Chaos, Solitons and Fractals 154 (2022) 111612

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

Nonlinear Science, and Nonequilibrium and Complex Phenomena

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

Dynamical time series embeddings in recurrent neural networks

Gonzalo Uribarri*, Gabriel B. Mindlin

IFIBA, CONICET and Departamento de Física, FCEyN, UBA, Buenos Aires, Argentina

ARTICLE INFO

Article history:
Received 8 June 2021
Revised 20 September 2021
Accepted 2 November 2021
Available online 28 November 2021

Keywords:
Recurrent neural networks
Time series
Dynamical systems
Embedding
Forecasting

ABSTRACT

Time series forecasting has historically been a key research problem in science and engineering. In recent years, machine learning algorithms have proven to be a very successful data-driven approach in this area. In particular, Recurrent Neural Networks (RNNs) represent the state-of-the-art algorithms in many sequential tasks. In this paper we train Long Short Term Memory networks (LSTM), which are a type of RNNs, to make predictions in time series corresponding to the observation of a single variable of a chaotic system. We show that, under certain conditions, networks learn to generate an embedding of the data in their inner sate that is topologically equivalent to the original strange attractor. Remarkably, this resembles standard forecasting methods from nonlinear science in which the time series is embedded in a multi-valued space using Takens's delay embedding mechanism.

1. Introduction

Time series modeling is a key area of scientific research. Forecasting the future state of a given sequence is a critical part of real-world problems in a wide range of areas: from climate and biology to traffic and finance [1–5].

In many applications time series come from systems whose variables obey a set of differential equations. This is often the case in physics and engineering. To fully describe these systems, it would be ideal to find the governing set of equations, but we are usually unable to do that. This may be because we do not know the underlying mechanisms of interaction, or because we have no access to measure all the variables characterizing the system under study. In either case, we have to use some kind of data-driven approach to model the time series data.

During the 80 s and 90 s dynamicists developed a series of tools to characterize and forecast time series from dynamical systems [6–13]. These methods proved particularly well suited for time series coming from non-linear and chaotic systems, which were problematic for classical statistical approaches. The core idea behind many of these methods is to first make an embedding of the time series and then study the system in this new representation. An embedding is a multi-valued sequence of points which could be mapped into the original flow of the dynamical system by means of a smooth and invertible change of coordinates, i.e. is topologically equivalent to it. Making this reconstruction of the flow was possible by means of the Takens's theorem [14,15]. Recently, it has been shown that autoencoders, a type of dimensionality reduction algorithm, can also be used to find proper embeddings for chaotic time series [16,17].

Suppose we have a time series data $\{x_1, x_2, \ldots, x_N\}$ generated by regularly inspecting the value of a variable in a dynamical system with a sampling interval τ. The central hypothesis behind Taken's theorem is that the number of elements needed to determine the future of the series, N_1, is smaller than the total number of elements in the series, $N_1 \ll N$. That means that the information contained in a segment of the series $\{x_k, x_{k+1}, \ldots, x_{N_1+k-1}\}$ is enough to determine the following segment $\{x_{k+1}, x_{k+1}, \ldots, x_{N_1+k}\}$. In this way, we can represent the points $\{x_i\}$ in the time series by points in R^{N_1}. This process generates an embedded manifold, a submanifold of R^{N_1} to which the time series will be restricted. Takens's theorem states that, if the manifold holding the flow is of dimension d, a sub-set of $N_1 \geq 2d+1$ points are sufficient to generate an embedding. But this embedding is not optimal or unique. In most cases a lower value of N_1 can lead to a correct embedding if the time lag between the points, τ, is adequate [8]. Notice that in real world applications where the times series data is noisy, if the value of τ is not large enough, the variation of the signal will be dominated by noise and the embedding will fail to spam a good representation of the data, even when $N_1 = 2d+1$.

Nowadays, due to the increasingly availability of computing power and data, data-driven modeling is dominated by machine learning algorithms, in particular neural networks [18,19]. These "black-box" models are used to make predictions in all kinds of time series, without differentiating their nature [20–24]. Out of the different types of networks available, LSTM (Long Short-Term Memory) is one of the most successful for working with time se-

* Corresponding author at: IFIBA, CONICET and Departamento de Física, FCEyN, UBA, Ciudad Universitaria, pabellon 1, cp 1428, Buenos Aires, Argentina.
E-mail address: gonzalo@df.uba.ar (G. Uribarri).

https://doi.org/10.1016/j.chaos.2021.111612
0960-0779/

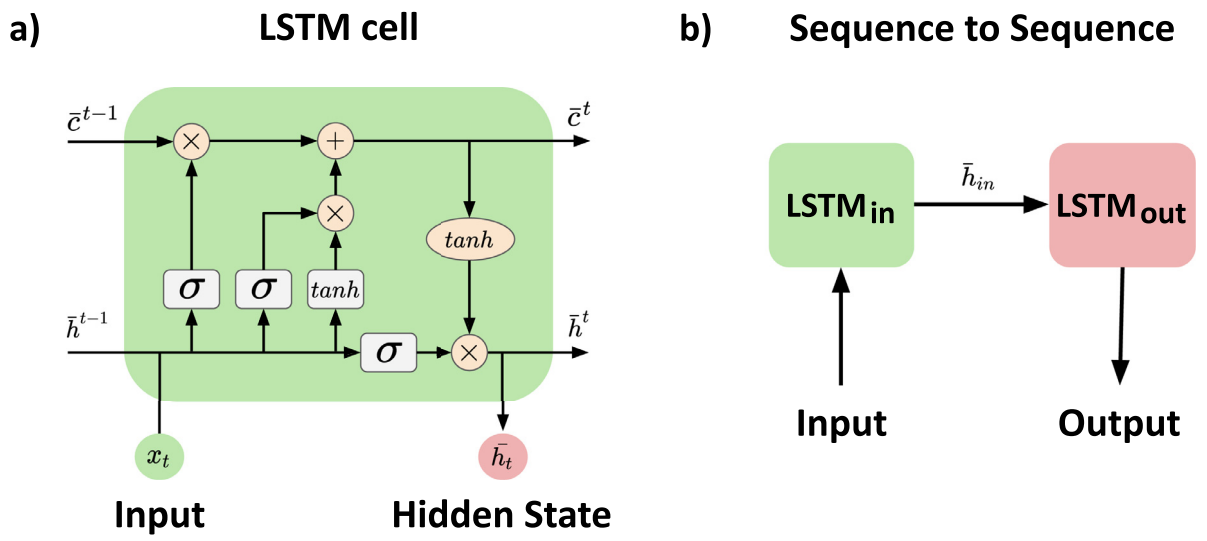

Fig. 1. a) Diagram of an LSTM cell. gray boxes indicate a neural layer, while orange circles indicate pointwise operations. The hidden state, h, and the cell state, c, carry information from one step of the network to the next. b) Diagram of sequence to sequence architecture. The input cell (encoder) reads and encodes the input sequence into a state vector $\bar{h}_{in}$, this vector is then passed to the output cell (decoder) to produce predictions. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ries [25–28]. As a Recurrent Network, it is designed to capture temporal correlations of an input signal by sequentially processing its elements. At each step, the network reads one element of the series, the input value x_t, and updates its inner states to new values, $\bar{h}_t$ and $\bar{c}_t$ (See Fig. 1a). Each input value x_t can be a scalar or a vector depending on the problem data, while the dimension of the inner states is an important hyperparameter to choose. LSTMs are used to build sequence to sequence architectures, which represent the state-of-the-art approach in many sequential tasks [29–36]. In this kind of architecture, an input LSTM cell (aka encoder) reads and encodes the input sequence into a state vector, $\bar{h}_{in}$, this vector is then passed to the output LSTM cell (aka decoder) to produce predictions (See Fig. 1b).

In recent years, an increasing number of studies have investigated the predictive power of recurrent neural networks when trained to predict time series coming from dynamical systems, showing the capability of this type of networks to capture complex nonlinear dynamics. In [28, 37, 38] and [39] the authors investigated the long-term performance of LSTMs on nonlinear systems by evaluating the accuracy and stochastic characteristics of the predicted signal. In [40] the authors tested LSTMs predictions outside the state space used for training by exploring dynamical properties of the solutions such as bifurcations and characteristic exponents. In [41, 42] and [43] authors compare the performance of different types of architectures (MLPs, RNNs, LSTMs and GRUs) in predicting systems with nonlinear dynamics.

In this work, we train LSTM networks in a sequence to sequence architecture to perform the task of predicting chaotic time series generated from a Rössler system. We choose to work with a chaotic system because, even when the system is low-dimensional, the generated time series never repeats itself and the task of predicting its future state is highly not trivial. We analyze the encoding of the signal made by the LSTM cell in the hidden space, $\bar{h}_{in}$, and check whether the reconstructed flow in this multi-valued environment is equivalent to that of the original dynamical system. To determine this equivalence, we inspect the topological organization of the unstable orbits coexisting with the chaotic attractor by means of the Linking Numbers (See subSection 2.3 Linking Numbers). Then, we will compute the topological organization of the same segments in the hidden space representation. We will interpret the match between the topological organization of the represented segments and the actual orbits as a signature of equivalence between the original and the reconstructed flows. Notice that our analysis differs from previous work on the topic in one key aspect: instead of focusing on analyzing the predicted signal, we investigate the internal representation of the data made by the recurrent network.

We show that, under certain conditions, these networks generate a topologically correct embedding of the time series in the hidden space of the input LSTM cell. Remarkably, this result suggests that the process learned by the recurrent neural network to make predictions resembles the standard forecasting methods developed by the dynamicists, in which the time series is embedded in a multi-valued space to be studied. We also discuss the conditions needed for this to happen and how knowing that the flow in the hidden space is equivalent to the original can help to interpret the forecasting process learned by recurrent neural networks.

The work is organized as follows. In the second section, we describe the dataset generation process, the architecture of the trained networks and the topological indeces that we use to compare flows. The third section, Results, presents the results obtained for two different training scenarios and analyses the conditions under which the flow in the hidden space turns out to be an adequate embedding of the training data. In the final section, Discussion, we discuss the implications of our finding in the context of an ongoing expansion of the use of neural networks in data-driven prediction of time series.

2. Methods

2.1. Dataset

The first step to generate our dataset was to integrate a Rössler system,

$$\frac{dx}{dt} = -y - z$$

$$\frac{dy}{dt} = x + ay$$

$$\frac{dz}{dt} = b + z(x - c)$$

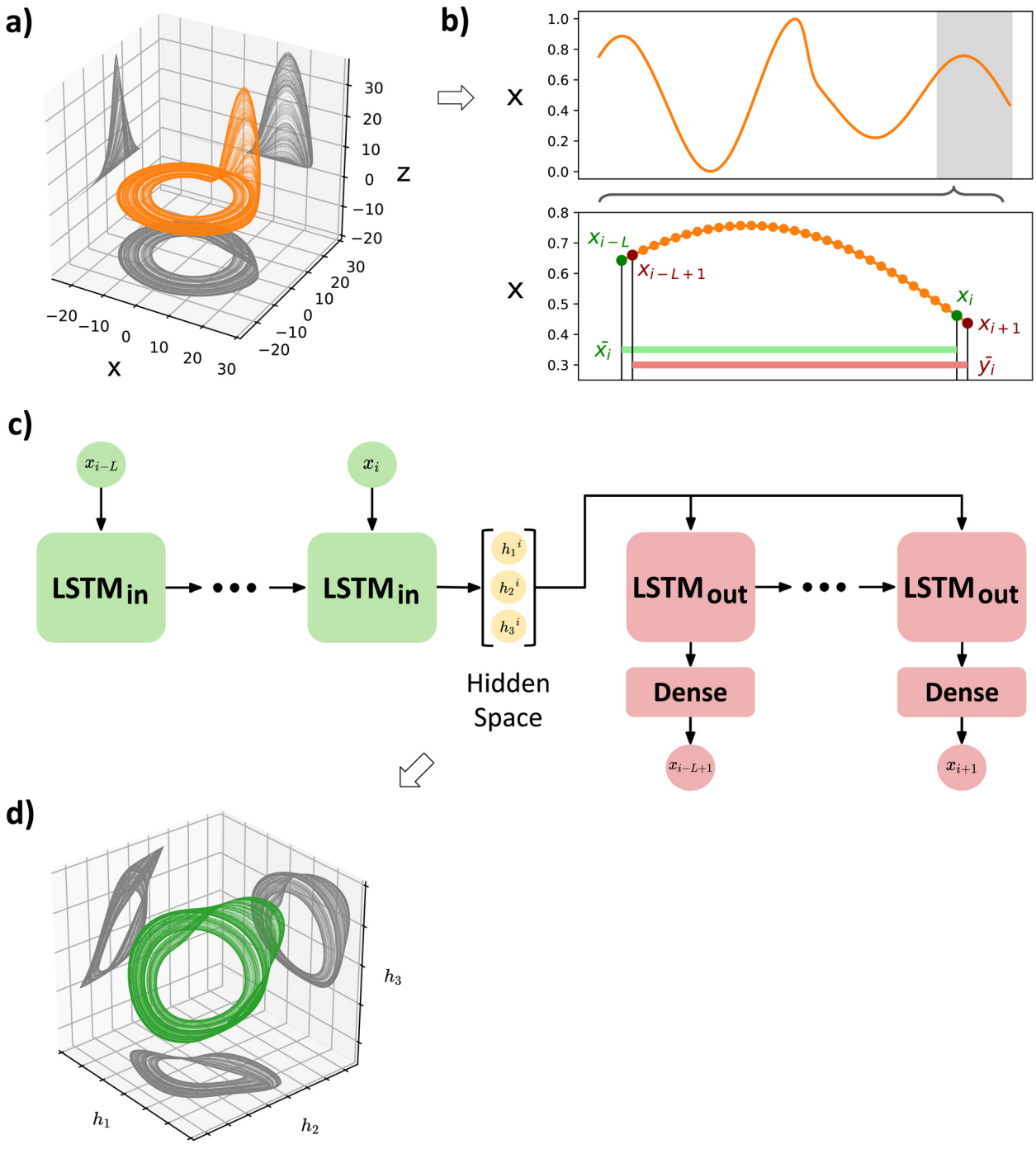

Fig. 2. Embeddings generated by LSTMs networks. a) We simulated a strange attractor using a Rössler dynamical system. b) Time series made from values of the x variable were used to train the neural network: for each training instance, L consecutive points $\{x_{i-L}, \ldots, x_i\}$ were used as input and the following L consecutive points $\{x_{i-L+1}, \ldots, x_{i+1}\}$ as the desire output. c) The architecture of our network. One input LSTM cell reads the input data and generates the embedding, then an output LSTM cell connected to a dense layer constituted by one single output neuron generates the forecast. The cost function being minimized is the Mean Square Error between this output and the actual data. d) Example of a reconstructed attractor in the Hidden Space.

with $a = b = 0.1$ and $c = 14$ using a fourth order Runge-Kutta method. The time series corresponds to the scalar value of the x variable, sampled every $\tau = 0.08$ time units (see Fig. 2a). To simulate more realistic data and add robustness to the results, we added noise to the time series in the training process. The added noise was Gaussian white noise, with zero mean, and a standard deviation equivalent to 2% of the standard deviation of x (different for each trained model).

In order to train the network, we partitioned the time series in segments of $L = 32$ points: $\{x_1, x_2, \ldots, x_{32}\}$, $\{x_2, x_3, \ldots, x_{33}\}, \ldots$. Each of these segments include data corresponding to a time span of approximately half the duration of the period-1 orbit of the attractor. We then took pairs of subsequent segments: the first to be used as the network input, $\bar{x} \in \mathbb{R}^{32}$, and the second one as the desire output, $\bar{y} \in \mathbb{R}^{32}$ (see Fig. 2b). Given the dimensionality of the attractor, Takens's theorem guarantees that each of these segments has enough information to unequivocally represent a particular state of the system.

The result of this process was 62,461 pairs of $(\bar{x}, \bar{y})$ elements, the input and target datasets $(\bar{X}, \bar{Y})$. Finally, a train-test split was

performed on the data. We use the data from the first 80% of the time series to train the network, and the data from the last 20% to test it.

2.2. Neural networks

The "unrolled" architecture of our sequence to sequence network is shown in figure (see Fig. 2c). The input LSTM cell reads the 32 elements of one input sequence, $\bar{x}_i \in \mathbb{R}^{32}$, and generates an encoding, $\bar{h}^i \in \mathbb{R}^H$, which is equal to the last value of its hidden state. Then an output LSTM cell receives this state and sequentially generates 32 hidden states. These states are read by an output neuron (indicated as Dense in see Fig. 2c) that transform each of this states into a one-dimensional value, generating the predicted output $\bar{y}_i^{pred} \in \mathbb{R}^{32}$.

We choose the dimension of the hidden state, H, in both LSTM cells to be 3. This is the optimal dimension to generate a proper embedding and allow us to use a topological index called Linking Number to study the encoding. We used hyperbolic tangent as activation function for both LSTMs cells and linear activation for the neuron at the output.

To implement the neural networks, we used Keras 2.3.1 with Tensorflow 1.15.2 as backend. The loss function minimized during the training was the Mean Square Error (MSE) between the desire output $\bar{y}$ and the network exit $\bar{y}_{pred}$. We used the Adam algorithm as optimizer, with learning rate $lr = 0.001$ and exponential decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$ [44]. A batch size of 64 was used and the models were trained for 201 epochs. A systematic and exhaustive search in the hyperparameter space was not performed to minimize the MSE in the test set.

2.3. Linking number

A strange attractor is a non-trivial invariant subset of the phase space of a dynamical system towards which a set of initial conditions is attracted. The topology of a strange attractor can be precisely quantified by studying the unstable periodic orbits coexisting with it [45]. If the attractor is three-dimensional, the organization of the orbits can be characterized by how they are knotted, and how they wind around each other. This topological feature is described by a numerical invariant called the Linking Number.

We use the Linking Numbers between the periodic orbits as a way to unequivocally determine whether the reconstructed attractor in the hidden space was equivalent to the original. We developed a function in Python to compute the Linking Number between two oriented curves in three dimensions, the source code is available on a public repository [16,46].

In order to compare it with the encoding performed by the recurrent networks, we need to compute the characteristic Linking Numbers between a set of periodic orbits in the Rössler attractor. Ideally, one should compute the topological organization of all the orbits, but it has been shown that low period orbits carry the information necessary to constrain the template organizing the complete flow [9,45]. To that end, we identified segments which were good approximations of three low period unstable periodic orbits. These orbits coexist with the chaotic solution obtained in the numerical simulations of our dynamical system (see Fig. 3a). Then, we calculated the Linking Numbers between these approximations of the orbits numerically, obtaining: *Linking* $(P1, P2) = -1$, *Linking* $(P1, P3) = -1$, *Linking* $(P2, P3) = -2$. We refer as $P1$ to the period-1 solution, $P2$ to the period-2 solution, and $P3$ to one of the two period-3 solutions in the Rössler attractor.

3. Results

3.1. Predicting the following segment

First, we trained 70 sequence-to-sequence models according to the process described in the previous section. Note that, despite following the same procedure, each training process is different due its stochastic nature (initialization of parameters, noise generation and shuffling of the training set). In Fig. 3c we present the result of the training process for 201 epochs. In red and blue, evolution of the Mean Square Error during the training for test and training sets; lines indicate the median and shadows the Interquartile Range. Upon completion of the training process, the Mean Square Error of training and test sets have reached a plateau for most models.

In order to study the encoding of the time series made by the LSTM network, we took the activation of the hidden neurons in the last step of the input LSTM, ${h_1}^i$, ${h_2}^i$and ${h_3}^i$, as a multi-valued environment. In Fig. 2d we display the flow generated by the training set in this multi-valued environment for one of the networks after competition of the training process. We evaluated the topological organization of the orbits in the hidden space by computing the linking numbers of the periodic orbits in this new representation. In Fig. 3b, we display the reconstruction for the three segments approximating the periodic solutions. In the case we used as example, the network training results in a correct embedding: the topological organization of the periodic orbits in the hidden space is identical to that of the original orbits (Fig. 3a).

We examined the topological structure of the orbits in the hidden space of all the trained models at different epochs during the training process and compute the proportion of the models with flows equivalent to that of the original attractor. The percentage of models with correct topology as a function of training epoch is displayed in Fig. 3c. Notice how the number of models with correct topologies increases as the networks learn to make better predictions and the MSE between the predicted and the real target segment decreases. After the first epoch, only 11% of the models lead to the right topological organization, whereas after 201 epochs 94% of them do.

The 6% of models with wrong topology were models that did not reach a good minimum of the cost function during training phase. There is a significant difference between the Mean Squared Error of the test set in these models, $4.94\ (4.25 - 5.21) \times 10^{-4}$, and the ones with correct topology, $1.59(1.29 - 2.12)\ \times 10^{-4}$. Values are presented as median (interquartile range).

In this first numerical experiment, the recurrent networks were trained with the objective of, given an input segment $\{x_{i-L}, \ldots, x_i\}$, having to predict the following segment $\{x_{i-L+1}, \ldots, x_{i+1}\}$, with $L = 32$. It is remarkable that, in order to achieve this, the LSTM generates an encoding in the hidden space that is topologically equivalent to the original attractor. In the text below, we discuss the reasons and conditions for this to happen.

In light of Takens's theorem, we know that both the input and target data, $(\bar{X}, \bar{Y})$, lay on manifolds which are embeddings of the original Rössler Attractor in $\mathbb{R}^L$. Let's call $\mathcal{M}_{in} \subset \mathbb{R}^{32}$ and $\mathcal{M}_{out} \subset \mathbb{R}^{32}$ to the input and target manifolds. So, our objective function, $\mathcal{F}$, is a one-to-one mapping from points on $\mathcal{M}_{in}$ to points on $\mathcal{M}_{out}$, i.e. $\mathcal{F} : \mathcal{M}_{in} \rightarrow \mathcal{M}_{out}$. By training our model, f_θ, we are trying to find a set of parameters, θ^*, so that our recurrent neural network is a good approximation of our objective function $\mathcal{F}$ in terms of the Mean Squared Error. But our network has an important constrain: the information about the input has to be carried by the hidden state $\bar{h} \in \mathbb{R}^H$. So, for f_θ be able to do a one-to-one mapping from $\mathcal{M}_{in}$ to $\mathcal{M}_{out}$, H has to be equal or bigger than the dimension of the original attractor. In our case, $H = 3$ is the optimal integer to fulfill this condition.

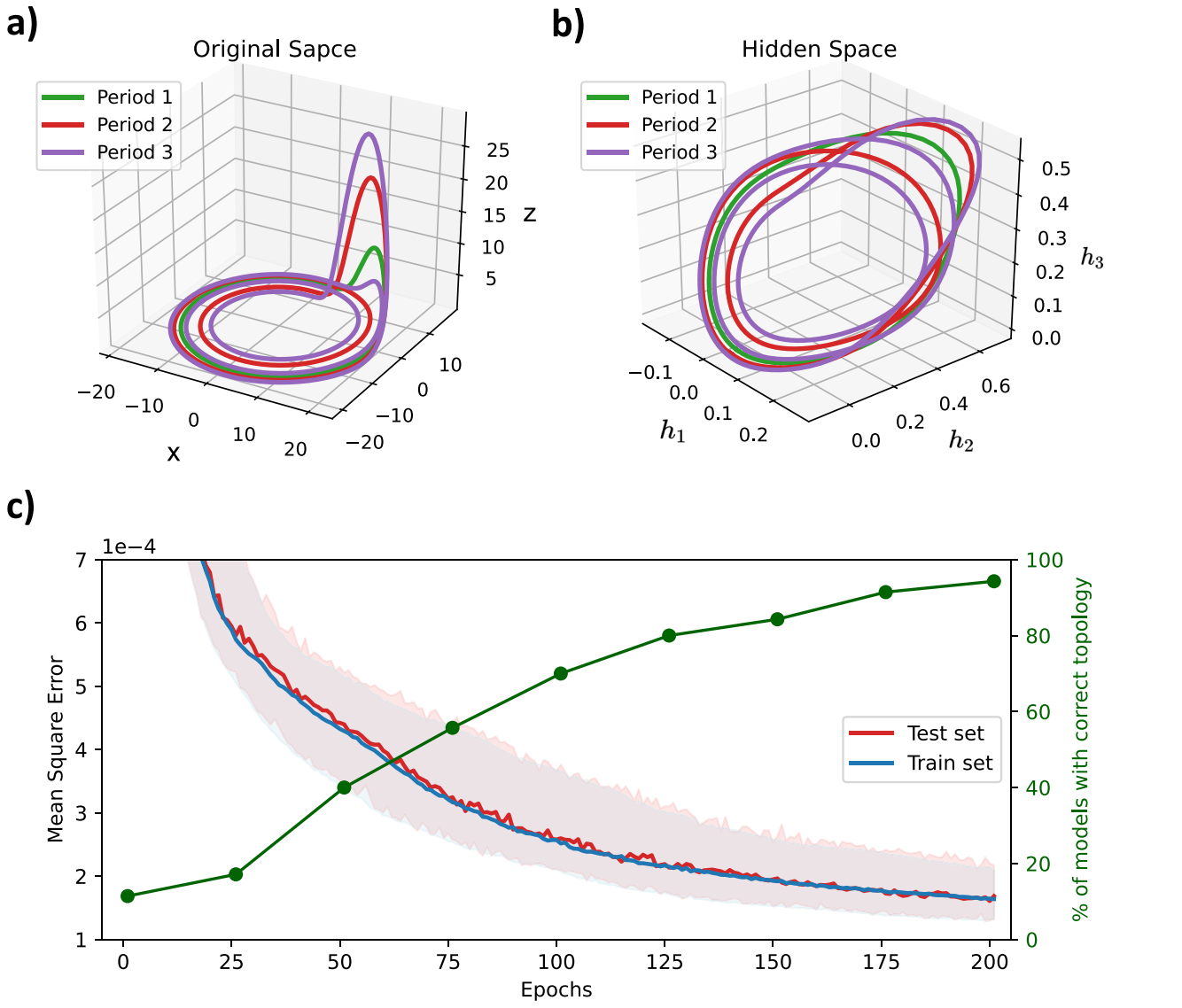

Fig. 3. Results. a) Comparison between the original attractor, and one reconstructed in the hidden space. We display the approximations of the periodic orbits in the original flow, and in the reconstructed one. In this example, both for the original and the reconstructed attractors, these orbits present linking numbers: *Linking* $(P1, P2) = -1$, *Linking* $(P1, P3) = -1$ and *Linking* $(P2, P3) = -2$. b) Result of the training process for 70 models. In red and blue, evolution of the Mean Square Error during the training for test and training sets, lines indicate the median and shadows the Interquartile Range. In green, we show the percentage of models whose topological organization of the unstable orbits in hidden space is the same as that of the original attractor. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

If the network f_θ is a good approximation of the objective function $\mathcal{F}$, it will also generate a one-to-one mapping between the input and the target data, $\mathcal{M}_{in}$ and $\mathcal{M}_{out}$. This means that, if f_θ is complex enough and properly trained, it will ensure a bijection between $\mathcal{M}_{in}$ and the representation of the data in the hidden space, $\mathcal{M}_h$, which acts as a bottleneck for the information flowing from the input to the output. This bijection is the key condition needed for the encoding made by the LSTM to be an embedding of the original attractor.

Attractors can also be characterized by the Maximal Lyapunov Exponent (MLE), a quantity that measures the rate of separation of infinitesimally close trajectories and is related to the notion of predictability for a dynamical system [47]. As an additional analysis, we computed the MLE for the original Rössler attractor, $L = 0.074$, and for the reconstructed attractors in the hidden space. We found $L = 0.074$ (0.072–0.077) for models where training led to a correct topology and $L = 0.084$ (0.081–0.087) for models where the training led to wrong topologies. We can then state that the models that reached a good minimum during training present a value for the largest Lyapunov exponent close to the one computed for the original attractor. However, this subject requires further investigation. Future work can explore different metric properties, such as the fractal dimension and the relative metric entropy, on attractors with different parameter values and noise levels [48–50].

3.2. Predicting N steps ahead

It is interesting to explore what happen when, instead of predicting the following segment, we train a sequence to sequence recurrent network to predict certain number of steps ahead in the time series. This is a conventional task these kind of networks are trained to achieve.

We used the exact same network and training scheme as in the previous section, but this time for each input segment $\bar{x}_i$ of $L = 32$ points, $\{x_{i-L}, \ldots, x_i\}$, the target output $\bar{y}_i$ was composed by the following N points in the time series, $\{x_{i+1}, \ldots, x_{i+N}\}$. We

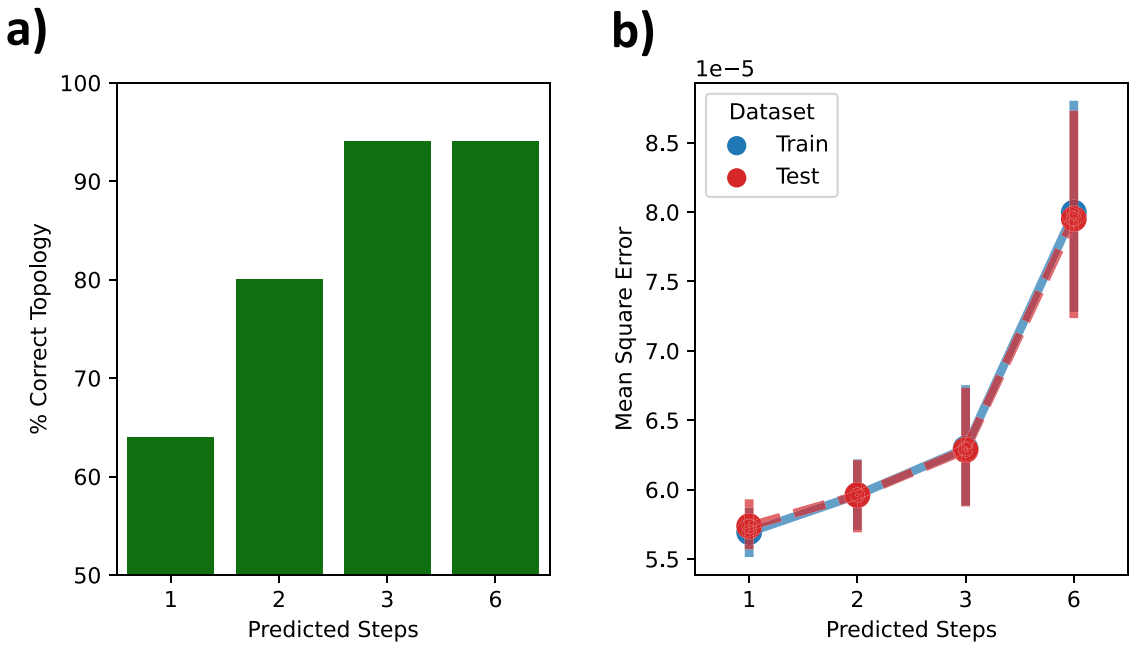

Fig. 4. Results of LSTM sequence-to-sequence models trained to predict 1, 2, 3 and 6 steps ahead, 50 networks were trained for each value. a) The percentage of models whose topological organization of the unstable orbits in hidden space is the same as that of the original attractor at the end of the training process. b) Mean Square Error of the prediction for training and test sets.

explored different number of steps ahead in the prediction, $N = 1, 2, 3$ and 6. We trained 50 models for each of these values.

In the same way that we did in the previous section, we evaluated the topological structure of the periodic orbits in the hidden space for all the trained models. We computed the percentage of models with correct organizations at the end of the training process for the different numbers of N explored. The results are shown in Fig. 4a. For $N = 1$, only 64% of the models present a correct topological organization in the hidden space after 201 epochs. But this proportion increases when the numbers of steps ahead increases. For $N = 3$ and $N = 6$, the percentage of models with correct topology reaches 95%. In Fig. 4b we present the Mean Squared Error in the test and train sets for the different values of N used. Notice that the task of predicting further steps becomes increasingly difficult to accomplish for the neural network.

If we compare the Mean Squared Error in the test set between the models with wrong topology with that of the models with correct topology for $N = 1$, there is no significant difference. For models with wrong topology the MSE is 5.57 $(5.45 - 6.06) \times 10^{-5}$, while for models with correct topology is 5.64 $(5.41 - 6.40) \times 10^{-5}$. This means that even models that reached a good minimum of the cost function during the training phase end up with wrong topologies.

Unlike in the case described in the previous section, in this case the points in the target dataset $\bar{Y}$ do not unequivocally define a state of the system for all cases. Depending on the choice for the value of N, the target manifold $\mathcal{M}_{out} \subset \mathbb{R}^N$ may or may not be an embedding of the original attractor. Let's analyze the cases $N = 1$ and $N = 2$. We know that, because of its dimension, the target manifold $\mathcal{M}_{out} \subset \mathbb{R}^N$ cannot be a homeomorphism of the original Rössler attractor for $N = 1$ or $N = 2$. For the same reason, we also know that the function $\mathcal{F}$ it is not a one-to-one mapping between points of $\mathcal{M}_{in}$ and points on $\mathcal{M}_{out}$. Thus, for these values of N, a model f_θ that is a good approximation of $\mathcal{F}$ does not necessarily imply a bijection between $\mathcal{M}_{in}$ and the representation of the data in the hidden space $\mathcal{M}_h$. This is why an important proportion of models end up with an incorrect organization of the periodic orbits in the hidden space, regardless of how well they are accomplishing the desire task.

4. Discussion

In the context of a growing use of neural networks as data-driven models in a wide variety of fields, our work adds to the efforts of interpreting the prediction mechanisms behind their success. In particular, we show that, if the time series data comes from a dynamical system, a recurrent neural network in sequence-to-sequence configuration learns to make a proper embedding of the signal in its hidden space. This process is analogous to the one traditionally performed by dynamics using the mechanisms proposed by Takens, where the non-trivial problem was to find a suitable sampling interval τ for the embedding to be optimal.

We decided to use LSTM cells as it is one of the most widely used recurring network variants, but the results can be extended to sequential models using other types of cells (classic RNNs, GRUs). Note that the discussion about the correctness of the embedding and its relation with the target dataset and the dimensionality of the encoding does not depend on the particular characteristics of the model f_θ with which we try to approximate the objective function F.

Furthermore, recurrent neural networks can not only be used to work with series of scalar values. Although in this work we refer to the simplest possible architecture, recurrent cells can be combined with other types of networks (dense, CNNs) to process more complex sequential data, such as multivariate time series or video.

We believe that our findings will be of great interest to people working with recurrent neural networks in all kinds of dynamical systems [28,38,43,51-53]. Our result shows that, if the number of predicted time steps used to train the network is sufficient to unequivocally define a state of the system, the representation of the data in the hidden space of the recurrent neural network is a flow topologically equivalent to that of the underlying dynamical system. Otherwise, if the predicted steps used to train the model are not sufficient to discern a particular state of the system, a correct

representation in the hidden state is not guaranteed, and this is the case regardless whether the network does a good job in terms of prediction error.

Having a correct representation of the time series in the hidden space, i.e. a proper embedding, allows us to access a phase portrait equivalent to that of the original system, which is usually unknown. For example, when using recurrent neural networks to predict the evolution of fluids (as made in [54–58]) the low-dimensional dynamics in the hidden space can be studied as a proxy for the dynamics of active modes involved in the actual problem under investigation. Future work will investigate whether a correct representation of the flow in the hidden space also contributes to having a more robust and accurate long-term prediction of the system's behavior.

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.

Acknowledgements

This work was supported by grants from UBACyT (Universidad de Buenos Aires) and from ANCyT (Argentina). Data and codes are available upon request. G.U. thanks Federico Barone for useful comments. G.M. dedicates this work to the memory of Tito Arecchi, whose passion for science, arts, literature, and the city of Florence, were and will always remain, an immense inspiration. "*En tu honor, Tito, salud*".

References

[1] Mudelsee M. Trend analysis of climate time series: a review of methods. Earth Sci Rev 2019;190:310–22.
[2] Stoffer, D.S., & Ombao, H. (2012). Special issue on time series analysis in the biological sciences.
[3] Ghosh B, Basu B, O'Mahony M. Multivariate short-term traffic flow forecasting using time-series analysis. IEEE Trans Intell Transp Syst 2009;10(2):246–54.
[4] Vlahogianni EI, Karlaftis MG, Golias JC. Short-term traffic forecasting: where we are and where we're going. Transportation Research Part C: Emerging Technologies 2014;43:3–19.
[5] Andersen TG, Bollerslev T, Christoffersen PF, Diebold FX. Volatility forecasting (No. w11188). National Bureau of Economic Research 2005.
[6] Grassberger P, Schreiber T, Schaffrath C. Nonlinear time sequence analysis. Int J Bifurc Chaos 1991;1(03):521–47.
[7] Casdagli M. Nonlinear prediction of chaotic time series. Physica D 1989;35(3):335–56.
[8] Mindlin GB, Solari HG, Natiello MA, Gilmore R, Hou XJ. Topological analysis of chaotic time series data from the Belousov-Zhabotinskii reaction. J Nonlinear Sci 1991;1(2):147–73.
[9] Mindlin GM, Gilmore R. Topological analysis and synthesis of chaotic time series. Physica D 1992;58(1–4):229–42.
[10] Sugihara G, May RM. Nonlinear forecasting as a way of distinguishing chaos from measurement error in time series. Nature 1990;344(6268):734–41.
[11] Sugihara G. Nonlinear forecasting for the classification of natural time series. Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences 1994;348(1688):477–95.
[12] Abarbanel HD, Brown R, Kadtke JB. Prediction in chaotic nonlinear systems: methods for time series with broadband Fourier spectra. Phys Rev A 1990;41(4):1782.
[13] Abarbanel H. Analysis of observed chaotic data. Springer Science & Business Media; 2012.
[14] Takens F. Detecting strange attractors in turbulence. In: Dynamical systems and turbulence, warwick 1980. Berlin, Heidelberg: Springer; 1981. p. 366–81.
[15] Deyle ER, Sugihara G. Generalized theorems for nonlinear state space reconstruction. PLoS ONE 2011;6(3):e18295.
[16] Uribarri G, Mindlin GB. The structure of reconstructed flows in latent spaces. Chaos: An Interdisciplinary Journal of Nonlinear Science 2020;30(9):093109.
[17] Gilpin W. Deep reconstruction of strange attractors from time series. Adv Neural Inf Process Syst 2020:33.
[18] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–44.
[19] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016.
[20] Lim, B., & Zohren, S. (2020). Time series forecasting with deep learning: a survey. arXiv preprint arXiv:2004.13408.
[21] Gamboa, J.C.B. (2017). Deep learning for time-series analysis. arXiv preprint arXiv:1701.01887.
[22] Sezer OB, Gudelek MU, Ozbayoglu AM. Financial time series forecasting with deep learning: a systematic literature review: 2005–2019. Appl Soft Comput 2020;90:106181.
[23] Wang H, Lei Z, Zhang X, Zhou B, Peng J. A review of deep learning for renewable energy forecasting. Energy Convers Manage 2019;198:111799.
[24] Yu, R., Li, Y., Shahabi, C., Demiryurek, U., & Liu, Y. (2017, June). Deep learning: a generic approach for extreme condition traffic forecasting. In Proceedings of the 2017 SIAM international Conference on Data Mining (pp. 777-785). Society for Industrial and Applied Mathematics.
[25] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput 1997;9(8):1735–80.
[26] Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.
[27] Olah, C. (2015). Understanding lstm networks.
[28] Sangiorgio M, Dercole F. Robustness of LSTM neural networks for multi-step forecasting of chaotic time series. Chaos, Solitons Fractals 2020;139:110045.
[29] Zaytar MA, El Amrani C. Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. Int J Comput Appl 2016;143(11):7–11.
[30] Zhu L, Laptev N. Deep and confident prediction for time series at uber. In: 2017 IEEE International Conference on Data Mining Workshops (ICDMW). IEEE; 2017. p. 103–10.
[31] Park SH, Kim B, Kang CM, Chung CC, Choi JW. Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture. In: 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE; 2018. p. 1672–8.
[32] Du S, Li T, Horng SJ. Time series forecasting using sequence-to-sequence deep learning framework. In: 2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP). IEEE; 2018. p. 171–6.
[33] Du S, Li T, Yang Y, Gong X, Horng SJ. An LSTM based encoder-decoder model for MultiStep traffic flow prediction. In: 2019 International Joint Conference on Neural Networks (IJCNN). IEEE; 2019. p. 1–8.
[34] Xiang Z, Yan J, Demir I. A rainfall-runoff model with LSTM-based sequence–to-sequence learning. Water Resour Res 2020;56(1):e2019WR025326.
[35] Lyu P, Chen N, Mao S, Li M. LSTM based encoder-decoder for short-term predictions of gas concentration using multi-sensor fusion. Process Saf Environ Prot 2020;137:93–105.
[36] Kao IF, Zhou Y, Chang LC, Chang FJ. Exploring a Long Short-Term Memory based Encoder-Decoder framework for multi-step-ahead flood forecasting. J Hydrol (Amst) 2020;583:124631.
[37] Lee T, Shin JY, Kim JS, Singh VP. Stochastic simulation on reproducing long-term memory of hydroclimatological variables using deep learning model. J Hydrol (Amst) 2020;582:124540.
[38] Madondo M, Gibbons T. Learning and modeling chaos using lstm recurrent neural networks. In: MICS 2018 Proceedings; 2018.
[39] Vlachas PR, Pathak J, Hunt BR, Sapsis TP, Girvan M, Ott E, et al. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. Neural Netw 2020;126:191–217.
[40] Cestnik R, Abel M. Inferring the dynamics of oscillatory systems using recurrent neural networks. Chaos: An Interdisciplinary Journal of Nonlinear Science 2019;29(6):063128.
[41] Shopov V, Markova V. Identification of Non-linear Dynamic System. In: 2019 International Conference on Information Technologies (InfoTech). IEEE; 2019. p. 1–3.
[42] de Jesús, Serrano-Pérez J, Fernández-Anaya G, Carrillo-Moreno S, Yu W. New Results for Prediction of Chaotic Systems Using Deep Recurrent Neural Networks. Neural Processing Letters 2021;53(2):1579–96.
[43] Gonzalez J, Yu W. Non-linear system modeling using LSTM neural networks. IFAC-PapersOnLine 2018;51(13):485–9.
[44] Kingma, D.P., & Ba, J. (2014). Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980.
[45] Gilmore R, Lefranc M. The topology of chaos. Weinheim, Germany: Wiley-VCH; 2003.
[46] Our software implementation to compute linking numbers is publicly available at https://github.com/gon-uri/linking_number.
[47] Eckmann JP, Ruelle D. Ergodic theory of chaos and strange attractors. The theory of chaotic attractors 1985:273–312.
[48] Kaplan, J.L., & Yorke, J.A. (1979). Chaotic behavior of multidimensional difference equations. In Functional differential equations and approximation of fixed points (pp. 204-227). Springer, Berlin, Heidelberg.
[49] Frederickson P, Kaplan JL, Yorke ED, Yorke JA. The Liapunov dimension of strange attractors. J Differ Equ 1983;49(2):185–207.
[50] Anishchenko VS, Astakhov S. Relative kolmogorov entropy of a chaotic system in the presence of noise. Int J Bifurc Chaos 2008;18(09):2851–5.
[51] Vlachas PR, Byeon W, Wan ZY, Sapsis TP, Koumoutsakos P. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 2018;474(2213):20170844.
[52] Hassanzadeh P, Chattopadhyay A, Palem K, Subramanian D. Data-driven prediction of a multi-scale Lorenz 96 chaotic system using a hierarchy of deep learning methods: reservoir computing, ANN, and RNN-LSTM. In: APS Division of Fluid Dynamics Meeting Abstracts; 2019. p. C17–009.
[53] Li T, Wu T, Liu Z. Nonlinear unsteady bridge aerodynamics: reduced-order modeling based on deep LSTM networks. J Wind Eng Ind Aerodyn 2020;198:104116.

[54] Wang Z, Xiao D, Fang F, Govindan R, Pain CC, Guo Y. Model identification of reduced order fluid dynamics systems using deep learning. Int J Numer Methods Fluids 2018;86(4):255–68.
[55] Mohan, A.T., & Gaitonde, D.V. (2018). A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks. arXiv preprint arXiv:1804.09269.
[56] Wiewel S, Becher M, Thuerey N. Latent space physics: towards learning the temporal evolution of fluid flow. In Computer graphics forum 2019;38(2):71–82.
[57] Hasegawa K, Fukami K, Murata T, Fukagata K. CNN-LSTM based reduced order modeling of two-dimensional unsteady flows around a circular cylinder at different Reynolds numbers. Fluid Dyn Res 2020;52(6):065501.
[58] Maulik R, Mohan A, Lusch B, Madireddy S, Balaprakash P, Livescu D. Time-series learning of latent-space dynamics for reduced-order model closure. Physica D 2020;405:132368.