Data driven nonlinear dynamical systems identification using multi-step CLDNN

Cite as: AIP Advances **9**, 085311 (2019); https://doi.org/10.1063/1.5100558
Submitted: 18 April 2019 • Accepted: 02 August 2019 • Published Online: 19 August 2019

Qi Teng and Lei Zhang

Data driven nonlinear dynamical systems identification using multi-step CLDNN

Cite as: AIP Advances 9, 085311 (2019); doi: 10.1063/1.5100558
Submitted: 18 April 2019 • Accepted: 2 August 2019 •
Published Online: 19 August 2019

Qi Teng and Lei Zhang[a)]

AFFILIATIONS

School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing 210046, P.R.China

[a)]Electronic mail: leizhang@njnu.edu.cn

ABSTRACT

In many cases, the equations that dynamical systems are based on are unknown and hard to model and predict. On the other hand, machine learning algorithms are based on the data of a solution as it evolves and do not need equations. In the era of abundant data, using machine learning technology to discover accurate mathematical models of dynamical systems directly from time series data becomes increasingly important. Recently, a multi-step deep neural networks (multi-step DNN) model without need of direct access to temporal gradients is proposed, which can accurately learn the evolution from a given set of observed data, identify nonlinear dynamical systems, and forecast future states. However, the architecture lacks the capability to capture long term temporal dependencies from dynamical time-series data. In the paper, based on the multi-step time-stepping schemes, we proposed a new CLDNN model which combine convolutional layer, long short-term memory layer and fully connected layer, to address the aforementioned weakness. The effectiveness of our model is tested for several benchmark problems involving the identification and prediction of complex, nonlinear and chaotic dynamics. The experiment results show that the multi-step CLDNN has better identification and prediction performance than the multi-step DNN. The research provides possible corroboration for developing new deep learning based algorithms for nonlinear system identification.

I. INTRODUCTION

Dynamical systems,[1] which contain many mathematical models, play a key role in deepening our understanding for complex physical world and improving our ability to forecast the future development track for a given process. From the flow rate of water in a pipeline to the amount of fish in spring, the mathematical modelling of the dynamics has yielded a set of tools to analyze the current state of the system which depends on the past and to predict possible states in the future. Often the above tools are in form of differential equations that originate from first physical principles. However, the complicacy of dynamical systems impedes our complete understanding and makes the first principles approach unworkable in most cases. Under this circumstance, scholars have to empirically postulate and simplify dynamical system models based on a given set of observations, such as models of tumor growth,[2] stock prices,[3] and social dynamics.[4] In the era of abundant data, how to discover sophisticated and accurate mathematical models of dynamical systems directly from data becomes increasingly important.

The answer to the above question requires the well-established knowledge of the system identification field,[5,6] which can be classified as white-box and black-box approaches depending on whether the first principle model is fully or not admissible. More recently, there were many white-box approaches including symbolic regression,[7] compressive sensing,[8,9] sparse regression,[10] to determine the underlying architecture of nonlinear dynamical system derived from data. These interpretable models can detect the overall parametric form of the governing equation. However, the above approaches have to depend on the libraries of basis function and therefore have difficulty in selecting appropriate basis function to identify the nonlinear dynamical systems from time-series data. Until recently, the approaches based on black box such as NARMAX,[11] neural networks,[12,13] genetic algorithms[14] and Laplacian spectral analysis[15] were proposed, which could predict the future system states from a given set of data describing the present and past states, while disambiguating the choice of basis function.

Later, Raissi et al. proposed a novel approach, which combine the classical multi-step family of time-stepping schemes

with deep neural networks (multi-step DNN), to approximate dynamical systems. This is an important innovation in practice as the approach need not to commit to a particular class of basis function by using a richer family of function approximators. Since the time derivatives are discretized using classical time-stepping rules, the multi-step DNN approach does not require direct access or approximations to temporal gradients.[16] However, the single deep neural network (DNN) has some limitations in its modeling capabilities, which causes the unsatisfactory predictive performance (such as chaotic dynamics). In particular, the traditional multi-step DNN architecture lacks the capability to capture the temporal dependencies from nonlinear dynamical time-series data. On the other hand, in deep learning fields such as speech recognition and text classification, some researchers have begun to utilize the complementarity of convolutional neural network (CNN) and long short-term memory (LSTM), such as CLDNN,[17] to address the aforementioned weakness.

Inspired by the multi-step DNN and CLDNN, we propose a new method of nonlinear systems identification that combines CNN, LSTM, DNN and classical linear multi-step method derived from numerical analysis into one unified architecture (multi-step CLDNN). For machine learning research, some combinations of different models are useful and have been widely utilized to improve the experiment performance. For example, the CLDNN itself is a combination of CNN, LSTM and DNN, which has been proven very useful for many machine learning tasks. We use CNN to automatically extract abstract features and LSTM to model temporal dependencies in time series data, and then DNN to map the function f representing the evolution of dynamical systems. In contrast to the multi-step DNN, our approach can effectively discover the dynamical dependencies in a given set of temporal data-snapshots, and return a more accurate results. Similar to the multi-step DNN, our method does not need to directly approximate the temporal gradients due to a given set of discretized time or choose a particular class of basis function. Moreover, in contrast to the multi-step DNN, exploring temporal information and capturing time dependencies are our key breakthrough, which allow our approach to have better prediction performance.

The rest of paper is organized as follows. In Section III, the multistep CLDNN model is tested through several benchmark experiments consisting of the two-dimensional damped oscillator, the chaotic Lorenz system, the Hopf bifurcation, and the experiment of Fluid flow behind a cylinder. Section IV summarizes our work and discusses future work.

II. MODEL ARCHITECTURE

The purpose of our model is to forecast and approximate dynamical systems in a given site of observed data. Fig. 1 shows the structure of our model. Firstly, the observed data is fed into the convolutional layer to extract the abstract features and then the output is fed into a rectified linear units (ReLU) function to improve the representation ability of neural network. The ReLU function rectifies the values of the inputs less than zero thereby forcing them to zero and eliminating the vanishing gradient problem observed in the earlier types of activation function, such as the Sigmoid and tanh activation function. On the other hand, the ReLU represents a nearly linear function, which is easier to optimize with

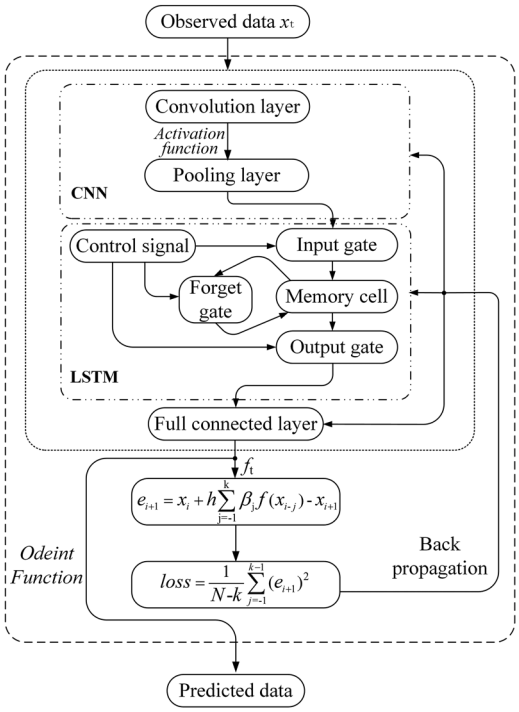

FIG. 1. The framework of the multi-step CLDNN.

gradient-descent methods.[18] We pass the output of ReLU function to pooling layer to reduce the size of the matrix. Next, the output of pooling layer is fed into LSTM layer to learn the long-range dependencies of time series data. Finally, the output of LSTM layer is fed into one fully connected layer to learn the evolution of dynamics $f(x(t))$. However, the $f(x(t))$ is not exact forecast and therefore need a continuous process of optimization and training. Actually the odeint model is the function of scipy library to solve numerically differential equation. We utilize odeint model to solve the evolution of dynamics f(x(t)) and indirectly get the predicted data. In particular, we use the linear multi-step method from numerical analysis to obtain local truncation error, and its mean squared function is used as loss function. Through the back propagation, the weighting parameters are updated to make the predictions of dynamical systems more accurate. Furthermore, by changing the parameters setting, we also explore the most accurate results according to different linear multi-step schemes.

Next, we explain in detail how to utilize the multi-step scheme to obtain loss function. Firstly, we consider the form of nonlinear structural dynamics.

$$\frac{dx(t)}{dt} = f(x(t)) \tag{1}$$

The vector $x(t) \in R^N$ represents the state of dynamical systems at time t and the function $f(x(t))$ denotes the dynamical constraints that define the evolution of the system, such as Newton's second law. Usually, the vector $x(t)$ is contaminated with noise which makes the Eq. 2 do not hold exactly. Instead, the noisy dynamics can be expressed as:

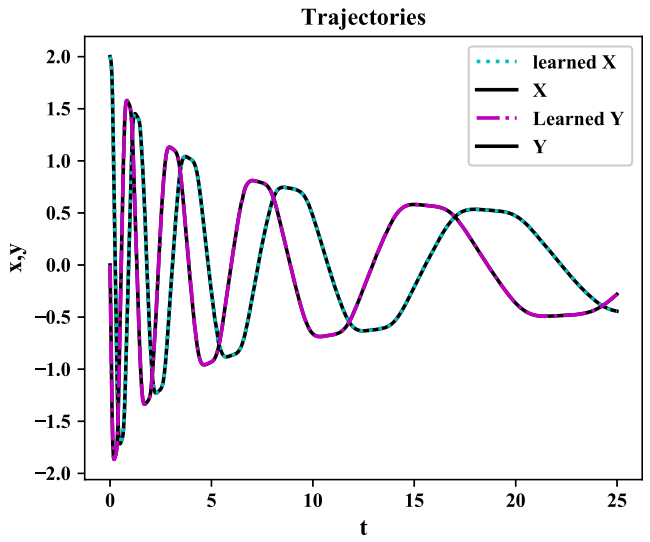

FIG. 2. The black solid line represents the exact dynamics and the dotted line denotes the predicted dynamics. Here the noise strength is 0.

$$\frac{d(\tilde{x}(t))}{dt} = \frac{d(x(t) + T \cdot \lambda)}{dt} = f(x(t)) \tag{2}$$

where T is modeled as a matrix of independent identically distributed Gaussian entries with zero mean and the λ denotes noise intensity.[8] Our goal is to determine the derivative $f(x(t))$ when the system is mingled with noise at several time $t_1, t_2, \ldots, t_N$, and seek the form of underlying dynamical system. We conclude the general form of linear multi-step method with K steps as follows:[19,20]

$$x_i = \sum_{j=0}^{k} \alpha_j x_{i-j} + \Delta t \sum_{j=-1}^{k} \beta_j f(x_{i-j}),\ i = k, \ldots, N-1 \tag{3}$$

The x_{i-j} represents the state of the dynamical system $x(t_{i-j})$ at time t_{i-j}. Note that different choices of the coefficient α_{i-j} and β_{i-j} mean different multi-step methods, such as Simpson method, Hamming method, and Adams method, etc. In this paper, we utilize the Adams method including Adams-Bashforth (AB) methods, Adams-Moulton (AM) methods, and Backward differentiation formula (BDF), as shown in Eq. 4:

$$x_{i+1} = x_i + \Delta t \sum_{j=-1}^{k} \beta_j f(x_{i-j}),\ i = k, \ldots, N-1 \tag{4}$$

Here, we are interested in truncation error directly from the equivalent change of Eq. 4.

$$e_{i+1} = x_i + \Delta t \sum_{j=-1}^{k} \beta_j f(x_{i-j}) - x_{i+1},\ i = k, \ldots, N-1 \tag{5}$$

The e_{i+1} is the truncation error of the predicted systems. Note that, our goal is to train our network via continually minimizing the mean squared error function derived from the Eq. 5. The cost function is presented as follows:

$$l = \frac{1}{N-k} \cdot \sum_{j=-1}^{k} e_{i+1}^2 \tag{6}$$

III. EXPERIMENT AND COMPARISON

A. Two-dimensional damped oscillator

We choose the two-dimensional damped harmonic oscillator as our first experimental object to compare the accuracy and robustness of low-dimensional dynamical systems between our method and the multi-step DNN. The dynamical equation are shown in Eqs. 7:

$$\begin{cases} \dot{x} = -0.1x^3 + 2.0y^3, \\ \dot{y} = -2.0x^3 - 0.1y^3. \end{cases} \tag{7}$$

Firstly we set the initial condition $[x_0, y_0]^T = [2, 0]$ and collect the data from $t = 0$ to $t = 25$ with a time-step $\Delta t = 0.01$. We use one convolution layer to extract the features of dynamics from the states (x, y), each with 200 kernels of size 1*3. As for pooling strategy we use max-pooling with size =1*2. Then the above output is fed into LSTM layer, where the LSTM layer has 256 hidden units. Finally, the output of the LSTM layer is fed into one fully connected

TABLE I. Comparison between the different linear multi-step scheme with different number of steps K. Here, the multi-step DNN's data in Table I are from Ref. 16.

Approach	Axis	Scheme	K=1	K=2	K=3	K=4	K=5
Our	X	AM	3.6e-03	6.5e-03	8.4e-03	6.0e-03	8.1e-03
	X	AB	1.3e+00	2.7e-02	3.0e-02	1.5e-02	1.1e-02
	X	BDF	1.2e+00	9.3e-03	8.7e-03	1.5e-02	1.9e-02
	Y	AM	2.9e-03	6.1e-03	8.1e-03	5.5e-03	7.4e-03
	Y	AB	1.3e+00	2.5e-02	2.7e-02	1.5e-02	1.0e-02
	Y	BDF	1.2e+00	7.5e-03	8.3e-03	1.4e-02	1.5e-02
Multi-step DNN	X	AM	8.8e-03	1.2e-02	1.6e-02	6.3e-03	1.1e-02
	X	AB	1.5e+00	3.1e-02	1.2e-01	4.3e-02	1.2e-02
	X	BDF	1.3e+00	8.8e-03	1.3e-02	1.4e-02	1.7e-02
	Y	AM	8.8e-03	1.0e-02	1.6e-02	5.8e-03	1.1e-02
	Y	AB	1.5e+00	3.0e-02	9.7e-02	3.5e-02	1.2e-02
	Y	BDF	1.3e+00	8.6e-03	9.9e-03	1.4e-02	1.5e-02

TABLE II. The optimum strategy comparison between the multi-step DNN and multi-step CLDNN.

	Optimum scheme	$\triangle t$	K	$l_2(x)$	$l_2(y)$
Multi-step DNN	AM	0.01	4	6.3e-03	5.8e-03
Ours	AM	0.01	1	3.6e-03	2.9e-03

TABLE III. The l_2 error between the predicted dynamics and the exact dynamics for the state (x, y) for different number of units and filters. Here, the noise strength is 0 and the default time step $\triangle t$ is 0.01.

Hidden size	Axis	Filter number		
		64	128	256
32	X	4.9e-03	2.3e-03	9.8e-04
128	X	1.9e-02	5.2e-02	4.8e-02
256	X	8.2e-02	1.0e-01	4.7e-01
32	Y	5.6e-03	2.3e-03	8.7e-04
128	Y	1.4e-02	4.7e-03	4.2e-02
256	Y	7.3e-02	9.5e-02	4.2e-03

layer with 2 neural units to predict the evolution of the systems. In particular, for the multi-step schemes, we utilize Adams-Moulton with K=1 step. Note that all experiments were conducted on a workstation with CPU AMD 2600 Six-Core processor, 12 GB memory, and a NVIDIA GTX1060 GPU with 5GB memory. The Fig. 2 illustrates the feasibility of our approaches via comparing the exact and predicted trajectories of the system.

In order to determine the optimum strategy, we choose the different linear multi-step schemes with different number of steps K including AB methods, AM methods, and BDF method, as shown in Table I. By computing the l_2 error between the exact dynamics and the learned dynamics, we seek the optimum strategy when the linear multi-step scheme is AM with step K=1. We compared the experimental results of our model with the multi-step DNN according to the performance of optimum strategy, as shown in Table II. We can clearly see that our optimum strategy outperforms the optimum strategy of the multi-step DNN in the accuracy aspect.

We change the number of filters and hidden states to explore the network parameters' influence on the model, as shown in Table III. According to experimental results, the increase of filter numbers can improve the accuracy owing to extracting more abstract features and exploring more details affecting the trajectories of dynamics. On the other hand, too many hidden units cause overfitting and make the predictive performance decrease. We also test the robustness of our method under different noise strength, as shown in Fig. 3(a) and 3(b). According to the experimental results, the performance of the multi-step CLDNN exceeds that of the multi-step DNN and is more accurate under different noise strength.

B. Chaotic Lorenz system

In order to test the multi-step CLDNN model in the applicability of chaotic dynamical systems, we choose to employ Chaotic

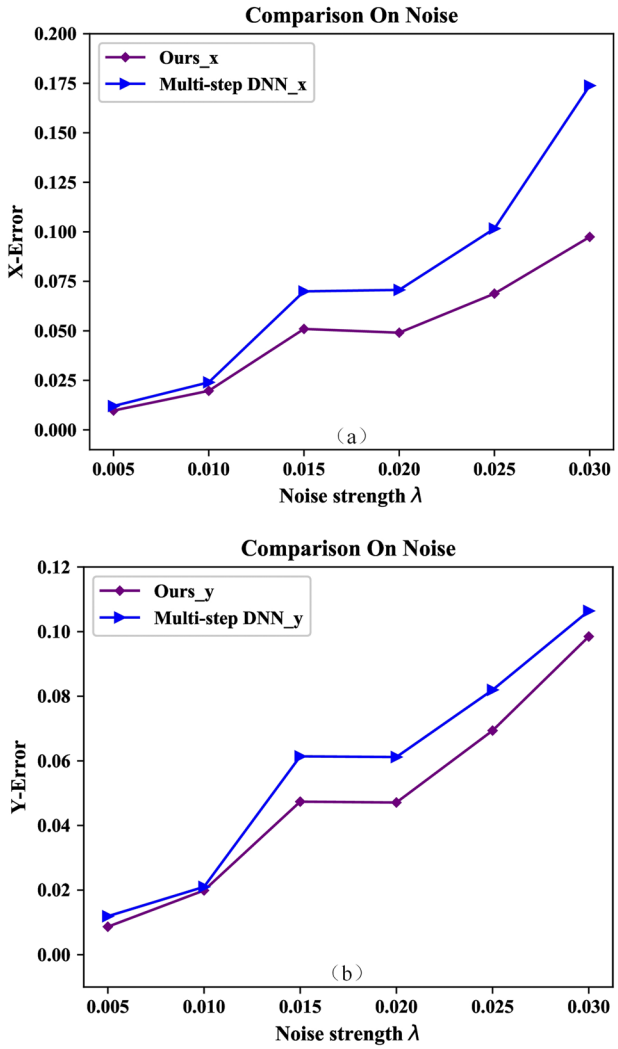

FIG. 3. The blue lines represent the trend of the multi-step DNN'sl2error and the purple lines denote the trend of multi-step CLDNN's l_2 error under different noise strength. The l_2 error of x, y axis are shown in Fig. 3(a) and (b).

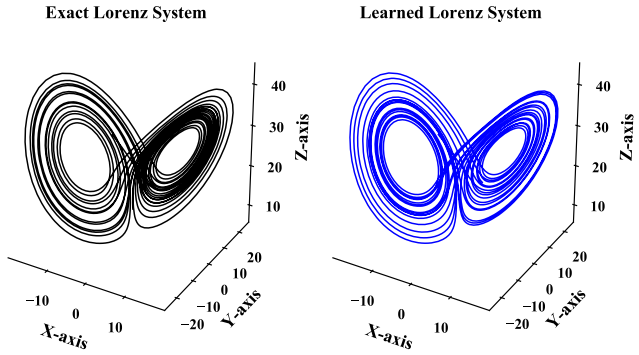

FIG. 4. The left black solid line represents the exact dynamics and the right blue line represents the predicted dynamics. Note that the noise strength is 0.

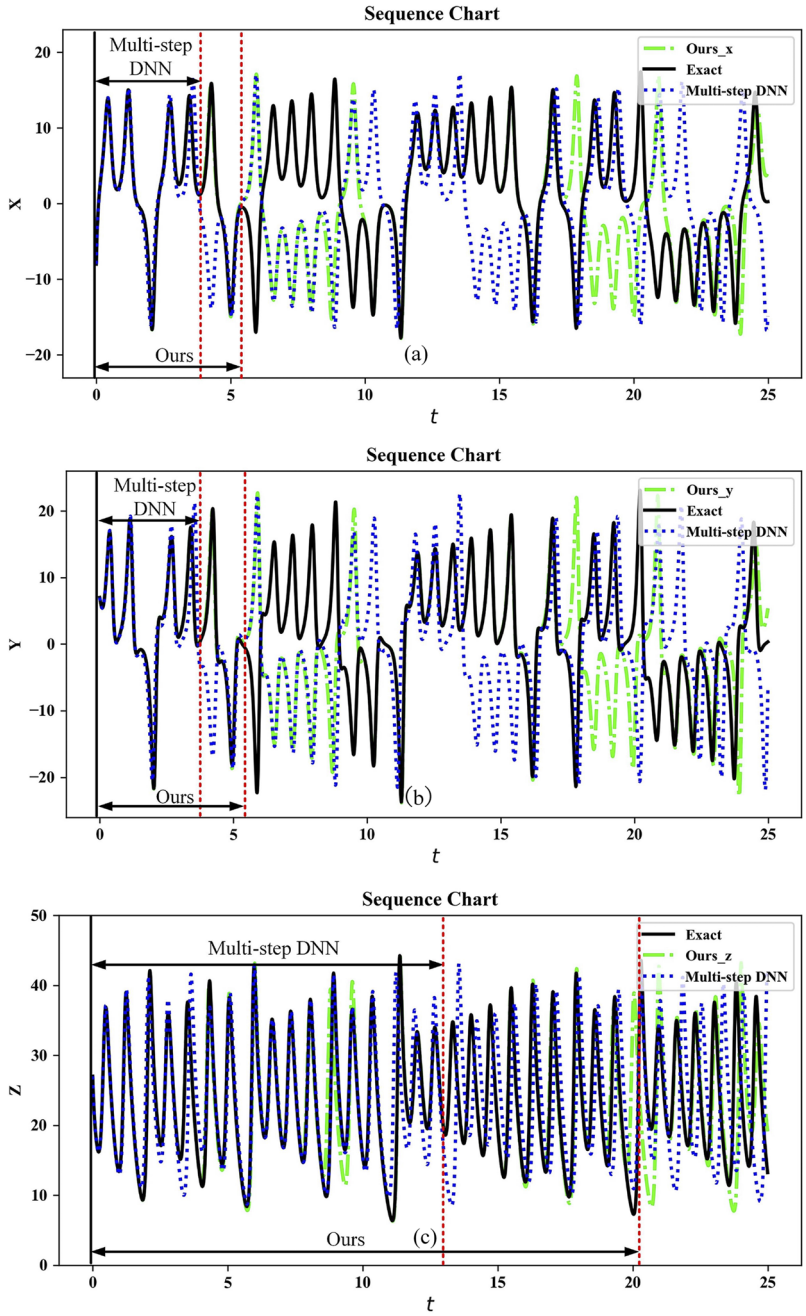

FIG. 5. The solid black lines represent the exact Lorenz systems trajectories and the dotted lime lines represent the trajectories of the multi-step CLDNN, and then the dotted blue lines denote the multi-step DNN's. The abscissas denotes time *t* and the ordinate denotes (x, y, z) state from top to bottom, as shown in Fig. 5(a), (b) and (c). Note that the data from the parameters setting of Ref. 16.

Lorenz system. Furthermore, the dynamics is fairly sensitive to the initial condition. The equation of Lorenz system[16,21] is expressed as:

$$\begin{cases} \dot{x} = 10(y - x), \\ \dot{y} = x(28 - z) - y, \\ \dot{z} = xy - \left(\frac{8}{3}\right)z. \end{cases} \tag{8}$$

Firstly we set the initial condition $[x_0, y_0, z_0]^T = [-8, 7, 27]^T$ and collect the data from $t = 0$ to $t = 25$ with a time-step $\Delta t = 0.01$. We utilize one CNN layer with 1D convolution kernel to extract the abstract features of dynamical systems from the observed data (x, y, z). On the other hand, we use max-pooling layer with size equal to 2 to reduce the size of matrix. Specifically, for the convolution layer, convolution kernel sizes is 3 and the filter numbers are 200. Then the output of CNN layer is fed into LSTM layer, where the LSTM layer has 256 hidden units. Finally, the output of LSTM layer is fed into one fully connected layer with 3 neural units to predict the evolution of systems. For the multi-step schemes, we utilize Adams-Moulton with K=1 step. The comparison

TABLE IV. The comparison of the l_2 error between the multi-step CLDNN and multi-step DNN. Note that the training data is noiseless and the experimental data of multi-step DNN is derived from the parameters setting of Ref. 16.

Approach	X-axis	Y-axis	Z-axis
multi-step DNN	1.40	1.43	3.54
Ours	1.14	1.15	2.00

between the exact dynamics and the learned dynamics are shown in Fig. 4. The multi-step CLDNN can learn the chaotic Lorenz system and accurately capture the form of the finite dimensional attractor.

Although the bi-stable structure of attractor is well learned at the very start, the small deviation of parameters may be accumulated and enlarged so that the subsequent trajectories hardly approximate the exact trajectories. The comparison between the exact trajectories, multi-step DNN predicted trajectories and our predicted trajectories are shown in Fig. 5. According to experimental results, in contrast to multi-step DNN, our approach can predict further trajectories of chaotic Lorenz system. Fig. 5(a) shows that our method can precisely forecast the time-series before $t < 5.2$ compared to the multi-step DNN' s time distance $t < 4$. The sequence diagram in y-axis and z-axis, Fig. 5(b) and Fig. 5(c), also demonstrate the capability of our method for reliably predict the dynamics compared to multi-step DNN. Table IV shows the l_2 error between the exact trajectories and the learned trajectories for the multi-step CLDNN and the multi-step DNN.

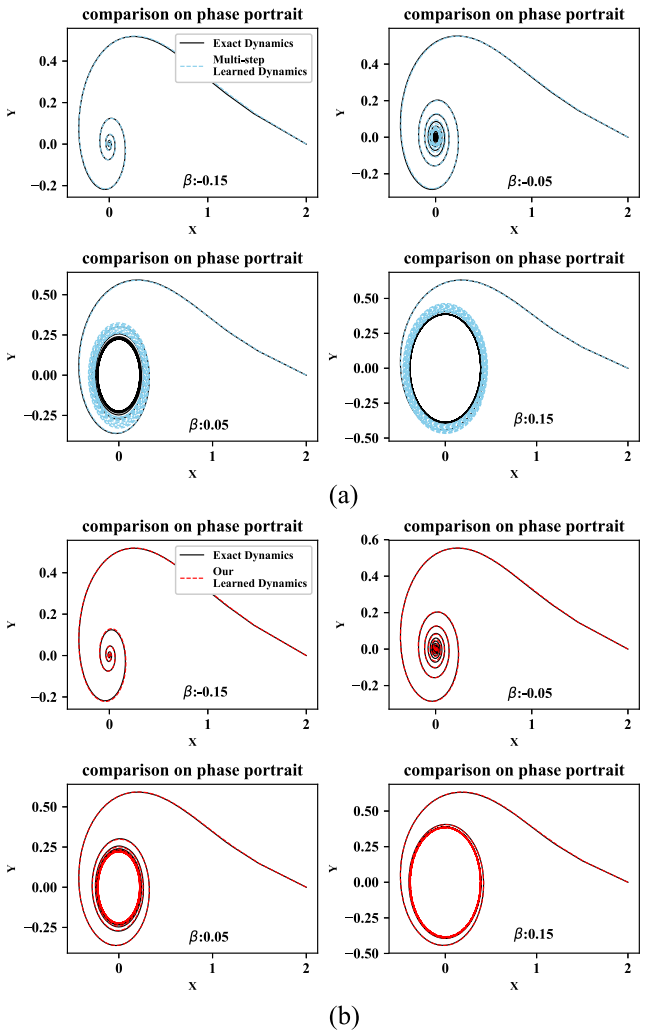

FIG. 6. The Supercritical Poincar'e–Andronov–Hopf bifurcation. The Fig. 6(a) represent our experimental results and the Fig. 6(b) represent the multi-step DNN's. Each subgraph has different parameters β in the Fig. 6(a) and the Fig. 6(b).

TABLE V. The Supercritical Poincar'e–Andronov–Hopf bifurcation: the l_2 error between the exact dynamics and the learned dynamics for the experiment.

	β	x	y
Multi-step DNN	−0.15	2.2e-02	2.3e-02
	−0.05	2.0e-01	1.4e-01
	0.05	2.0e-01	2.4e-01
	0.15	1.1e-01	1.2e-01
Ours	−0.15	1.4e-02	1.8e-02
	−0.05	5.6e-02	7.5e-02
	0.05	5.1e-02	5.6e-02
	0.15	1.8e-02	1.8e-02

C. Hopf bifurcation

For chaotic dynamics, the Hopf bifurcation occurs at a critical point, and the systems switch from stable state to unstable state with a limit cycle pictured as a closed curve in phase space. In the last experiment, we choose the Supercritical Poincar'e–Andronov–Hopf bifurcation[22] to illustrate the performance with respect to identifying parameterized dynamics, and the form of the system is shown in Eq. 9:

$$\begin{cases} \dot{x} = \beta x - y - x\left(x^2 + y^2\right), \\ \dot{y} = x + \beta y - y\left(x^2 + y^2\right). \end{cases} \tag{9}$$

We set the initial condition $[x_0, y_0]^T = [2, 0]$ and collect the data from t=0 to t=75 with a time-step $\Delta t = 0.01$. We use one convolution layer to extract the abstract features of dynamics, each with 200 kernels of size 1*3. As for pooling strategy we use max-pooling with size =1*2 to reduce the size of matrix. Then the above output is fed into LSTM layer, where the LSTM layer has 256 hidden units. Finally, the output of LSTM layer is fed into one fully connected layer with 2 hidden units to predict the evolution of systems. Specially, for the multi-step schemes, we utilize Adams-Moulton with K=1 step. The Fig. 6 shows that our model can more precisely capture the attractor than the multi-step DNN when the parameter $\beta < 0$. On the other hand, the Fig. 6(a) shows that multi-step DNN does not predict the limit cycle as an asymptotically stable state very well, whereas our approach can precisely the limit cycle of zero amplitude when the parameter $\beta > 0$, as shown in Fig. 6(b). Specific data is shown in Table V.

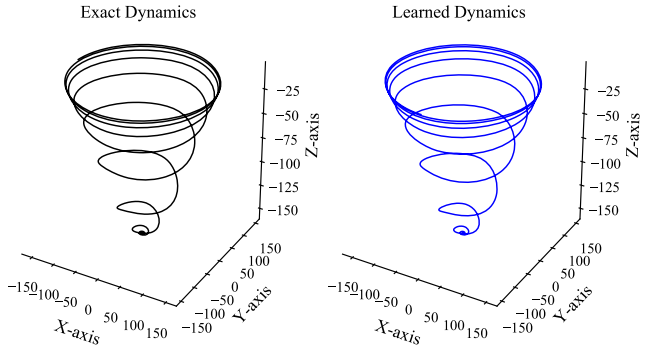

FIG. 7. The left black solid line represents the exact dynamics and the right blue line represents the predicted dynamics. Note that the noise strength is 0.

D. Fluid flow behind a cylinder

In the experiment, studies are conducted on the dynamics of fluid flow behind a cylinder using the same experiment setup in the Ref. 16. As for the network parameter setting, we utilize one CNN layer with 1D convolution kernel to extract the abstract features of dynamical systems from the proper orthogonal decomposition coefficients (x, y, z) with a time step of $\Delta t = 0.02$. On the other hand, we use max-pooling layer with size equal to 2 to reduce the size of matrix. Specifically, for the convolution layer, convolution kernel sizes is 3 and the filter numbers are 128. Then the output of CNN layer is fed into LSTM layer, where the LSTM layer has 256 hidden units. Finally, the output of LSTM layer is fed into one fully connected layer with 3 neural units to predict the evolution of systems. For the multi-step scheme, we utilize Adams-Moulton with K=1 step. The Fig. 7 shows that our method can not only capture the form of the dynamics but also accurately predict the transient dynamics connecting the unstable steady state with the mean of the limit cycle. The specific l_2 error for the multi-step DNN and multi-step CLDNN are shown in Table VI.

Actually, for all experiment A, B, C and D, the full connected layer (FC) can be used to represent and predict f. In experiment B and C, the prediction performance is especially shown for chaotic and bifurcation dynamics. The experiment results are tested in several benchmark systems. For accuracy, the comparisons between two methods are made and the possible improvements are measured. See Table II: (x, y):(%42.8, %50), Table IV: (x, y, z): (%18.6, %19.6, %43.5), and Table V:(%36.4, %21.7). The improvement should be significant in the machine learning fields. On the other hand, an obvious improvement of prediction performance can be seen in chaotic dynamics, due to the use of the LSTM layers.

TABLE VI. The comparison of the l_2 error between the multi-step CLDNN and multi-step DNN. Note that the training data is noiseless and the experimental data of multi-step DNN is derived from the parameters setting of Ref. 16.

Approach	X-axis	Y-axis	Z-axis
multi-step DNN	6.8e-01	6.8e-01	4.2e-02
Ours	3.4e-01	3.4e-01	6.5e-03

IV. CONCLUSION

In the paper, we have proposed a novel model to predict the nonlinear dynamical systems from time-series data. The proposed model combines CNN, LSTM, DNN and classical multi-step method derived from numerical analysis to learn dynamical systems. In the model, we use CNN to automatically extract abstract features and LSTM to model temporal dependencies, and then DNN to map the function f representing the evolution of dynamical systems. A key step of the proposed model is to use the multi-step method to obtain cost function for training the CLDNN. We compare our model with the multi-step DNN model and find that our model not only can perform better than multi-step DNN in dynamics prediction, particularly chaotic dynamics, but also can achieve excellent results for identifying parameterized dynamics. The multi-step CLDNN avoids choosing appropriate basis function and approximating the temporal gradients in a given set of discretized time. Because one LSTM layer is added, the model has better capability to capture temporal dependencies from time series data. In the future work, we will design an attention mechanism to focus on interesting parts of the observed data. In addition, this method can reduce the negative influence of noise for nonlinear system identification.

ACKNOWLEDGMENTS

The work was supported by the National Science Foundation of China (No. 61203237) and the Industry-Academia Cooperation Innovation Fund Projection of Jiangsu Province (BY2016001-02).

REFERENCES

[1] E. Ott, *Chaos in dynamical systems* (Cambridge university press, 2002).
[2] F. Michor, Y. Iwasa, and M. A. Nowak, "Dynamics of cancer progression," Nature Reviews Cancer **4**, 197 (2004).
[3] R. A. Ajayi and M. Mougoué, "On the dynamic relation between stock prices and exchange rates," Journal of Financial Research **19**, 193–207 (1996).
[4] C. Castellano, S. Fortunato, and V. Loreto, "Statistical physics of social dynamics," Reviews of Modern Physics **81**, 591 (2009).
[5] K. J. Åström and P. Eykhoff, "System identification—A survey," Automatica **7**, 123–162 (1971).
[6] R. Pintelon and J. Schoukens, *System identification: A frequency domain approach* (John Wiley & Sons, 2012).
[7] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," Science **324**, 81–85 (2009).
[8] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," Proceedings of the National Academy of Sciences **113**, 3932–3937 (2016).
[9] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," Science Advances **3**, e1602614 (2017).
[10] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, "Model selection for dynamical systems via sparse regression and information criteria," Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **473**, 20170009 (2017).
[11] S. A. Billings, *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains* (John Wiley & Sons, 2013).
[12] A. Cochocki and R. Unbehauen, *Neural networks for optimization and signal processing* (John Wiley & Sons, Inc., 1993).
[13] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," IEEE Transactions on Neural Networks **1**, 4–27 (1990).

[14]K. Kristinsson and G. A. Dumont, "System identification and control using genetic algorithms," IEEE Transactions on Systems, Man, and Cybernetics **22**, 1033–1046 (1992).
[15]D. Giannakis and A. J. Majda, "Nonlinear Laplacian spectral analysis for time series with intermittency and low-frequency variability," Proceedings of the National Academy of Sciences **109**, 2222–2227 (2012).
[16]M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Multistep neural networks for data-driven discovery of nonlinear dynamical systems," arXiv preprint arXiv:1801.01236 (2018).
[17]T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE, 2015), pp. 4580–4584.
[18]C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," arXiv preprint arXiv:1811.03378 (2018).
[19]S. Gottlieb, "On high order strong stability preserving Runge-Kutta and multi step time discretizations," Journal of Scientific Computing **25**, 105–128 (2005).
[20]T. Simos, "Optimizing a class of linear multi-step methods for the approximate solution of the radial Schrödinger equation and related problems with respect to phase-lag," Open Physics **9**, 1518–1535 (2011).
[21]T. Yang, L.-B. Yang, and C.-M. Yang, "Impulsive control of Lorenz system," Physica D: Nonlinear Phenomena **110**, 18–24 (1997).
[22]J. E. Marsden and M. McCracken, *The Hopf bifurcation and its applications* (Springer Science & Business Media, 2012), Vol. 19.