Prediction of Chaotic Time Series Based on LSTM, Autoencoder and Chaos Theory

Nguyen Duc Huy[2(✉)] and Duong Tuan Anh[1,2]

[1] Department of Information Technology, HCMC University of Foreign Languages and Information Technology, Ho Chi Minh City, Vietnam
anh.dt@huflit.edu.vn

[2] Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam
ndhuy13@gmail.com

Abstract. Time-series forecasting, especially in a chaotic system, is a critical problem because its application is ubiquitous in several real-world fields, namely finance, environment, traffic, meteorology, industry, etc. In literature, there are many proposed methods for chaotic time series forecasting, but it is still challenging to yield a high predictive accuracy due to the chaotic characteristic which is very sensitive on the initial condition. In this work, we propose a fusion approach that takes advantage of chaos theory to represent time series data into phase space and combines autoencoder (AE) with Long Short-Terms Memory (LSTM) networks. First of all, the task of phase-space reconstruction starts with determining appropriate time lag and embedding dimension for the input time series. Next, autoencoder, which is constructed by LSTM cells, takes responsibility for latent-feature extraction through an unsupervised learning task and feeds the extracted data into LSTM-based forecaster. The experimental results on seven datasets including both synthetic and real-world chaotic time series reveal that our proposed method outperforms other forecasting methods using only stacked autoencoder, LSTM with or without chaos theory.

Keywords: Chaos · Phase space reconstruction · LSTM · Autoencoder · Chaotic time series forecasting

1 Introduction

Chaotic time series prediction is involved in various practical areas such as finance, traffic, environment, meteorology, geology, industry, etc. Chaos is one of the characteristics of time series, indicating the sensitivity of a chaotic system when has a small change of initial condition, as known as the Butterfly Effect. Several studies indicate that chaos theory can be utilized to yield better predictive results in chaotic time series forecasting.

However, it is still challenging to make accurate results in chaotic time series forecasting. The conventional methods using statistical and mathematical techniques (for

C. V. Phan and T. D. Nguyen (Eds.): ICCASA 2022, LNICST 475, pp. 141–155, 2023.
https://doi.org/10.1007/978-3-031-28816-6_11

instance, moving average, exponential smoothing, ARIMA model), k-nearest- neighbors algorithm, Multi-Layer-Perceptron (MLP) neural networks, Radial-Basis-Function (RBF) Networks and Support Vector Machines (SVMs), do not yield reliable prediction accuracy in case of time series with chaotic characteristics.

Taking advantage of the advances of deep learning in the past decade, several researchers have applied deep neural networks, such as Deep Belief Networks (DBN), Stacked Autoencoder (SAE), Long Short-Term Memory (LSTM) in forecasting chaotic time series. Through experiments in these works it is found out that Deep Neural Networks (DNN) can bring out better predictive performance. Some remarkable studies which apply DNN in chaotic time series forecasting can be listed as follows.

Kuremoto et al., in 2014, introduced a DBN model which stacks several restricted Boltzmann Machines (RBM) and one multi-layer perceptron (MLP) network to forecast chaotic time series [1]. For a specific kind of chaotic time series, namely short-term passenger flow in the China railway system, Zhang et al. [2] proposed a method which incorporates LSTM network and chaos theory in 2018. In 2019, Xu et al. [3] applied Continuous Deep Belief Neural Network combined with chaos theory for daily urban water demand prediction. Yang and Shen, in 2020 [4] proposed a method for chaotic time series forecasting by means of Differencing Long Short Term Memory (D-LSTM) and chaos theory. After that, more generally, the study of Phien et al. [5] in 2021 revealed that LSTM is more effective for chaotic time series prediction than DBN even though both use phase-space reconstruction in chaos theory. Sangiorgio et al. [6] in 2020 also combined chaos theory and LSTM for multi-step ahead forecasting in chaotic time series which brought out good prediction accuracy. Xu et al. in 2022 [7] combined SAE and particle swarm optimization for multivariate chaotic time series forecasting.

Recently, there have been some research works which integrate autoencoder, a kind of deep neural network to LSTM in order to enhance the predictive accuracy of LSTM in time series forecasting. Li et al. in 2018 [8] proposed a fusion approach which utilizes sparse autoencoder to extract features and LSTM-based forecaster to predict water quality. Heryadi in 2018 [9] studied a hybrid method which combines autoencoder and LSTM for short-term weather forecasting. Hoa et al. in 2021 [10] used a hybrid method which combines LSTM-based autoencoder and LSTM-based forecaster to predict foreign exchange rate which provided reliable prediction results.

Inspired by the main ideas from the two forecasting methods ([8–10]), in this work, we aim to apply these ideas in the context of chaotic time series. That means we attempt to combine LSTM-based autoencoder, LSTM and chaos theory to improve one- step-ahead prediction in chaotic time series. The main idea is that the proposed approach first transforms the univariate time series data to phase space and extracts latent features by the encoder from the unsupervised-trained autoencoder. After that, an LSTM-based forecaster is trained to get a one-step-ahead predicted value. To be more general and diverse, our study evaluates the proposed approach on two kinds of chaotic time series datasets including synthetic datasets (Lorenz, Mackey-Glass, and Rossler system) and real-world datasets (daily foreign exchange rate of AUD/USD, EUR/USD, monthly mean total sunspot number, IBM daily stock closed price). We compare the proposed method to three other deep neural network methods, such as LSTM with or without chaos theory, autoencoder with phase space reconstruction. To examine the prediction results, we use Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) as performance measures. The experimental results reveal that our proposed approach brings out a better predictive accuracy than the three other methods in terms of the three above mentioned metrics on the seven tested datasets.

The structure of the paper is as follows. In Sect. 2, we provide some basic backgrounds about chaotic theory, LSTM, and autoencoder (AE). In Sect. 3, the approach of combining chaos theory, autoencoder and LSTM forecaster for chaotic time series forecasting is introduced. Section 4 describes the experimental results which compare the proposed method to LSTM without and with chaos theory, and SAE with chaos theory. Finally, the paper in concluded in Sect. 5.

2 Background

2.1 LSTM and Bidirectional LSTM

Long Short-Term Memory (LSTM) was proposed by Hochreiter and Schmidhuber in 1997 [11]. LSTM is an improved variant of Recurrent Neural Network (RNN) designed specifically for sequence data to overcome long-term dependencies and exponential (or vanishing) gradient problem in RNN. The main idea of LSTM is cell state that has ability to keep or forget information from previous steps. The cell state is constructed by three gates named input gate, output gate, and forget gate. Forget gate takes responsibility of remembering the most useful information and forgetting the rest. Both input and output gate control when input signal and output signal should be processed. With this structure, LSTM unit was possible to handle long-term dependencies. Figure 1 depicts the architecture of each LSTM block (cell).

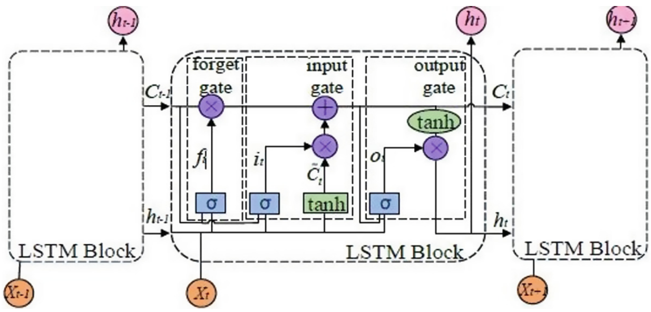

Fig. 1. Long short term memory cell

The activation of each gate is depicted by equations follow:

$$f_t = \sigma(W_f[C_{t-1}, h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i[C_{t-1}, h_{t-1}, x_t] + b_i) \quad (2)$$

$$\hat{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_f * \hat{C}_t \quad (4)$$

$$o_t = \sigma(W_o[C_t, h_{t-1}, x_t] + b_o) \quad (5)$$

$$h_t = o_t * \tanh(C_t) \quad (6)$$

where x_t is input vector, h_t is output vector of hidden layer, f_t is forget gate vector, C_t is cell state vector, i_t is input gate vector, o_t is output gate vector, σ is the sigmoid function, *W, b* respectively denote the weight and bias vector of each gate.

Bidirectional LSTM (Bi-LSTM) [12] is an extension of LSTM model in which two LSTMs are applied to the input data in two diferent directions, forward and backward. Applying the LSTM twice helps to improve learning long-term dependencies and thus consequently will improve the accuracy of the model ([12–14]). Figure 2 presents the structure of Bi-LSTM networks compared to LSTM.

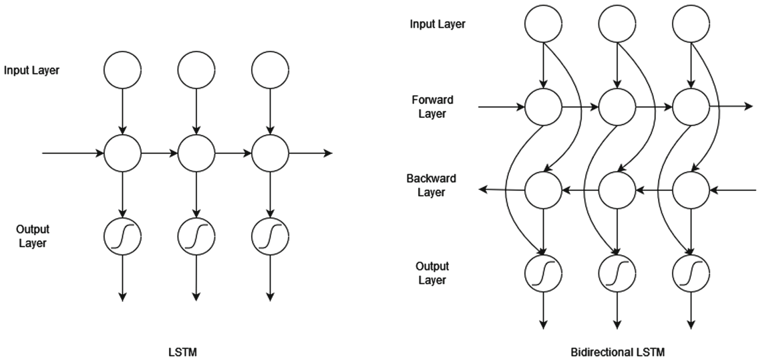

Fig. 2. LSTM and bidirectional LSTM

2.2 Autoencoder and Stacked Autoencoder

Autoencoder

An autoencoder (AE) is a special case of neural networks in which the input is the same as the output. It performs an unsupervised learning task. There are three components in AE including the *encoder*, the *code* and the *decoder*. In AE, the encoder and the decoder may have more than one layer. Firstly, AE compresses an input to a vector representation named code by the encoder. The code is regenerated to the input afterwards. In other words, the output is reconstructed from the code using the decoder. Because of its characteristics, AE is often used to extract latent features, reduce dimensions and denoise the input data in an unsupervised manner. Figure 3 depicts the architecture of an autoencoder in which both encoder and decoder consist of two layers. The input and output are almost the same, indicating that the code can be represented for input and reconstructed by the decoder. Therefore, after finishing the training step, the encoder is put out separately and used as a feature extractor.

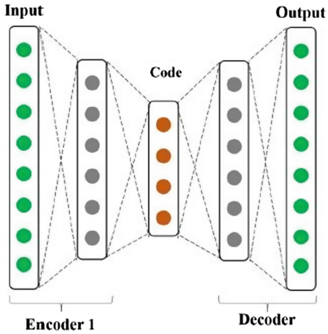

Fig. 3. The structure of an autoencoder model

Stacked Autoencoder
When more than one autoencoder are stacked, a stacked autoencoder (SAE) model is created. In SAE, except input of the first AE, the input of the next AE is taking the output of the previous one. More clearly, considering SAEs with n layers, the first layer is trained as an autoencoder with inputs from training set. After obtaining the first hidden layer, the output of the mth hidden layer is used as the input of the (m + 1)th hidden layer. In this way, multiple autoencoders can be stacked hierarchically. More clearly, considering SAEs with l layers, the first layer is trained as an autoencoder,

Sometimes, SAE architecture can also be applied to forecasting task. In this situation, we need to add a standard predicted layer (for example: fully connected layer) on the top most layer. This is illustrated in Fig. 4.

2.3 Chaos Theory

A chaotic time series is an irregular motion in deterministic system. A common practice in predicting the chaotic time series is to reconstruct the phase space of the system using delay space embedding method described as follows.

Given a single-dimensional time series x_t, where $t = 1, 2,.., N$, the method of time delays can be applied to contruct the phase space [16]. In this method, a corresponding phase space can be formed by assigning an element of the time series *xt* and its successive delays as coordinates of a multi-dimensional point.

$$X_t = \left\{x_t, x_{t+\tau}, x_{t+2\tau}, \ldots, x_{t+(m-1)\tau}\right\} \tag{7}$$

where each X_t is a data point in phase space, τ is referred to the *time lag* while m is termed *embedding dimension*. The dimension of new phase space m is consisdered as the optimal

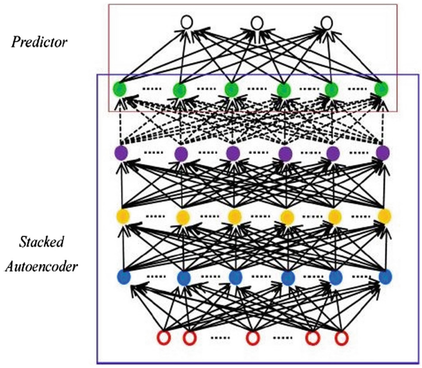

Fig. 4. The structure of SAE for forecasting ([15])

dimension for recovering the object without distorting any of its topological properties, thus it may be different from the true dimension of the space where the object lies. Takens proposed the idea of embedding dimension lower bound, that is $m \geq 2D + 1$, D is the strange attractor dimension ([16]). For each time series, both the τ and m parameters must be estimated.

To select a suitable time delay τ, we can use the *mutual information* method proposed by Fraser and Swinney [17]. To determine the minimum sufficient embedding dimension m we can use the *false nearest neighbor* method proposed by Kennel et al. [18].

Moreover, to check whether a time series has chaotic characteristics or not, we need to compute the largest Lyapunov index. A time series which has a finite positive maximal Lyapunov index is a chaotic time series. There exists a method proposed Rosenstein et al. [19] to calculate the maximal Lyapunov index from a given time series.

3 The Proposed Forecasting Method

Now, we present our proposed approach named AELSTM_C (Auto-Encoder combined with LSTM network and Chaos theory) for chaotic time series forecasting. This approach takes advantage of phase space reconstruction from chaos theory and combines two types of deep neural networks: autoencoder (AE), and LSTM. Specifically, the autoencoder network is constructed simply with an LSTM-based encoder and followed by an LSTM-based decoder. In our proposed model, each encoder layer or decoder layer consists of several LSTM cells.

With the combination of AE and LSTM, the capacity of LSTM to memorize long-term sequence data can be utilized to capture the temporal characteristics and cumulative effects of data. At the same time, autoencoder is used to extract the hidden features in the data and these features are better to overcome the shortcomings of LSTM which is easy to forget recent data, so as to further improve the predictive accuracy of LSTM model.

In training process, the proposed forecasting method performs the following steps.

Step 1: For each input chaotic time series, we determine delay time τ and embedding dimension m by computing Average Mutual Information (AMI) and using False Nearest

Neighbors (FNN) algorithm respectively. Given a one-dimensional time series x_t, where $t = 1, 2, \ldots N$. The vector of m-dimensional phase space is represented as follows, where X is the input data of next step, while the target values are in Y:

$$X = \begin{bmatrix} x_1 & x_{1+\tau} & \cdots & x_{1+(m-1)\tau} \\ x_2 & x_{2+\tau} & \cdots & x_{2+(m-1)\tau} \\ \vdots & \vdots & & \vdots \\ x_{N-1} & & \cdots & x_{N-1+(m-1)\tau} \end{bmatrix}, Y = \begin{bmatrix} x_{2+(m-1)\tau} \\ x_{3+(m-1)\tau} \\ \vdots \\ x_N \end{bmatrix} \quad (8)$$

Depending on each dataset, an appropriate data preprocessing method will be chosen afterward. In order to check the chaotic characteristic of a time series, the maximal Lyapunov index would be calculated in this step.

Step 2: The reconstructed data is split into a training part and a testing part with the percentage of 80 and 20, respectively. The unsupervised stage uses an LSTM-based autoencoder to encode and rebuild the input. Each sample in X is both input and target in the autoencoder training process. Then, the encoder is separated and acts like a feature extractor that produces an encoded vector for each input sample from X.

To enhance prediction performance and capture meaningful information, we concatenate the extracted features with the current value and other hand-engineered features like difference, percentage of change.

Step 3: In the supervised stage, the output of the previous step becomes the input which is fed into the forecaster constructed by an LSTM network. The losses are calculated by respectively comparing with the target from Y.

Figure 5. Illustrates the workflow of our proposed method for chaotic time series forecasting.

In testing process, we use the encoder from training process and do not need to retrain. The extracted vector that is generated from each sample by the encoder, concatenates with other features and feed to forecaster afterward to get a prediction result.

We also propose another model for chaotic time series forecasting in which LSTM network is replaced with Bi-Directional LSTM both in autoencoder and forecaster. This model is called AEBiLSTM_C (Auto-Encoder combined with Bi-LSTM network and Chaos theory).

4 Evaluation Experiments

In this evaluation experiment, we compare our two proposed methods (AELSTM_C and AEBiLSTM_C) for chaotic time series forecasting with three other deep neural network methods. As for the test datasets, we use seven univariate chaotic time series datasets including not only synthetic but also real-life datasets.

The Mean Absolute Error (MAE), the Root Mean Square Error (RMSE), the Mean Absolute Percentage Error (MAPE) are used as performance measures in this study. The formulas of the three evaluating measures are given as follows:

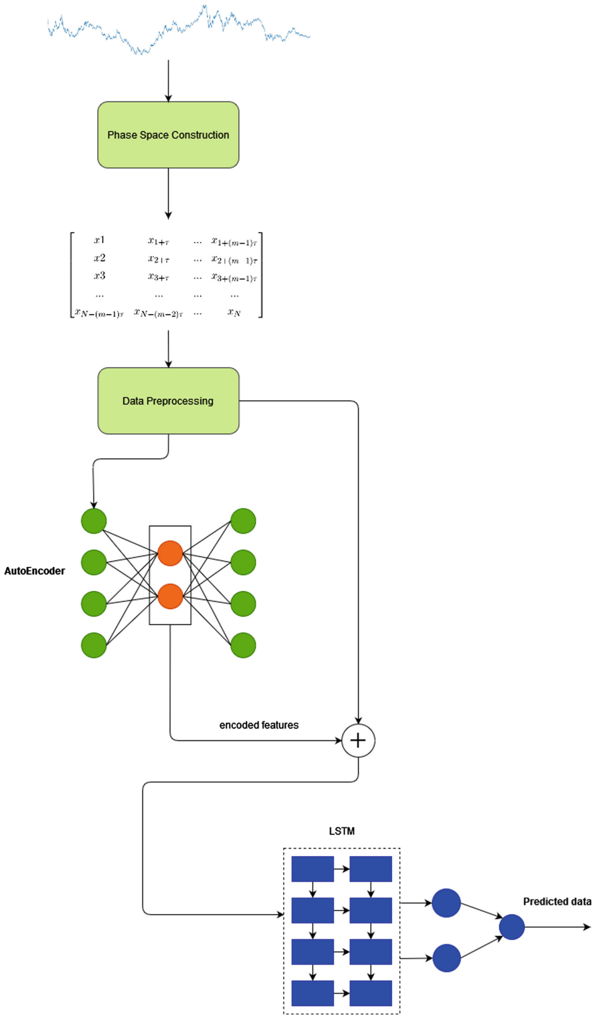

Fig. 5. A block diagram demonstrating the proposed forecasting method

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}\left|\hat{y}_t - y_t\right| \tag{9}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2} \tag{10}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{\left|\hat{y}_t - y_t\right|}{y_t} \tag{11}$$

in which n is the length of time series, $\hat{y}_t$ is the predicted value at time point t and y_t denotes the actual value in time point t.

The use of these metrics represents various aspects to assess the forecasting models. The first two are absolute evaluation metrics while the last one (MAPE) is a relative metric. The MAPE is a scale-invariant statistic that expresses error as a percentage. The model has higher predictive accuracy when MAE, MSE or MAPE is much closer to 0.

4.1 Datasets and Parameter Estimation

As mentioned before, there are three synthetic and four real life chaotic time series datasets that are used in this experiment. All these datasets have been used in literature as benchmark datasets due to their chaotic characteristics. The detailed description of the datasets is as follows.

1. This dataset is computed from the Lorenz system which is represented by three differential equations as in (12).

$$\begin{cases} \frac{dx}{dt} = a(y - x) \\ \frac{dx}{dt} = xy - cz) \\ \frac{dx}{dt} = x(b - z) - y \end{cases} \tag{12}$$

   where $a = 10$, $b = 28$, $c = 8/3$. In this study, we only use x-axis data containing 1000 points.
2. This dataset is computed from the Mackey-Glass system which is represented by the following differential equation:

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1 + x^{c}(t-\tau)} - bx(t) \tag{13}$$

   where $a = 0.2$, $b = 0.1$, $c = 10$, $r = 17$ and $x0 = 1.2$. This time series dataset consists of 1001 data points.
3. This dataset is computed from the Rossler system, which is represented by the following differential equations:

$$\begin{cases} \frac{dt}{dt} = -z - y \\ \frac{dy}{dt} = x + ay) \\ \frac{dz}{dt} = b + z(x - c) \end{cases} \tag{14}$$

   where $a = 0.15$, $b = 0.2$ and $c = 10$. This time series dataset consists of 8192 data points.
4. This dataset consists of the daily closed price of the exchange rate between AUD (Australian Dollar) and USD (US Dollar) from 2nd January 1990 to 31st

   December 2019. This dataset is denoted as AUDUSD, which contains 7820 data points.

5. This dataset consists of the daily closed price of the exchange rate between EUR (Euro) and USD (US Dollar) from 2nd January 1990 to 31st December 2019. This dataset is denoted as EURUSD, which contains 7820 data points.
6. This dataset consists of monthly smoothed total sunspot numbers from January 1824 to December 2018. It is a widely used benchmark dataset in chaotic time series forecasting. It has 2340 data points.
7. This dataset consists of the daily closed price of IBM stock from 2nd January 2002 to 31st December 2020, which contains 4784 values.

The seven datasets are divided into training-validation and testing parts with the corresponding proportions of 80%, and 20%, respectively. Figure 6 illustrates the plots of seven datasets which the first part (in blue) indicates training-validation part, and the rest is testing part.

In this study, we utilize *nonlinearTseries* [20] and *tseriesChaos* [21] package in R software to determine time lag and embedding dimension for each dataset. Besides, we use Scikit-learn for data preprocessing, and Keras [22] (based on Tensorflow) for training the deep learning models.

Table 1 reports the appropriate parameters that we choose in this work including time lag τ, embedding dimensions m for each dataset, and data preprocessing technique by experiment.

In this work, before feeding to autoencoder (AE), a data preprocessing technique is used depending on each dataset including min-max normalization, zero-mean normalization or just using original data.

Our proposed forecasting methods (AELSTM_C and AEBiLSTM_C) consist of two components, one for each stage: feature extraction and forecasting. Encoder and decoder in AE consist of two layers. LSTM-based forecaster also consists of two layers. The architecture parameters of the two proposed forecasting methods are described in Table 2.

The autoencoder is trained using Backpropagation algorithm with some supporting techniques, such as learning rate scheduler and early stopping. The forecaster is trained using Backpropagation algorithm with some supporting techniques, such as learning rate scheduler, early stopping and dropout.

Table 1. Parameters in phase-space reconstruction for seven datasets.

	Lorenz	Mackey-Glass	Rossler	AUDUSD	EURUSD	Sunspots	IBM
τ	4	4	4	107	59	38	102
m	4	5	4	7	6	6	6

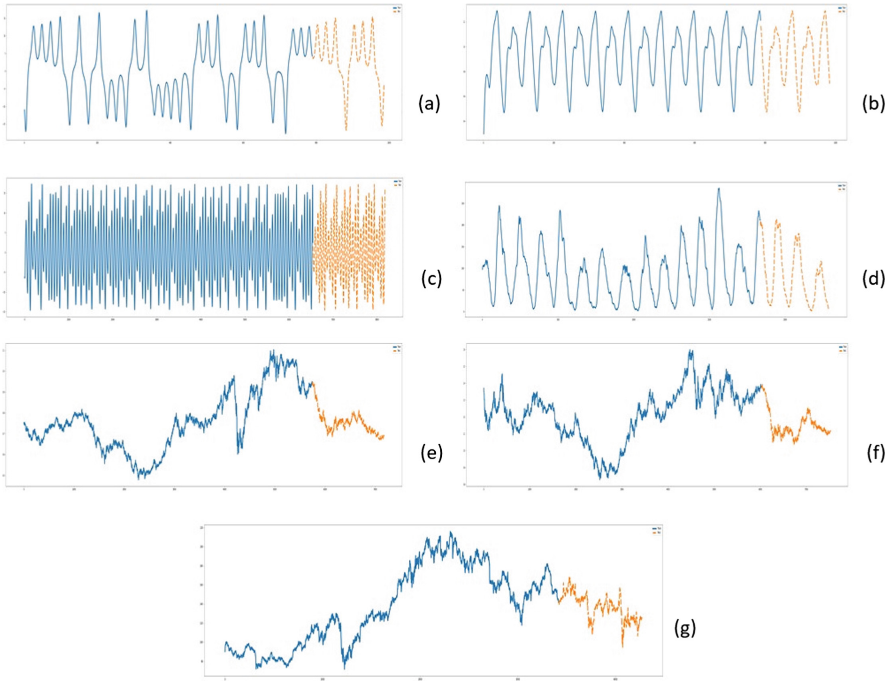

Fig. 6. The plots of seven time series (a) Lorenz, (b) Mackey-Glass, (c) Rossler, (d) Sunspots,(e) AUDUSD, (f) EURUSD and (g) IBM

Table 2. Architecture parameters for two proposed methods

Method	LSTM-based Autoencoder	Forecaster
AELSTM_C	Encoder: 128 LSTM units, 64-LSTM units Decoder: 64 LSTM units, 64 LSTM units	128 LSTM units, 64 LSTM units, 50- units fully connected, 1-unit output
AEBiLSTM_C	Encoder: 128 LSTM units, 64 Bi-LSTM units Decoder: 64 Bi-LSTM units, 64 LSTM units	128 Bi-LSTM units, 64 Bi-LSTM units, 50-units fully connected, 1-unit output

To evaluate the predictive accuracy of our two proposed forecasting methods, we compare these two methods to three other deep neural network methods including LSTM without chaos theory, LSTM with chaos theory, and SAE with chaos theory (denoted as LSTM, LSTM_C, SAE_C, respectively). Notice that SAE_C can be understood as a forecaster contains stacked autoencoder architecture and uses phase space data as input. The architecture parameters of the three comparative methods are shown in Table 3.

Table 3. Architecture parameters for three other comparative methods

Method	Architecture
LSTM	128-unit LSTM, 64-unit LSTM, 50-unit fully connected, 1-unit output
LSTM_C	128-unit LSTM, 64-unit LSTM, 50-unit fully connected, 1-unit output
SAE_C	Encoder: 128-LSTM units, 64- LSTM units, Decoder: 64-LSTM units, 64-LSTM units Followed by: 50-unit fully connected, 1-unit output

4.2 Experimental Results

The experiments in this work focus on one-step-ahead forecasting. The prediction errors on seven chaotic time series datasets in terms of MAE, RMSE and MAPE are reported in Table 4, Table 5, and Table 6, respectively.

In Table 4, Table 5, and Table 6, the best result value on each dataset is highlighted in bold. In terms of MAE, AELSTM_C yields the best results on 5 out of 7 datasets and AEBiLSTM_C yields the best results on 2 out of 7 datasets. In terms of RMSE, AELSTM_C yields the best results on 4 out of 7 datasets and AEBiLSTM_C yields the best results on 3 out of 7 datasets. In terms of MAPE, AELSTM_C gives the best results on 4 out of 7 datasets and AEBiLSTM_C provides the best results on 3 out of 7 datasets.

Table 4. MAE Prediction errors of the five methods on 7 datasets

	AEBiLSTM_C	AELSTM_C	SAE_C	LSTM_C	LSTM
Lorenz	0.002707229	**0.002457134**	0.03475358	0.041782404	1.123163643
Mackey- Glass	**0.000746647**	0.000844273	0.005041643	0.005561022	0.028847694
Rossler	0.003117215	**0.002675147**	0.013910623	0.022279623	0.87368712
AUDUSD	0.003394172	**0.003367457**	0.003457074	0.003466233	0.005352983
EURUSD	**0.004242299**	0.004248584	0.004275216	0.004353091	0.006650565
Sunspots	1.326593694	**1.291807013**	1.365681218	1.844001427	2.648827568
IBM	1.532437417	**1.527935701**	1.547949677	1.547214839	2.318986811

From Table 6, the reduction in the magnitude of MAPE values between AELSTM_C and LSTM_C and between LSTM_C and LSTM can be derived and reported in Table

7. In comparing the AELSTM_C with LSTM_C, the percentage of reduction in MAPE varies from 87.99% to 1.37%. In comparing the LSTM_C with LSTM, the percentage of reduction in MAPE ranges from 96.95% to 32.29%.

Based on the experimental results in Table 4, Table 5, Table 6 and Table 7, we can derive the following findings:

– In all seven tested datasets, the two proposed forecasting methods (AEBiLSTM_C and AELSTM) always bring out the lowest prediction errors and the performance improvement in comparison to the three other deep neural network methods (LSTM_C, SAE_C

Table 5. RMSE prediction errors of the five methods on 7 datasets

	AEBiLSTM_C	AELSTM_C	SAE_C	LSTM_C	LSTM
Lorenz	0.003779549	**0.003233493**	0.051545547	0.064413727	1.365926947
Mackey- Glass	**0.000967249**	0.0010554	0.006046002	0.00676352	0.034668973
Rossler	**0.004590836**	0.00569002	0.026299925	0.04621249	1.045916098
AUDUSD	0.00444350	**0.004426192**	0.004511058	0.004545271	0.006839612
EURUSD	**0.00574289**	0.005749826	0.005796582	0.005851447	0.008764376
Sunspots	1.877174774	**1.830231368**	1.912499436	2.563664491	3.577799998
IBM	2.312525518	**2.30859582**	2.338475677	2.335779534	3.362016134

Table 6. MAPE Prediction errors of the five methods on 7 datasets

	AEBiLSTM_C	AELSTM_C	SAE_C	LSTM_C	LSTM
Lorenz	0.119799658	**0.107294015**	0.67290555	0.824846866	27.04564231
Mackey- Glass	**0.079898069**	0.09816394	0.561337052	0.600053668	3.342857122
Rossler	**0.104063821**	0.126757365	0.646663646	0.948263934	53.52557816
AUDUSD	0.44939974	**0.445757483**	0.458010878	0.457661599	0.700437479
EURUSD	**0.370720357**	0.371291928	0.373560396	0.379574627	0.578685863
Sunspots	2.597493668	**2.559844591**	2.649975593	3.560925701	5.41588127
IBM	1.16278277	**1.159223622**	1.176135408	1.175343294	1.735965376

Table 7. % Reduction of MAPE values between AELSTM_C and LSTM_C and between LSTM_C and LSTM

	% Reduction of AELSTM_C over LSTM_C	% Reduction of LSTM_C over LSTM
Lorenz	87.9923	96.9502
Mackey-Glass	83.6408	82.0497
Rossler	86.6328	98.2384
AUDUSD	02.6013	34.6605
EURUSD	02.1822	34.4073
Sunspots	28.111	34.2500
IBM	01.371	32.2950

and LSTM) are remarkable. This demonstrates that the proposed methods are the best among the five comparative methods over all seven datasets.

- AutoEncoder as a feature extraction unit, phase-space reconstruction for handling chaotic time series and LSTM as a time series forecaster mainly contribute to the improvement of prediction accuracy. It is clear that LSTM_C is much better than LSTM (i.e. LSTM without chaos theory) and SAE_C is better than LSTM_C.
- Compared to AELSTM_C, in some cases, the applying of Bi-LSTM can yield a slight improvement in prediction accuracy.

5 Conclusion and future work

Chaotic time series forecasting is still a challenge because of the extremely complicated characteristics of chaotic time series. Therefore, improving the performance of chaotic time series forecasting is especially important. In this work, we propose a hybrid deep neural network method for the one-step-ahead prediction in chaotic time series which combines chaos theory, latent-feature extraction ability of autoencoder and memorization characteristics of LSTM. Experimental results over three synthetic datasets and four real-life datasets of chaotic time series show that our two proposed approaches (AELSTM_C and AEBiLSTM_C) outperform the three other methods (LSTM_C, SAE_C and LSTM) in terms of forecasting accuracy. Moreover, the experiment results also confirm that chaos theory, AutoEncoder can be combined with LSTM network to improve forecasting performance in chaotic time series.

Deep Neural Network hybrid approach, such as AELSTM_C or AEBiLSTM_C, is a suitable alternative for chaotic time series forecasting. As for future work, we would like to improve our model in some following directions:

- We intend to apply other variants of stacked autoencoder to enhance feature extraction.
- We plan to extend our proposed forecasting methods for multi-step ahead prediction in chaotic time series ([6]).
- We will explore attention mechanism in LSTM ([23]) which can concentrate on influential data points in chaotic time series forecasting.

References

1. Kuremoto, T., Obayashiand, M., Kobayashi, K., Hirata, T. and Mabu, S.: Forecasting chaotic time series data by DBNs. In: 7th International Congress on Image and Signal Processing (2014)
2. Zhang, Y., Zhu, J., Zhang, J.: Short-term passenger flow forecasting based on phase space reconstruction and LSTM. In: Jia, L., Qin, Y., Suo, J., Feng, J., Diao, L., An, M. (eds.) EITRT 2017. LNEE, vol. 482, pp. 679–688. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-7986-3_69
3. Xu, Y., Zhang, J., Long, Z., Lv, M.: Daily urban water demand forecasting based on chaotic theory and continuous deep belief neural network. Neural Process. Lett. **50**(2), 1173–1189 (2018). https://doi.org/10.1007/s11063-018-9914-5
4. Yang, C.H., Shen, H.Y.: Analysis and prediction of chaotic time series based on deep learning neural networks. In: International Conference on System Science and Engineering (ICSSE), Kagawa, Japan, pp. 1–9 (2020)

5. Phien, N.N., Anh, D.T., Platos, J.: A comparison between deep belief network and LSTM in chaotic time series forecasting. In: Proceedings of International Conference on Machine Learning and Machine Intelligence (MLMI), HangZhou, China, pp. 157–163 (2021)
6. Sangiorgio, M., Dercole, F.: Robustness of LSTM neural networks for multi-step forecasting of chaotic time series. Chaos, Solitons Fractals, vol. 139, p. 110045 (2020)
7. Xu, X., Ren, W.: A hybrid model of stacked autoencoder and modified particle swarm optimization for multivariate chaotic time series forecasting. Appl. Soft Comput. **116**, 108321 (2022)
8. Li, Z., Peng, F., Niu, B., Li, G., Wu, J., Miao, Z.: Water quality prediction model combining sparse auto-encoder and LSTM network. IFAC Pap. Online **51**(17), 831–836 (2018)
9. Heryadi, Y.: Learning hierarchical weather data representtion for short-term weather forecasting using autoencoder and Long Short Term Memory models. In: Proceedings of ACIIDS, pp. 373–384 (2019)
10. Hoa, T.V., Anh, D.T., Hieu, D.N.: Foreign exchange rate forecasting using autoencoder and LSTM networks. In: Proceedings of International Conference on Intelligent Information Technology (ICIIT), pp. 22–28. ACM, Ho Chi Minh City (2021)
11. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
12. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process **45**, 2673–2681 (1997)
13. Siami-Namini, S., Tavakoli, N., Namin, A.S.: A comparative analysis of forecasting financial time series using ARIMA, LSTM, and BiLSTM. In: Proceedings of IEEE BigData, Los Angeles, CA, USA (2019)
14. Abduljabbar, R.L., Dia, H, Tsai, P.W.: Unidirectional and bidirectional LSTM models for short term traffic prediction. J. Adv. Transp. 589075, 16 (2021)
15. Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y.: Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. **16**(2), 865–873 (2015)
16. Takens, F.: Detecting strange attractors in turbulence. In: Rand, D., Young, L.-S. (eds.) Dynamical Systems and Turbulence, Warwick 1980. LNM, vol. 898, pp. 366–381. Springer, Heidelberg (1981). https://doi.org/10.1007/BFb0091924
17. Fraser, A.M., Swinney, H.L.: Independent coordinates for strange attractors from mutual information. Phys. Rev. A **33**(2), 1134–1140 (1986)
18. Kennel, M.B., Brown, R., Abarbanel, H.D.I.: Determining embedding dimension for phase-space reconstruction using a geometrical construction. Phys. Rev. A **45**(6), 3403–3411 (1992)
19. Rosenstein, M.T., Collins, J.J., Luca, C.J.D.: Reconstruction expansion as a geometry- based framework for choosing proper delay times. PHYSICA D **73**, 82–98 (1994)
20. NonlinearTseries in R. https://cran.r-project.org/web/packages/nonlinearTseries. Accessed 2021
21. tseriesChaos in R. https://cran.r-project.org/web/packages/tseriesChaos/tseries. Accessed 2021
22. Cholett, F.: Keras. http://keras.io. Accessed 2021
23. Du, S., Li, T., Yang, Y., Horng, S.J.: Multivariate time series forecasting via attention- based encoder-decoder framework. Neuro Comput. **388**, 269–279 (2020)