Model-free prediction of multistability using echo state network

Mousumi Roy,1 Swarnendu Mandal,1 Chittaranjan Hens,2 Awadhesh Prasad,3 N.V. Kuznetsov,4,5,6 and Manish Dev Shrimali1

1)Department of Physics, Central University of Rajasthan, Ajmer, Rajasthan, India - 305817
2)Center for Computational Natural Sciences and Bioinformatics, International Institute of Information Technology, Gachibowli, Hyderabad, India - 500 032
3)Department of Physics & Astrophysics, University of Delhi, Delhi, India - 110 007
4)Department of Applied cybernetics, Saint Petersburg University, St. Petersburg, Russia
5)Department of Mathematical Information Technology, University of Jyvaskyla, Jyvaskyla, Finland
6)Institute for Problems in Mechanical Engineering of the Russian Academy of Sciences, St. Petersburg, Russia

(Dated: 2 September 2022)

In the field of complex dynamics, multistable attractors have been gaining a significant attention due to its unpredictability in occurrence and extreme sensitivity to initial conditions. Co-existing attractors are abundant in diverse systems ranging from climate to finance, ecological to social systems. In this article, we investigate a data-driven approach to infer different dynamics of a multistable system using echo state network (ESN). We start with a parameter-aware reservoir and predict diverse dynamics for different parameter values. Interestingly, machine is able to reproduce the dynamics almost perfectly even at distant parameters which lie considerably far from the parameter values related to the training dynamics. In continuation, we can predict whole bifurcation diagram significant accuracy as well. We extend this study for exploring various dynamics of multistable attractors at unknown parameter value. While, we train the machine with the dynamics of only one attractor at parameter p, it can capture the dynamics of co-existing attractor at a new parameter value p + Δp. Continuing the simulation for multiple set of initial conditions, we can identify the basins for different attractors. We generalize the results by applying the scheme on two distinct multistable systems.

The occurrence of multistability is evident to exist in many daily life complex dynamical systems like climate, financial market, ecological systems and many more. Often the unpredictable behavior of such systems lead to undesired scenario. Hence, it is important to analyse the dynamical behaviors of these systems. But often we do not have the exact mathematical model to mimic the behavior of many such systems. In this study, we use a data-driven machine learning approach to infer the dynamics of multistable systems. We discuss that, a few time series data with its corresponding system parameters are enough to capture essence of complex dynamics for a broad parameter range. Moreover, training the system for a given parameter, it is possible to identify the presence of unknown co-existing attractors for a different parameter even with their basin of attraction.

I. INTRODUCTION

In the field of non-linear dynamics, multistable attractors possess different stable dynamics at a fixed parameter set solely depending on the initial conditions. Its dynamical uncertainty and extreme dependency on the initial conditions make it difficult to acquire knowledge of the overall system dynamics. Additionally, no prior knowledge of the basins corresponding to different stable dynamics makes it harder while working with such systems. Basin can be smooth or riddled-like. In reality, multistable attractors are present in diverse systems, such as, ecology, financial, laser, etc. Because of its presence in various fields, the nature of its dynamics, as well as the basins of such multistable attractors, are significantly important to be identified. Most importantly, as in most of these cases, the exact mathematical form of the system is unknown. Since last few years, a large number of theoretical and numerical studies have been making a great effort to shed light on this dimension. However, most of the studies are focused on the route of controlling multistability within the system. Despite these enormous attempts by researchers, any straightforward method for model-free prediction of different dynamics and basins is still rare.

Motivated from this backdrop, in this study we address this issue using a model-independent machine learning technique. Here, we use a simple yet extremely powerful tool Echo State Network (ESN) which is popular for its training cost efficiency and architectural simplicity. Based on the basic mechanism, various form of ESN is available targeting to make a balance between efficiency and simplicity of the machine architecture. Because of its uncomplicated application procedure, ESN appears as one of the effective tools in diverse areas, ranging from neuroscience to robotic, speech recognition to language processing, network connectivity inference to epidemic spreading, etc. In this context, ESN also takes a significant role in predicting diverse dynamical characteristics, such as, chaotic time series prediction, Lyapunov exponent, cluster synchronization, burst synchronization, spatio-temporal dynamics, etc. Another significant direction is to explore the way of optimizing training procedures and choosing hyperparameters of ESN. Despite these, researchers use several physical systems as a reservoir as well, developing an intriguing field of physical reservoir.
Some recent studies have shown the identification of sudden dynamical changes using ESN depending on system parameters. For example, in a coupled system, ESN can predict the critical coupling strengths at which the first order transition occurs. Furthermore, Xiao et al. have shown the prediction of amplitude death due to a parameter drift. Recently, another interesting study provided a different scheme for training, where they make scaling of the system parameter and get diverse dynamics by switching the parameter at different values, which leads to identifying period doubling route towards chaos. A recent study has also explored different attractors of multistable attractors using a machine learning architecture. However, the prediction accuracy can be improved.

Based on the previous investigations and to fill the gap in the field of multistability, we exploit a data-driven machine learning approach to locate the basins and diverse dynamics of the systems exhibiting more than one stable attractor at a particular parameter set. We proceed for this study in two directions using ESN: The first one is the identification of various dynamics using only a few system parameters and corresponding time series for training. To be specific, we train the machine with three time series data along with their respective parameter values to build a parameter-aware reservoir and can predict the dynamics for a range of parameter with great accuracy. In contrast to an earlier attempt, here we show the machine’s ability to infer the dynamics at distant parameters from the training data even when the training dynamics are less complex, for example, mere periodic. The second dimension is to further extend this study with an attempt to identify different dynamics and the corresponding basins of multistable attractors. Unlike a previous attempt in our study, we use a different architecture that significantly improves the level of accuracy in prediction. Furthermore, the natural unpredictability and initial condition dependency of multistable systems in real life make it quite difficult to deal with those systems without having any prior knowledge. Inspired by this backdrop, in this study we use a model-free data-driven technique to classify different attractors as well as their basins of a multistable system. We train the ESN using the dynamics of only one attractor and predict the others’ dynamics, and basins for every attractor as well, for an unknown system parameter value.

We arrange the article as follows: in Sec. II we elaborate discuss the basic architecture of ESN that we use for this investigation. Sec. III is devoted to the results. This section starts with a description of the original dynamics of a system that we use for the study. Next, we arrange all the results such as the prediction of different dynamics corresponding to different parameter values, and obtain the bifurcation diagram. Furthermore, we identify the basin of multistable attractors depending on a parameter drift. Furthermore, we verify all the results with another model exhibiting bistability between a stable oscillator and a stable fixed point. Finally, we conclude this study with sec. IV.

II. ESN ARCHITECTURE

The reservoir dynamics is described with the following map as follows:

\[ r(t+1) = (1-a)r(t) + a \tanh[\mathcal{W}_{\text{res}}r(t) + \mathcal{W}_{\text{in}}u(t)] \]

where, \( r(t) \) represents the state of the reservoir at time \( t \). \( N_{\text{res}} \)

is the dimension of the reservoir. \( \mathcal{W}_{\text{res}} \) is a random sparse matrix carrying the connection weights among \( N_{\text{res}} \) number of nodes, having spectral radius \( \rho \). The leaking parameter \( a \) can take values from 0 to 1. It controls the trade-off between the information of reservoir dynamics and non-linear functional value of the input data. We consider an \( m \)-dimensional time series data for training, therefore, \( u(t) \) is the \( m+1 \)-dimensional input data along with the corresponding system parameter value. To incorporate the information of the parameter through training data we take \( u(t) \) as the dynamics along with the corresponding parameter value, where, the first \( m \) elements are the dynamics of the input signal \( \hat{u}(t) \) at time \( t \) and the last element is the corresponding parameter, i.e.

\[ u(t) = [\hat{u}(t) \; \rho]^T \]

\( \mathcal{W}_{\text{in}} \) is the matrix that makes connections within the input data and the reservoir. We randomly choose the elements of \( \mathcal{W}_{\text{in}} \) from the uniformly distributed interval \([-\sigma, \sigma]\). We form \( \mathcal{W}_{\text{in}} \) in such a way that the information of the input data would be equally distributed to the nodes of the reservoir, that means the information of each dimension of input goes to \( \frac{1}{N_{\text{res}}} \) number of reservoir nodes. However, the information of the parameter is connected to all the nodes of reservoir. That makes the reservoir aware about the relation between the dynamics and the respective parameter.

We train the machine for \( N_p \) different parameter values one by one and successively store the values of

\[ r(t) = [r_1(t), r_2(t), r_3(t), \ldots, r_{N_{\text{res}}}(t)] \]

following the conventional way.

\[ \tilde{r}(t) = \begin{cases} r_n(t) & \text{if } n \text{ is odd} \\ r_n^2(t) & \text{if } n \text{ is even} \end{cases} \]

We store the values of \( \tilde{r}(t) \) within the matrix \( \mathcal{R} \) after ignoring \( t_{\text{trans}} \) time points as the transient dynamics of reservoir. Therefore, \( \mathcal{R} \) is a matrix with dimension \( N_{\text{res}} \times N_p(t_f - t_{\text{trans}}) \) for a training data of length \( t_f \). After calculating \( \mathcal{R} \), we can get the readout weight matrix from the system of linear equation

\[ \mathcal{W} = \mathcal{W}_{\text{out}}\mathcal{R}, \]

where, \( \mathcal{W} \) is the matrix having targeted data sets with dimension \( m \times N_p(t_f - t_{\text{trans}}) \). For this case, the target output is the state of the system at next time step. Now, in order to minimize the error between \( \mathcal{W} \) and \( \mathcal{R} \), we apply Ridge regression and obtain the read-out matrix \( \mathcal{W}_{\text{out}} \) in the following way,

\[ \mathcal{W}_{\text{out}} = \mathcal{W} \mathcal{R}^T (\mathcal{R} \mathcal{R}^T + \beta I)^{-1} \]
where \( b \) is the regularization parameter used to avoid the issue of over-fitting. After completing a successful training, the machine is now ready to make a prediction depending only on the parameter values.

In the prediction process, we replace only the parameter value with the new one and ESN produces the corresponding time series in an auto-iterative way only on the basis of \( W_{\text{out}} \) and the new parameter value using the following equations:

\[
r(t + 1) = (1 - a)r(t) + a \tanh [W_{\text{res}}.r(t) + W_{\text{in}}.v(t)],
\]

\[
v(t) = [\tilde{v}(t) \ p_{\text{new}}]^T
\]

\[
\tilde{v}(t) = W_{\text{out}} \tilde{r}(t),
\]

where

\[
\tilde{r}(t) = \begin{cases} 
  r_n(t) & \text{if } n \text{ is odd} \\
  r_o(t) & \text{if } n \text{ is even}
\end{cases}
\]

Repeating Eqs. (5)-(8), we obtain the machine generated predicted time series at the new parameter values. We start the prediction process from the initial condition where the system possesses a bounded solution. One trivial choice is to start from one of the points lying on the trajectory of the training data. Before starting the generation of predicted time series, the reservoir needs to warm-up for some iterations at new parameter space to get rid of the effect of reservoir state initialization. To do this, we repeat Eqs. (5)-(7) for \( w_a \) time points and let the machine produce the predicted time series \( \tilde{v}(t) \) following the above discussed method.

This scheme involves certain hyperparameters, namely spectral radius \( (\rho) \), leaking parameter \( (a) \), range of the non-zeros values of \( W_{\text{in}} \) matrix \( (\sigma) \) and regularization parameter \( (b) \). We optimize these parameters before running predictions for different tasks following a simultaneous optimization technique\(^{3,12}\). We use the root-mean-square error (RMSE) of predicted time series of length \( N_p \) with respect to its actual target, averaged over sufficient number of reservoir realizations, as the loss function to be minimized for the optimization process.

After obtaining a set of optimal hyperparameters, we build the final machine using those hyperparameters and now machine is ready to predict different dynamics depending on the parameter values and initial conditions.

III. RESULTS

To illustrate the previously mentioned reservoir-computing scheme, in this section we consider two models having multistable attractors. For both the models, we train the machine with the dynamics of only one attractor and try to predict the dynamics depending on diverse parameter values and initial conditions. We always choose oscillatory dynamics over the steady-state from the bi-stable system for training to make the reservoir well-informed about the dynamical information of the original system.

**A. Multistable system**

We consider a non-linear system\(^{5}\) having multistability as follows:

\[
\begin{align*}
\dot{x} &= y \\
\dot{y} &= x \\
\dot{z} &= -y + 3y^2 - x^2 - xz + \alpha + \epsilon u \\
\dot{u} &= -ku - \epsilon(z - \beta)
\end{align*}
\]

Depending on the parameters \( \alpha, \beta \) and \( \epsilon \), system (9) exhibits several dynamics. For \( \alpha = -0.02 \), overall dynamics of the system is shown on a \((\beta - \epsilon)\) parameter space (see Fig.1(a)). Here, depending on the values of \( \epsilon \) and \( \beta \), the system does not possess any fixed point in region I, highlighted with grey color, and in the other regions (II, III, IV) it generates a fixed point as follows,

\[
x^* = \pm \sqrt{\alpha + \frac{\epsilon^2 \beta}{k}}, \ y^* = 0, z^* = 0, u^* = \frac{\epsilon \beta}{k}
\]

Clearly, on satisfying the condition \((\alpha + \frac{\epsilon^2 \beta}{k}) > 0 \ i.e. \alpha > -\frac{\epsilon^2 \beta}{k}\), the system has a real fixed point. Region II (marked by yellow), III (marked by green) and IV (marked by orange) are classified based on the stability of the fixed point. In the region III, the fixed point is unstable producing an oscillatory dynamics and region IV depicts the parameter space of stable steady state. Furthermore, in region II, system has multistability with unbounded state, stable steady state and stable oscillatory dynamics. In this continuation, in Fig.1(b), we plot the oscillatory dynamics for region (III) and and in Fig.1(c), the steady state dynamics is shown as in region (IV).

**B. Prediction of dynamics at various parameters**

For this study, we fix the parameter at \( \beta = 0.15 \) and choose \( \epsilon \) as the varying parameter. Here, we train ESN with the
FIG. 2. (a)-(c) The training dynamics which are used to train ESN. We use periodic dynamics for $\varepsilon = 0.4, 0.38, 0.36$. (d)-(f) The original dynamics those we are trying predict using ESN for diverse values of $\varepsilon$. (g)-(i) Corresponding predicted dynamics. We observe that, machine can capture diverse dynamics extremely well which are not at all revealed to the machine during training. Machine is only aware of the periodic dynamics with different amplitude and corresponding system parameters. Any other information is not disclosed from the earlier. Interestingly, depending only on the new system parameter value, machine is able to predict more complex dynamics like chaos (see fig.2(g)) as well as simple dynamics like steady state (see fig.2(i)) extremely well even at distant parameters from the training data.

C. Prediction of bifurcation diagram and entropy

We extend this study for the whole range of parameter space. We follow the same training scheme discussed in the previous section for predicting the dynamics of the system for a large set of values of $\varepsilon$. In Fig3(a), we plot the original bifurcation diagram and in Fig3(c), we plot the bifurcation diagram obtained from the ESN predicted dynamics. Here, we train the machine only with the dynamics at $\varepsilon = 0.4, 0.38, 0.36$ (see Figs2(a), (b), (c)) and the blue vertical lines in Fig3(a). Now, machine can easily predict the dynamics throughout the whole parameter space. Here, machine is not only able to predict the dynamics when the parameter is quite far from the training parameter, interestingly, it can also capture the bifurcation scenario of period doubling
route towards chaos. The transition points are captured almost perfectly. Therefore, we reproduce the whole bifurcation diagram of the system with simple ESN architecture and only a small number of periodic training dynamics.

Furthermore, We calculate the entropy profile with respect to $\epsilon$ from the original and predicted time series data as depicted in Fig.4(b) and Fig.4(d). The entropy for a given trajectory is defined as

$$H = - \sum_{i=1}^{n} p_i \log(p_i),$$

where, $p_i$ is the nonzero probability of $i^{th}$ state of the phase space being occupied when the whole state space volume is discretized by sufficiently small unit volumes. $n$ is the total number of states with nonzero probability. We observe that the entropy profile of the predicted dynamics matches with that of the original system very well, which verifies the degree of phase space occupancy of the predicted trajectory for the whole range of the parameter $\epsilon$.

D. Prediction of basin for unknown attractors

We continue this study with the prediction of basin for different attractors using the same architecture of the reservoir. Here, the system have a multistable region in the $(\beta - \epsilon)$ parameter space as shown in Fig.4(a), highlighted with the yellow color. We randomly choose a parameter set $(\beta, \epsilon) = (0.15, 0.4)$ from the bistable region for identifying basins. In Fig.4(d), we plot basin of the original system. Three different colored basins exhibit three different dynamics (time series are presented in Figs.4(a), (b), (c)). For the initial condition in region I$_{o}$ (shown by teal color), system blows up towards infinity (see Fig.4(a)). Region II$_{o}$ (highlighted with green color) produces oscillating dynamics which is depicted in Fig.4(b). Lastly, region III$_{o}$ gives a stable steady state dynamics as shown in Fig.4(c).

To start the prediction of basin through ESN, one should notice that, the chosen training parameter set is not very far from the parameter at which the basins are to be identified. Considering the complexity of multistable system, the basin prediction task at a completely unknown parameter is a quite challenging task itself. However, one can always increase this parameter drift by sacrificing the accuracy and using larger training data set with greater computational cost. Only one type of dynamics (feasible to choose the oscillatory dynamics) we use for training and no other information of the system is revealed to machine. Here, we choose the oscillatory dynamics at the parameter values $\epsilon = 0.43$, $0.42$, $0.41$ as the training data and want to predict the basins for all the three different dynamics at $\epsilon = 0.4$. We follow the same procedure for training as previously mentioned and after a successful training, we start predicting the dynamics at $\epsilon = 0.4$. Here, we fix the machine hyper-parameters at $\rho = 0.06$, $\sigma = 0.5057$, $a = 0.385$, $b = 5 \times 10^{-8}$. Now, to identify the basins we start the calculations from the space initial conditions $(x_0, y_0) \in [0, 15] \times [-35, 5]$. Interestingly, we find three different predicted dynamics depending only on the initial conditions. First one is the unusual dynamics like Fig.4(g), which is not bounded (approaches near $x = 25$) into the usual range of the system trajectories. Second one is the oscillation (as shown in Fig.4(h)). Here, we need to focus on the transient dynamics, because in Fig.4(h), the oscillation starts immediately after starting the prediction. On the other hand, in Fig.4(i), we find the trajectory initially generates negligible oscillations (almost steady state) for large time points and after that it produces oscillations. For this study, we take 2500 time points as the threshold for classifying two different dynamics. Producing subthreshold oscillation (we choose $x < 1.4$) for initial 2500 time points fells it into the third type of dynamics (i)). Based on these three types of predicted dynamics, we obtain Fig.4(e), the predicted basin. We find the machine is able to identify the basins for unbounded dynamics very well (Region I$_{p}$ in Fig.4(e)). Region II$_{p}$ in predicted basin, ESN generated dynamics is same as the original system dynamics. Lastly, for region III$_{p}$ exhibits dynamics as shown in Fig.4(i). Here, machine is not generating a stable steady state dynamics, but it initially produces transient small oscillations, depending on that we can separate region III$_{p}$ in Fig.4(e).

Furthermore, in the process of generating reservoir some randomness enter into the system. To overcome the impact of randomness over the obtained results and to check the robustness of our scheme, we repeat the same investigation for a sufficient times (say N) for independently generated reservoir keeping the ESN hyperparameter fixed. Depending on num-

FIG. 4. Basin prediction for system (9). (a), (b), (c) Three original dynamics are presented for three different region of the (d) original basin. Region I$_{o}$, region II$_{o}$ and region III$_{o}$ correspond to (a) unbounded solution, (b) oscillation and (c) steady state. (e) Predicted basin and the corresponding predicted dynamics for region I$_{p}$, region II$_{p}$, region III$_{p}$ are shown in (g), (h), (i). True positive ratio for the basin prediction is depicted in (f).
FIG. 5. Prediction of the bifurcation diagram and entropy. (a), (b), (c) Oscillatory dynamics for training are shown on \( x - y \) phase space. (d) Original bifurcation diagram and (e) original entropy are plotted with respect to the parameter \( \alpha \). Three blue lines in (a) represent the parameter values of the training dynamics. Corresponding predicted bifurcation diagram and the entropy are plotted in (f) and (g) respectively.

number of correct prediction (say M). We calculate the quantity, True positive ratio = \( \frac{M}{N} \). If the ratio goes to unity, it means every time machine is able to produce the right prediction and value of this ratio is zero means machine is not able to predict the dynamics correctly for a single time. Intermediate values of the ratio shows the level of producing the correct prediction by ESN depending on the intrinsic randomness. Here, we take N=100. In Fig.5(f), we plot the true positive ratio at every initial conditions for the whole space on a 0 (yellow) to 1 (blue) scale. Machine is able to predict the whole space of the basin quite perfectly. Only near boundaries when transition occurs from one dynamics to other, the accuracy level of the predicted dynamics is at lower side. Therefore, we can conclude that excluding the boundary regions, over the space of whole initial conditions basins for different dynamics are considerably well predicted.

E. Another bistable system

We generalize the results by considering another system\textsuperscript{32} having bistability as follows:

\[
\begin{align*}
\dot{x} &= yz + \alpha \\
\dot{y} &= x^2 - y \\
\dot{z} &= 1 - 4x
\end{align*}
\]

where, for \( \alpha < 0 \) the system possesses an unstable fixed point, but for \( \alpha > 0 \) the system enters into a bistable regime, where a stable fixed point \( \left( \frac{1}{3}, \frac{1}{16}, -16\alpha \right) \) and a stable oscillator coexist.

Similarly for this model, we train the machine with a few simple dynamics of the system with the parameter values of \( \alpha \) and let the machine predict the dynamics for whole range of \( \alpha \). Here, we take the periodic dynamics (see Figs.5(a), (b), (c)) for \( \alpha = 0.03, 0.032, 0.034 \) to train ESN, which are also shown by the three blue vertical lines in Fig.5(d). The corresponding hyperparameters are as follows: \( \rho = 0.5057, \sigma = 0.35, a = 0.25, b = 5 \times 10^{-6} \). After completing the training procedure, we change the new parameter values to the machine input, and continuing the same process for the whole parameter range, we obtain the predicted bifurcation diagram depending only on the values of \( \alpha \). In Fig.5(d), we plot the original bifurcation diagram and Fig.5(f) is the corresponding machine predicted diagram. We observe that machine identifies the transition towards higher periodic order quite accurately. Moreover, we plot the corresponding entropy of the original and the machine predicted data in Fig.5(e) and in Fig.5(g) respectively. The original and machine predicted entropy follow almost similar path with respect to \( \alpha \).

The system possesses a bistability between two stable dynamics: (i) stable oscillation (see Fig.5(a)) and (ii) stable steady state (see Fig.5(b)). We plot the corresponding original basin in Fig.5(c) over the initial conditions \( (x_0, z_0) \in [-0.5, 1.0] \times [-6, 3] \), where region I\textsubscript{o} resembles the initial conditions for oscillating dynamics and region II\textsubscript{o} is the basin for steady state. The original basin is corresponding to \( \alpha = 0.03 \). For training we take the oscillation dynamics at \( \alpha = 0.033, 0.032, 0.031 \). In this case we set the machine hyperparameters as: \( \rho = 0.5057, \sigma = 0.06, a = 0.385, b = 9 \times 10^{-7} \). Now ESN generates two types of dynamics on the space of initial conditions. One oscillatory dynamics (shown in Fig.5(f)) and one stable steady state (shown in Fig.5(g)). Now, we separate the basins based on these two dynamics, and finally we obtain the predicted basin for this system in Fig.5(d). Now, to verify the robustness under the system’s intrinsic randomness, we plot the true positive ratio for the basin in Fig.5(e) and similar to the previous system we find that near the boundary true positive ratio remains low. However, excluding the boundary region machine is able to capture two separate dynamics considerably well depending only on the initial conditions.
FIG. 6. Basin prediction for system (12). Time series of the original dynamics (a) Oscillation and (b) fixed point dynamics corresponding to two different region of the (c) original basin. (d) Predicted basin with the corresponding predicted dynamics (f) oscillation and (g) steady state. (e) True positive ratio for the prediction of the basin.

IV. CONCLUSION

In this study, we investigated a strategy to identify diverse dynamics in a system through a parameter-aware ESN approach. Using this technique, learning from a few training dynamics ESN can build the dynamics for any parameter value even if it lies considerably far from the training parameter set. We elaborately explained the whole training and prediction process for obtaining the diverse predicted dynamics. Without being exposed to any other additional information, machine is able to train itself in such a way that it can capture the nature of complex dynamics for a broad range of parameter values. Exploiting this scheme, we are able to draw the predicted bifurcation diagram which almost follows the original dynamics. Another significant direction of this study is to identify the basins for multistable attractors. It is well known that there are several systems appearing in many areas possess bistability or multistability for a certain parameter space. Also, without any prior knowledge of the basin these systems can produce various undesired situations in practice. In this context, we gave a recipe for locating basins of multiple co-existing attractors using the dynamical information of only one of them. Training with only one time series at few parameter values each is enough for the machine to be trained such that it can predict the dependency of the system dynamics on the initial conditions. Machine is able to generate different dynamics if we start the prediction process from different initial conditions and eventually the basin is also predicted without putting any extra effort. In this context, a very recent study has attempted to show the utility of a different machine learning scheme for locating an unseen attractors of a multistable system. Thus, machine learning approach for predicting multistability contributes greatly in better understanding of complex dynamical systems.

CONFLICT OF INTEREST

The authors have no conflict to disclose.

AUTHOR CONTRIBUTIONS

M.R: Conceptualization (equal); Methodology (equal); Formal analysis (lead); Investigation (lead); Software (lead); Visualization (lead); Validation (equal); Writing – original draft (lead). S. M.: Conceptualization (equal); Methodology (equal); Formal analysis (lead); Investigation (lead); Software (lead); Visualization (lead); Validation (equal); Writing – original draft (lead). C.H.: Conceptualization (equal); Methodology (equal); Validation (equal); Writing – review and editing (equal). A.P.: Conceptualization (equal); Supervision (equal); Writing – review and editing (equal). N.V.K.: Supervision (equal); Writing – review and editing (equal). M.D.S.: Conceptualization (equal); Methodology (equal); Validation (equal); Supervision (equal); Writing – review and editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ACKNOWLEDGEMENT

M.R. and M.D.S. are financially supported by Department of Science and Technology (DST), India under the Indo-Russian Joint Research Programme (No. INT/RUS/RSF/P-18). S.M. and M.D.S. acknowledge SERB, Department of Science and Technology (DST), India (DST - SERB (CRG/2021/003301)). C.H. is supported by DST-INSPIRE-Faculty grant (Grant No. IFA17-PH193).

REFERENCES

1. U. Feudel, “Complex dynamics in multistable systems,” Int. J. Bifurc. Chaos 18, 1607–1626 (2008).
2. T. Tanaka and T. Aoyagi, “Multistable attractors in a network of phase oscillators with three-body interactions,” Phys. Rev. Lett. 106, 224101 (2011).
3. C. Hens, S. K. Dana, and U. Feudel, “Extreme multistability: Attractor manipulation and robustness,” Chaos 25, 053112 (2015).
4. C. Hens, R. Banerjee, U. Feudel, and S. Dana, “How to obtain extreme multistability in coupled dynamical systems,” Phys. Rev. E 85, 035202 (2012).
5. U. Chaudhuri and A. Prasad, “Complicated basins and the phenomenon of amplitude death in coupled hidden attractors,” Phys. Lett. A 378, 713–718 (2014).
A. N. Pisarchik and U. Feudel, “Control of multistability,” Phys. Rep. 540, 167–218 (2014).

E. Shahaj and M. D. Shrimlal, “Controlling multistability with intermittent noise,” Chaos Solit. Fractals. 160, 112187 (2022).

H. Jaeger, “The “echo state” approach to analysing and training recurrent neural networks-with an erratum note,” Bonn, Germany: German National Research Center for Information Technology GMD Technical Report 148, 13 (2001).

H. Cui, X. Liu, and L. Li, “The architecture of dynamic reservoir in the echo state network,” Chaos 22, 033127 (2012).

Q. Ma, L. Shen, W. Chen, J. Wang, J. Wei, and Z. Yu, “Functional echo state network for time series classification,” Inf. Sci. 373, 1–20 (2016).

J. D. Gauthier, E. Bollt, A. Griffith, and W. A. Barbosa, “Next generation reservoir computing,” Nat. Commun. 12, 1–8 (2021).

J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, “Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach,” Phys. Rev. Lett. 120, 024102 (2018).

G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, “Recent advances in physical reservoir computing: A review,” Neu. Net. 115, 100–123 (2019).

H.-H. Kim and J. Jeong, “Decoding electroencephalographic signals for direction in brain-computer interface using echo state network and gaussian readouts,” Comput. Biol. Med. 110, 254–264 (2019).

A. E. Hramov, V. A. Maksimenko, and A. N. Pisarchik, “Physical principles of brain–computer interfaces and their applications for rehabilitation, robotics and control of human brain states,” Phys. Rep. 918, 1–133 (2021).

H. Hauser, “Physical reservoir computing in robotics,” in Reservoir Computing (Springer, 2021) pp. 169–190.

M. D. Skowronski and J. G. Harris, “Automatic speech recognition using a predictive echo state network classifier,” Neural Netw. 20, 414–423 (2007).

M. D. Skowronski and J. G. Harris, “Noise-robust automatic speech recognition using a predictive echo state network,” IEEE Transactions on Audio, Speech, and Language Processing 15, 1724–1730 (2007).

A. Banerjee, J. D. Hart, R. Roy, and E. Ott, “Machine learning link inference of noisy delay-coupled networks with optoelectronic experimental tests,” Phys. Rev. X 11, 031014 (2021).

A. Banerjee, S. Chandra, and E. Ott, “Inference of c. elegans neural network structure from calcium fluorescence time series data with reservoir computing,” Bulletin of the American Physical Society (2022).

S. Ghosh, A. Senapati, A. Mishra, J. Chattopadhyay, S. K. Dana, C. Hens, and D. Ghosh, “Reservoir computing on epidemic spreading: A case study on covid-19 cases,” Phys. Rev. E 104, 014308 (2021).

T. W. Weng, H. Yang, C. Gu, J. Zhang, and M. Small, “Synchronization of chaotic systems and their machine-learning models,” Phys. Rev. E 99, 042203 (2019).

F. Borra, A. Vulpiiani, and M. Cencini, “Effective models and predictability of chaotic multiscale systems via machine learning,” Phys. Rev. E 102, 052203 (2020).

T. L. Carroll, “Using reservoir computers to distinguish chaotic signals,” Phys. Rev. E 98, 052209 (2018).

J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, “Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data,” Chaos 27, 121102 (2017).

S. Saha, A. Mishra, S. Ghosh, S. K. Dana, and C. Hens, “Predicting bursting in a complete graph of mixed population through reservoir computing,” Phys. Rev. Res. 2, 033338 (2020).

M. Roy, A. Senapati, S. Poria, A. Mishra, and C. Hens, “Role of assortativity in predicting burst synchronization using echo state network,” Phys. Rev. E 105, 064205 (2022).

R. S. Zimmermann and U. Parlitz, “Observing spatio-temporal dynamics of excitable media using reservoir computing,” Chaos 28, 043118 (2018).

M. Lukoševičius, “A practical guide to applying echo state networks,” in Neural networks: Tricks of the trade (Springer, 2012) pp. 659–686.

G. Mandal, S. Sinha, and M. D. Shrimlal, “Machine-learning potential of a single pendulum,” Phys. Rev. E 105, 054203 (2022).

H. Fan, L.-W. Kong, Y.-C. Lai, and X. Wang, “Anticipating synchronization with machine learning,” Phys. Rev. Res. 3, 023237 (2021).

R. Xiao, L.-W. Kong, Z.-K. Sun, and Y.-C. Lai, “Predicting amplitude death with machine learning,” Phys. Rev. E 104, 014205 (2021).

J. Z. Kim, Z. Lu, E. Nozari, G. J. Pappas, and D. S. Bassett, “Teaching recurrent neural networks to infer global temporal structure from local examples,” Nat. Mach. Intell. 3, 316–323 (2021).

A. Röhm, D. J. Gauthier, and I. Fischer, “Model-free inference of unseen attractors: Reconstructing phase space features from a single noisy trajectory using reservoir computing,” Chaos 31, 103127 (2021).

A. Griffith, A. Pomerance, and D. J. Gauthier, “Forecasting chaotic systems with very low connectivity reservoir computers,” Chaos 29, 123108 (2019).

J. Yperman and T. Becker, “Bayesian optimization of hyper-parameters in reservoir computing,” arXiv preprint arXiv:1611.05193 (2016).

C. E. Shannon, “A mathematical theory of communication,” The Bell system technical journal 27, 379–423 (1948).

X. Wang and G. Chen, “A chaotic system with only one stable equilibrium,” Commun. Nonlinear. Sci. Numer. Simul. 17, 1264–1272 (2012).

D. J. Gauthier, I. Fischer, and A. Röhm, “Learning unseen coexisting attractors,” arXiv preprint arXiv:2207.14133 (2022).