AIP Advances

Data driven governing equations approximations using attention based multistep neural networks

Cite as: AIP Advances **10**, 065227 (2020); https://doi.org/10.1063/5.0015600
Submitted: 29 May 2020 . Accepted: 30 May 2020 . Published Online: 19 June 2020

Qi Teng, Jiaqi Wang, Zeyu Ding, Lei Zhang, and Zhenyu Wang

COLLECTIONS

Paper published as part of the special topic on Chemical Physics, Energy, Fluids and Plasmas, Materials Science and Mathematical Physics

View Online Export Citation CrossMark

ARTICLES YOU MAY BE INTERESTED IN

Data driven nonlinear dynamical systems identification using multi-step CLDNN
AIP Advances **9**, 085311 (2019); https://doi.org/10.1063/1.5100558

Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensor observation data
Physics of Fluids **32**, 015113 (2020); https://doi.org/10.1063/1.5136351

Learning the tangent space of dynamical instabilities from data
Chaos: An Interdisciplinary Journal of Nonlinear Science **29**, 113120 (2019); https://doi.org/10.1063/1.5120830

Data driven governing equations approximations using attention based multistep neural networks

Cite as: AIP Advances 10, 065227 (2020); doi: 10.1063/5.0015600
Submitted: 29 May 2020 • Accepted: 30 May 2020 •
Published Online: 19 June 2020

Qi Teng,[1] Jiaqi Wang,[1] Zeyu Ding,[1] Lei Zhang,[1,2,a)] and Zhenyu Wang[1]

AFFILIATIONS

[1] School of Electrical and Automation Engineering, Nanjing Normal University, Nanjing 210046, People's Republic of China
[2] The Institute of Information and Control Engineering Technology, Nanjing Normal University, Nanjing 210046, People's Republic of China

[a)] Author to whom correspondence should be addressed: leizhang@njnu.edu.cn

ABSTRACT

During the past decade, deep learning has represented the biggest research trend in the field of machine learning, which provides new powerful tools to interrogate high dimensional time series data in a way that has not been possible before. With the recent success in natural language processing, one would expect widespread adaptation to problems like time series forecasting and classification. After all, both involve processing sequential data. However, to this point, research on their adaptation to time series problems has remained limited. Recently, a multi-step time-stepping scheme without the need of direct access to temporal gradients has been proposed, which can accurately identify nonlinear dynamical systems from time series data. In this paper, we combined an attention mechanism with a deep model in a multi-step time-stepping scheme to perform nonlinear system identification and forecasting tasks. To our knowledge, this is the first paper to use attention based models to deal with nonlinear system identification and forecasting problems. The attention weights on rows select those variables that are helpful for forecasting, which can enhance the information across multiple time steps to capture temporal information. The experiment results indicate that the attention based model in a multi-step time-stepping scheme has better identification and prediction performance for nonlinear time series identification and forecasting problems.

I. INTRODUCTION

The physical laws extracted from experimental data play a critical role in many science and engineering applications, and time series data collected from experiments often obey unknown governing equations. From the fluid flow behind a cylinder to the movement of the pendulum under the influence of gravity, the mathematical models of dynamics derived from the observed data have yielded a set of methods that aim to analyze the current state of system depending on the past and to forecast the possible state in the future. The methods are usually in the form of differential or partial differential equations originating from the classical first physical principles derivations. However, in most cases, we do not have any prior knowledge about the underlying physical processes or the system is too complex, which makes the first principles approach infeasible. In order to identify accurate expressions for the dynamics[1] in the form of differential equations, there has been a substantial research effort toward improving the process of the data driven model discovery.

It has been a challenge to reconstruct a nonlinear dynamical system[2] from time series data without any prior knowledge. Such "model free" approaches for the identification and prediction of dynamical systems mainly include white-box and black-box techniques.[3,4] Recently, there has been a growing interest in white-box approaches (see, e.g., symbolic regression,[5,6] heterogeneous multiscale method,[7] and sparse regression[8,9]). Although the above white-box methods can return more interpretable models that uncover the over-all parametric form of the potential governing equation, these methods inevitably require an appropriate basis function to obtain sparse representation for dynamics. More research efforts were devoted to black-box (see, e.g., NARMAX,[10] neural networks,[11,12] and genetic algorithms[13,14]) approaches that avoid the choice of the basis function to perform the system identification tasks of directly observed data.

System identification using the deep learning method has emerged as a viable alternative for traditional black-box approaches. Recently, Raissi *et al.* blended the classical tools from numerical analysis, namely multi-step time-stepping deep neutral networks, to propose a novel multi-step deep neural network (multi-step DNN) approach.[15] Due to the discretized time derivatives using the classical time-stepping rules, the multi-step DNN approach does not require direct access or approximations to temporal gradients avoiding the choice of the appropriate basis function. On the other hand, attention models have recently powered significant progress in natural language processing (NLP).[16] With this recent success in NLP, one would expect widespread adaptation to problems like time series forecasting and classification. After all, both are involved in processing sequential data. However, research on the adaptation of attention to time series problems has remained limited. Deep models with attention provide a compelling alternative to address the time series problems. To be specific, the attention weights tend to select those variables that are most helpful for forecasting.

In this paper, inspired by this notion, attention approaches are introduced into the system identification problems based on the aforementioned multi-step time-stepping schemes. Embedding directly an attention mechanism into the multi-step DNN is hard because the computation of the local feature has to depend on the convolutional layers. To solve the problem, we embed the attention mechanism into a classical CLDNN (convolution neural network, long short term memory, deep neural network) framework[17] that includes convolution neural network (CNN), long short-term memory (LSTM), and DNN layers. There are two reasons to choose the CLDNN frameworks: First, computing the local features from the data has to depend on the CNN layers; second, both CNNs and LSTM have shown improvements over DNNs across a wide variety of deep learning tasks. CNNs, LSTMs, and DNNs are complementary in their modeling capabilities, as CNNs are good at reducing frequency variations and extracting local features, LSTMs are good at temporal modeling, and DNNs are appropriate for mapping global features. The main contribution of our works is embedding the attention mechanism into a classical CLDNN framework in multi-step time-stepping schemes. To the best of our knowledge, this is the first paper to leverage the attention mechanism to deal with system identification using the deep learning method. By comparing multi-step DNN, multi-step, and attention-based multi-step CLDNN, we validate the improvement induced by the attention mechanism in several benchmark experiments.

The remainder of this manuscript is structured as follows: In Sec. II, we propose our model architecture and explain how to combine the attention module and CLDNN frame in the multi-step time-stepping schemes. In Sec. III, the comparisons between multi-step DNN, multi-step, and attention-based multi-step CLDNN are made in several benchmark experiments consisting of the chaotic Lorenz system, the Rossler system, and the Hopf bifurcation. Section IV summaries and draws conclusions.

II. MODEL ARCHITECTURE

Without the loss of generality, we consider the nonlinear dynamical system of the form

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

in which $x(t) \in R^D$ are the state variables that denote the state of the system at time t and the function f describes the evolution of the system. The main aim of our model is to identify and predict underlying dynamical systems (1) from the given measurements of the state $x(t)$ of the system at several time instances t_1, t_2, . . ., t_N. The model consists of fundamental CLDNN pipelines and the attention submodule, as shown in Fig. 1. The traditional deep learning methods tend to learn equally from the whole observed data. The advantage of our attention mechanism is that it can identify salient data parts and enhance their influence, while suppressing the unimportant information. The parameters of this neural network can be learnt according to the multi-step time-stepping schemes. To embed the attention mechanism, the CLDNN pipeline with CNNs, LSTMs, and DNNs is introduced for replacing the DNN pipeline in the aforementioned multi-step DNN. We first discuss the method of attention by the computation of the compatibility function, and then conclude the modification to the network architecture and training method in multi-step time-stepping schemes.

A. Attention module

To focus on the interesting parts from the observed data, the main task of the attention module is to implement a compatibility computation between the local feature vectors extracted from the third, fourth, and fifth convolutional layers and the global feature vectors at the end of the pipeline. Specifically, the attention module contains the compatibility function, softmax function, and weighting function. First, we utilize the compatibility function to implement the dot product between g and l_i^c as a measure of their compatibility,

$$c_i^c = \langle l_i^c, g \rangle,\ i \in \{1, \ldots, n\}. \tag{2}$$

According to Eq. (1), we can obtain a special matrix, which records the compatibility score $C(l^c, g) = \{c_1, c_2, \ldots, c_n\}$. Actually, the relative magnitude of the scores at the corresponding matrix position indicates the contribution value that predicts the evolution of the dynamical systems, whereafter, the compatibility scores are

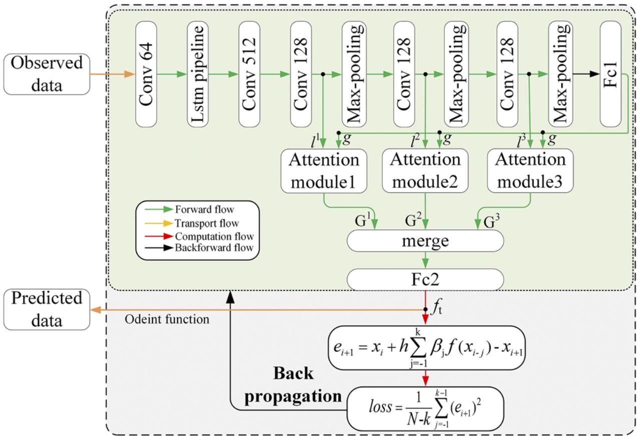

FIG. 1. The framework model.

normalized by a softmax function as follows:

$$a_i^c = \frac{\exp(c_i^c)}{\sum_j^n \exp\left(c_j^c\right)}. \tag{3}$$

The normalized compatibility scores $A^c = \{a_1^c, a_2^c, \ldots, a_n^c\}$ are then fed into the weighting function to produce a single vector G^c for each layer c by element-wise weighted averaging,

$$g^c = \sum_{i=1}^{n} a_i^c \cdot l_i^c. \tag{4}$$

In particular, we obtain a set of bran-new global vectors g^c to replace traditional global vectors, and then the global vectors g^c are merged into one vector to obtain the comprehensive data compared to the single updated feature data G^c. Finally, the full connected (FC) layer mainly plays a role in mapping the shape of the observed data.

B. Fundamental CLDNN

As shown in Fig. 1, the fundamental model that the attention submodules are based on is a deep CLDNN that consists of convolutional, LSTM, and fully connected layers. First, the observed data are fed into the convolutional pipeline with $1 * 1$ kernel size to increase the dimension of the input space, and then, the output is fed into a rectified linear units (ReLU) function to enhance the representation capability of the neural network. In particular, for most of the CNN layers, the CNN is used to reduce the frequency variation and extract the local feature. In the $1 * 1$ CNN layer, the CNN is used to increase the output dimension of this layer, which aims to match the input dimension of the subsequent LSTM layers (referring to Raissi *et al.*'s method). Actually, the $1 * 1$ Condv has been widely used to increase or decrease the dimension in many other deep learning literature,[18] and then, the output of each layer is transformed using a ReLU activation function. Here, the LSTM is applied right after the $1 * 1$ convolutional layers, which is used to learn the temporal relationship contained in the time series data. Next, the second and third convolution layers are connected to one another without any intervening pooling layer to obtain the high-resolution data, which helps the subsequent convolution layers to extract more desired local feature maps denoted by $l^c = \{l_1^c, l_2^c, \ldots, l_n^c\}$ for a given convolutional layer $c = \{1, 2, 3\}$. The third, fourth, and fifth convolution layers are all followed by pooling layers and the last pooling layer is passed into the FC layer to obtain the global feature vector g. The attention module contains a key compatibility function that can capture pivotal information from the observed data and suppress the unnecessary or misleading information. Next, the above outputs updated feature data G is expressed as $G = \left\{G^1, G^2, G^3\right\}$ to replace the traditional global features g. Then, all G^c vectors are merged into one vector to obtain the comprehensive data compared to the single updated feature data G^c. Next, the above output is fed into the last FC layer to map the function f representing the evolution of dynamical systems. Here, we utilize the linear multi-step method (LMM) from numerical analysis to obtain the local truncation error, and its mean squared function is regarded as the loss function. Finally, back propagation will optimize the weight parameters of the proposed model to make the predictions of dynamical systems more accurate.

C. Loss function acquisition

To learn the system f, let us consider how to leverage the multi-step scheme to obtain the loss function. Referring to the multi-step DNN, the general form of the linear multi-step method with K steps for the Eq. (1) can be given as

$$x_{i+1} = \sum_{j=0}^{k-1} \alpha_j x_{i-j} + \Delta t \sum_{j=-1}^{k-1} \beta_j f(x_{i-j}), \; i = k-1, \ldots, N-1. \tag{5}$$

In fact, different choices for parameters α_j and β_j lead to different linear multi-step method (LMM) schemes. In this paper, according to Eq. (5), our attention model is used to get the evolution of the system f via implicit Euler multi-step schemes, which can solve the evolution of dynamics and get learned the state of the system $x(t_{n-k})$ at time t_{n-k}. Then, we proceed to generate the trajectories of the learned dynamics via using the odeint function in the scientific computational library SciPy. In particular, we can obtain the local truncation error via the implicit trapezoid formula method, which is then optimized via the gradient descent method to train the network. In our case, the trapezoidal formula is used,

$$x_{i+1} = x_i + \frac{1}{2} \cdot \Delta t \cdot (f(x_i) + f(x_{i+1})), \; i = 0, \ldots, N, \tag{6}$$

in which K = 1, $\alpha_0 = -1$, $\alpha_1 = 1$, $\beta_{-1} = 0.5$, and $\beta_0 = 0.5$. Referring to the trapezoidal formula, the truncation error is equivalently transformed into

$$\begin{aligned} e_{i+1} &= x_{i+1} - \left(\sum_{j=0}^{k-1} \alpha_j x_{i-j} + \Delta t \sum_{j=-1}^{k-1} \beta_j f(x_{i-j}) \right), \\ i &= k-1, \ldots, N-1 \end{aligned} \tag{7}$$

in which the e_{i+1} denotes the truncation error of the predicted dynamics. The weight parameters of the proposed attention-based CLDNN can be learned via continually minimizing the mean squared error function as the cost function derived from Eq. (6). The form of the cost function is as follows:

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

III. EXPERIMENT AND COMPARISON

A. Chaotic Lorenz system

We choose the chaotic Lorenz system,[15,19] as our first experiment object to seek the identification of chaotic systems evolving on a finite dimensional attractor and make comparison between the multi-step DNN, the multi-step CLDNN, and our approach concerning the accuracy of the chaotic systems prediction. The equation of the Lorenz system is expressed as

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

Here, we choose $[x_0, y_0, z_0]^T = [-8, 7, 27]^T$ as our initial conditions and collect data from $t = 0$ to $t = 25$ with a time-step

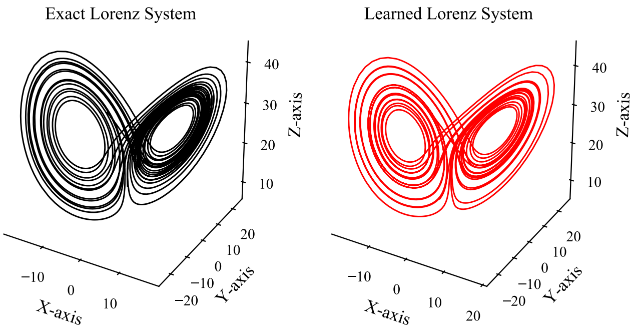

FIG. 2. The black phase portrait on the left represents the exact Lorenz system and the red phase portrait represents the learned dynamics.

Δt = 0.01. In particular, the work platform is based on the server with the central processing unit (CPU) Intel Core i7-6850k Six-Core processor, 64 GB memory, and NVIDIA RTX2080Ti GPU with 11 GB memory. Figure 2 not only shows the ability of the model to accurately capture the attractor of the Lorenz system but also illustrates the feasibility of predicting dynamical trajectories. We make a comparison between trajectories of the exact and the identified systems for different LMM schemes. According to Table I, it can be seen that the Adams–Moulton with k = 1 scheme performs better than the Adams–Bashforth and BDF methods. Similarly, the Adams–Moulton with k = 1 scheme consistently outperforms the Adams–Bashforth and BDF methods in accordance with Refs. 15 and 17. Without the loss of generality, the LMM schemes in this paper are set to Adams–Moulton with k = 1 scheme for all the experiments.

On the other hand, compared with the CLDNN method, there is a significant decrease from 2.23M to 1.41M by the use of our method in terms of FLOPs. To make the comparison more fair, the DNN method is set to the same hidden layers and neurons as our method's ones. It can be seen that the DNN method need more computation cost than ours, accompanied by a performance drop of 43%. Actually, the DNN method has to rely on deeper network layers along with more complicated dynamics due to its limitation in modeling and generalization ability, which leads to a worse prediction performance in the Lorenz system or Rossler system.

TABLE I. Comparison between the different linear multi-step schemes with different numbers of steps K.

	X-axis	Scheme	k = 1	k = 2	k = 3
Attention based CLDNN	X	AM	0.99	1.36	1.39
	X	AB	1.54	1.37	1.41
	X	BDF	1.55	1.37	1.48
	Y	AM	1.02	1.36	1.38
	Y	AB	1.51	1.35	1.39
	Y	BDF	1.53	1.37	1.47
	Z	AM	0.29	0.50	0.49
	Z	AB	0.48	0.47	0.50
	Z	BDF	0.51	0.38	0.41

Although the bi-stable structure of the attractor is well captured, the chaotic Lorenz system is rather sensitive to the initial conditions, and any small deviation may cause the system coefficients to diverge exponentially as the time evolves. In this regard, we compare the predicted trajectories between the multi-step DNN, the multi-step CLDNN, and our approach and then plot the sequence chart, as shown in Fig. 3. For the Rossler system and Hopf bifurcation experiments, our method and the methods used in Refs. 15 and 17 can fit the true trajectory well. To demonstrate the superiority of our method, we use the L_2 error metric, i.e., accuracy to perform comparisons. In the Lorenz experiment, our method can track the true trajectory longer than other methods. Figure 3(a) shows that our approach can accurately approximate the trajectory before t < 8, while the multi-step DNN's and the multi-step CLDNN's forecasted time-series can only attain t = 4 and t = 5.2, respectively. Similarly,

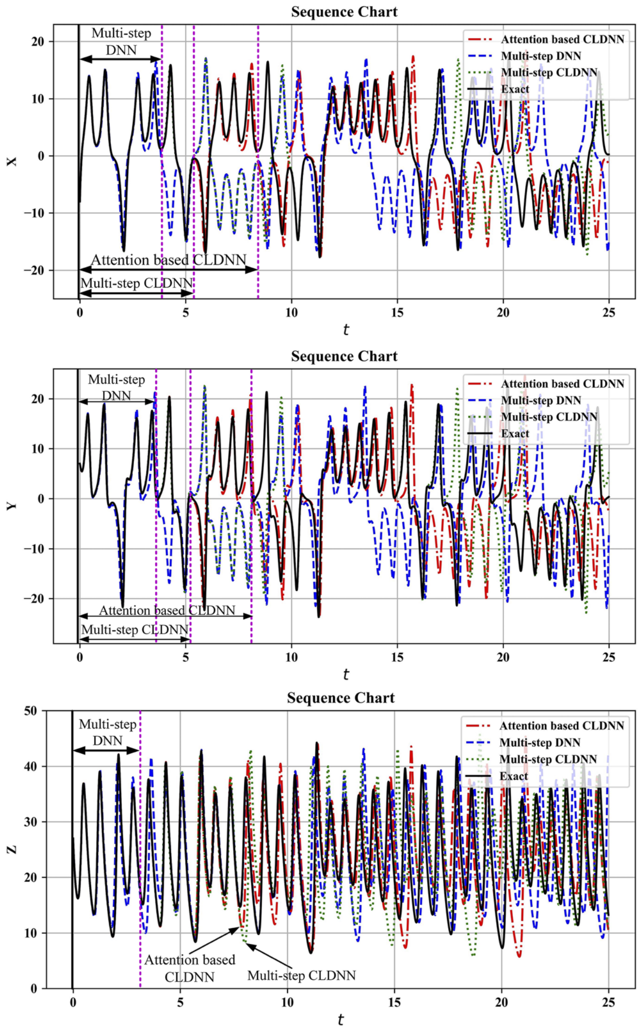

FIG. 3. The comparison of the trajectories about the Lorenz system between the multi-step DNN, the multi-step CLDNN, and our approach. Note that the relevant parameters setting is derived from Refs. 15 and 17, respectively.

TABLE II. The L_2 error between the exact dynamics and the predicted dynamics. Note that the relevant parameters setting for the multi-step DNN and the multi-step CLDNN is derived from Refs. 15 and 17, respectively.

Approach	X-axis	Y-axis	Z-axis	Length	FLOPs
Multi-step DNN	1.40	1.43	3.54	4.0	2.51M
Multi-step CLDNN	1.24	1.24	0.36	5.2	2.23M
Attention based CLDNN	0.99	1.02	0.29	8.0	1.41M

both Figs. 3(b) and 3(c) illustrate that our architecture can more reliably predict the dynamics. According to the experimental results, our method can predict longer trajectories than the multi-step DNN and the multi-step CLDNN. In particular, we make L_2 error analysis between the exact dynamics and the predicted dynamics, as shown in Table II.

B. Rossler system

Although the Rossler system is a simple nonlinear system, it still has a chaotic behavior with a period-doubling bifurcation phenomenon. Furthermore, this Rossler system is fairly sensitive to the initial conditions. The equation of the Rossler system is shown below,

$$\begin{cases} \dot{x} = -y - z, \\ \dot{y} = x + 0.2y, \\ \dot{z} = 0.2 + z(x - 10). \end{cases} \tag{9}$$

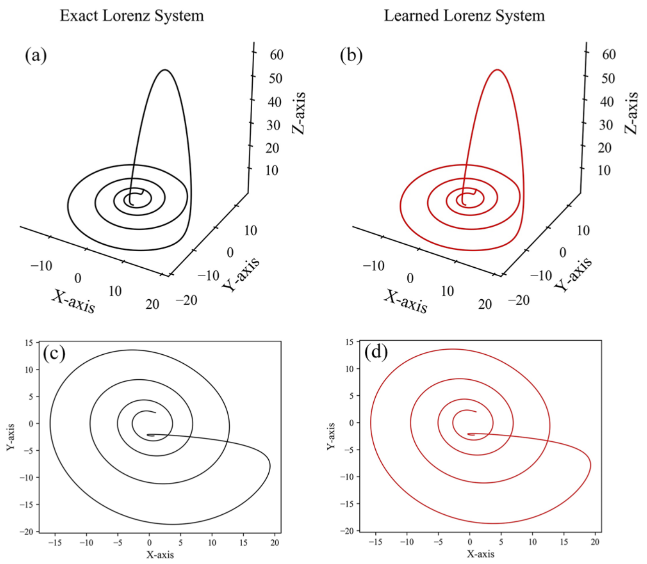

FIG. 4. The Rossler system: The subgraph (a) denotes the exact dynamics and the subgraph (c) is the corresponding x–y planar graph. Similarly, the subgraph (b) denotes our learned dynamics and the subgraph (d) is the corresponding x–y planar graph.

In the experiment, the initial conditions are set to $[x, y, z]^T = [1.0, 2.0, 3.0]^T$ and the dynamics data are collected from $t = 0$ to $t = 25$ with a time-step $\Delta t = 0.01$. For the multi-step scheme, we utilize Adams–Moulton with $K = 1$ step. As depicted in Fig. 4, our learned system can accurately predict the evolution of the dynamical system. When the prediction capability of a model is insufficient, the dynamical system parameters may have deviations that cause the system switch from periodicity to aperiodicity. For the above problems, we compare the predicted trajectories between the multi-step DNN, multi-step CLDNN, and our approach, as shown in Fig. 5. The multi-step CLDNN can accurately identify the trajectories of the Rossler system, and likewise our approach can also approximate the

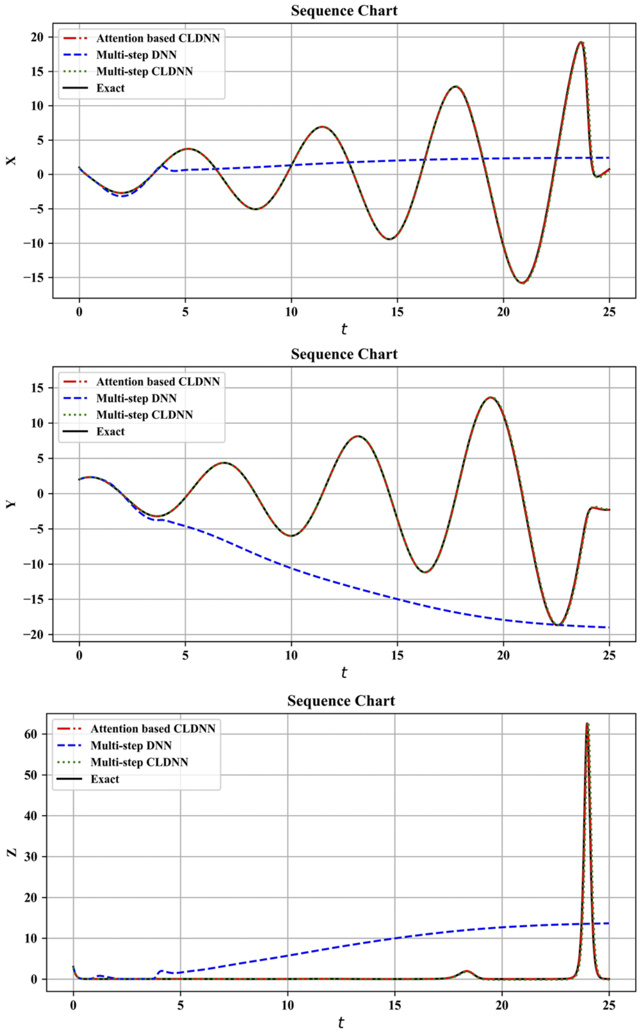

FIG. 5. The comparison of the trajectories about the Rossler system between the multi-step DNN, the multi-step CLDNN, and our approach. The abscissae denote time t and the ordinates represent (*x*, *y*, *z*) states from top to bottom. Note that the relevant parameters setting for the multi-step DNN and the multi-step CLDNN is derived from Refs. 15 and 17, respectively.

TABLE III. The L_2 error between the exact dynamics and the predicted dynamics. Note that the relevant parameters setting for the multi-step DNN and the multi-step CLDNN is derived from Refs. 15 and 17, respectively.

Approach	X-axis	Y-axis	Z-axis
Multi-step DNN	1.02×10	1.90×10	1.52×10
Multi-step CLDNN	5.05×10^{-2}	2.79×10^{-2}	2.40×10^{-1}
Attention based CLDNN	1.56×10^{-2}	9.51×10^{-3}	6.86×10^{-2}

exact trajectories, but the accurately predicted trajectories of multi-step DNN can only be attained at t = 4.2. Specifically, we make L_2 error analysis between the exact dynamics and the predicted dynamics as shown in Table III. Table III clearly shows that our model is superior to multi-step DNN and multi-step CLDNN in terms of prediction accuracy.

C. Hopf bifurcation

In practice, many real-world systems are sensitive to the parameters, which may cause the system state change or bifurcate when the parameter crosses a critical value. In this experiment, we choose the Supercritical Poincar'e–Andronov–Hopf bifurcation to elucidate the capability of identifying parameterized dynamics and the equation is shown below,

$$\begin{cases} \dot{x} = \beta x - y - x(x^2 + y^2), \\ \dot{y} = x + \beta y - y(x^2 + y^2). \end{cases} \tag{10}$$

We use $[x_0, y_0]^T = [2, 0]^T$ as the initial conditions and collect data from $t = 0$ to $t = 75$ with a time-step $\Delta t = 0.01$. For the multi-step scheme, we utilize Adams–Mouton with $K = 1$ step. In particular, different parameter values μ are set to test the capability of our method for identifying parameterized dynamics. Figure 6 clearly depicts that our method can correctly capture the transition from the fixed point for μ to the limit cycle of zero amplitude compared with the multi-step DNN and the multi-step CLDNN. It is noted that our approach can ideally reproduce the limit cycle for $\mu > 0$, while the multi-step DNN can only approximate the limit cycle due to the imprecise parameters of learned dynamics, as seen in Table III.

In fact, the attention mechanism plays a key role in capturing pivotal information from time-series data and concentrating the valuable information for the object dynamics. However, the multi-step DNN and multi-step CLDNN tend to treat all information even the negative information equally and fail to capture the more essential information, which leads to an incomplete and imprecise system identification. After several benchmark experiments, we make comparison between the three methods for experimental accuracy and the possible improvements are measured. See Table II: (x, y, z): (30.5%, 27.7%, 35%), Table III: (x, y, z): (98.4%, 99.5%, 98.48%), and Table IV: (β, x, y): (−0.15%, 56.5%, 71.3%), (−0.05%, 97.0%, 96.1%), (0.05%, 98.0%, 98.0%), (0.15%, 93.9%, 94.5%). According to the above data, the improvement should be prominent, especially in the machine learning fields.

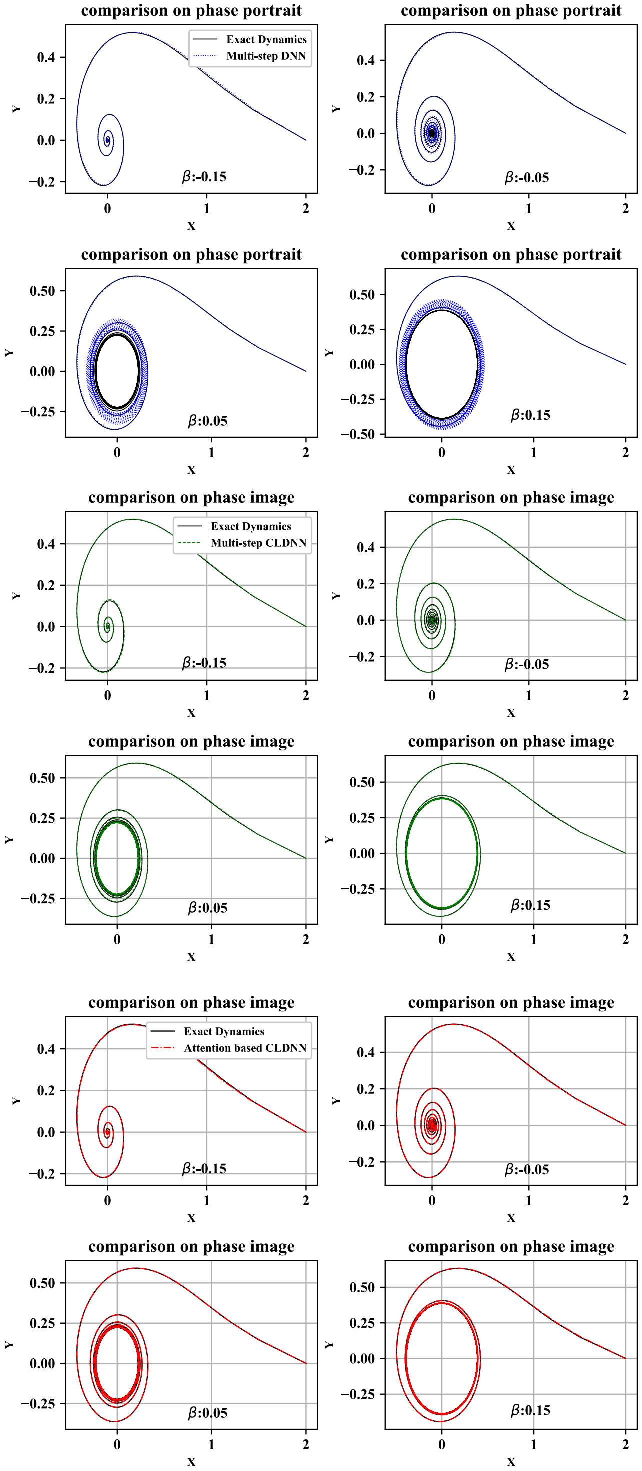

FIG. 6. The Supercritical Poincar'e–Andronov–Hopf bifurcation. The Hopf system is compared to the corresponding learned dynamics for different methods at different parameter values μ. Note that the relevant parameters setting for the multi-step DNN and the multi-step CLDNN is derived from Refs. 15 and 17, respectively.

TABLE IV. The L_2 error between the exact dynamics and the predicted dynamics. Note that the relevant parameters setting for the multi-step DNN and the multi-step CLDNN is derived from Refs. 15 and 17, respectively.

	B	X	y
Multi-step DNN	−0.15	2.2×10^{-2}	2.3×10^{-2}
	−0.05	2.0×10^{-1}	1.4×10^{-1}
	0.05	2.0×10^{-1}	2.4×10^{-1}
	0.15	1.1×10^{-1}	1.2×10^{-1}
Multi-step CLDNN	−0.15	1.4×10^{-2}	1.8×10^{-2}
	−0.05	5.6×10^{-2}	7.5×10^{-2}
	0.05	5.1×10^{-2}	5.6×10^{-2}
	0.15	1.8×10^{-2}	1.8×10^{-2}
Attention based CLDNN	−0.15	9.6×10^{-3}	6.6×10^{-3}
	−0.05	6.0×10^{-3}	5.4×10^{-3}
	0.05	4.1×10^{-3}	4.8×10^{-3}
	0.15	6.7×10^{-3}	6.6×10^{-3}

IV. CONCLUSION

In summary, we have demonstrated an attention-based deep neutral network to predict the nonlinear dynamical systems from time-series data. The proposed model combines the deep neural network (DNN), long short-term memory (LSTM), and convolution neural network (CNN) to learn dynamical systems from the observed data. In the model, we leverage the attention mechanism to seek the potential valuable information while suppressing irrelevant information, and then use DNN to map the function f denoting the evolution of dynamical systems. In particular, we utilize the linear multi-step scheme from numerical analysis to obtain local truncation for training our model. Compared with the multi-step DNN, the learned dynamics from our model can more accurately predict the chaotic dynamics and capture the attractor due to the ability of focusing on the salient information. Besides, our methodology also avoids approximating the temporal gradients from discretized time derivatives. The research provides possible corroboration for the application of the attention mechanism in nonlinear system identification using the deep learning technology.

ACKNOWLEDGMENTS

The work was supported by the Natural Science Foundation of Jiangsu Province under Grant No. BK20191371.

DATA AVAILABILITY

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

REFERENCES

[1] E. Ott, *Chaos in Dynamical Systems* (Cambridge University Press, 2002).

[2] J. D. Farmer, "Chaotic attractors of an infinite-dimensional dynamical system," Physica D **4**, 366–393 (1982).

[3] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: A unified overview," Automatica **31**, 1691–1724 (1995).

[4] K. Worden, C. Wong, U. Parlitz, A. Hornstein, D. Engster, T. Tjahjowidodo, F. Al-Bender, D. Rizos, and S. Fassois, "Identification of pre-sliding and sliding friction dynamics: Grey box and black-box models," Mech. Syst. Signal Process. **21**, 514–534 (2007).

[5] L. Billard and E. Diday, "Symbolic regression analysis," *Classification, Clustering, and Data Analysis* (Springer, 2002), pp. 281–288.

[6] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," Science **324**, 81–85 (2009).

[7] E. Weinan, B. Engquist, and Z. Huang, "Heterogeneous multiscale method: A general methodology for multiscale modeling," Phys. Rev. B **67**, 092101 (2003).

[8] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, "Model selection for dynamical systems via sparse regression and information criteria," Proc. R. Soc. A **473**, 20170009 (2017).

[9] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Data-driven discovery of partial differential equations," Sci. Adv. **3**, e1602614 (2017).

[10] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains* (John Wiley & Sons, 2013).

[11] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," IEEE Trans. Neural Networks **1**, 4–27 (1990).

[12] S. Chen, S. Billings, and P. Grant, "Non-linear system identification using neural networks," Int. J. Control **51**, 1191–1214 (1990).

[13] M. Gen and L. Lin, "Genetic algorithms," in *Wiley Encyclopedia of Computer Science and Engineering* (Wiley–Interscience, 2007), pp. 1–15.

[14] K. Kristinsson and G. A. Dumont, "System identification and control using genetic algorithms," IEEE Trans. Syst., Man, and Cybern. **22**, 1033–1046 (1992).

[15] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Multistep neural networks for data-driven discovery of nonlinear dynamical systems," arXiv:1801.01236 (2018).

[16] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "ABCNN: Attention-based convolutional neural network for modeling sentence pairs," Trans. Assoc. Comput. Linguist. **4**, 259–272 (2016).

[17] Q. Teng and L. Zhang, "Data driven nonlinear dynamical systems identification using multi-step CLDNN," AIP Adv. **9**, 085311 (2019).

[18] M. Lin, Q. Chen, and S. Yan, "Network in network," arXiv:1312.4400 (2013).

[19] T. Yang, L.-B. Yang, and C.-M. Yang, "Impulsive control of Lorenz system," Physica D **110**, 18–24 (1997).