




Deep Learning for analyzing chaotic dynamics in biological time series: Insights from frog heart signals

Carmen Mayora-Cebollero^{a,*} , Flavio H. Fenton^b , Molly Halprin^b, Conner Herndon^b, Mikael J. Toye^b, Roberto Barrio^a 

^a IUMA, Computational Dynamics group and Department of Applied Mathematics, Universidad de Zaragoza, Zaragoza, Spain

^b School of Physics, Georgia Institute of Technology, Atlanta, USA

ARTICLE INFO

Communicated by J. Hu

Keywords:

Deep Learning
Chaos
Cardiac dynamics
Experimental data

ABSTRACT

The study of experimental data is a relevant task in several physical, chemical and biological applications. In particular, the analysis of chaotic dynamics in cardiac systems is crucial as it can be related to certain pathological arrhythmias. When working with short and noisy experimental time series, some standard techniques for chaos detection cannot provide reliable results because of such data characteristics. Moreover, when small datasets are available, some Deep Learning techniques cannot be applied directly (that is, using part of the data to train the network, and using the trained network to analyze the remaining dataset). To overcome all these limitations, we propose an automatic algorithm that combines Deep Learning and some selection strategies based on a mathematical model of the same nature of the experimental data. To demonstrate its performance, we test it with experimental data obtained from ex-vivo frog heart experiments, achieving highly accurate results.

1. Introduction

Nowadays, the analysis and study of experimental data is becoming more and more crucial to develop mathematical models and classify different possible behaviors. In data from real experiments, various features are common, such as differing lengths in the time series, possibly a very short amount of data in most cases, or the impossibility of measuring all the variables of the system (such as latent variables). In this study, we focus on a first attempt to automate the analysis of chaotic dynamics in cardiac time series (in our case, frog heart signals) using Deep Learning techniques and a periodically paced cardiac action potential map [1,2] to verify validity, as iterative maps remain a valuable tool for studying chaos in cardiac cell models [3–5]. Chaos dynamics in the heart [6–8] has been proposed to exist at two levels, in the ECG for healthy individuals [9,10] and in tissue level during fibrillation [11–13] or during fast pacing [14,15], as well as higher order periods [16,17]. Therefore, the methods presented here could help in the analysis and classification of chaos at different regimes as well as in different animal species, including humans. Moreover, characterizing chaos in these contexts is essential for developing targeted strategies and designing more effective therapeutic interventions, particularly those based on chaos control techniques for the termination of arrhythmias [18–21].

Various measures of complexity such as Lyapunov exponents [22–24], Permutation Entropy [25–28], and the 0-1 test for chaos [29] are often used to perform chaos analysis (distinguish regular (e.g., periodic or quasiperiodic) and chaotic behavior) of dynamical systems [30–37]. However, as these techniques are usually based on large time series, they sometimes present problems when used with real-world data as recordings are generally short and noisy [38,39]. In some cases, preprocessing tasks can be useful to deal with such kind of datasets [40,41]. Moreover, when dealing with large experimental datasets it is necessary to have an automatic algorithm for chaos analysis since human intervention on the entire dataset is not feasible. Recently, some authors have used Deep Learning (Artificial Neural Networks) to detect chaos in a dynamical system [42–45]. In particular, in [45], it was illustrated that using Deep Learning with just partial information, such as single-variable time series (without any previous preprocessing technique), is enough to obtain a complete understanding of the dynamical behavior (approximation of the full Lyapunov exponents spectrum). Could Deep Learning (DL) be applied to analyze chaotic dynamics in an experimental dataset? Notice that from the dynamical systems point of view, a chaos analysis consists of detecting if a time series has regular or chaotic behavior. From the DL perspective, it is a classification task. When applied to cardiac

* Corresponding author.

Email address: cmayora@unizar.es (C. Mayora-Cebollero).

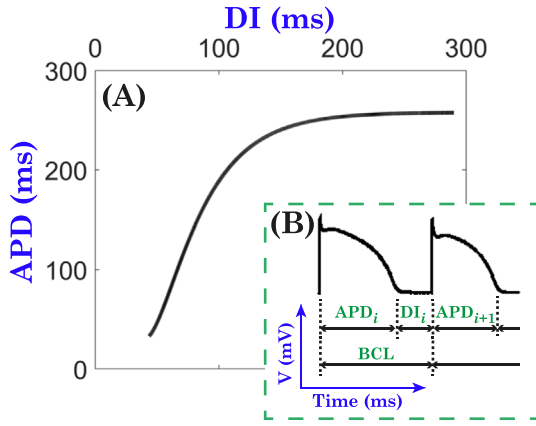


Fig. 1. (A) APD restitution curve fitted to the kinetics of the Beeler–Reuter model. (B) Scheme to illustrate the APD, DI and BCL concepts.

tissue data, such an approach could potentially enable the prediction and characterization of complex experimental dynamics [46,47].

During pacing of experimental or simulated cardiac cell or tissue, trans-membrane voltage signals, called action potentials, can be obtained that appear as shown in panel (B) of Fig. 1. These signals can be characterized by an Action Potential Duration (APD), a Diastolic Interval (DI), and a Basic Cycle Length (BCL). The APD is the duration from the onset of cell depolarization to the completion of repolarization, given a certain threshold, where most commonly a value between 75 % and 90 % of repolarization is used. The DI is then the time interval from the end of an APD to the onset of the next one, for the same threshold, which is essentially the recovery time of the cell. The BCL is the time interval between consecutive sinus beats or the pacing cycle length used to stimulate the heart when experiments are conducted.

Using the APD and DI data, an APD restitution curve can be constructed to predict the APD given a particular period of stimulation. This is a one-dimensional map that has been shown to be useful for predicting complex dynamics [48] and has been demonstrated experimentally to predict period-doubling bifurcations [49]. Usually, it is a monotonically increasing function, but some experiments have shown that it could be biphasic [50,51], which could lead to chaotic dynamics [52–54]. Mathematically, it is represented with a discrete equation of the form $APD_{i+1} = f(BCL - APD_i) = f(DI_i)$. In panel (A) of Fig. 1 we have the APD restitution curve fitted to the kinetics of the Beeler–Reuter model [1]. The APD restitution curve can be more complex, being a function of the history of the stimulation pacing protocol due to memory [55–58] and of calcium concentrations in the cells [59,60].

Some recent studies have used Machine Learning techniques to predict chaotic time series from simulated and experimental data [46,47, 61,62], however, to our knowledge they have not been used for classification of chaos and approximation of Lyapunov exponents. In this paper, we combine the use of the classical Logistic map information [63] to train Artificial Neural Networks (ANNs), and the Beeler–Reuter APD heart map model [1,64] to verify the validity and select the most suitable trained Artificial Neural Network. With these elements we build an algorithm to detect chaos in experimental time series and we apply it to real-world data obtained from live frog hearts. An important point is to realize how the use of basic information data, i.e. the use of the basic and classical Logistic map as training data, combined with the a posteriori selection of the network using a heart map (in this case) allows us to correctly detect the dynamics. This is a *de facto* proof of the universality of the chaos dynamics information regardless of the problem being studied.

This paper is organized as follows. In Section 2, we describe the developed DL algorithm to perform chaos analysis of heart time series

(Sections 2.1–2.4 are devoted to explain in detail all the steps of the algorithm, and in Section 2.5 we provide the pseudocode). In Section 3, we apply this algorithm to experimental datasets obtained from frog heart dynamics (in Section 3.1 the approved animal protocol and experimental setup are described, in Section 3.2 we show the performance of the algorithm in the datasets, and in Section 3.3 we carry out a brief comparison with standard techniques). Finally, in Section 4 we draw some conclusions.

PyTorch [65] has been used to perform all the DL experiments in this work.

2. DL algorithm for analyzing chaotic dynamics in biological time series

In this section we propose a new algorithm to analyze chaotic dynamics in biological (cardiac) time series. The algorithm consists of four steps:

Step 1: DL Setup and Training. 10 randomly initialized recurrent-like Artificial Neural Networks with the architecture proposed in [43] are trained for 2,000 epochs (early stopping and validation data are used) with time series of length 1,000 from the Logistic Map (data creation as explained in [43,66]).

Step 2: DL for Analyzing Chaotic Dynamics in a Mathematical Model. A test analysis is performed in a mathematical model with the same nature as the experimental data (in our case, we consider an APD heart map [1]) with each Artificial Neural Network trained in Step 1.

Step 3: Selection of the Most Suitable DL Network. Some criteria are applied to the results of Step 2 to automatically detect which is the most suitable network to perform the analysis of chaotic dynamics in biological (heart) time series.

Step 4: DL for Analyzing Chaotic Dynamics of Experimental Data. Analysis of chaotic dynamics is performed, using the network chosen in Step 3, on the experimental data (in our case, APD time series from frog heart signals).

In Sections 2.1–2.4, all the steps of the algorithm are explained in detail. In Section 2.5, the pseudocode of the algorithm can be found. A complete schematic diagram of the four steps is depicted in Fig. 2.

2.1. Step 1: DL setup and training

Deep Learning (DL) is the branch of Machine Learning that uses Artificial Neural Networks (ANNs) to learn from data with different levels of abstraction. Since the introduction of ANNs, several architectures have been proposed and widely used for various applications [47,67–69]. One type is the Recurrent Neural Network (RNN) usually applied for sequential processing, which makes it an ideal choice when working with time series. Basic RNNs present several training problems as exploding/vanishing gradients and catastrophic forgetting. On the one hand, some authors have recently proposed alternative training algorithms based on dynamical systems theory to avoid gradient drawbacks [70]. On the other hand, to try to alleviate those problems, new architectures as Long Short-Term Memory (LSTM) cells [67] or Gated Recurrent Units (GRUs) [71] have been developed. For our purpose we will use an architecture based on stacked LSTM cells with a final classification layer.

A Long Short-Term Memory (LSTM) cell (Fig. 3) is a DL architecture that processes information over time. At time step t , its inputs are an external data point $x(t)$ (usual input data for an ANN), a cell state $c(t-1)$, and a hidden state $h(t-1)$; and the outputs are the updated cell state $c(t)$, the updated hidden state $h(t)$, and the usual output $y(t)$ of an ANN (we take it equal to $h(t)$). The states $c(\cdot)$ and $h(\cdot)$ are devoted to keep information from previous time steps and are updated according to

$$c(t) = f(t) \otimes c(t-1) + i(t) \otimes g(t),$$

$$h(t) = o(t) \otimes \tanh(c(t)),$$

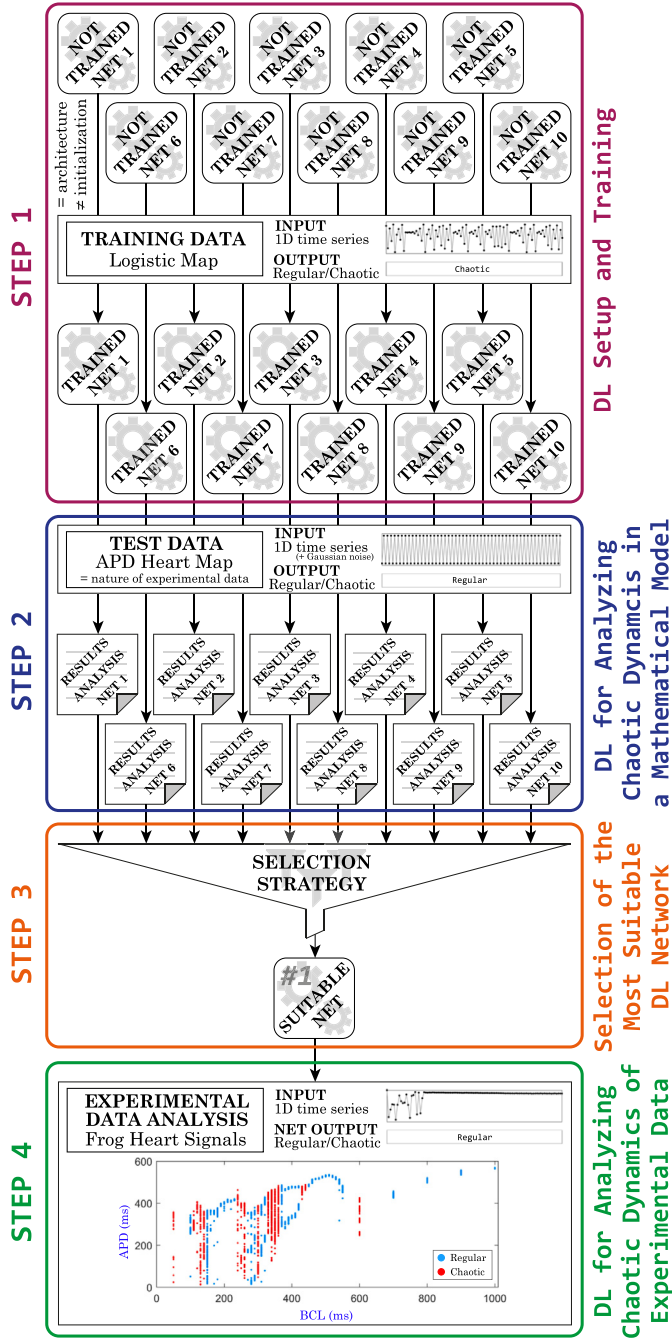


Fig. 2. Schematic diagram of the algorithm for analyzing chaotic dynamics in experimental time series.

where \otimes is the element-wise product, and $f(t)$, $g(t)$, $i(t)$ and $o(t)$ are given by

$$f(t) = \sigma(W_f^{[x]}x(t) + W_f^{[h]}h(t-1) + b_f),$$

$$g(t) = \tanh(W_g^{[x]}x(t) + W_g^{[h]}h(t-1) + b_g),$$

$$i(t) = \sigma(W_i^{[x]}x(t) + W_i^{[h]}h(t-1) + b_i),$$

$$o(t) = \sigma(W_o^{[x]}x(t) + W_o^{[h]}h(t-1) + b_o).$$

In these previous expressions, $\sigma(z) = 1/(1 + \exp(-z))$ and $\tanh(z) = 2\sigma(2z) - 1$ are the activation functions (sigmoid and hyperbolic tangent, respectively), and $W_*^{[x,h]}$ and b_* ($*$ $\in \{f, g, i, o\}$) are the trainable parameters (weights and biases, respectively) of the network. Note that

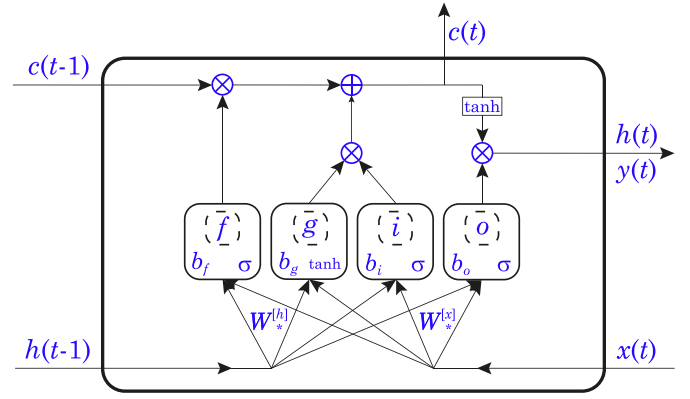


Fig. 3. Scheme of an LSTM cell.

because of the application of the sigmoid activation function, f , i and o act as gates that screen the information and memory of the network.

The ANN that we use for chaos analysis (introduced in [43]) has two stacked trainable LSTM cells (both are unidirectional with bias term and states of dimension 4) followed by a trainable linear classification layer of two neurons (one for each class: regular and chaotic) whose input is the last hidden state of both LSTM cells. Finally the softmax activation function, given by $\text{softmax}_i(z) = \exp(z_i) / (\sum_{j=1}^n \exp(z_j))$ with $z \in \mathbb{R}^n$, is applied to the output of the classification layer (to transform output values to scores, as usual for classification DL tasks). A graphical representation of the DL architecture used for chaos detection in this work is depicted in Fig. 4.

To fit the aforementioned trainable parameters (weights and biases), the network has to be trained using data (in our case, from the Logistic map). In fact, the training process consists of finding the value of the trainable parameters that minimizes the loss function (that measures the discrepancy between the network output and the expected output or label) for the given training dataset. As working with a classification task, we consider the standard cross-entropy loss function, and we use L^2 -regularization (weight decay $\beta = 10^{-5}$) to avoid overfitting. Therefore, the function to minimize is

$$-\sum_{j=1}^B \log(\text{NN}(x^j)) + \beta \sum_{k=1}^K w_k^2,$$

where $\text{NN}(x^j)$ is the network output (once the softmax function has been applied) for input x^j , K is the number of trainable parameters, w_k represents a trainable parameter (a weight or a bias), and B is the batch size (for optimization purposes, the training dataset is divided into disjoint subsets known as batches). The applied optimizer is Adam [72] with learning rate of 0.008. The network is trained for 2,000 epochs using early stopping technique (that is, the final network has the value of the trainable parameters that provides the lowest value for the validation dataset, a set of samples not present in the training set, during training process).

A key point related to the training process of the ANN is the selection of the training data. In our case, as the experimental data consists of short time series, we try to use basic and generic discrete data obtained from one of the most well-known discrete maps, the Logistic map [63]. One interesting fact in dynamical systems theory is the universality of numerous dynamical phenomena, like Feigenbaum constants [73], chaos dynamics and so on. Therefore, as training data we use generic data to avoid particularizing the training process.

The Logistic map [63] is a well-known discrete one-dimensional dynamical system that presents great dynamical richness (period-doubling bifurcation, chaos, ...) despite its simplicity. It is given by

$$x_{i+1} = \alpha x_i (1 - x_i), \quad (1)$$

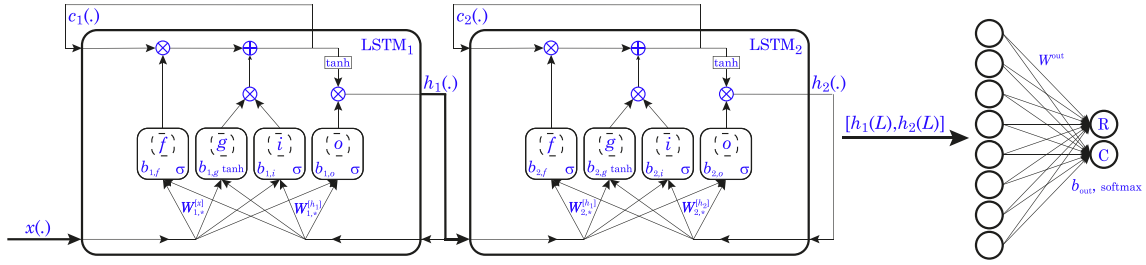


Fig. 4. Graphical representation of the DL architecture used in this work (L is the length of the input time series, R corresponds to regular class, and C to chaotic one). See the text for more details.

where x_i is the variable at the i -th iteration, and α is the bifurcation parameter (that can be interpreted as the growth rate of the system). The Logistic map is constructed in such a way that the variable only takes values in the interval $[0, 1]$. As it is a one-dimensional map, the Lyapunov exponent (LE) [22] can be easily computed applying equation

$$LE = \frac{1}{T} \sum_{j=1}^T \log |\alpha(1 - 2x_j)|, \quad (2)$$

where the expression inside the absolute value corresponds to the derivative of the right-hand side of Eq. (1) with respect to the system variable x , \log is the natural logarithm, and T corresponds to a large enough value of the iteration number. An appropriate number of transient iterations has to be computed prior to applying the formula.

In panel (A) of Fig. 5 we have represented the bifurcation diagram of the Logistic map when $\alpha \in [1, 4]$ and $x_0 = 0.5$. In this bifurcation diagram the dynamics of the Logistic map can be seen clearly: For small values of α the dynamics converge into an equilibrium point, later there is a cascade of period-doubling bifurcations which result in chaotic dynamics for values of α larger than 3.5 approximately. In panel (B), LEs have been depicted. We can check that for regular behavior (equilibrium points and periodic orbits) the LE is negative or zero, and it is positive for chaos. For comparison purposes, we also use other two chaos detection methods that have been used for time series classification in literature. In panel (C), we have depicted the (normalized) results obtained by applying the Permutation Entropy (PE) method (Antropy [74], an open-source Python package, has been applied). Notice that to apply the Permutation Entropy technique [25,28], we have to set the permutation order n and the delay τ . To that goal, we follow the parameter selection procedure of Ref. [28]: As we are working with a one-dimensional discrete dynamical system and a time series length of 1,000, we fix the permutation order to $n = 5$ (maximum n value such that the length of the time series is greater than $5n$!) and the delay to $\tau = 1$. For the Permutation Entropy, lower values indicate less complexity, hence more regular behavior, whereas higher values indicate greater complexity, associated with more chaotic or irregular dynamics. In panel (D), we have represented the results provided by the 0-1 test for chaos (MATLAB implementation in [75] has been used), where values around 0 indicate regular behavior, and those that are approximately 1 mean chaotic behavior. By comparing plots (B), (C), and (D), where different complexity measures have been applied, we observe that the results are consistent across them (see red vertical dashed lines that indicate the boundary between regular and chaotic regions).

To train the network with early stopping technique, the data is divided into three datasets: Training dataset contains data to learn from, validation dataset prevents overfitting, and test set checks the performance of the trained network. Each dataset contains time series and their corresponding LE computed with the classical technique in [22] (that is used as label to check the behavior detected by the ANN). To obtain the time series, 12,000 time steps are computed with Eq. (1) and the last 1,000 correspond to the time series. To compute the LE, 12,000

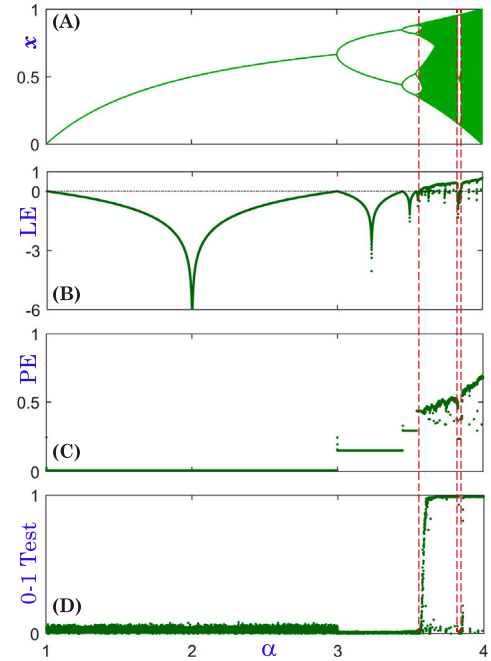


Fig. 5. α -parametric line of the Logistic map ($x_0 = 0.5$). (A) Bifurcation diagram. (B) Lyapunov exponents (LEs). (C) Permutation Entropy (PE). (D) 0-1 test for chaos. The red vertical dashed lines across panels (A–D) indicate the boundary between regular and chaotic regions to show that the results provided by all techniques are consistent.

time steps are obtained with Eq. (1), first 1,000 points are discarded as transient, and Eq. (2) is applied to the remaining ones.

To obtain the training dataset, we create two raw datasets, one with initial condition $x_0 = 0.5$ and other with $x_0 = 0.9$. In both sets, the time series have length 1,000 and the parameter α takes 12,000 equidistant values in $[0, 4]$. Data is screened deleting similar samples (two time series are similar if their distance in infinity norm is less than 10^{-4}), and finally 2,000 regular and 2,000 chaotic samples are selected randomly to build such training dataset. For validation set, a raw dataset with time series of length 1,000, initial condition $x_0 = 0.75$, and the parameter α taking 12,000 equidistant values in $[0, 4]$ is created. It is screened and 1,000 time series of each dynamical behavior are selected randomly to obtain the final validation dataset. The test set is created similarly to validation dataset but with $x_0 = 0.8$ as the initial condition.

As shown in [43], if we train 10 randomly initialized ANNs with the aforementioned architecture and the built train and validation sets (trained networks are checked with the test set), a powerful tool to detect chaos in the Logistic map is obtained (accuracy greater than 95 % in all datasets and in all the trained networks).

Table 1

Summary of the parameters related to the network architecture and its training process.

DL Architecture
LSTM cell + LSTM cell + Classification layer
LSTM
Input size = 1
Hidden size = 4
Number of layers = 2 (two stacked LSTM cells)
Bias = True
Bidirectional = False
Classification (linear) layer
Input size = 8 (last hidden state of both LSTM cells)
Output size = 2 (2 classes: regular, chaotic)
Activation function: Softmax
Loss Function
Cross Entropy Loss + L^2 -regularization
Weight decay (β) = 10^{-5}
Optimizer
Adam optimizer
Learning rate = 0.008
Training
2,000 epochs + Early stopping
Training data. Logistic map: $x_{i+1} = \alpha x_i (1 - x_i)$
$x_0 \in \{0.5, 0.9\}$, $\alpha \in [0, 4)$
Time series length = 1,000
2,000 regular samples + 2,000 chaotic samples
Validation data. Logistic map: $x_{i+1} = \alpha x_i (1 - x_i)$
$x_0 = 0.75$, $\alpha \in [0, 4)$
Time series length = 1,000
1,000 regular samples + 1,000 chaotic samples

All the parameters involved in the implementation of the network architecture and its training process are summarized in Table 1.

Remark 1. Equivalent experiments have been carried out for different values of the time series length and the total number of epochs during training, but values 1,000 and 2,000, respectively, seem to give the best results. Notice that in [43], length 1,000 was also proposed as the best option to train networks for a chaos detection task.

2.2. Step 2: DL for analyzing chaotic dynamics in a mathematical model

Our final objective (Step 4) is to analyze chaotic dynamics in heart time series. We remark that the 10 ANNs have been trained with data from the Logistic map (Step 1), not with heart-like data. Moreover, the time series that will be analyzed in Step 4 correspond to experimental recordings, that in general are short and noisy. Therefore, in this step we use some heart dynamics information, a simple cardiac map model, to perform several numerical tests on all trained ANNs. This is an important point as it connects generic and universal dynamics of the classical Logistic map with a more specific cardiac map model. That is, the training is in some way universal, but the selection of the particular ANN is done by a more specific mathematical model focused on the nature of the experimental data.

The APD restitution curve (which describes the dynamics of a single cardiac cell) fitted to the kinetics of Beeler–Reuter model [64] gives rise to this discrete equation [1] (named as Beeler–Reuter APD heart map in what follows):

$$\text{APD}_{i+1} = 258 + 125 \exp(-0.068(\text{DI}_i - 43.54)) - 350 \exp(-0.028(\text{DI}_i - 43.54)),$$

where APD_{i+1} is the Action Potential Duration (APD, duration from the onset of cell depolarization to the completion of repolarization) of the $(i + 1)$ -th stimulus and $\text{DI}_i = n \text{BCL} - \text{APD}_i$ is the Diastolic Interval (DI,

recovery time of the cell) of previous responding stimulus. Therefore, the map equation can be rewritten as

$$\text{APD}_{i+1} = 258 + 125 \exp(-0.068(n \text{BCL} - \text{APD}_i - 43.54)) - 350 \exp(-0.028(n \text{BCL} - \text{APD}_i - 43.54)), \quad (3)$$

where BCL is the Basic Cycle Length (BCL), that is, the time between two consecutive pacing stimuli, and n is the parameter block (lower $n \in \mathbb{N}$ such that $n \text{BCL} - \text{APD}_i \geq \text{DI}_{\min}$, with DI_{\min} the minimum DI, whose value is set to 43.54 ms). Notice that the BCL can be considered as the bifurcation parameter as it is independent of discrete time.

Since the Beeler–Reuter APD heart map is a one-dimensional map, the Lyapunov exponent [22] can be obtained by applying the formula

$$\text{LE} = \frac{1}{T} \sum_{j=1}^T \log \left| 8.5 \exp(-0.068(n \text{BCL} - \text{APD}_j - 43.54)) - 9.8 \exp(-0.028(n \text{BCL} - \text{APD}_j - 43.54)) \right|, \quad (4)$$

where the expression inside the absolute value corresponds to the derivative of the right-hand side of Eq. (3) with respect to the system variable APD, \log is the natural logarithm, and T corresponds to a sufficiently large value of the iteration number. An appropriate number of transient iterations must be calculated before applying the formula.

In panel (A) of Fig. 6, we have depicted the bifurcation diagram of the Beeler–Reuter APD heart map for $\text{BCL} \in [55, 400)\text{ms}$ and $\text{APD}_0 = 240\text{ms}$. If we move from higher to lower BCL values in such a bifurcation diagram we can see equilibrium points, a period-doubling bifurcation, and a 2:1 block (the cell responds one of each two pacing stimuli) followed by a period-doubling, chaos, and higher-order blocks (see [1] for more dynamical information). In panel (B), the LEs calculated with Eq. (4) have been represented. In this panel it can be seen that regular behavior (equilibrium points and periodic orbits) corresponds to negative or zero LEs, and chaos to positive LE values. In panel (C), we have depicted the (normalized) results obtained applying Permutation Entropy (Antropy [74], an open-source Python package, has been applied). As indicated for the Logistic map, to apply the Permutation Entropy technique we have to set the permutation order and the delay. As we are also working with a one-dimensional discrete dynamical system and a time series length of 1,000, as clarified before, we consider permutation order 5 and delay 1. For this technique, lower values indicate less complexity, hence more regular behavior, whereas higher values indicate greater complexity, associated with more chaotic or irregular dynamics. In panel (D), we have represented the results provided by the 0-1 test for chaos (MATLAB implementation in [75] has been used), where values around 0 indicate regular behavior, and those that are approximately 1 mean chaotic behavior. By comparing plots (B), (C), and (D), where different complexity measures have been applied, we observe that the results are consistent across them (see red vertical dashed lines that indicate the boundary between regular and chaotic regions).

At this point, we are going to perform a chaos analysis of one BCL-parametric line (see Fig. 6) of this APD heart map using the networks trained in Step 1. Such BCL-parametric line has been obtained taking 12,000 equidistant values for the BCL in the interval $[55, 400)\text{ms}$ and initial condition $\text{APD}_0 = 240\text{ms}$. Eq. (3) is applied for 20,500 time points. First 500 points are discarded as transient, and the remaining ones are used for the LEs computation with Eq. (4). The time series obtained for each BCL value are constructed with the last 1,000 time points. Later, a component-wise Gaussian random noise with strength $\mu \in \{0, 0.5, 1.0\}$ is added to each time series (notice that when $\mu = 0$, no noise is added). The analyses of time series with Gaussian noise are repeated 100 times to give results as mean \pm standard deviation. Notice that the time series from the Logistic map take values in $[0, 1]$. This does not occur with this Beeler–Reuter APD heart map, so we normalize the data to the interval $[0, 1]$ before performing the DL analysis of chaotic dynamics (a random number sampled uniformly in such interval is assigned to constant time

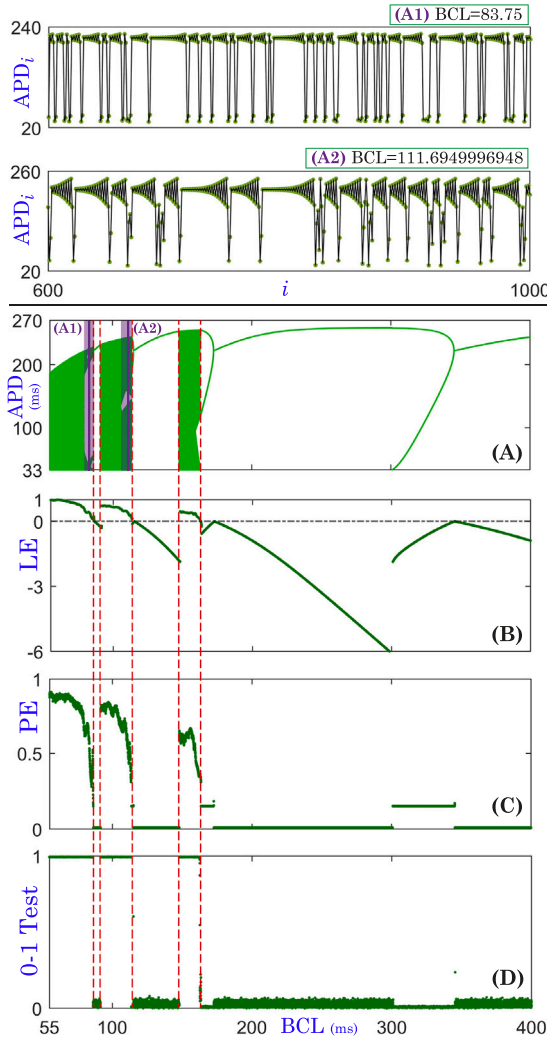


Fig. 6. BCL-parametric analysis of the Beeler–Reuter APD heart map ($APD_0 = 240\text{ms}$). (A) Bifurcation diagram. Purple shading corresponds to regions where the network fails the most in the classification between regular and chaotic dynamics. (A1) and (A2) are two of these incorrectly classified time series (in green we have the points of the time series, and we have joined such points with black segments for the ease of viewing). (B) Lyapunov exponents (LEs). (C) Permutation Entropy (PE). (D) 0-1 test for chaos. The red vertical dashed lines across panels (A–D) indicate the boundary between regular and chaotic regions to show that the results provided by all techniques are consistent.

series and a linear mapping between the range of non-constant samples to the interval $[0, 1]$ is done otherwise).

Remark 2. The initial condition and the times used for LEs computation in the APD heart map are taken as in [76].

Now, we present different accuracy indicators (adapted to the use of experimental data and the cardiac model) to provide information to select in *Step 3* the most appropriate DL network (among all computed in *Step 1*) to analyze heart data under noise and the absence of it.

The *Accuracy* (percentage of samples that have been correctly classified) is the measure proposed to check the performance of the networks. It is defined as

$$\text{Accuracy (\%)} = \frac{T_R + T_C}{T_R + T_C + F_R + F_C} \cdot 100,$$

where T_R and T_C are the number of true regular and true chaotic samples (that is, the number of samples that are correctly classified by the

network as regular and chaotic, respectively); and F_R and F_C are the false regular and false chaotic samples (that is, the number of samples that are incorrectly classified by the DL network as regular and chaotic, respectively). Notice that if the dataset in which this formula is applied is unbalanced (we do not have the same number of samples for each dynamical behavior) as occurs in our BCL-parametric line (19.708 % of the samples are chaotic according to the LE value, while the remaining 80.292 % have a regular behavior), this measure cannot be reliable. For example, in a dataset with a large number of regular and a small number of chaotic samples, even if all the chaotic samples are not detected correctly, the accuracy can be close to 100 % if all the regular samples are correctly classified, and this does not mean that the network is able to perform the chaos detection analysis properly. To avoid this problem, we use other variants of the accuracy, the *Accuracy Chaotic* and *Accuracy Regular* that compute the percentage of chaotic and regular samples, respectively, that are correctly classified. The corresponding formulas are as follows

$$\text{Accuracy Chaotic (\%)} = \frac{T_C}{T_C + F_R} \cdot 100,$$

$$\text{Accuracy Regular (\%)} = \frac{T_R}{T_R + F_C} \cdot 100.$$

Remark 3. If we consider chaotic as the positive class and regular as the negative one, accuracy chaotic corresponds to the sensitivity or recall (in %), and accuracy regular is the specificity (in %).

Notice that a regular dynamical behavior includes mainly two different types of dynamics: equilibrium points (EPs) and periodic orbits (POs). To complete our analysis we also compute the *Accuracy EPs* and *Accuracy POs*, that represent the percentage of equilibrium points and periodic orbits, respectively, that have been classified correctly as regular:

$$\text{Accuracy EPs (\%)} = \frac{T_{R|EPs}}{T_{R|EPs} + F_{C|EPs}} \cdot 100,$$

$$\text{Accuracy POs (\%)} = \frac{T_{R|POs}}{T_{R|POs} + F_{C|POs}} \cdot 100,$$

with $T_{R|EPs}$ and $T_{R|POs}$ the number of equilibrium points and periodic orbits, respectively, that have been classified correctly as regular; and $F_{C|EPs}$ and $F_{C|POs}$ the number of equilibrium points and periodic orbits, respectively, that are incorrectly classified as chaotic. In our parametric line, 79.803 % of the regular samples are equilibrium points and the remaining 20.197 % present periodic behavior, so we consider that the measures that we have just defined are important to check that with DL both dynamics are detected properly as regular.

Finally, we define an indicator to detect if the network detection is robust against noise. We refer to it as $\text{diff}_{0-\bar{\mu}}$ and it corresponds to the percentage of samples that the network does not detect with the same behavior in the analysis of data without noise ($\mu = 0$) and with noise strength $\bar{\mu} \neq 0$ (in our analysis $\bar{\mu} = 0.5$ or $\bar{\mu} = 1.0$). That is, if

$$S_{\mu} = \{\text{classification obtained from data with noise strength } \mu\},$$

then

$$\text{diff}_{0-\bar{\mu}} = 100 - \frac{\# [S_{\mu=0} \cap S_{\mu=\bar{\mu}}]}{\text{total number of samples}} \cdot 100,$$

where # is used to refer to the cardinality.

Remark 4. The tests performed in this paper with Beeler–Reuter APD heart map can also be performed with Lewis and Guevara APD heart map [76] and similar/equivalent results are obtained.

2.3. Step 3: Selection of the most suitable DL network

Once we have performed the DL chaos analysis of the APD heart map (*Step 2*) we have to apply some criteria to automatically select (without

direct human supervision) a robust network against noise that can properly perform the chaos detection in the Beeler–Reuter APD heart map. We expect that the selected network will be able to properly perform the DL chaos analysis in biological time series of frog heart signals.

The automatic selection criteria takes into account the accuracy-like measures and the indicator $diff_{0-\bar{\mu}}$ (for $\bar{\mu} \in \{0.5, 1.0\}$) defined in *Step 2*. In particular, our selection criteria is as follows: All accuracies (accuracy, accuracy chaotic, accuracy regular, accuracy EPs and accuracy POs) have to be greater than 75 % in mean (to ensure good performance in chaos detection task), and $diff_{0-0.5}$ and $diff_{0-1.0}$ have to be less than 1 % in mean (to ensure robustness in the detection against noise). Moreover, we impose that the standard deviation for all the measures and indicators has to be less than 1 % (to ensure that results are quite independent of randomness).

Notice that we are checking if any of the 10 trained networks (*Step 1*) has gone further, and besides having learned to detect chaos in the Logistic map, it is able to generalize its detection to discrete heart dynamics. Note that the ANNs have only been trained with dynamics from the Logistic map, the data was not preprocessed in any special way, and no extra dynamical information has been given to the networks, so the proposed task is not easy and we consider that the aforementioned criteria is reasonable.

In *Table 2* we present the results for the DL analysis of chaotic dynamics of the network that satisfies all the imposed criteria. In particular, for the cases with noise (second and third columns), the analyses have been performed 100 times due to randomness, and results correspond to mean \pm standard deviation.

Let us focus on the results of the without noise case ($\mu = 0$). Notice that the value of all the accuracies (except accuracy chaotic) are greater than 95 %. Notice that both types of regular behavior (equilibrium points and periodic orbits) are properly detected in almost all the cases. It is necessary to check why the accuracy chaotic is lower than the other accuracy measures (but still greater than 75 %). Most of the samples that are incorrectly detected as regular when their behavior is chaotic are located in the purple shaded boundary areas of the bifurcation diagram of the APD heart map in panel (A) of *Fig. 6*. In panels (A1) and (A2) of this figure we have an example of an incorrectly classified sample on each purple region (BCL = 83.75ms and BCL = 111.694996948ms, respectively). Notice that both samples are quite similar. As we can see, the general behavior is chaotic. However, it highlights the upper parts of the time series in which, compared with the general range of the APD values of the time series, there is not much variability. This is a behavior that we do not expect to be present in the Logistic map where the networks are trained. Taking into account that the data used to train the networks does not have extra dynamical information beyond that provided by the time series, and this behavior is not present, to require the network to detect it is very demanding.

For the analyses with noise (noise strength $\mu \in \{0.5, 1.0\}$), all the accuracy results are quite good and similar to those given by the without noise case (with a standard deviation less than 0.15 in all cases). This is confirmed by the $diff_{0-0.5}$ and $diff_{0-1.0}$ indicators that are lower than 0.5 % in mean and with standard deviation lower than 0.06, increasing its value with noise strength as expected.

2.4. Step 4: DL for analyzing chaotic dynamics of experimental data

So far, we have trained 10 randomly initialized recurrent-like Artificial Neural Networks using data from the Logistic map (*Step 1*), we have performed DL chaos analyses of an APD heart map with all the trained networks (*Step 2*), and we have applied some criteria on these analyses to select one network that performs well and is robust against noise (*Step 3*). Now, we have all the ingredients to attempt a DL chaos analysis of biological time series from heart dynamics.

When training the networks, the data comes from the Logistic map. As already highlighted before, this equation is constructed in such a way that time series values are in the interval $[0, 1]$. In general, experimental

Table 2

Results of the DL chaos analysis in the Beeler–Reuter APD heart map with the network chosen by the criteria applied in *Step 3*. Notice that $\mu = 0$ corresponds to the analysis without noise. Results for $\mu \in \{0.5, 1.0\}$ are given as mean \pm standard deviation for 100 trials.

	$\mu = 0$	$\mu = 0.5$	$\mu = 1.0$
Accuracy (%)	95.092	94.752 \pm 0.050	94.651 \pm 0.060
Accuracy chaotic (%)	76.195	75.951 \pm 0.071	75.552 \pm 0.112
Accuracy regular (%)	99.730	99.367 \pm 0.060	99.339 \pm 0.069
Accuracy EPs (%)	100	99.545 \pm 0.075	99.542 \pm 0.087
Accuracy POs (%)	98.666	98.664 \pm 0.009	98.541 \pm 0.037
$diff_{0-\bar{\mu}}$ (%)	–	0.361 \pm 0.049	0.493 \pm 0.059

data is not in the interval $[0, 1]$, so we have to normalize it to that interval (the trained networks will not be able to process data properly in whatever other rank). With the data already normalized, time series are given as input to the chosen network in *Step 3*. The output of the network will give us the information about the behavior (regular or chaotic) of such time series. Therefore, the defined algorithm allows performing chaos analysis of heart time series without human supervision.

Remark 5. Notice that the trained network is a recurrent-like neural network, therefore, the length of the input is not fixed and time series of whatever length can be processed. For example, with other architectures as the Multi-Layer Perceptron (obtained when perceptrons are stacked) the input size has to be constant.

2.5. Pseudocode of the algorithm for analyzing chaotic dynamics in biological time series

In *Algorithm 1* we present the pseudocode of the DL algorithm for analyzing chaotic dynamics in biological time series that we have described in *Sections 2.1–2.4*. For simplicity and better comprehension, we have used some abbreviations. In particular, with `num_nets` we refer to the number of ANNs that have the same architecture but random initialization (it is set to 10 as indicated in *Section 2.1*). `ANNarch` corresponds to the network architecture described in *Section 2.1*. `dataLM*` with $* \in \{\text{train, val, test}\}$ is the data from the Logistic map used to train, validate, and test the network, respectively (such data includes the time series `dataLM*time series` and the corresponding labels `dataLM*labels`). The other two datasets used in the algorithm are the BCL-parametric line of the Beeler–Reuter APD heart map referred as `dataBRBCL-line` (with `dataBRBCL-linetime series` the time series, and `dataBRBCL-linelabels` the labels), and the experimental dataset of heart dynamics indicated as `dataExperimentaltime series` (note that in this case we do not have the labels `dataExperimentallabels`). When using the notation `acc-like μ_j (i)` for $j = 1, 2, 3$ and $i = 1, \dots, \text{num_nets}$ (line 15 of the pseudocode), we consider that we have a list that includes the accuracy, the accuracy chaotic, the accuracy regular, the accuracy EPs and the accuracy POs of net i when noise with strength μ_j is added to the data. In addition, these values (as well as the values of the robustness indicator) are given as mean \pm standard deviation (mean \pm std). Therefore in the criteria of *Step 3* (see line 23), for example, condition `acc-like μ_j mean(i) > 75 %` indicates that all the elements of the list have to be greater than 75 % in mean.

Remark 6. Notice that in *Step 2*, for case $j = 1$, we have $\mu_j = \mu_1 = 0$, so in the part of the algorithm where Gaussian noise is added (line 12 of the pseudocode) the data does not change. Moreover, for this value of j in this step of the algorithm, the computation of the robustness indicator $diff_{\mu_1-\mu_j} = diff_{\mu_1-\mu_1}$ (line 16 of the pseudocode) is trivial as it will always be equal to 0 % (for this reason, it is not included in the criteria of *Step 3*, see line 23 of the pseudocode). An *if* statement could be added to avoid calculating it.

Algorithm 1 DL algorithm for analyzing chaotic dynamics in biological time series.

```

1: procedure STEP 1 (num_nets = 10, ANNarch, dataLMtrain, dataLMval, dataLMtest) ▷ ANNs Framework
2:   parallel for  $i = 1$  to num_nets
3:     Net( $i$ )  $\leftarrow$  RandomInitialization(ANNarch) ▷ Random initialization of ANN trainable parameters
4:     T-Net( $i$ )  $\leftarrow$  Train(Net( $i$ ), dataLMtrain, dataLMval) ▷ Training
5:     Test(T-Net( $i$ ), dataLMtest) ▷ Test
6:   return T-Net( $i$ )
7: end parallel for
8: end procedure
9: procedure STEP 2 ( $\mu_1 = 0, \mu_2 = 0.5, \mu_3 = 1.0$ , T-Net(1), ..., T-Net(num_nets), dataBRBCL-line) ▷ Analysis APD Heart Map
10:  parallel for  $i = 1$  to num_nets
11:    for  $j = 1$  to 3 do
12:      dataBRBCL-line,  $\mu_j$ time series  $\leftarrow$  dataBRBCL-linetime series +  $\mu_j \cdot \text{Gaussian}(0, 1)$  ▷ Add noise with strength  $\mu_j$ 
13:      N-dataBRBCL-line,  $\mu_j$ time series  $\leftarrow$  Normalization(dataBRBCL-line,  $\mu_j$ time series) ▷ Data normalization
14:      result $\mu_j$ ( $i$ )  $\leftarrow$  DLAnalysis(T-Net( $i$ ), N-dataBRBCL-line,  $\mu_j$ time series) ▷ DL chaos analysis
15:      acc-like $\mu_j$ ( $i$ )  $\leftarrow$  AccuracylikeMeasures(result $\mu_j$ ( $i$ ), N-dataBRBCL-linelabels) ▷ Accuracy-like measures
16:      diff $\mu_1 - \mu_j$ ( $i$ )  $\leftarrow$  RobustnessIndicator(result $\mu_1$ ( $i$ ), result $\mu_j$ ( $i$ )) ▷ Robustness indicator
17:      return acc-like $\mu_j$ ( $i$ ), diff $\mu_1 - \mu_j$ ( $i$ )
18:    end for
19:  end parallel for
20: end procedure
21: procedure STEP 3 (acc-like $\mu_j$ ( $i$ ), diff $[\mu_1=0]-\mu_j$ ( $i$ ) with  $i = 1, \dots, \text{num\_nets}$ ,  $j = 1, 2, 3$ ) ▷ Selection of suitable ANN
22:  parallel for  $i = 1$  to num_nets
23:    if acc-like $\mu_1$ ( $i$ ) > 75 % and acc-like $\mu_2$ mean( $i$ ) > 75 % and acc-like $\mu_3$ std( $i$ ) < 1 % and acc-like $\mu_3$ mean( $i$ ) > 75 % and acc-like $\mu_2$ std( $i$ ) < 1 % and diff $\mu_1 - \mu_2$ mean( $i$ ) < 1 % and diff $\mu_1 - \mu_2$ std( $i$ ) < 1 % and diff $\mu_1 - \mu_3$ mean( $i$ ) < 1 % and diff $\mu_1 - \mu_3$ std( $i$ ) < 1 % then
24:      suitableIndex  $\leftarrow i$  ▷ Index of the ANN that satisfies the criteria
25:      return suitableIndex
26:    end if
27:  end parallel for
28: end procedure
29: procedure STEP 4 (T-Net(suitableIndex), dataExperimentaltime series) ▷ Analysis Experimental Data
30:  parallel for  $k = 1$  to size(dataExperimentaltime series)
31:    N-dataExperimentaltime series( $k$ )  $\leftarrow$  Normalization(dataExperimentaltime series( $k$ )) ▷ Data normalization
32:    result( $k$ )  $\leftarrow$  DLAnalysis(T-Net(suitableIndex), N-dataExperimentaltime series( $k$ )) ▷ DL chaos analysis experimental data
33:  return result( $k$ )
34:  end parallel for
35: end procedure

```

3. Results of DL algorithm for analyzing chaotic dynamics in biological time series: Frog heart signals

In this section we explain how the experimental data (frog heart signals) is obtained (Section 3.1), we apply the previous detailed Algorithm 1 to analyze chaotic dynamics in such data (Section 3.2), and we briefly compare these results with those provided by standard techniques (Section 3.3).

3.1. Data. Frog cardiomyocyte recordings

In this subsection we explain in detail the protocol of the experiment and how it has been carried out to obtain the frog heart signals.

The frog heart experiments were performed following an approved Georgia Tech, Institutional Animal Care and Use Committee (IACUC) protocol # A100673. Frogs were euthanized via rapid decapitation followed by pithing of the brain and spinal cord which ensures the cessation of neural activity avoiding any pain. The heart is then quickly excised and cannulated via the aorta using a syringe filled with Tyrode solution, a blood substitute, to wash out all the blood (see Fig. 7). This process ensured a blood-clots free preparation and allowed the heart to be kept physiologically viable in a container at room temperature, also filled with Tyrode solution, throughout the experiment.

Transmembrane voltage signals were recorded as described in previous preparations [77,78]. Briefly, individual heart cells from the whole heart were impaled with a fine-tipped micro-electrode, which

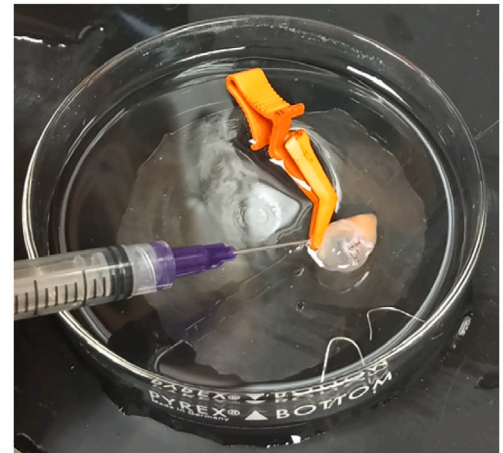


Fig. 7. Photograph taken during an experiment with a frog heart in Professor Flavio H. Fenton's laboratory at Georgia Tech.

was connected to a high-impedance amplifier to minimize signal loss. The amplified signals were routed to a data acquisition (DAQ) board interfaced with a computer for recording and analysis. This setup enabled precise voltage recordings of the electrical activity of the cardiac cells.

To induce variations in the Basic Cycle Length (BCL) of pacing, an isolated external current stimulation device was employed. The stimulating electrode was positioned about 0.5 cm from the recording micro-electrode and delivered currents at twice the excitation threshold. This ensured effective and consistent stimulation of the cardiac tissue for the experimental protocols.

We consider two experimental datasets (Dataset 1 and Dataset 2) obtained as explained before from two different frog hearts. Both sets of samples contain voltage time series with several APDs, each one with a different BCL. That is, the time series correspond to the Action Potential Duration of frog heart dynamics for different pacing rates. All signals were recorded after the BCL was left for several minutes of accommodation to that BCL.

In Dataset 1 we have 52 time series whose minimum length (number of points) is 15 and the maximum is 205. In this case, $BCL \in [50, 1000]$ (ms). In panel (A) of Fig. 9 (ignore colors for now) we have the bifurcation diagram obtained for this experimental dataset (x -axis is the pacing rate BCL and y -axis is the APD-time series). In panel (B) of this figure (ignore colors for now) we have the length (number of points) of each time series.

Dataset 2 contains 24 time series. Except for an unusual long time series of 521 points, the remaining ones have a minimum length (number of points) of 21 and a maximum of 138. For this dataset, $BCL \in [200, 1000]$ (ms). In panel (A) of Fig. 12 (ignore colors for now) we have the bifurcation diagram obtained for these experimental frog heart signals (x -axis is the pacing rate BCL and y -axis is the APD-time series). In panel (B) of such figure (ignore colors for now) we have the length (number of points) of each time series.

3.2. Results

We are going to apply the DL algorithm for chaos analysis of heart time series defined in Section 2 to some experimental data. That is, once Steps 1–3 (see Sections 2.1–2.3) have been carried out, and a network that we expect to be suitable for chaos detection has been obtained, we use it on the experimental datasets of frog heart signals defined in Section 3.1 (Step 4).

To check if the DL analysis of chaotic dynamics has been successful, we need to label the experimental dataset manually according to expert criteria. In general, we consider the asymptotic behavior to label the data. In particular, the rules taken into account for such classification are as follows:

- Rule a. If a time series seems to have long chaotic transient behavior (more than a quarter of the length) and only some regularity can be inferred at the end (see Fig. 8(A) for an example), it is considered that the behavior is chaotic. In this case, we cannot ensure that the asymptotic behavior is regular as not enough information is provided.
- Rule b. If a time series seems to have short chaotic transient behavior (less than a quarter of the length) and some regularity later (as in Fig. 8(B)), it is classified by the expert as regular.
- Rule c. If some regular windows are present in a general chaotic behavior (see Fig. 8(C)), chaotic behavior is assigned.
- Rule d. As we work with experimental data, the dynamics corresponding to equilibrium points are not constant values over time. We consider that a time series is an equilibrium point or represents regular behavior if when transient has ended (that is, in the last three quarters according to Rule b), the range of values taken by the variable before normalization is small (a difference less than 25 approximately) or a monotonically increasing or decreasing behavior can be inferred (see Fig. 8(D) for an example).

Taking into account these rules, in Dataset 1, 13 of the 52 samples have chaotic behavior, and the remaining 39 are regular. And in Dataset 2, we have that 20 of the 24 samples are regular and just 4 are chaotic.

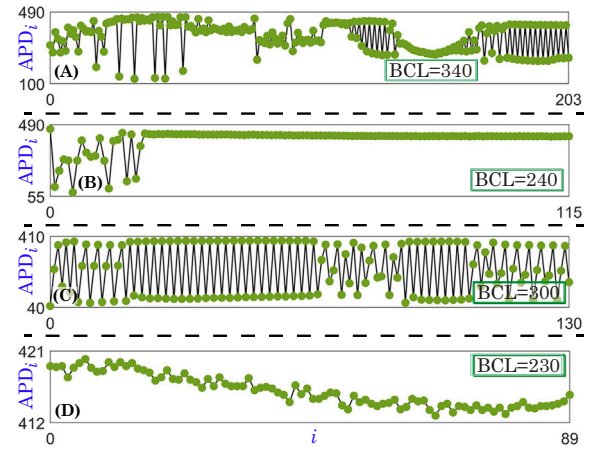


Fig. 8. Time series to illustrate the rules applied when labeling experimental data manually (expert criteria). (A) corresponds to Rule a, (B) to Rule b, (C) to Rule c, and (D) to Rule d (see the text for more details). In green we have the points that correspond to the time series, and we have joined such points with black segments for the ease of viewing.

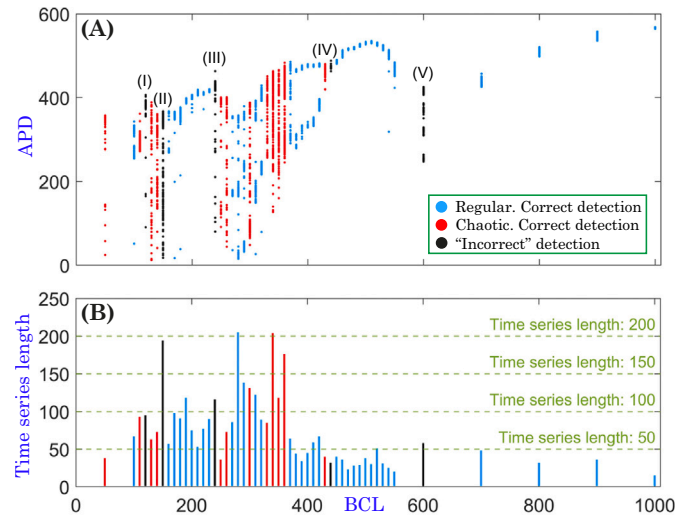


Fig. 9. Chaos analysis of experimental frog heart signals from Dataset 1 using the DL algorithm of Section 2. (A) Results of the chaos analysis of experimental time series shown in the bifurcation diagram. (B) Time series length (that is, number of time points). In both panels, colors correspond to the results of the DL classification: Blue is used for correct regular detection, red for correct chaotic detection, and black for “incorrect” detection. (I–V) correspond to the “incorrect” detections and the time series are drawn in Fig. 10.

First of all, we use the proposed algorithm to analyze chaotic dynamics in Dataset 1. In the bifurcation diagram of Fig. 9(A), the DL chaos analysis of experimental data of heart dynamics is depicted. In blue we have the samples that have been correctly detected as regular by the algorithm (compared with the expert criteria), in red those correctly classified as chaotic, and in black the “incorrectly” detected ones (that is, false regular and false chaotic detections). As we can see, just 5 of the 52 samples have been incorrectly classified (9.615 % of the samples). That is, the DL algorithm for chaos analysis has been successful in 90.385 % of the dataset (47 of 52 samples). As our dataset is not balanced (we have more regular than chaotic samples), it is important to compute the accuracy chaotic and accuracy regular. In particular, such values are 92.308 % (12 of 13 samples) and 89.744 % (35 of 39 time series), respectively. Notice that all the accuracy values are around 90 % of success, and it seems that the DL algorithm for chaos analysis of heart

Table 3

Results of the DL chaos analysis of the experimental data (frog heart signals) of Dataset 1 with the network obtained in the algorithm.

DATASET 1	Experimental results
Accuracy (%)	90.385 % (47 of 52 samples)
Accuracy chaotic (%)	92.308 % (12 of 13 samples)
Accuracy regular (%)	89.744 % (35 of 39 samples)

time series has worked properly. In Table 3 we have summarized such accuracy results.

In Fig. 9(B) we have represented the length, number of data points, of each experimental time series (x -axis corresponds to the BCL value of the sample, and the height of the line is the length). Colors correspond to the network classification: blue for correct regular detection, red for correct chaotic detection, and black for “incorrect detection”. The green dashed horizontal lines indicate length values 50, 100, 150 and 200, and have been added to facilitate visualization. As already mentioned when describing the experimental dataset, the time series have different lengths (minimum length 15 and maximum length 205). In particular, 40.385 % of the samples (21 of 52) contain at most 50 time points, and 80.769 % (42 of 52 samples) have at most 100. Notice that most of the samples are quite short, and this can complicate the task as not enough information can be provided to the network. However, it seems that the applied DL algorithm for chaos analysis is able to detect properly most of the regular and chaotic behavior regardless of the length.

Let us analyze in detail the “incorrectly” detected samples (in black in Fig. 9). We have represented them in Fig. 10. In green we have the points that correspond to the time series, and we have connected such points with black segments for a better visualization of the behavior. Notice that not all the samples have the same length, that is, the scale of x -axis is different for each time series. As can be seen in Fig. 10, samples (I), (III) and (IV) have a similar behavior: Short chaotic transient dynamics, asymptotically converging to an equilibrium point (regular). According to Rule *b* that we have used for expert classification of the experimental time series, the samples have been labeled as regular. The ANN detected them as chaotic. The whole time series are chaotic (as DL detected), but dynamically their asymptotic behavior can be considered as regular (as we labeled them). Sample (II), as can be seen in Fig. 10, seems to have some periodicity at the end, with long chaotic transient. Therefore, applying Rule *a* of the expert classification, the time series has been labeled as chaotic. However, the algorithm detected it as regular. The fifth incorrect detection corresponds to sample (V). As seen in Fig. 10, it can be difficult to classify such an orbit. In fact, this sample seems to correspond to an orbit inside or just after a period-doubling bifurcation, so it can be difficult to classify it as a periodic orbit with a high period, a quasi-periodic orbit, or a Feigenbaum chaotic orbit. As it seems to follow some periodicity along all the time points, it has been labeled as regular. The network detected it as chaotic with a score (“probability”) of 0.625 for this class (that is, with a high uncertainty). Note that all these “incorrectly” classified time series have regular and chaotic patterns, and so any classification cannot be accurate as there are too few data points to correctly classify them. Therefore, the results of the ANN cannot really be considered as a false result (and for this reason we have used quotation marks in the word incorrect when we refer to them).

Remark 7. Notice that in the case of sample (V) (see Figs. 9 and 10), the network assigned a score 0.625 for chaotic class and a score of 0.375 for regular class. The values are closer to 0.5 than to 1, what means that the ANN is not sure about its decision. If some human supervision wants to be added to the algorithm defined in Section 2, we can add a control to Step 4. It would be defined as follows: If the higher score of one sample for both categories is lower than $2/3$, such data point should be revised by an expert as the detection score is close to the classification border

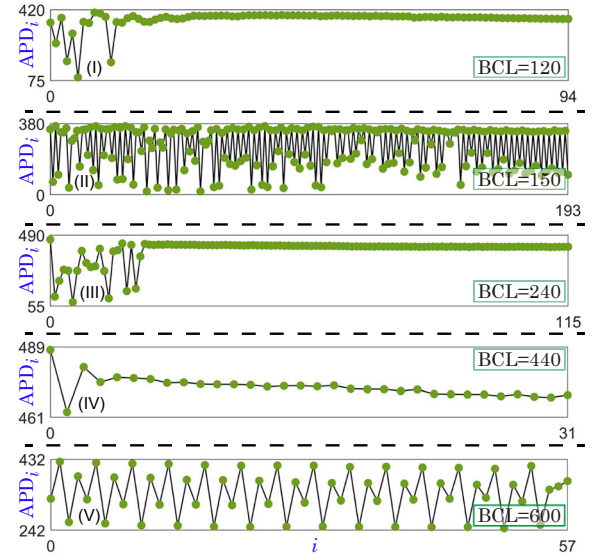


Fig. 10. Time series that have been “incorrectly” detected in Dataset 1. They correspond to samples (I)–(V) in black in Fig. 9. In green we have the points that correspond to the time series and we have joined such points with black segments for a better visualization of the behavior.

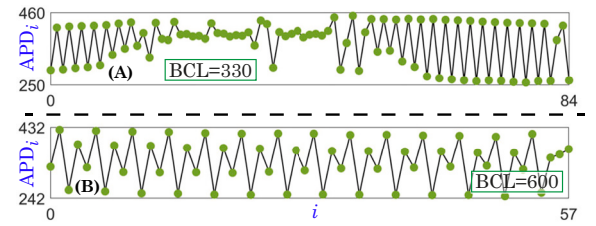


Fig. 11. Time series of Dataset 1 with detection score (“probability”) less than $2/3$ for the winning class. In green we have the dots that correspond to the time series and we have connected them with black segments for the ease of viewing.

(0.5 is the threshold to distinguish between regular and chaotic). Adding this control in our experimental dataset, just 3.486 % of the samples (2 of 52 samples) should be revised. These two samples have been represented in Fig. 11. As already indicated, one of these samples (sample (B)) corresponds to one of the samples “incorrectly” detected (see sample (V) of Fig. 10). The other one (sample (A)) was correctly classified by the algorithm. The network assigned a score 0.592 for the chaotic class and 0.408 for regular class. Notice that the time series seems to follow some periodicity at the beginning and the end of the recordings but in between it presents a chaotic behavior, so the doubt of the network could be totally justified. After this expert supervision, the final accuracy for the whole dataset would be 92.308 % (48 of the 52 samples), with accuracy regular increasing also to this value (36 of 39 samples).

Finally, we apply the proposed algorithm to Dataset 2. In the bifurcation diagram of Fig. 12(A), the DL analysis of chaotic dynamics of the experimental heart data is represented following the same code of colors of the study of Dataset 1 (blue for regular samples correctly detected comparing with expert criteria, red for correctly classified chaotic time series, and black for “incorrectly” classified signals). As summarized in Table 4, 91.667 % of the samples have been classified properly (22 of 24 time series), in fact, we obtain a value of 95 % for accuracy regular (19 of 20 samples) and 75 % for accuracy chaotic (3 of 4 samples). Notice that this accuracy chaotic value is not really high, but we have to take into account that there are only 4 chaotic samples in the dataset and just 1 has been misclassified (that is, small set of samples in which just

Table 4

Results of the DL chaos analysis of the experimental data (frog heart dynamics) of Dataset 2 with the network obtained in the algorithm.

DATASET 2	Experimental results
Accuracy (%)	91.667 % (22 of 24 samples)
Accuracy chaotic (%)	75.000 % (3 of 4 samples)
Accuracy regular (%)	95.000 % (19 of 20 samples)

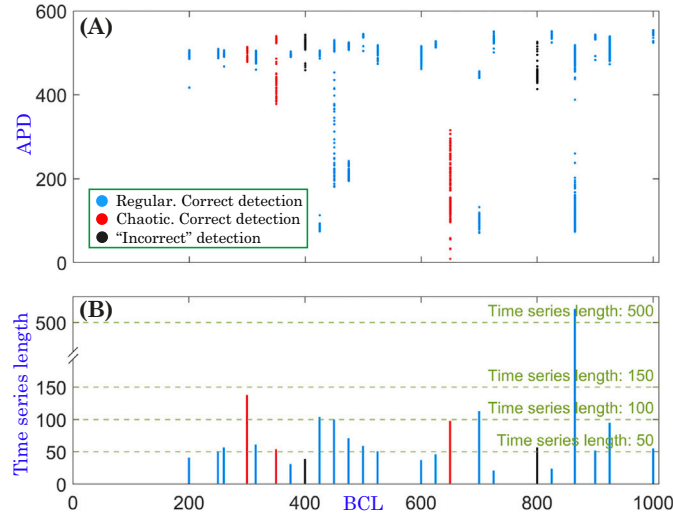


Fig. 12. Chaos analysis of experimental data of frog heart signals from Dataset 2 using the DL algorithm of Section 2. (A) Results of the chaos analysis of experimental time series shown in the bifurcation diagram. (B) Time series length (that is, number of time points). In both panels, colors correspond to the DL classification: Blue is used for correct regular detection, red for correct chaotic detection, and black for “incorrect” detection.

one error strongly degrades the results). Only two samples have been “incorrectly” classified in the whole dataset compared to the expert detection, both with a short number of points (length) as can be seen in Fig. 12(B). This panel illustrates that except for one long sample, the remaining signals generally have less than 100 points.

3.3. Brief comparison with other standard techniques

In this subsection we perform a brief study of the results provided by different standard techniques in the analysis of chaotic dynamics in the experimental frog heart signals of Dataset 1. This will allow us to check the performance of the algorithm proposed in the present work. The standard techniques that we apply are the Permutation Entropy (PE) [25,28] (Python package Antropy [74] has been used for the implementation) and the 0-1 test for chaos [29] (MATLAB implementation in [75] has been applied).

Permutation Entropy is considered a noisy but robust technique that measures the complexity of data even in the presence of dynamical and observational noise. The 0-1 test for chaos is a powerful and widely applied tool to distinguish between chaotic and regular behavior in dynamical systems. As shown in Figs. 5 and 6, both techniques provide accurate and meaningful results when applied to mathematical models, and they have been widely used in the literature [31,33]. Therefore, we consider them suitable tools to compare with our algorithm.

In Fig. 13 we have the analysis of chaotic dynamics provided by the expert classification in panel (A) (it is based on Rules a-d explained in Section 3.2 and it is considered as the ground truth), by our algorithm in panel (B) (this panel coincides with panel (A) of Fig. 9), by the Permutation Entropy (PE) technique in panel (C), and by the 0-1

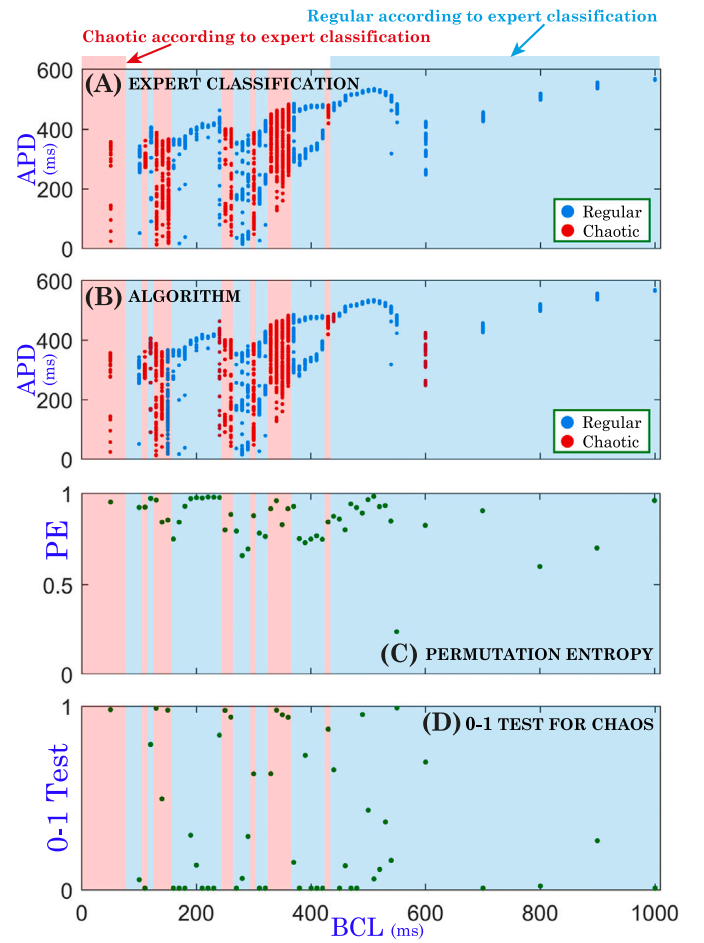


Fig. 13. Analysis of chaotic dynamics for the experimental data (frog heart signals) of Dataset 1 using different techniques. (A) Bifurcation diagram with the results provided by the expert classification. Blue corresponds to regular and red to chaotic dynamics. (B) Bifurcation diagram with the results provided by the DL algorithm proposed in this paper (it corresponds to panel (A) of Fig. 9). Blue corresponds to regular and red to chaotic dynamics. (C) Permutation Entropy (PE). (D) 0-1 test for chaos. Color shading follows the behavior obtained with the expert classification that we consider as ground truth to compare with the remaining techniques.

test for chaos in panel (D). Blue and red shading across all the panels correspond to the results provided by the expert classification that we consider as ground truth (blue corresponds to regular label and red to chaotic one). The Permutation Entropy technique depends on two parameters, the permutation order n and the delay τ , we have set them to 3 and 1, respectively, following the recommendations in [28]. For the permutation order, it is suggested to use the maximum n value such that the length of the time series is larger than $5n!$. As we have a large variability in the time series length across our experimental dataset, we consider $n = 3$ that satisfies the criterion for most of the samples. For the delay τ , we consider the recommendation indicated for one-dimensional dynamical systems, as working with single-variable time series and the corresponding mathematical model (APD heart map) would be one-dimensional.

If we focus on panel (C) of Fig. 13 corresponding to the (normalized) Permutation Entropy, at first sight we can see that values greater than 0.5 have been obtained for almost all the samples. This seems to provide not meaningful results (or at least results that are not easy to interpret). If we study the plot carefully, we can see that, for instance, for the regular samples in the fourth and fifth blue shadings (from left to right), the complexity is a bit lower than for the samples in the neighboring

red shading regions, showing that they are more regular and slightly aligned with the expert classification. However, this difference does not seem to be remarkable. On the other hand, we can see that most of the Permutation Entropy values for regular samples in the third and last blue shading regions are larger than those for the chaotic samples in the neighboring regions, providing results that would not allow us to classify the dynamical behavior easily. Therefore, we can infer that Permutation Entropy does not provide proper or easy to interpret results for our particular dataset.

If we take a look on panel (D) of Fig. 13 corresponding to the 0-1 test for chaos, we can see that for several samples (around 20 time series, that is, more than one third of the dataset) we obtain intermediate values, neither close to 0 nor to 1, not providing a dynamical classification for them. For the remaining ones, except for a couple of samples in the most-right blue shading, the classification is adequate. In this case, we can see that the 0-1 test for chaos cannot provide proper results in a wide part of the dataset because of the nature of the samples (short time series).

Comparing the performance of the standard techniques (Permutation Entropy in panel (C) of Fig. 13 and 0-1 test for chaos in panel (D)) and the performance of our algorithm (see panel (B) of Fig. 13 and Section 3.2 for more details), we can see that our proposed tool to analyze chaotic dynamics in experimental data is the more accurate (at least in our dataset). This could be mainly due to the nature of the data (very short noisy time series). That is, as it is also observed in literature, the classical techniques usually need sufficiently long time series, but this is not always possible like occurs in heart data. Therefore, new techniques like the one introduced here can help in the analysis of this kind of experimental data.

4. Conclusions

There is an increasing amount of experimental data in practice, so automatic techniques that would allow us to classify them properly are necessary. Real-world data often has some drawbacks, such as noisy and short recordings, or limited amount of samples. In addition, not all the variables of the real-world system can be available for recording. In this paper, we propose a Deep Learning-based algorithm to tackle the chaos analysis of biological time series. The algorithm is divided into different steps. We consider an Artificial Neural Network architecture based on Long Short-Term Memory cells with a final classification layer. The training data is taken from a basic and generic discrete dynamical system: the Logistic Map. Once 10 Artificial Neural Networks with that same architecture but different initialization of the trainable parameters have been trained with such generic data, a mathematical model with the same nature as the experimental data is used to perform a test (in our case of study, the Beeler-Reuter APD heart map). From this analysis, applying some selection criteria, one of the trained Artificial Neural Networks is selected. Such network seems to be adequate to perform the analysis of chaotic dynamics of biological time series (in our case, of frog heart signals).

The algorithm is tested on a set of experimental data obtained from frog heart signals in the laboratory of Professor Flavio H. Fenton. The accurate results of the paper highlights that the combination of Deep Learning techniques and mathematical models can be useful to face the analysis of chaotic dynamics in experimental data even when it consists of short time series (where other techniques, such as the computation of the maximum Lyapunov exponent would not work properly). In fact, the algorithm shows that Artificial Neural Networks trained in generic data can be used for particular problems once the most appropriate one has been selected using data from a problem similar or of the same nature as the particular problem we want to address. Moreover, this is a *de facto* proof that most dynamical phenomena are quite general and universal. However, the proposed algorithm has some limitations. One of them arises from the use of Deep Learning. Currently, this technique lacks general interpretability, which introduces a disadvantage into our algorithm that is not typically present in classical algorithms. On the

other hand, one of the crucial points of the algorithm is the use of a mathematical model that shares the nature of the experimental data. Although rare, it is possible that no suitable model is available for the type of experimental data to be analyzed. As future work, the authors plan to investigate how emerging branches of Deep Learning that incorporate physical information, such as Physics-Informed Neural Networks (PINNs) which are expected to provide more interpretability, could be applied to the analysis of experimental data.

CRedit authorship contribution statement

Carmen Mayora-Cebollero: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Flavio H. Fenton:** Writing – review & editing, Supervision, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Molly Halprin:** Investigation. **Conner Herndon:** Investigation. **Mikael J. Toye:** Investigation, Data curation. **Roberto Barrio:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

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.

Acknowledgments

RB and CMC have been supported by the Spanish research projects PID2021-122961NB-I00 and PID2024-156032NB-I00, and the European Regional Development Fund and Diputación General de Aragón E24-23R. RB has been supported by the European Regional Development Fund and Diputación General de Aragón LMP94-21. CMC has been supported by Spanish research project PID2022-140556OB-I00 and by Ministerio de Universidades de España with an FPU grant (FPU20/04039). FHF, MH, CH and MJT have been supported by NIH, United States grant 2R01HL143450.

Data availability

Data will be made available on request.

References

- [1] H.M. Hastings, F.H. Fenton, S.J. Evans, O. Hotomarglu, J. Geetha, K. Gittelson, J. Nilson, A. Garfinkel, Alternans and the onset of ventricular fibrillation, *Phys. Rev. E* 62 (3) (2000) 4043.
- [2] Y. Xie, G. Hu, D. Sato, J.N. Weiss, A. Garfinkel, Z. Qu, Dispersion of refractoriness and induction of reentry due to chaos synchronization in a model of cardiac tissue, *Phys. Rev. Lett.* 99 (11) (2007) 118101.
- [3] L.M. Muñoz, M.O. Ampofo, E.M. Cherry, Controllability of voltage-and calcium-driven cardiac alternans in a map model, *Chaos, An Interdiscip. J. Nonlinear Sci.* 31 (2) (2021).
- [4] R. Stenzinger, T. Scalvin, P. Morelo, M. Tragtenberg, Cardiac behaviors and chaotic arrhythmias in the Hindmarsh-Rose model, *Chaos, Solitons & Fractals* 175 (2023) 113983.
- [5] X. Wang, J. Landaw, Z. Qu, Intracellular ion accumulation in the genesis of complex action potential dynamics under cardiac diseases, *Phys. Rev. E* 109 (2) (2024) 024410.
- [6] A. Garfinkel, M. Spano, W. Ditto, J. Weiss, Controlling cardiac chaos, *Science (N.Y.)* 257 (1992) 1230–1235.
- [7] J.E. Skinner, A.L. Goldberger, G. Mayer-Kress, R.E. Ideker, Chaos in the heart: implications for clinical cardiology, *Bio/Technology* 8 (11) (1990) 1018–1024.
- [8] N. Thakor, Chaos in the heart: signals and models, in: *Proceedings of the 1998 2nd International Conference Biomedical Engineering Days*, 1998, pp. 11–18.
- [9] A.L. Goldberger, Is the normal heartbeat chaotic or homeostatic?, *Physiology* 6 (2) (1991) 87–91.
- [10] C.-S. Poon, C.K. Merrill, Decrease of cardiac chaos in congestive heart failure, *Nature* 389 (6650) (1997) 492–495.
- [11] A.L. Goldberger, V. Bhargava, B.J. West, A.J. Mandell, Some observations on the question: is ventricular fibrillation “chaos”?, *Phys. D* 19 (2) (1986) 282–289.
- [12] J.N. Weiss, A. Garfinkel, H.S. Karagueuzian, Z. Qu, P.-S. Chen, Chaos and the transition to ventricular fibrillation: a new approach to antiarrhythmic drug evaluation, *Circulation* 99 (21) (1999) 2819–2826.

- [13] V. Gupta, M. Mittal, V. Mittal, Chaos theory and ARTFA: emerging tools for interpreting ECG signals to diagnose cardiac arrhythmias, *Wirel. Pers. Commun.* 118 (4) (2021) 3615–3646.
- [14] D.R. Chialvo, J. Jalife, Non-linear dynamics of cardiac excitation and impulse propagation, *Nature* 330 (6150) (1987) 749–752.
- [15] D.R. Chialvo, R.F. Gilmour Jr, J. Jalife, Low dimensional chaos in cardiac tissue, *Nature* 343 (6259) (1990) 653–657.
- [16] A. Gizzi, E.M. Cherry, R.F. Gilmour Jr, S. Luther, S. Filippi, F.H. Fenton, Effects of pacing site and stimulation history on alternans dynamics and the development of complex spatiotemporal patterns in cardiac tissue, *Front. Physiol.* 4 (2013) 71.
- [17] S. Irvanian, I. Uzelac, A.D. Shah, M.J. Toye, M.S. Lloyd, M.A. Burke, M.A. Daneshmand, T.S. Attia, J.D. Vega, M.F. El-Chami, F.M. Merchant, E.M. Cherry, N.K. Bhatia, F.H. Fenton, Complex repolarization dynamics in ex vivo human ventricles are independent of the restitution properties, *Europace* 25 (12) (2023) eua350.
- [18] T. Lilienkamp, U. Parlitz, S. Luther, Taming cardiac arrhythmias: terminating spiral wave chaos by adaptive deceleration pacing, *Chaos, An Interdiscip. J. Nonlinear Sci.* 32 (12) (2022).
- [19] N. DeTal, A. Kaboudian, F.H. Fenton, Terminating spiral waves with a single designed stimulus: Teleportation as the mechanism for defibrillation, in: *Proceedings of the National Academy of Sciences*, vol. 119, 2022 (24) e2117568119.
- [20] S.A. Tyler, D. Mersing, F.H. Fenton, M.R. Tinsley, K. Showalter, Experimental studies of spiral wave teleportation in a light sensitive Belousov–Zhabotinsky system, *Chaos, An Interdiscip. J. Nonlinear Sci.* 34 (9) (2024).
- [21] Y.C. Ji, I. Uzelac, N. Otani, S. Luther, R.F. Gilmour Jr, E.M. Cherry, F.H. Fenton, Synchronization as a mechanism for low-energy anti-fibrillation pacing, *Heart Rhythm* 14 (8) (2017) 1254–1262.
- [22] J.H. Argyris, G. Faust, M. Haase, An Exploration of Chaos: an Introduction for Natural Scientists and Engineers, North Holland, 1994.
- [23] A. Wolf, J.B. Swift, H.L. Swinney, J.A. Vastano, Determining Lyapunov exponents from a time series, *Phys. D* 16 (3) (1985) 285–317.
- [24] M.T. Rosenstein, J.J. Collins, C.J. De Luca, A practical method for calculating largest Lyapunov exponents from small data sets, *Phys. D* 65 (1–2) (1993) 117–134.
- [25] C. Bandt, B. Pompe, Permutation entropy: a natural complexity measure for time series, *Phys. Rev. Lett.* 88 (2002) 174102.
- [26] J. Amigó, *Permutation Complexity in Dynamical Systems*, Springer, Berlin Heidelberg, 2010.
- [27] D. Stosic, D. Stosic, T. Stosic, B. Stosic, Generalized weighted permutation entropy, *Chaos, An Interdiscip. J. Nonlinear Sci.* 32 (10) (2022).
- [28] M. Riedl, A. Müller, N. Wessel, Practical considerations of permutation entropy, *Eur. Phys. J. Spec. Top.* 222 (2013) 249–262.
- [29] G.A. Gottwald, I. Melbourne, *The 0-1 Test for Chaos: A Review*, Springer, Berlin Heidelberg, Berlin, Heidelberg, 2016, pp. 221–247.
- [30] R. Barrio, S. Serrano, A three-parametric study of the Lorenz model, *Phys. D* 229 (1) (2007) 43–51.
- [31] J.A.E. Fouda, B. Bodo, G.M. Djeufa, S.L. Sabat, Experimental chaos detection in the Duffing oscillator, *Commun. Nonlinear Sci. Numer. Simul.* 33 (2016) 259–269.
- [32] R. Barrio, M. Martínez, E. Pueyo, S. Serrano, Dynamical analysis of early afterdepolarization patterns in a biophysically detailed cardiac model, *Chaos, An Interdiscip. J. Nonlinear Sci.* 31 (7) (2021).
- [33] R. Halfar, Dynamical properties of Beeler-Reuter cardiac cell model with respect to stimulation parameters, *Int. J. Comput. Math.* 97 (1–2) (2020) 498–507.
- [34] S.A. Shintani, Observation of sarcomere chaos induced by changes in calcium concentration in cardiomyocytes, *Biophys. Physicobiol.* 21 (1) (2024) e210006.
- [35] M.A. Islam, I.R. Hassan, P. Ahmed, Dynamic complexity of fifth-dimensional Henon map with Lyapunov exponent, permutation entropy, bifurcation patterns and chaos, *J. Comput. Appl. Math.* 466 (2025) 116547.
- [36] X. Ma, G. Litak, S. Zhou, Using 0–1 test to diagnose periodic and chaotic motions of nonlinear vortex-induced vibration energy harvesters, *Chaos, Solitons & Fractals* 192 (2025) 116036.
- [37] D. Tokar, F. Sommer, M.A. D’Esposito, A simple method for detecting chaos in nature, *Commun. Biol.* 3 (article number 11) (2020).
- [38] M. Paluš, Chaotic measures and real-world systems: does the Lyapunov exponent always measure chaos?, in: H. Kantz, J. Kurths, G. Mayer-Kress (Eds.), *Nonlinear Analysis of Physiological Data*, Springer, Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 49–66.
- [39] M. Zanin, F. Olivares, Ordinal patterns-based methodologies for distinguishing chaos from noise in discrete time series, *Commun. Phys.* 4 (1) (2021) 190.
- [40] A. Kumar, Light propagation through biological tissue: comparison between Monte Carlo simulation and deterministic models, *Int. J. Biomed. Eng. Technol.* 2 (4) (2009) 344–351.
- [41] A. Kumar, A.P.S. Chauhan, Robust feature extraction from omnidirectional outdoor images for computer vision applications, *Int. J. Instrum. Meas.* 10 (2025) 8–13.
- [42] A. Celletti, C. Gales, V. Rodríguez-Fernández, M. Vasilé, Classification of regular and chaotic motions in Hamiltonian systems with deep learning, *Sci. Rep.* 12 (1) (2022) 1–12.
- [43] R. Barrio, Á. Lozano, A. Mayora-Cebollero, C. Mayora-Cebollero, A. Miguel, A. Ortega, S. Serrano, R. Vigar, Deep Learning for chaos detection, *Chaos, An Interdiscip. J. Nonlinear Sci.* 33 (7) (2023) 073146.
- [44] J. Pathak, Z. Lu, B.R. Hunt, M. Girvan, E. Ott, Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data, *Chaos, An Interdiscip. J. Nonlinear Sci.* 27 (12) (2017).
- [45] C. Mayora-Cebollero, A. Mayora-Cebollero, R. Barrio, Á. Lozano, Full Lyapunov exponents spectrum with Deep Learning from single-variable time series, *Phys. D* 472 (2025) 134510.
- [46] S. Shahi, F.H. Fenton, E.M. Cherry, A machine-learning approach for long-term prediction of experimental cardiac action potential time series using an autoencoder and echo state networks, *Chaos, An Interdiscip. J. Nonlinear Sci.* 32 (6) (2022).
- [47] S. Shahi, F.H. Fenton, E.M. Cherry, Prediction of chaotic time series using recurrent neural networks and reservoir computing techniques: a comparative study, *Mach. Learn. Appl.* 8 (2022) 100300.
- [48] J. Nolasco, R.W. Dahlen, A graphic method for the study of alternation in cardiac action potentials, *J. Appl. Physiol.* 25 (2) (1968) 191–196.
- [49] M. Guevara, G. Ward, A. Shrier, L. Glass, Electrical alternans and period doubling bifurcations, *Comput. Cardiol.* 562 (1984) 167–170.
- [50] M.R. Franz, C.D. Szwedlow, L.B. Liem, J. Schaefer, Cycle length dependence of human action potential duration in vivo. Effects of single extrastimuli, sudden sustained rate acceleration and deceleration, and different steady-state frequencies, *J. Clin. Invest.* 82 (3) (1988) 972–979.
- [51] P. Szilágyi, T. Bányász, J. Magyar, G. Szigei, Z. Papp, A. Varró, P. Nánási, Intracellular calcium and electrical restitution in mammalian cardiac cells, *Acta Physiol. Scand.* 163 (2) (1998) 139–147.
- [52] M. Watanabe, N.F. Otani, R.F. Gilmour Jr, Biphasic restitution of action potential duration and complex dynamics in ventricular myocardium, *Circ. Res.* 76 (5) (1995) 915–921.
- [53] Z. Qu, J.N. Weiss, A. Garfinkel, Spatiotemporal chaos in a simulated ring of cardiac cells, *Phys. Rev. Lett.* 78 (7) (1997) 1387.
- [54] F.H. Fenton, E.M. Cherry, H.M. Hastings, S.J. Evans, Multiple mechanisms of spiral wave breakup in a model of cardiac electrical activity, *Chaos, An Interdiscip. J. Nonlinear Sci.* 12 (3) (2002) 852–892.
- [55] F.H. Fenton, S.J. Evans, H.M. Hastings, Memory in an excitable medium: a mechanism for spiral wave breakup in the low-excitability limit, *Phys. Rev. Lett.* 83 (19) (1999) 3964.
- [56] E.G. Tolkacheva, D.G. Schaeffer, D.J. Gauthier, W. Krassowska, Condition for alternans and stability of the 1:1 response pattern in a “memory” model of paced cardiac dynamics, *Phys. Rev. E* 67 (3) (2003) 031904.
- [57] J.J. Fox, E. Bodenschatz, R.F. Gilmour Jr, Period-doubling instability and memory in cardiac tissue, *Phys. Rev. Lett.* 89 (13) (2002) 138101.
- [58] E.M. Cherry, F.H. Fenton, Suppression of alternans and conduction blocks despite steep APD restitution: electrotonic, memory, and conduction velocity restitution effects, *Am. J. Physiol.-Heart Circ. Physiol.* 286 (6) (2004) H2332–H2341.
- [59] M.E. Díaz, S.C. O’neill, D.A. Eisner, Sarcoplasmic reticulum calcium content fluctuation is the key to cardiac alternans, *Circ. Res.* 94 (5) (2004) 650–656.
- [60] J.G. Restrepo, A. Karma, Spatiotemporal intracellular calcium dynamics during cardiac alternans, *Chaos, An Interdiscip. J. Nonlinear Sci.* 19 (3) (2009).
- [61] J. Pathak, B. Hunt, M. Girvan, Z. Lu, E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach, *Phys. Rev. Lett.* 120 (2) (2018) 024102.
- [62] B. Ramadevi, K. Bingi, Chaotic time series forecasting approaches using machine learning techniques: a review, *Symmetry* 14 (5) (2022) 955.
- [63] R.M. May, Simple mathematical models with very complicated dynamics, *Nature* 261 (5560) (1976) 459–467.
- [64] G.W. Beeler, H. Reuter, Reconstruction of the action potential of ventricular myocardial fibres, *J. Physiol.* 268 (1) (1977) 177–210.
- [65] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: an imperative style, high-performance Deep Learning library, *Adv. Neural Inf. Process. Syst.* 32 (2019).
- [66] R. Barrio, Á. Lozano, A. Mayora-Cebollero, C. Mayora-Cebollero, A. Miguel, A. Ortega, S. Serrano, R. Vigar, Dataset of Article: “Deep Learning for Chaos Detection”, 2023, <https://doi.org/10.17632/k4x675k5dm.3>. <https://data.mendeley.com/datasets/k4x675k5dm/3>.
- [67] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780.
- [68] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.
- [69] F. Liu, X. Zhou, J. Cao, Z. Wang, H. Wang, Y. Zhang, Arrhythmias classification by integrating stacked bidirectional LSTM and two-dimensional CNN, in: *Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD 2019 April 14–17, 2019, Proceedings, Part II*, vol. 23, Springer, Macau, China, 2019, pp. 136–149.
- [70] R. Engelken, Gradient flossing: improving gradient descent through dynamic control of Jacobians, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, S. Levine (Eds.), *Advances in Neural Information Processing Systems*, vol. 36, Curran Associates, Inc, 2023, pp. 10412–10439.
- [71] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, *arXiv preprint arXiv:1406.1078*, 2014.
- [72] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, *arXiv preprint arXiv:1412.6980*, 2014.
- [73] M.J. Feigenbaum, Quantitative universality for a class of nonlinear transformations, *J. Stat. Phys.* 19 (1) (1978) 25–52.
- [74] R. Vallat, Antropy: entropy and complexity of time-series in Python (v0.1.9), 2025. <https://github.com/raphaelvallat/antropy>.
- [75] P. Matthews, 0-1 Test for Chaos, MATLAB Central File Exchange, 2025, <https://www.mathworks.com/matlabcentral/fileexchange/25050-0-1-test-for-chaos>.
- [76] T.J. Lewis, M.R. Guevara, Chaotic dynamics in an ionic model of the propagated cardiac action potential, *J. Theor. Biol.* 146 (3) (1990) 407–432.

- [77] E.M. Cherry, F.H. Fenton, Visualization of spiral and scroll waves in simulated and experimental cardiac tissue, *New J. Phys.* 10 (12) (2008) 125016.
- [78] E.M. Cherry, F.H. Fenton, A tale of two dogs: analyzing two models of canine ventricular electrophysiology, *Am. J. Physiol.-Heart Circ. Physiol.* 292 (1) (2007) H43–H55.

Author biography

Carmen Mayora-Cebollero received her B.S. and Ph.D. degrees in Applied Mathematics from Universidad de Zaragoza, Spain, in 2020 and 2025, respectively. Her research interests include dynamical systems, machine learning, optimization techniques, cardiac dynamics, and mathematical neuroscience.

Flavio H. Fenton received the B.S. degree in Theoretical Physics from the Universidad Nacional Autonoma de Mexico, Mexico, in 1990, and the M.S. and Ph.D. degrees in Physics from the Northeastern University, USA, in 1993 and 1999, respectively. Currently, he is a professor in the School of Physics at the Georgia Institute of Technology, USA. His research interests include excitable media, complex systems, and pattern formation, using a combined approach of theory, experiments, and computer simulations.

Molly Halprin majored in Biomedical Engineering at the Georgia Institute of Technology, USA, in 2020. Her research interests include excitable media and experiments.

Conner Herndon received his B.S., M.S., and Ph.D. degrees in Physics at the Georgia Institute of Technology, USA, in 2015, 2016 and 2021, respectively. Currently, he is working as Manager of Clinical Intelligence at Intuitive and is the co-founder of My Data Unlocked in Atlanta, USA. His research interests include experiments and excitable media.

Mikael J. Toye received his B.S. degree in Physics at the Clemson University, USA, in 2018. He is currently a Ph.D. student in Physics at the Georgia Institute of Technology, USA. His research interests include excitable media and complex systems.

Roberto Barrio received his B.S. and Ph.D. degrees in Applied Mathematics from Universidad de Zaragoza, Spain, in 1992 and 1997, respectively, where he is currently a Full Professor at the Department of Applied Mathematics. His current research interests include optimization techniques, machine learning, numerical analysis, mathematical neuroscience, cardiac dynamics, dynamical systems and computational dynamics.