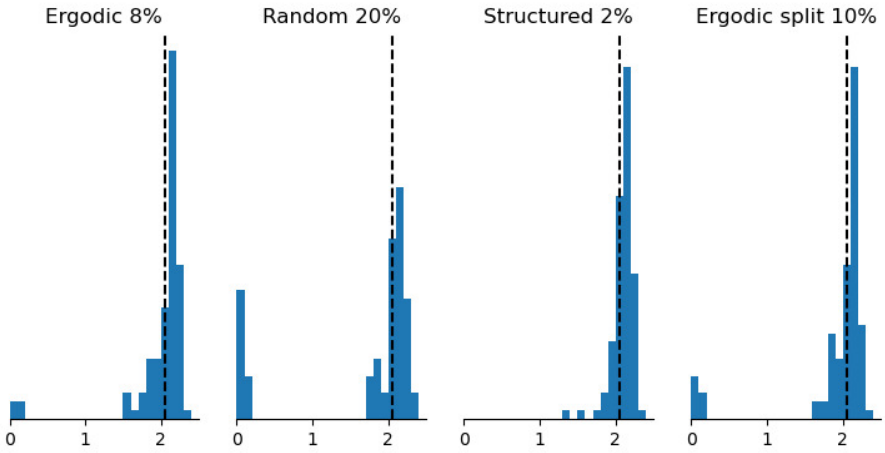


Figure 6: Radial distribution of the t-SNE projected model parameters. The LSTM memory is here initialized at random.

numerical conditioning of the training steps and prevents degenerate gradients, [10].

The contrast in performance between models learned from $\mathcal{T}_{\text{ergo}}$ and $\mathcal{T}_{\text{split}}$, which rely on the *same* information, leads to identify a key point. The LSTM model trained with $\mathcal{T}_{\text{ergo}}$ learns to best fit one long trajectory. In the case of $\mathcal{T}_{\text{split}}$, it learns to best fit 9 short trajectories, which however, when stitched together, match the $\mathcal{T}_{\text{ergo}}$ long trajectory. Yet, their performances are consistently very different in the null initialization situation, emphasizing that the initial memory of the retained recurrent network (LSTM) dramatically affects the resulting performance. In the case of $\mathcal{T}_{\text{ergo}}$, the impact of an incorrect initial memory is diluted within a long trajectory and weakly affects the learning quality. When short trajectories are considered instead, the relative impact of an incorrect initialization is stronger and can introduce a significant bias, hence a poor resulting model.

This observation is further supported by the model learned from $\mathcal{T}_{\text{FP}}$. While it also relies on samples originating from 9 distinct, non sequential, segments, its performance is good, as evidenced by Fig. 4. What is unique about this particular training set is that the samples are from segments of trajectories originating from (unstable) fixed points of the Lorenz'63. These points of special interest are associated with a locally linear dynamics, and this allows the incorrect initial memory to be progressively wiped out in time without significantly affecting the predicted dynamics. In this regard, fixed points are privileged training elements which allow to reliably learn memory-based models by initializing the training in a configuration where the memory does not have a harmful impact.

6 Concluding discussion

In this work, we have performed a thorough analysis of some aspects inherent to the process of learning a complex physical system from observational data. More specifically, we have considered the following issues: (i) What is the impact of the initial state of the retained model on its resulting performance? (ii) Is there a minimum amount of data needed to obtain a robust model able to generalize? (iii) Is it possible to go beyond this limitation thanks to some knowledge about the system?

The first issue is related to the objective of discussing best-practice guidelines for practitioners. The others are more fundamental. The second is related to what is known for classical embedding approaches, which is bound to fail when time-series data are not enough to reconstruct the entire phase-space. The last one explores perspectives to find a physics-informed learning approach capable of overcoming some of the intrinsic limitations of data-driven supervised learning.

We have considered the classical Lorenz'63 system as the benchmark problem, because of its simplicity and

hence the possibility to carry out an exhaustive campaign of simulations. Moreover, the system is known in details and allows to investigate the importance of singular points of the dynamics. For the learning method, we have considered an LSTM, an established recurrent structure often used in such context.

In these memory-based models, flushing an initially incorrect memory is crucial for obtaining a relevant and faithful model able to generalize its prediction beyond the sole situations encountered during the training step. This issue may be particularly severe if one learns from a set of short-termed observations. Our numerical experiments show clearly that there is an impact of the initialization of the memory on the quality of the predictions provided by the LSTM models. As a matter of fact, it should be always initialized as random to avoid bias in the learning since, in all cases, we have obtained better performance in such a configuration. In particular, when the initial condition is random, we have found little difference in the results obtained by training on a long trajectory or using exactly the same data but randomly shuffled in several short trajectories, consistently with the problem of exploring the phase-space of a dynamical system.

Generally speaking, we have found that, in absence of prior information on the system at hand besides an estimated upper bound of its dimension, the ergodic theory, and notably the Kac lemma, provides a criterion on the minimal amount of data necessary to train a model. This amount of data grows exponentially with the dimension of the attractor. Models trained on a smaller amount of training data have been shown to consistently lead to poor performance, as illustrated both via the time-series prediction and the resulting model estimated dimension assessment criteria.

Remarkably, numerical experiments have neatly shown that starting from the fixed points of the dynamical system constituted a workaround in learning the dynamics of the system from situations where the memory is essentially harmless to the learning process. Furthermore, this choice leads to consistent and accurate models even with a decreased amount of data with respect to what prescribed by ergodic theory. The information of the structure of the dynamics appears therefore key to improve the learning process.

On-going efforts further explore the idea that an additional benefit of training a model from fixed points is that the resulting dataset encompasses a wide range of solution regimes, from a linear behavior close to the fixed points where the non-linearities are comparatively weak, to a full nonlinear regime when the state reaches the attractor. This essentially allows to disentangle the contribution of the different terms of the underlying system and to sequentially focus the learning effort on some solution regimes from specific parts of the training set. In this sense, this is reminiscent of a form of curriculum learning approach.

While we here focused on the Lorenz'63 system, we believe the findings of this work to be widely applicable and to provide some evidence-based good practice for data-driven modeling of physical systems. Work in progress concerns a finer analysis of the interplay between the structure of the dynamical system and the learning process in order to explain why fixed points are so informative, in the hope to propose an active learning strategy aimed at leveraging the underlying physics of a dynamical system.

Acknowledgement

This work was funded by the French Agence Nationale de la Recherche via the *Flowcon* project (ANR-17-ASTR-0022). The last author gratefully acknowledges stimulating discussions with Alex Gorodetsky.

References

[1] G. Boffetta, M. Cencini, M. Falcioni, and A. Vulpiani. Predictability: a way to characterize complexity. *Physics reports*, 356(6):367–474, 2002.

[2] F. Borra, A. Vulpiani, and M. Cencini. Effective models and predictability of chaotic multiscale systems via machine learning. *Physical Review E*, 102(5):052203, 2020.

[3] S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.

[4] J. P. Crutchfield and B. S. McNamara. Equation of motion from a data series. *Complex systems*, 1(417-452):121, 1987.

[5] M. de Hoop, R. Baraniuk, J. Bruna, M. Campillo, H. Jasperson, S. Mallat, T. Nguyen, and L. Seydoux. Unsupervised learning for identification of seismic signals. In *Geophysical Research Abstracts*, volume 21, 2019.

[6] J.-P. Eckmann and D. Ruelle. Ergodic theory of chaos and strange attractors. *Rev. Mod. Phys.*, 57:617–656, Jul 1985.

[7] A. Eftekhari, H. L. Yap, M. B. Wakin, and C. J. Rozell. Stabilizing embedology: Geometry-preserving delay-coordinate maps. *Physical Review E*, 97(2):022222, 2018.

[8] W. Garnett and L. Campbell. *The Life of James Clerk Maxwell.* 1882.

[9] F. A. Gers, D. Eck, and J. Schmidhuber. Applying lstm to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri-01*, pages 193–200. Springer, 2002.

[10] M. M. Ghazi, M. Nielsen, A. Pai, M. Modat, M. J. Cardoso, S. Ourselin, and L. Sørensen. On the initialization of long short-term memory networks. *ArXiv*, 1912.10454, 2019.

[11] J. F. Gibson, J. D. Farmer, M. Casdagli, and S. Eubank. An analytic approach to practical state space reconstruction. *Physica D: Nonlinear Phenomena*, 57(1):1 – 30, 1992.

[12] R. Gilmore, J.-M. Ginoux, T. Jones, C. Letellier, and U. S. Freitas. Connecting curves for dynamical systems. *Journal of Physics A: Mathematical and Theoretical*, 43(25):255101, 2010.

[13] J.-M. Ginoux and C. Letellier. Flow curvature manifolds for shaping chaotic attractors: I. Rössler-like systems. *J. Phys. A: Math. Theor.*, 42:285101, 2009.

[14] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning.* MIT press, 2016.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[17] M. Kac. *Probability and related topics in physical sciences*, volume 1. American Mathematical Soc., 1959.

[18] H. Kantz and T. Schreiber. *Nonlinear time series analysis*, volume 7. Cambridge university press, 2004.

[19] K. Kashinath, M. Mustafa, A. Albert, J.-L. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, A. Manepalli, D. Chirila, R. Yu, R. Walters, B. White, H. Xiao, H. A. Tchelepi, P. Marcus, A. Anandkumar, P. Hassanzadeh, and Prabhat. Physics-informed machine learning: case studies for weather and climate modelling. *Phil. Trans. Roy. Soc. A*, 379:20200093, 2021.

[20] N. Kuznetsov, T. Mokaev, O. Kuznetsova, and E. Kudryashova. The Lorenz system: hidden boundary of practical stability and the Lyapunov dimension. *Nonlinear Dyn.*, 102:713–732, 2020.

[21] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of atmospheric sciences*, 20(2):130–141, 1963.

[22] G. Paladin and A. Vulpiani. Anomalous scaling laws in multifractal objects. *Physics Reports*, 156(4):147–225, 1987.

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

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

[25] H. Poincaré. *Les méthodes nouvelles de la mécanique céleste*, volume 3. Gauthier-Villars et fils, 1899.

[26] M. Raissi, P. Perdikaris, and G. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019.

[27] T. Sauer, J. A. Yorke, and M. Casdagli. Embedology. *Journal of statistical Physics*, 65(3-4):579–616, 1991.

[28] F. Takens. Detecting strange attractors in turbulence. In D. Rand and L.-S. Young, editors, *Dynamical Systems and Turbulence, Warwick 1980*, pages 366–381, Berlin, Heidelberg, 1981. Springer Berlin Heidelberg.

[29] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

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

[31] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, J. Pfrommer, A. Pick, R. Ramamurthy, M. Walczak, J. Garcke, C. Bauckhage, and J. Schuecker. Informed Machine Learning - A Taxonomy and Survey of Integrating Knowledge into Learning Systems. *IEEE Transactions on Knowledge and Data Engineering*, 2021. accepted.

[32] H. Whitney. Differentiable manifolds. *Annals of Mathematics*, 37(3):645–680, 1936.

[33] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, and P. Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 394:56–81, 2019.