Contents lists available at ScienceDirect

Physica A

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

Prediction of chimera in coupled map networks by means of deep learning

Sidney T. da Silva [a,c,*], Ricardo L. Viana [b,f], C.A.S. Batista [e], Antonio M. Batista [a,d]

[a] *Pós-Graduação em Ciências/Física, Universidade Estadual de Ponta Grossa, Ponta Grossa, PR, Brazil*
[b] *Departamento do Física, Universidade Federal do Paraná, Curitiba, PR, Brazil*
[c] *Departamento do Química, Universidade Federal do Paraná, Curitiba, PR, Brazil*
[d] *Departamento de Matemática e Estatística, Universidade Estadual de Ponta Grossa, Ponta Grossa, PR, Brazil*
[e] *Center for Marine Studies, Federal University of Paraná, Brazil*
[f] *Instituto de Fisica da USP, Brazil*

ARTICLE INFO

Article history:
Received 2 October 2022
Received in revised form 28 November 2022
Available online 5 December 2022

Keywords:
Machine learning
Deep learning
Reservoir computing
Multilayer perceptron
Coupled map lattices
Chimera states

ABSTRACT

Chimera states are Spatio-temporal patterns in coupled oscillator arrays, in which incoherent domains coexist with coherent ones. To characterize chimeras, however, is a nontrivial problem since it is difficult to distinguish between coherent domains and incoherent domains. A useful tool for this task is machine learning, in particular deep learning techniques like reservoir computing and multilayer perceptrons. In this work we use these quantifiers in order to identify chimera states in logistic map lattices with non-local coupling. We compare our results from machine learning techniques with more conventional characterizations, such as Lyapunov exponents and a local order parameter.

1. Introduction

Chimera states are spatio-temporal patterns in coupled oscillator arrays, in which incoherent domains coexist with coherent ones [1]. Even before its nomenclature, chimera states were observed in arrangements of forced-coupled Duffing oscillators [2] and complex Ginzburg–Landau equations not locally coupled [3]. However, non-local coupling is not the only way to achieve chimera states, since they can also occur due to spatially modulated delayed feedback coupling [4]. The presence of chimera states has been identified in network models [4,5] such as the Kuramoto model of coupled oscillators [6], neural networks [7] and lattices of coupled oscillators of van der Pol- Duffing [8]. Chimera states have also been observed in experiments involving a liquid crystal spatial light modulator, which controls the polarization properties of an optical wavefront. It is an experimental realization of a network of nonlocally coupled maps [9]. Other observations of chimera states were in populations of coupled chemical oscillators [10], coupled metronomes [11,12] and electronic oscillators [13].

For a one-dimensional chain of coupled oscillators, the chimera states are typically characterized by a coherent domain coexisting close to an incoherent one [14]. Batista et al. [15], through the order parameter(which is defined in Section 2.3), developed a quantifier that determines the ratio between coherent and non-coherent oscillators. Through this ratio, they quantified when a state can be chimera or no, limiting this ratio to a certain interval. In [16,17], the authors use machine learning techniques to classify coherent, non-coherent and chimera states. For this they use non-recurring classifiers.

[*] Corresponding author at: Departamento do Química, Universidade Federal do Paraná, Curitiba, PR, Brazil.
E-mail address: sidneyt.silva82@gmail.com (S.T. da Silva).

https://doi.org/10.1016/j.physa.2022.128394
0378-4371/© 2022 Elsevier B.V. All rights reserved.

In this work, we use a modern recurrent deep learning network that is widely used in time series both for use as classifiers and to predict new values of the time series. This is thanks to the ability to keep the memory of entries and thereby learn the dynamics of these series. And unlike other classifiers that often require extensive data handling such as feature extraction using other quantifiers, this network only needs the time series without any other quantifiers. The great advantage in using deep learning is its efficiency and versatility, managing to learn patterns and dynamic characteristics that go beyond the human perspective, resulting in a much more efficient process of characterizing spatio-temporal pattern.

This paper is organized as follows. Section 2 introduces the coupled chaotic logistic map lattice with finite range coupling and quantifiers. Section 3 introduces the artificial neural network. Section 4 presents our results. The last Section contains our conclusions.

2. Coupled map networks

We build a coupled logistic map network with two different types of nonlocal coupling: chemical coupling and mean field coupling. Many investigations of spatiotemporal dynamics in discrete time have been used local coupling, in which each map $x \longrightarrow f(x)$ is coupled to its nearest neighbors, in the form

$$x_{n+1}^{(i)} = (1-\varepsilon)f(x_n^{(i)}) + \frac{\varepsilon}{2}\{f(x_n^{(i+1)}) + f(x_n^{(i-1)})\}, \tag{1}$$

where $x_n^{(i)}$ is the state variable at discrete time n and belonging to a chain of N identical systems, such that $i = 1, 2, \ldots, N$; ε standing for the coupling strength. This type of coupling is also called laplacian and represents the discretization of a second derivative with respect to the position along a one-dimensional lattice.

In contrast, in a network of globally coupled maps, each map is coupled by means of a "mean field" generated by all other sites, regardless of their relative position or to a chemical coupling where the interaction is mediated by the rapid diffusion of some chemical into the medium in which systems are embedded.

2.1. Logistic map with chemical coupling

This type of non-local coupling appears in models of dynamical systems (henceforth represented by maps) whose interaction is mediated by the rapid diffusion of some chemical in the medium in which the systems are embedded. In this case, each site releases a chemical substance at a rate that depends on its own dynamics. The dynamics of other maps are affected by the local chemical concentration. Kuramoto showed that as long as the diffusion is fast enough, the coupling in a spatial dimension is not local, the relative coupling strength decreases exponentially with the lattice position with a decay rate γ. The network is given by

$$x_{n+1}^{(i)} = f(x_n^{(i)}) + C\varepsilon \sum_{j=1}^{(N-1)/2} \frac{1}{e^{2\gamma j}}(f(x_n^{(i+j)}) - 2f(x^i) + f(x_n^{(i-j)})), \tag{2}$$

with

$$C = \frac{2}{\sum_{j=1}^{(N-1)/2} e^{\gamma j}}, \tag{3}$$

where r is the nonlinearity, ε is the coupling force (coupling laplacian [18]) and C is the term responsible for the chemical coupling. When $\gamma \longrightarrow \infty$, the coupling is local and when $\gamma \longrightarrow 0$, the coupling is global (all to all). The x_n^i is the discrete-time state variable n and belongs to a chain of N identical systems with periodic boundary condition ($x_n^{i\pm N} = x_n^i$), with $i = 1, 2, \ldots, N$.

We assume that the local dynamics is governed by the logistic map $f(x) = rx(1-x)$ with $r = 3.8$, for which the uncoupled maps exhibit chaotic behavior (this value is held constant unless indicated otherwise). It is known that, if the coupling is local, there is high-dimensional space–time chaos [19]. For nonlocal coupling, however, many chimera states have been observed, with different wavenumbers for the coherent domains, as the coupling parameters (radius r and force ε) are varied.

2.2. Logistic map with mean field coupling

A model that considers the relative coupling strength as decreasing with the lattice position in a power-law fashion with exponent α is given by:

$$x_{n+1}^{(i)} = f(x_n^{(i)}) + \frac{\varepsilon}{\eta(\alpha)} \sum_{j=1}^{(N-1)/2} \frac{1}{j^{\alpha}}(f(x_n^{(i+j)}) - 2f(x^i) + f(x_n^{(i-j)})), \tag{4}$$

where the normalization factor is $\eta(\alpha) = 2\sum_{j=1}^{(N-1)/2} j^{\alpha}$.

Mean field [20] is a type of global coupling in which each site is influenced by the average effect of all sites in the network. The limiting behavior as α varies from zero to infinity is the same as before. The local dynamics is also governed by the logistic map $f(x) = rx(1-x)$ with $r = 3.8$. x_n^i is the discrete-time state variable n and belongs to a chain of N identical systems with periodic boundary condition ($x_n^{i\pm N} = x_n^i$) with $i = 1, 2, \ldots, N$.

If $\alpha = \infty$ only the terms with $j = 1$ survive in the summation, and only the nearest neighboring sites contribute to the coupling and $\alpha \longrightarrow 0$, the coupling is global (all to all).

2.3. Quantifiers

One way to analyze our results obtained from an algorithm is to compare them with some dynamic quantifiers. The first quantifier used to characterize each of these patterns is the local order parameter R_i [21]. It is a quantifier that determines the degree of coherence of the lattice. We introduce a phase for the jth map from the following definition:

$$\sin \psi_j = \frac{2x^j - \max_j\{x^j\} - \min_j\{x^j\}}{\max_j\{x^j\} - \min_j\{x^j\}}, \tag{5}$$

for $j = 1, \ldots, N$. The values $\max_j\{x^j\}$ and $\min_j\{x^j\}$ are, respectively, the maximum and minimum values of the state variable for a network (snapshot), so that a spatial half-cycle is mapped in the phase interval $[-\pi/2, \pi/2]$. The magnitude of the local order parameter is defined as:

$$R_i = \lim_{N \longrightarrow \infty} \frac{1}{2\delta(N)} \left| \sum_{j \in C} e^{i\psi_j} \right|, \tag{6}$$

for ($i = 1, \ldots, N$). The sum is restricted to the range of j-values,

$$C : \left| \frac{j}{N} - \frac{i}{N} \right| \leq \delta(N), \tag{7}$$

where $\delta(N) \longrightarrow 0$ when $N \longrightarrow \infty$.

For the $R_i \approx 1.0$ the sites belong to coherent domains. When R_i takes values less than 1.0, the sites are in inconsistent domains.

We define a new parameter known as the P degree of coherence, which determines the number of sites with $R_i \approx 1.0$. In relation to the size of the lattice, we add all sites with $R_i \approx 1.0$ and divided by the size of the network (snapshot).

Another quantifier widely used to determine the dynamics of a lattice of coupled maps is the Lyapunov spectrum [22] $\{\lambda_i\}_{i=1}^N$, where λ_i is greater than 0 for a chaotic dynamic. The normalized sum of the positive Lyapunov exponents is a numerical estimate for the Kolmogorov–Sinai density (KS).

3. Bidirectional deep read echo state (BDES) networks

We consider a neural network with the objective of classification. The network is trained to classify the time series into one of the groups. For the classification model, we use two coupled neural networks. A recurrent network with echo state (ESN) [23,24] coupled with a multilayer perceptron network (Fig. 1) and using Principal Component Analysis (PCA) [25] as a dimension reducer.

3.1. Reservoir computing

The reservoir computing (RC) has been used for modelling nonlinear time series. In the learning context, state echoes (ESNs) are more common in (RC) models, where the input sequence is projected into a larger space through the use of the non-linear reservoir. Learning is accomplished through the application of simple linear techniques in the space of the reservoir.

The proposed architecture is adapted from [26], called bidirectional deep read ESN (BDESN), which combines the speed of the RC with the trainable precision of RNNs. The model is equipped with a bidirectional reservoir. Bidirectional architectures have been successfully applied in RNNs to extract temporal resources from the time series that have a very long time dependency.

The BDESN is utilized for the classification of time series $\mathbf{x} = \{\mathbf{x}_t\}_{t=0}^T$ labeled with class $\mathbf{c}$ through the following procedure. We first project the time series with smaller dimension $\mathbf{x}(t)$ to a larger space through the reservoir ($\mathbf{h}(t)$). Then a dimension reduction algorithm projects the reservoir outlet into a smaller space represented by the state vector $\mathbf{r_x}$. Finally, a multilayer perceptron (MLP) classifies the vector representative of $\mathbf{x}$, as shown in Fig. 1.

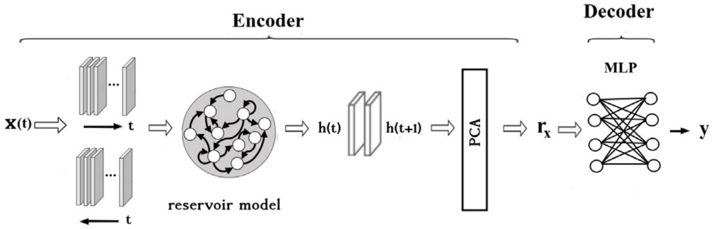

Fig. 1. It represents a deep network composed of a computational reservoir followed by a dimensional reduction (PCA) and a reading layer formed of a multilayer perceptron network trained for classification.

3.2. Reservoir

The reservoir acts as an encoder that generates the input representation in a larger space. This state produced by the reservoir brings all the dynamic information from the original input. This encoding is performed using the weights $\theta_{enc} = \{W^i, W^h\}$. The dynamics of this process is performed by the following equation

$$\mathbf{h}(t) = (1-\alpha) + \alpha f(\mathbf{W}^h \mathbf{h}(t-1) \\ + \mathbf{W}^i \mathbf{x}(t) + \eta(t)) \,, \tag{8}$$

where $\mathbf{h}(t)$ is the time-dependent internal state, which combines the current input $\mathbf{x}(t)$ with the previous state $\mathbf{h}(t-1)$. The function f is a non-linear activation function (tanh), $\mathbf{W}^h$ is the sparse matrix that defines the recurrent self-connects in the reservoir, and $\mathbf{W}^i$ defines the incoming connections. Both matrices are randomly generated and are not trained. The behavior of the reservoir is mainly controlled by five hyper parameters, that are: the size of the states N, the spectral radius ρ of W^h, the dimensioning of the inputs ω, the hyper leakage parameter α, and the noise η which is used for regularization in the reservoir. The η term represents additive white Gaussian noise with spherical covariance matrix and unit standard deviation. By means of an optimal fit of these hyper parameters, the reservoir produces rich dynamics and its internal states can be used to solve many prediction and classification tasks. The state generated by the reservoir $\mathbf{h}(t)$, after all inputs have been processed, is a high-dimensional representation that incorporates the temporal dependencies of $\mathbf{x}$. Since the reservoir exchanges its internal stability with a memory at time T [27], the state tends to lose information from the initial times. To get around this problem, we feed the same reservoir with the inverse order of the time series $\mathbf{x}' = \{\mathbf{x}_{T-t}\}_{t=0}^{T}$ and generate a new state $\mathbf{h}(t)'$ that is more influenced with the first inputs. The final resultant state is obtained by concatenating the two states, $\overline{\mathbf{h}}_T = [\mathbf{h}(t); \mathbf{h}(t)']$. The bidirectional reservoir has recently been used for time series prediction [28]. From the sequence of the RNN states generated over time,

$$\mathbf{H} = [\overline{h}(1), \ldots, \overline{\mathbf{h}}(T)], \tag{9}$$

it is possible to extract a representation $\mathbf{r}_X = r(\mathbf{H})$ of the input $\mathbf{x}$.

The $\mathbf{r}_X$ vector brings us all the information about the characteristics of the $\mathbf{x}$ input, in this case the $\mathbf{r}_X$ vector is formed by the weights and bias learned when a later state is generated by the previous state, as shown in the equation

$$\mathbf{h}(t+1) = \mathbf{W}_r \mathbf{h}(t) + \mathbf{b}_r \,, \tag{10}$$

where $\mathbf{r}_X = \{\mathbf{W}_r, \mathbf{b}_r\} \in \mathbb{R}^{R(R+1)}$ and R is the number of neurons that form the reservoir.

In summary, all dynamic characteristics of the input are represented by the vector $\mathbf{r}_X$, which is formed by the weights $\{\mathbf{W}_r, \mathbf{b}_r\}$. After constructing $\mathbf{r}_X$, we can decode in the output space, which are the y classes for the classification case (Fig. 3(a)) or for the regression case (Fig. 1). This decoding can be performed by

$$y = g(\mathbf{r}_X, \theta_{dec}), \tag{11}$$

where g is a multilayer perceptron network (Fig. 1) and the θ_{dec} weights to be learned.

As the reservoir has a large dimension due to the number of neurons, this takes an overfit and computational resources. The PCA [25] dimensionally reduces the states, showing a better performance. The PCA aims to reduce the feature space, choosing a space that better separates these features, facilitating the decision surface.

iii. Multilayer perceptron (MLP)

These state vectors with their reduced dimensions become the input vectors to the network (MLP), where the classification or regression takes place. The weights of these layers will undergo adjustments during training. At this point, a normal training of an MLP network is made.

These deep MLPs are known for their generalizability and adaptability, important characteristics for the problem at hand. Nowadays, deep layer networks can be efficiently trained using sophisticated regularization techniques and pre-training techniques that help to avoid overfit and null or explosive gradient problem.

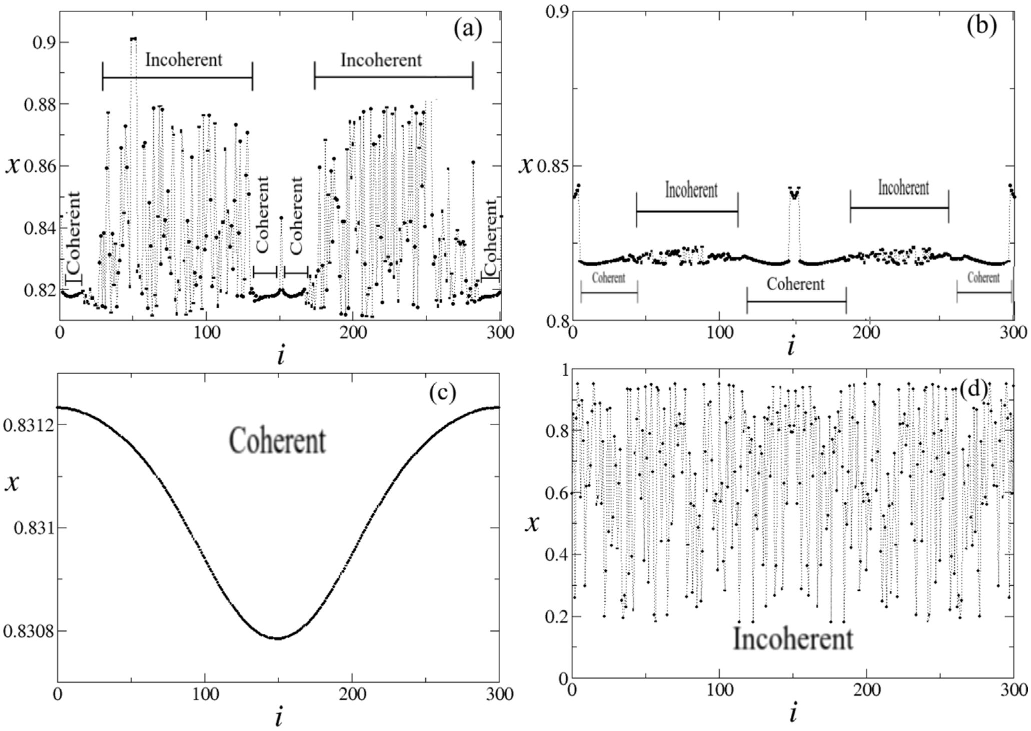

Fig. 2. Snapshots of chimera examples in coupled map networks. For each time series, we use a Gaussian initial condition with a transient of $n = 2\,000$ iterated and chemical coupling, where we observe coherent and non-coherent states for a network with $N = 300$. We consider (a) $r = 3.8$, $\varepsilon = 0.2$ and $\gamma = 0.02$, (b) $r = 3.58$, $\varepsilon = 0.1$ and $\gamma = 0.0029$, (c) $r = 2.8$, $\varepsilon = 0.8$ and $\gamma = 0.029$ and (d) $r = 3.58$, $\varepsilon = 0.4$ and $\gamma = 1.4$. The panels (a) and (b) are examples of spatio-temporal patterns with incoherent and desynchronized domains coexist with coherent(chimera). In the panel (c), we see a periodic spatio-temporal pattern with only coherent domain and the panel (d) exhibits chaos spatio-temporal pattern.

In our architecture, we use a reading layer formed by an MLP network with three hidden layers of 400 neurons each one and an input layer with 500 neurons. On the hidden layers of MLP, we consider dropout= 0.2 [29] and "Greedy Layer-Wise" [30] as pre-training of the network. The optimizer and error function were "adam [31]" and "MSE(mean square error)", respectively. The best hyper parameters of the reservoir are: $N = 500$, $\alpha = 0.96$, $\rho = 0.95$, $\omega = 0.9$, $\eta = 0.0011$ and PCA $= 100$, which is used in the learning process.

4. Results and discussions

Our network is trained through time series with chimera state profiles (as examples of Figs. 2(a)–(b)), series with periodic behavior (Fig. 2(c)) and chaotic behavior (Fig. 2(d)). We use a network of coupled maps with two different types of coupling, chemical coupling and mean field coupling. For the evolution of these maps, we consider the logistic map. To compare the results found by our network, we calculate two quantifiers in the same parameter space, the order parameter and the Kolmogorov–Sinai density.

The main objective of our algorithm is to characterize the regions in the parameter space, where chimeras are formed. For this, we train our neural network with three different classes: the first class (labeled as class 0) containing 705 periodic time series, second class (labeled as class 1) containing 1601 chimera state time series and 820 time series with chaotic dynamics. These datasets are divided into 70% for training, 5% for validation and 25% for testing. We compute the performance measurement through the confusion matrix (Fig. 3). The performance is evaluated based on three main measurement performances in RNN models, which are the accuracy, precision and recall.

$$\begin{aligned} Accuracy &= \frac{(TP + TN)}{(TP + TN + FP + FN)}, \\ Precision &= \frac{TP}{(TP + FP)}, \\ Recall &= \frac{TP}{TP + FN}, \\ F_1 score &= \frac{2 * Precision * Recall}{Precision + Recall}. \end{aligned} \tag{12}$$

Table 1
Confusion matrix.

	Predicted positive	Predicted negative
Current Positive	TP	FN
Current Negative	FP	TN

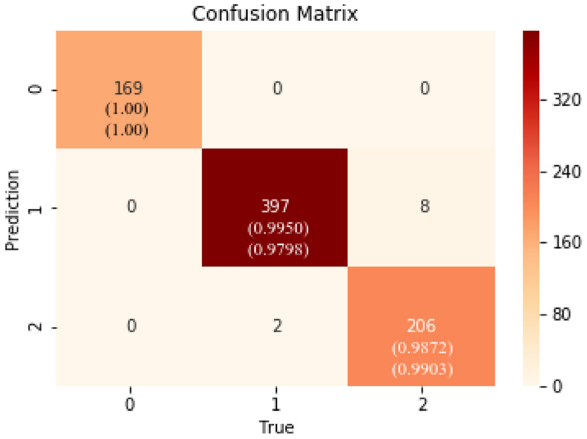

Fig. 3. Confusion matrix of test data. 0 - periodic time series, 1- chimera state time series, and 2 - time series with chaotic dynamics. The top values represent precision and low values represent recall for each time series, respectively.

where TP(True Positive), FN(False Negative), FP(False Positive) and TN(True Negative). These measurements are described using the confusion matrix that considers a two-class classification problem, as illustrated in Table 1. The main diagonal values are the correctly predicted values while the off-diagonal values are the wrongly predicted ones.

In neural networks, an important rule is the choice of hyper parameters for a better performance in the classification model without suffering overfit. After choosing these hyper parameters, using Bayesian optimization, we can apply our network for the classification problem. The best hyper parameters of the reservoir are: $N = 500$, $\alpha = 0.96$, $\rho = 0.95$, $\omega = 0.9$, $\eta = 0.0011$ and PCA=100. For the classification layer, the architecture of the MLP network (Fig. 1) is made with three hidden layers with 500, 400 and 400 neurons. For the regularization, we consider a dropout of 20% with an adaptive learning rate of 0.01 and with a minibatch of 32. In the output layer, we apply the softmax function, and for the gradient, we use adam [31]. In the reservoir (BDESN), we utilize the hyper parameters: $N = 500$, $\alpha = 0.96$, $\rho = 0.95$, $\omega = 0.9$, $\eta = 0.0011$ and PCA=100. After the training phase, the best accuracy is 98.72% in the test data with 98.76% f1 score.

Fig. 4 shows the validation in the training phase, this procedure is important due to fact that it shows if the network is suffering overfitting or underfitting. In this phase, in each epoch, the network is trained and updated with the data separated for training, after updating its parameters (at the end of each epoch). We apply the network in the training and validation data for the prediction of its classes. Comparing the predicted classes with the real classes, we determine the accuracy as a function of each training epoch. It is important to note that the separate data for validation are the unknown data of the algorithm, while the training data are the data used in the network update, so the accuracy of the training data tends to be higher. Overfitting occurs when the accuracy of training data continues to improve over time, but there is a gradual decrease in validation accuracy. The training phase where this starts to occur and where overfiiting starts, and this is not desirable. We can observe that there is a convergence of the accuracy of the data used for the training as the data separated for the validation, in function of each training epoch(epoch 30), showing that our network has a good generalization.

After the training phase of our models, we can now apply this network in the parameter space to the two coupled map networks. For each pair of parameters, we compute the time series and use this series in the artificial neural network, consequently the algorithm predicts which class this series belongs to.

In the first model, we consider a chemically coupled logistic map. In Fig. 5 for $r = 3.8$, we observe that the region where the network classifies as a chimera state is immersed in a region with chaotic dynamics. The region that contains the chimeras has the order parameter ranging from (0.4 to 0.9) and its KS density ranging from (0.12 to 0.48). These regions, which the network classified as chimeras, have a mixture of sites with chaotic and periodic dynamics (Fig. 5(b)) and a mixture of synchronized and desynchronized oscillators (Fig. 5(c)). In Fig. 6, we observe that (γ/ε), becomes almost constant as a function of the coupling.

In Fig. 7, we obtain the prediction of our network by varying r and ε, keeping γ fixed. We see that one of the regions where our algorithm classified as chimera is a transition region between chaotic and periodic dynamics. There are also regions with chimera profiles within both the chaotic region and the periodic region. Looking at the order parameter (Fig. 7(c)) and the KS density (Fig. 7(b)), the regions where chimeras are formed have a dynamics with chaotic oscillators

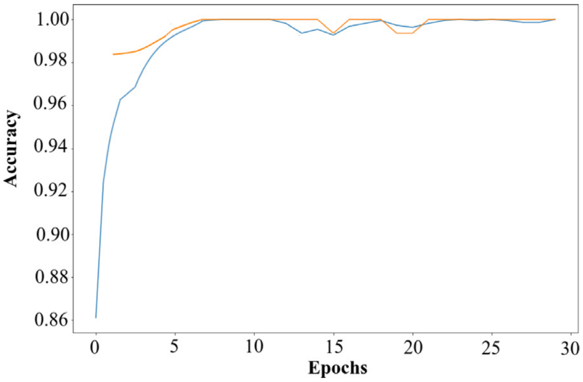

Fig. 4. Validation phase plot, where the blue curve represents the accuracy of the data used in the training and the yellow curve is the accuracy of the data used in the validation.

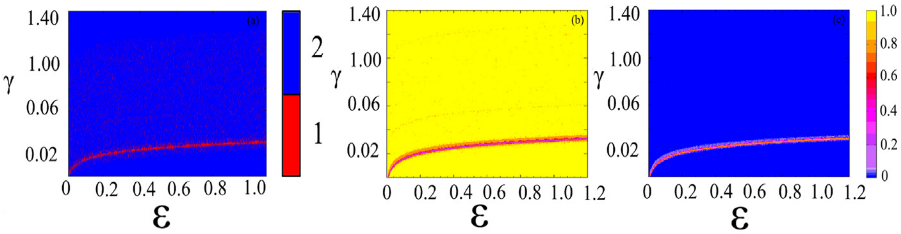

Fig. 5. Plot for the chemically coupled logistic map, where we vary ε and γ with $r = 3.8$. For each time series, we use a Gaussian initial condition with a transient of $n = 2\,000$ iterated. The panel (a) displays the classes predicted by the network, where class 1 (color red) is chimera and class 2 (color blue) is chaotic. The panel (b) shows the KS density for the same pairs of parameters and the panel (c) exhibits the order parameter. The color bar is for the three figures.

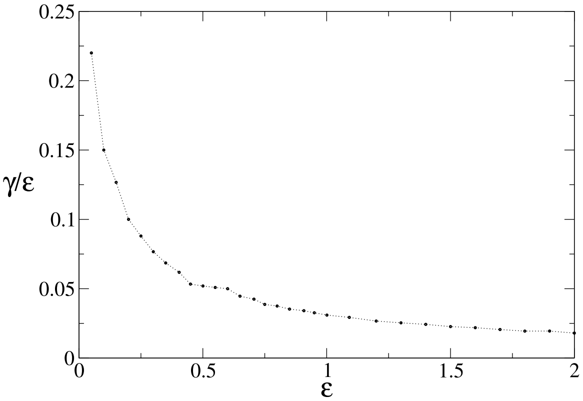

Fig. 6. Ratio between γ/ε as a function of the growth of ε. We see that the ratio decreases when ε increases. The values of these parameters are the values present in the red region highlighted in Fig. 4.

mixed with periodic oscillators ($0.48 > KS > 0.12$) and synchronized oscillators mixed with unsynchronized oscillators ($0.8 > P > 0.4$).

For the second model of networks of coupled maps (Logistic map with midfield coupling), we again compute the parameter space. In Fig. 8, we vary ε and α for $r = 3.8$. As we know that for r equal to 3.8 in uncoupled logistic map, only chaotic dynamics are formed, this is confirmed in Fig. 8 and this profile is maintained even for low coupling, as predicted by network and shown by the quantifiers $KS > 0$ and $P \approx 0$. As we increase the coupling values (ε), we observe the formation of chimeras interspersed with chaotic and periodic regions, where for larger values of ε, periodic dynamics begin to predominate. Again, we identify that the regions where our network classified as being chimera are regions that separate the chaotic dynamics from the periodic dynamics, that is, a kind of transition between these two dynamic behaviors. These same characteristics are observed when we vary ε and r, keeping α fixed (Fig. 9).

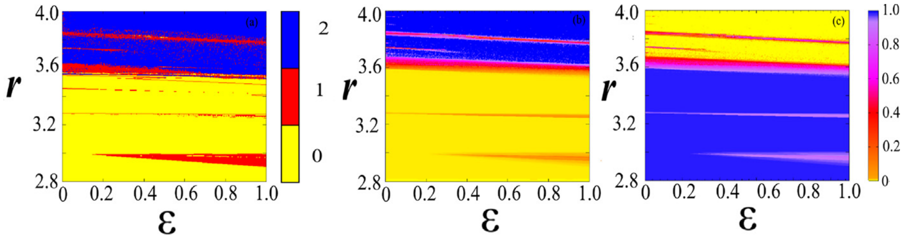

Fig. 7. Plot for the chemically coupled logistic map, where we vary ε and r with $\gamma = 0.029$. For each time series, we use a Gaussian initial condition with a transient of $n = 2\,000$ iterated. The panel (a) display the classes predicted by the network, where class 0 is periodic time series, class 1 is chimera and class 2 is chaotic. The panel (b) shown display the KS density for the same pairs of parameters and the panel (c) exhibits the order parameter. The color bar is for the order parameter and also for KS.

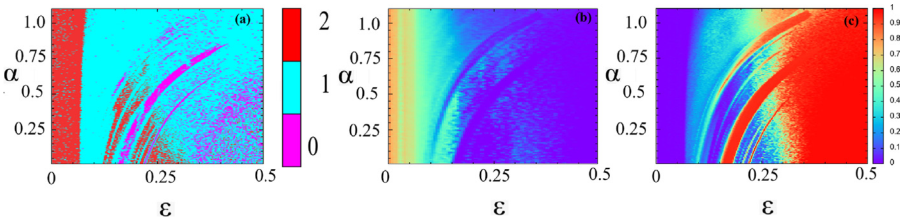

Fig. 8. Plot for the chemically coupled logistic map, where we vary ε and α for $r = 3.8$. For each time series, we use a Gaussian initial condition with a transient of $n = 2\,000$ iterated. The panel (a) shown the classes predicted by the network, where class 0 is periodic time series, class 1 is chimera and class 2 is chaotic. The panels (b) and (c) display the entropy KS and order parameter, respectively. The color bar is for the order parameter and also for KS.

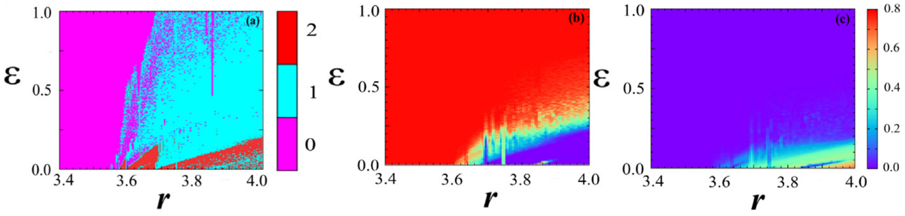

Fig. 9. Plot for the chemically coupled logistic map, where we vary ε and r for $\alpha = 1.08$. For each time series, we use a Gaussian initial condition with a transient of $n = 2\,000$ iterations. The panel (a) exhibits the classes predicted by the network, where class 0 is periodic time series, class 1 is chimera and class 2 is chaotic. The panels (b) and (c) display the entropy KS and order parameter.

Comparing the predicted classes with the values of entropy and the order parameter, we observe an agreement between them. In the regions with chaotic behavior, the is KS density greater than zero ($KS \approx 0.6$) with the parameter of order close to zero and in the regions classified as periodic, the entropy is close to zero and the parameter of order close to 1, as expected. The regions where the network exhibit chimera states, looking at the quantifiers, we calculate the KS density with values above zero, but below 0.4 and with the order parameter between 0.4 to 0.9. This shows that the chimera state has desynchronized and synchronized oscillators, as well as the desynchronized oscillators have chaotic dynamics.

5. Conclusions

We observe that our method is able to detect chimera states with great precision. We also show that the chimera state is always found in the transition region between the periodic and chaotic states. To evaluate the predictions of our algorithm, we complement our analysis using two dynamic quantifiers (order parameter and entropy KS). Comparing the regions of the phase space, where our algorithm classified as chimera states, with the two dynamic quantifiers, we conclude that the chimera is composed of a chaotic dynamics together with a periodic dynamics and the chaotic regions have a desynchronized behavior while the periodic regions have a synchronized behavior.

Our main result is to show that recurrent neural networks are able to learn the dynamic behavior of a time series, even for series with more complex behaviors such as chaotic series and chimera state. In this work, considering different time series (periodic, chimera and chaotic state), our network was able to learn the dynamics of the classes and predict them with great precision.

CRediT authorship contribution statement

Sidney T. da Silva: Conceived of the presented idea, Developed the theory, Performed the calculations, Dynamics of the chimeras, Guide the findings of this work, Discussed the results, Contributed to the final manuscript. **Ricardo L. Viana:** Conceived of the presented idea, Check the methods, Give suggestions for improving the work, Dynamics of the chimeras, Guide the findings of this work, Discussed the results, Contributed to the final manuscript. **C.A.S. Batista:** Conceived of the presented idea, Check the methods, Give suggestions for improving the work, Dynamics of the chimeras, Guide the findings of this work, Discussed the results, Contributed to the final manuscript. **Antonio M. Batista:** Conceived of the presented idea, Check the methods, Give suggestions for improving the work, Dynamics of the chimeras, Guide the findings of this work, Discussed the results, Contributed to the final manuscript.

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.

Data availability

No data was used for the research described in the article.

Acknowledgments

The authors received partial financial support from the following agencies: CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), Brazil, FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), Brazil, CAPES (Coordenação de Pessoal de Ensino Superior), Brazil, and Fundação Araucária (Estado do Paraná), Brazil.

References

[1] D.M. Abrams, S.H. Strogatz, Phys. Rev. Lett. 93 (2004) 174102.
[2] D.K. Umberger, E. Ott, C. Grebogi, B. Afeyan, Phys. Rev. A 39 (1989) 9.
[3] Y. Kuramoto, D. Battogtokh, Nonlinear Phenom. Complex Syst. 5 (2002) 380.
[4] R.G. Andrzejak, G. Ruzzene, I. Malvestio, Chaos 27 (2017) 053114.
[5] O.E. Omelchenko, Y.L. Maistrenko, P.A. Tass, Phys. Rev. Lett. 100 (2008) 044105.
[6] M.S. Santos, J.D. Szezech Jr., A.M. Batista, I.L. Caldas, R.L. Viana, S.R. Lopes, Phys. Lett. A 379 (2015) 2188.
[7] M.S. Santos, J.D. Szezech Jr., F.S. Borges, K.C. Iarosz, I.L. Caldas, A.M. Batista, R.L. Viana, J. Kurths, Chaos Solitons Fractals 101 (2017) 86.
[8] D. Dudkowski, Y. Maistrenko, T. Kapitaniak, Chaos 26 (2016) 116306.
[9] A.M. Hagerstrom, T.E. Murphy, R. Roy, P. Hövel, I. Omelchenko, E. Schöll, Nat. Phys. 8 (2012) 658.
[10] M.R. Tinsley, S. Nkomo, K. Showalter, Nat. Phys. 8 (2012) 662.
[11] E.A. Martens, S. Thutupalli, A. Fourrière, O. Hallatschek, Proc. Natl. Acad. Sci. USA 110 (2013) 10563.
[12] T. Kapitaniak, P. Kuzma, J. Wojewoda, K. Czolczynski, Y. Maistrenko, Sci. Rep. 4 (2014) 6379.
[13] L.B. Gambuzza, A. Buscarino, S. Chessari, L. Fortuna, R. Meucci, M. Frasca, Phys. Rev. E 90 (2014) 032905.
[14] D.M. Abrams, S.H. Strogatz, Internat. J. Bifur. Chaos Appl. Sci. Engrg. 16 (2006) 21.
[15] C.A.S. Batista, R.L. Viana, Physica A 526 (2019) 120869.
[16] N. Kushwaha, N.K. Mendola, S. Ghosh, A.D. Kachhvah, S. Jalan, Front. Phys. 9 (2021) 147.
[17] M.A. Ganaie, S. Ghosh, N. Mendola, et al., Chaos 30 (6) (2020) 063128.
[18] K. Kumari, P. Gupta, Praveen, G. Shanker, Phys. Sci. Int. J. 9 (2016) 1–10.
[19] M. Wolfrum, O.E. Omel'chenko, S. Yanchuk, Y.L. Maistrenko, Chaos 21 (2011) 013112.
[20] C.A.S. Batista, R.L. Viana, Chaos Solitons & Fractals 131 (2020) 109501.
[21] I. Omelchenko, Y. Maistrenko, P. Hövel, E. Schöll, Phys. Rev. Lett. 106 (2011) 234102.
[22] K. Kaneko, Physica D 34 (1989) 1–41.
[23] M. Lukosevicius, H. Jaeger, Comput. Sci. Rev. 3 (2009) 127.
[24] Filippo M. Bianchi, S. Scardapana, S. Lokse, R. Jenssen, IEEE Trans. Neural Netw. Learn. Syst. 32 (2021) 2169.
[25] A. Tharwat, Int. J. Appl. Pattern Recognit. 3 (2016) 197.
[26] F.M. Bianchi, S. Scardapane, S. Lokse, R. Jenssen, Bidirectional deep-readout echo state networks, in: European Symposium on Artificial Neural Networks, 2018.
[27] F.M. Bianchi, L. Livi, C. Alippi, Investigating echo state networks dynamics by means of recurrence analysis, IEEE Trans. Neural Netw. Learn. Syst. 29 (2018) 427–439.
[28] A. Rodan, A.F. Sheta, H. Faris, Bidirectional reservoir networks trained using SVM+ privileged information for manufacturing process modeling, Soft Comput. 21 (2017) 6811–6824.
[29] P. Baldi, P. Sadowski, Understanding dropout, Adv. Neural Inf. Process. Syst. 26 (2013).
[30] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Adv. Neural Inf. Process. Syst. 153 (2007).
[31] R. Liu, T. Wu, B. Mozafari, Adam with bandit sampling for deep learning, Adv. Neural Inf. Process. Syst. 33 (2020) 5393–5404.