Prediction of chaotic attractors in quasiperiodically forced logistic map using deep learning

J. Meiyazhagan\textsuperscript{1} and M. Senthilvelan\textsuperscript{1,}\textsuperscript{*}

\textsuperscript{1}Department of Nonlinear Dynamics, Bharathidasan University, Tiruchirappalli - 620 024, Tamilnadu, India

Abstract

We forecast two different chaotic dynamics of the quasiperiodically forced logistic map using the well-known deep learning framework Long Short-Term Memory. We generate two data sets and use one in the training process and the other in the testing process. The predicted values are evaluated using the metric called Root Mean Square Error and visualized using the scatter plots. The robustness of the Long Short-Term Memory model is evaluated using the number of units in the layers of the model. We also make multi-step forecasting of the considered system. We show that the considered Long Short-Term Memory model performs well in predicting chaotic attractors upto three steps.

\textsuperscript{*} Correspondence to: velan@cnld.bdu.ac.in

arXiv:2203.11151v1 [cs.LG] 18 Mar 2022
I. INTRODUCTION

Recently, Machine Learning (ML) and Deep Learning (DL) models have been used in various fields of physics [1–3]. In the study of dynamics of nonlinear systems, ML and DL algorithms are extensively used for the prediction and discovery of the behaviour of the chaotic and complex systems. For example, they have been used to identify chimera states [4, 5], in the replication of chaotic attractors [6], using symbolic time series for network classification [7], separating chaotic signals [8], learning dynamical systems in noise [9] and in the prediction of extreme events [10–15]. Very recently, the authors of Ref. [16] have considered Hénon map and used a ML algorithm, namely Artificial Neural Network (ANN), to study the extreme events in it. The authors have focussed on binary classification and classified the data points as extreme and non-extreme [16].

In our studies, we consider logistic map with quasiperiodic forcing and predict the time series of the system which is not continuous. The system exhibits chaos in two different regimes. We predict both the chaotic attractors of this system with the help of the DL framework, namely Long Short-Term Memory (LSTM). The logistic map with quasiperiodic forcing is described by the following equations, namely [17, 18]

\[ x_{n+1} = \alpha[1 + \epsilon \cos(2\pi\phi_n)]x_n(1 - x_n), \]  
\[ \phi_{n+1} = \phi_n + \omega \pmod{1}, \]  

where \( \epsilon \) and \( \omega = (\sqrt{5} - 1)/2 \) are the forcing amplitude and irrational driving frequency respectively. The authors in Ref. [17] redefined the driving parameter as \( \epsilon' = \epsilon/(4/\alpha - 1) \) to study the dynamics of the system in the regimes of \( 0 \leq x \leq 1, 0 \leq \phi \leq 1 \) and \( 0 \leq \epsilon \leq 1. \) The schematic phase diagram [17] of the system is given in Fig. 1. The system shows various dynamic behaviours, namely periodic, strange nonchaotic and chaotic attractors which can be characterized by the nonzero Lyapunov exponent \( \Lambda \) [17], where

\[ \Lambda = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} \ln |\alpha[1 + \epsilon \cos(2\pi\phi_i)](1 - 2x_i)|. \]  

From Fig. 1 we can notice the interesting behaviour of the considered system which has two chaotic regimes, namely \( C_1 \) and \( C_2. \) The \( C_1 \) regime is the continuation of the chaotic regime in the logistic map for \( \epsilon = 0 \) at the end of the period-doubling cascade, at \( \alpha = 3.5699... \). The chaos in \( C_2 \) regime is due to low nonlinearity and large amplitude forcing [17]. Our aim
FIG. 1: Schematic phase diagram of the quasiperiodically forced logistic map. $C_1$ and $C_2$ are two different chaotic regimes.

is to predict chaotic attractors in both the regime using LSTM model since it is capable of forecasting the data which is in the form of a sequence.

We organize our work as follows. In Sec. 2, we discuss the generation of training and testing data. In Sec. 3, we consider a DL framework called LSTM and train it using training set data and predict the test set data. The performance of the LSTM model is discussed in Sec. 4. We present the conclusion in Sec. 5.

II. DATA PREPARATION

Generating data is the foremost task in prediction because prediction is done only by learning the relationship between the given data. We calculate the value of $x$ for $10^5$ iterations using equation (1) in both the regimes $C_1$ and $C_2$. This discrete space data is then converted into supervised learning data by taking $x_n$ as input and $x_{n+1}$ as output. The chaotic attractors in the both regimes $C_1$ and $C_2$ are shown in Fig. 2. The Figs. 2 (a) & (b) corresponding to the regime $C_1$ and Figs. 2 (c) & (d) correspond to the $C_2$ regime. The values of the parameters are taken as (a) $\alpha = 3.6, \epsilon' = 0.5$, (b) $\alpha = 3.9, \epsilon' = 1.0$, (c) $\alpha = 3.0, \epsilon' = 1.0$ and (d) $\alpha = 3.1, \epsilon' = 0.8$. We divide the data into two parts: (i) training set
FIG. 2: Chaotic attractors in two different regimes $C_1$ and $C_2$. (a) $\alpha = 3.6, \epsilon' = 0.5$ and (b) $\alpha = 3.9, \epsilon' = 1.0$ correspond to $C_1$. (c) $\alpha = 3.0, \epsilon' = 1.0$ and (d) $\alpha = 3.1, \epsilon' = 0.8$ correspond to $C_2$. The points in blue (colour online) denoting the training set data and red (colour online) denoting the test set data.

and (ii) test set. Training set data are used during the training process of the DL model and test set data are used for the evaluation of the ability of the DL model. In Fig. 2 the blue dots are the data used for training purpose and the red coloured data are used for testing. We use $6 \times 10^4$ data as training set data and $4 \times 10^4$ data as test set data.

These two sets of data are rescaled using min-max normalization which is given by the formula [19],

$$x_{i, \text{rescaled}} = a + \frac{(x_i - x_{\text{min}})(b - a)}{x_{\text{max}} - x_{\text{min}}}, \quad i = 1, 2, 3, \ldots, n,$$

(3)

where $x_{\text{min}}$ and $x_{\text{max}}$ are the minimum and maximum value of the data set respectively. We fix $a = -1$ and $b = +1$ in order to scale the data between $-1$ and $+1$. During the testing phase, this preprocessing scaling step is reversed after obtaining the output from the DL model in order to compare the results with the actual data.
III. DEEP LEARNING FRAMEWORK: LONG SHORT-TERM MEMORY

When the data is in a sequential form one can make use of the Recurrent Neural Networks (RNN) [20] which is a type of ANN. For the present study we consider a DL framework known as LSTM [21] which is a special kind of RNNs. In recent years, LSTM framework has proven to be capable of forecasting time series of the chaotic systems even when there are extreme events in the time series [10, 13, 14]. The main feature that differentiates LSTM from the other RNNs is that the latter has only one activation function for the neurons that is tanh but in the case of the former, a sigmoid function is used for recurrent activations and tanh is used for the activation of neurons. The sigmoid activation function is defined by [22],

\[
\sigma(z) = \frac{1}{1 + e^{-z}}.
\]

(4)

We construct the LSTM model in the following way. We consider two LSTM layers each having 16 units in it and followed by a layer which has one neuron for output. During the training, we give both the input and the corresponding output to the model, that is we give \(x_n\) as the input and \(x_{n+1}\) as the output. By doing this, the model will learn the nonlinear relations between the given data. After training, the learned model is used to forecast the data steps. During the testing phase, we feed only the input data and ask the model for the corresponding output. The predicted values at the output given by the LSTM model are compared with actual values to determine the efficiency of the model in forecasting the chaotic attractors of the considered system.

IV. RESULTS AND DISCUSSION

To visualize the performance of the considered DL model, predicted data are plotted over the actual data in Fig. 3. Black dots (colour online) denote the actual value and green dots (colour online) denote the predicted value. The data in Figs. 3 (a) & (b) correspond to \(C_1\) regime and Figs. 3 (c) & (d) correspond to \(C_2\) regime. From the plots, we can see that relatively all predicted data coincide with the actual data. To have a clear understanding of the efficiency of the model we calculate the Root Mean Square Error (RMSE) value using the formula,

\[
RMSE = \sqrt{\frac{1}{N_{\text{Test}}} \sum_{i=1}^{N_{\text{Test}}} (\hat{Y}_i^{\text{Test}} - Y_i^{\text{Test}})^2}.
\]

(5)
where $\hat{Y}_{i}^{Test}$, $Y_{i}^{Test}$ and $N_{Test}$ denote the predicted values, actual values and total number of data in the test set respectively. We make use of the scatter plots which are plotted by taking actual values in the $x$-axis and predicted values in the $y$-axis (see Fig. 4). From Figs. 4 (a) & (b) we can see that the RMSE values for the regime $C_1$ are 0.015 and 0.012 respectively for the parameter values $\alpha = 3.6, \epsilon' = 0.5$ and $\alpha = 3.9, \epsilon' = 1.0$. The outcome of the scatter plots almost fit in straight line, thereby indicating that the difference between predicted and actual values are very low. From Figs. 4 (c) & (d) we can see that the the results of the second regime $C_2$ are calculated as 0.013 and 0.004 respectively for the parameter values $\alpha = 3.0, \epsilon' = 1.0$ and $\alpha = 3.1, \epsilon' = 0.8$. The scatter plots for the test set data of regime $C_2$ also show very little scatter points, thereby indicating the best fit of predicted data with the actual data.
A. Effect of model architecture

To study the effect of model architecture on the performance of the considered model we vary the number of units and analyse the performance based on the RMSE values. For this purpose, we change the units in both LSTM layers and train the model. Then each trained model is evaluated using the test set data. The outcome is shown in Fig. 5. For the $C_1$ regime, we evaluate the model with the data corresponding to $\alpha = 3.6, \epsilon' = 0.5$ and plot the results in Fig. 5(a). For the $C_2$ regime, we evaluate the model with the data corresponding to $\alpha = 3.0, \epsilon' = 1.0$ and plot the results in Fig. 5(b). The RMSE value changes while varying the number of units in the LSTM layers.
FIG. 5: RMSE values for various number of units in LSTM layers. (a) and (b) corresponds to the regimes $C_1$, $\alpha = 3.6$, $\epsilon' = 0.5$ and $C_2$, $\alpha = 3.0$, $\epsilon' = 1.0$ respectively.

**B. Multi-step forecasting**

Now, we consider the task of multi-step forecasting. To do this, while preparing the supervised learning data, instead of having only one future step value, we take more than one value at the output. For this we consider the data in both the regimes $C_1$ ($\alpha = 3.6$, $\epsilon' = 0.5$) and $C_2$ ($\alpha = 3.0$, $\epsilon' = 1.0$). The results of multi-step forecasting are shown in Fig. 6. From this figure we can infer that in the forecasting of multi-steps two and three, the considered model outperformed our expectations in the prediction task, the plots have fewer scatter points and the RMSE values are in admissible range. But for steps four and five, the model failed to give accurate values in both the regimes. It can be seen from Figs. 6 (c), (d), (g) and (h), the points are scattered much when making forecasting with fourth and fifth steps.

**V. CONCLUSION**

In this work, we have considered the logistic map with quasiperiodic forcing. The system exhibits chaos in two different regimes. We employed a DL framework LSTM, for the prediction of two different chaos. For this, we have generated $10^5$ data totally and used $6 \times 10^4$ data for training and the remaining $4 \times 10^4$ data for the purpose of testing. We forecast the chaos corresponding to the two regimes $C_1$ and $C_2$. The outcome of the experiments
FIG. 6: Scatter plots with RMSE values for the multi-step forecasting. (a)-(d) correspond to $C_1$ regime $\alpha = 3.6, \epsilon' = 0.5$ and (e)-(f) correspond to $C_2$ regime $\alpha = 3.0, \epsilon' = 1.0$.

are evaluated using the performance metric RMSE value and they are analyzed through the scatter plots which have been plotted between the predicted value and actual value. Further, we have checked the effect of the number of units of the LSTM layers on the performance of the model. In this connection, we have done multi-step forecasting in order to predict more than one future value of the considered map. From the obtained results, we conclude that the developed LSTM framework can be used for forecasting the chaotic dynamics of the discrete system, namely quasiperiodically forced logistic map described by Eq. [1].

ACKNOWLEDGEMENTS

JM thanks RUSA 2.0 project for providing a fellowship to carry out this work. MS acknowledges RUSA 2.0 project for providing financial support in procuring a high-performance GPU server which highly assisted this work.

[1] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Rev. Mod. Phys. 91, 045002 (2019).
[2] A. Choudhary, J. F. Lindner, E. G. Holliday, S. T. Miller, S. Sinha, and W. L. Ditto, Phys. Rev. E 101, 062207 (2020).

[3] S. T. Miller, J. F. Lindner, A. Choudhary, S. Sinha, and W. L. Ditto, Chaos, Solitons & Fractals: X 5, 100046 (2020).

[4] G. Barmparis, G. Neofotistos, M. Mattheakis, J. Hizanidis, G. Tsironis, and E. Kaxiras, Phys. Lett. A 384, 126300 (2020).

[5] M. A. Ganaie, S. Ghosh, N. Mendola, M. Tanveer, and S. Jalan, Chaos 30, 063128 (2020).

[6] J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott, Chaos 27, 121102 (2017).

[7] A. Panday, W. S. Lee, S. Dutta, and S. Jalan, Chaos 31, 031106 (2021).

[8] S. Krishnagopal, M. Girvan, E. Ott, and B. R. Hunt, Chaos 30, 023123 (2020).

[9] S. Mukhopadhyay and S. Banerjee, Chaos 30, 103125 (2020).

[10] J. Meiyazhagan, S. Sudharsan, and M. Senthilvelan, Eur. Phys. J. B 94, 1 (2021).

[11] J. Meiyazhagan, S. Sudharsan, A. Venkatesan, and M. Senthilvelan, Eur. Phys. J. Plus 137, 1 (2022).

[12] V. Pyragas and K. Pyragas, Phys. Lett. A 384, 126591 (2020).

[13] A. Ray, T. Chakraborty, and D. Ghosh, Chaos 31, 111105 (2021).

[14] S. N. Chowdhury, A. Ray, A. Mishra, and D. Ghosh, J. Phys. Complex. 2, 035021 (2021).

[15] A. Asch, E. Brady, H. Gallardo, J. Hood, B. Chu, and M. Farazmand, arXiv:2111.04857 (2021).

[16] M. Lellep, J. Prexl, M. Linkmann, and B. Eckhardt, Chaos 30, 013113 (2020).

[17] A. Prasad, V. Mehra, and R. Ramaswamy, Phy. Rev. E 57, 1576 (1998).

[18] J. Heagy and S. Hammel, Physica D: Nonlinear Phenomena 70, 140 (1994).

[19] L. Al Shalabi, Z. Shaaban, and B. Kasasbeh, Journal of Computer Science 2, 735 (2006).

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Nature 323, 533 (1986).

[21] S. Hochreiter and J. Schmidhuber, Neural Comput. 9, 1735 (1997).

[22] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning (MIT Press, 2016).