IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 17, 2020, accepted August 28, 2020, date of publication September 1, 2020, date of current version September 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020801

Deep Hybrid Neural Network and Improved Differential Neuroevolution for Chaotic Time Series Prediction

WEIJIAN HUANG, YONGTAO LI, AND YUAN HUANG
School of Information and Electrical Engineering, Hebei University of Engineering, Handan 056038, China

Corresponding author: Yongtao Li (lyotard@163.com)

This work was supported in part by the Natural Science Foundation of Hebei Province under Grant F2015402077, and in part by the Project of the University Science and Technology Research Youth Fund of Hebei Province under Grant QN2018073.

ABSTRACT Chaos is widespread in non-linear systems such as finance, energy, and weather. In the chaos system, a variable changing with time generates a chaotic time series, which contains a wealth of information about the non-linear system, and it is helpful for us to analyze and understand chaos systems. Traditional hybrid models for chaotic time series prediction based on neural networks have problems such as low prediction accuracy and difficulty in determining the network topologies. In recent years, the chaotic time series prediction has attached the attention of researchers in the area of deep learning. In this paper, we use a deep hybrid neural network (DHNN) based on convolutional neural network (CNN), gated recurrent unit (GRU) network, and attention mechanism to predict chaotic time series. Besides, we use the idea of neuroevolution to optimize the topologies of the DHNN. In the DHNN, we use CNN to capture spatial features from phase space reconstruction of chaotic time series. Then, we combine spatial features with the original chaotic time series. GRU extracts the spatio-temporal features from the combined sequence, and an attention mechanism with a non-linear activation function is designed to capture critical spatio-temporal features. Besides, we propose an improved differential evolution (IDE) algorithm to optimize the topologies of the DHNN, including the filter sizes of CNN and the number of hidden neurons of GRU. We develop the IDE with an adaptive mutation operator and dynamic chaos crossover operator, which can improve convergence speed and reduce optimization time. In this paper, we use the theoretical Lorenz datasets, monthly mean total sunspot datasets, and the actual coal-mine gas concentration datasets to verify the prediction accuracy of the proposed prediction model. Experimental results have shown that the proposed prediction model performs well in chaotic time series forecasting.

INDEX TERMS Chaotic time series prediction, convolutional neural network, gated recurrent unit, attention mechanism, improved differential evolution, neuroevolution.

I. INTRODUCTION

Chaotic time series prediction (CTSP) is involved in various domains of social and natural sciences, such as copper metal price, oilfield water injection, wind power, and rainfall [1]–[4]. Over the last decade, CTSP has been applied to the study of blood glucose, disease, and gait in humans [5]–[7]. Besides, CTSP also has been applied to cyber-information tasks such as retweeting [8], information diffusion [9], and DoS and DDoS attack detection [10]. The application of CTSP in the real world is becoming more significant and more widespread.

Theoretical and empirical studies reported in the literature suggest that the hybrid model is one of the best ways to improve the accuracy of time series forecasting [11], [12]. A hybrid network model combined support vector machine (SVM) and echo state mechanism (ESM) was proposed to CTSP [13]. Ardalani *et al.* [14] proposed a hybrid Elman-NARX neural network to forecast the chaotic time series. Said Jadid *et al.* developed an unscented Kalman filter and NARX neural network to analyze and predict the Lorenz time series [11]. Combined with the smoothing approach considering the entropic information, a noisy forecast

The associate editor coordinating the review of this manuscript and approving it for publication was Hisao Ishibuchi.

method was applied to chaotic rainfall time series [4]. Nhabangue *et al.* proposed a functional link extreme learning machine to CT SP [15], and Xu *et al.* applied a hybrid regularized echo state network to forecast multivariate CTSP [16]. Yan *et al.* developed a hybrid empirical mode decomposition and neural network for Maritime Time Series Prediction [17]. These hybrid models have performed well in CTSP. More recently, deep learning algorithms such as long short-term memory neural network (LSTM) [18], convolutional neural network (CNN) [19], and hybrid CNN-LSTM neural network have been applied to CTSP [20]–[22]. YanLi *et al*. applied hybrid empirical mode decomposition, adaptive regrouped, and LSTM to forecast port cargo thro-ughput time series [23]. Compared to the hybrid machine learning model, the hybrid deep learning model has a better performance [22].

In the last few years, a particular hybrid model named neuro-evolution has once again caught the attention of researchers. Unlike other hybrid models, neuroevolution can be used to design neural networks [24], [25]. Genetic algorithms (GA) and evolutionary strategies (ES) have yielded excellent performance in optimizing the topology and weights of neural networks [26], [27]. Through neuroevolution, we can determine the appropriate network structure for a specific problem and achieve excellent predictive performance [28]. At present, the research of evolutionary algorithms for neuroevolution is continuously developing.

In previous studies, we observed that attention mechanism had made more exceptional performance on the tasks of sequence models, such as machine translation and textual entailment [29], [30]. The attention mechanism can extract key spatio-temporal features from spatial and temporal features [29]. Besides, it also can solve some long-term memory problems [31]. Thus, it is taken into account while using neural networks to extract temporal features from sequence models. More recently, attention mechanism has been applied to predict time series. Youru *et al.* introduced an evolutionary attention learning approach to transfer shared parameters of LSTM [32], and a multistage attention network is designed to capture the influence information and the variation law of data over time [33]. Yao *et al.* proposed a dual-stage attention-based recurrent neural network (DA-RNN) to address long-term temporal dependencies and select the relevant driving series to make predictions [34]. Yeqi *et al.* developed a dual-stage two-phase attention-based recurrent neural network (DSTP-RNN) for long-term and multivariate time series prediction, which can capture spatio-temporal correlations and long-term temporal dependencies [35]. In deep learning, an attention mechanism with function mapping is designed to capture mutation information on the target time series, which can process the fusion of historical hidden state and cell state information for LSTM [36].

The hybrid models mentioned in the previous literature were only considered the spatial or temporal characteristics of chaotic time series. In this paper, a deep hybrid neural network based on deep learning is proposed to CTSP, which considers both spatial and temporal. In the proposed model, spatial characteristics are acquired by CNN, and gated recurrent unit (GRU) [37] is used to extract temporal characteristics. We apply the differential evolution (DE) [38] algorithm to design the topologies of hybrid neural network and search appropriate time-steps for forecasting. However, we observed that simple DE and adaptive DE [39] have a slow convergence speed and long running times. Thus, we improve DE by changing the mutation operator and crossover operator. In section II, we describe the specific improvements in detail. Of course, the attention mechanism is used to extract spatial-temporal features from hybrid deep learning neural networks.

The paper is organized as follows. We introduce the hybrid neural network, CNN, GRU, attention mechanism, and improved DE in section II. In section III, we describe the specific details of the experiments. In section IV, we analyze and discuss the experimental results. The conclusion is summarized in section V.

II. HYBRID MODEL

As shown in Fig.1, various kinds of neural networks play different roles in the proposed hybrid model. The reconstructed phase space of the chaotic time series contains spatial features of the chaos system, while the original sequence also contains rich temporal features. Therefore, the CNN model is used to capture the spatial features of the reconstructed phase space, and GRU extracts the spatio-temporal features under the spatial features. The attention model is used to capture the critical spatio-temporal features. Meanwhile, we use the improved DE to design topology of the hybrid network, including the kernel sizes of the CNN and the number of hidden neurons for the GRU neural network. In this paper, we make a one-step prediction, and time-steps (the number of data used to forecast) often affects the prediction accuracy. Therefore, we use improved DE to search fitting lookback for forecasting. The details are described as follows.

A. CONVOLUTIONAL NEURAL NETWORK

Convolutional neural network(CNN) is a specialized kind of deep learning neural networks which can process data with known grid-like topologies [19], [40]. It is widely to use CNN in the fields of time series analysis, computer vision, and natural language processing[40]. CNN can categorize into 1-D (dimension), 2-D, and 3-D convolution by processing the different data streams. In this paper, we use a 1-D convolutional neural network, which is widely using in the fields of time series analysis and natural language processing [41]. As shown in Fig.2(a), the main parts of the simple 1-D CNN include the essential input and output layers, the convolutional and pooling layers are the most critical layer, and the fully connected layer is necessary. In 1-D CNN, we can understand the function of convolution as extracting the translation features of the data in a particular direction, where the essence of the operation of the convolution is the circular multiplication and summing, which is expressed by

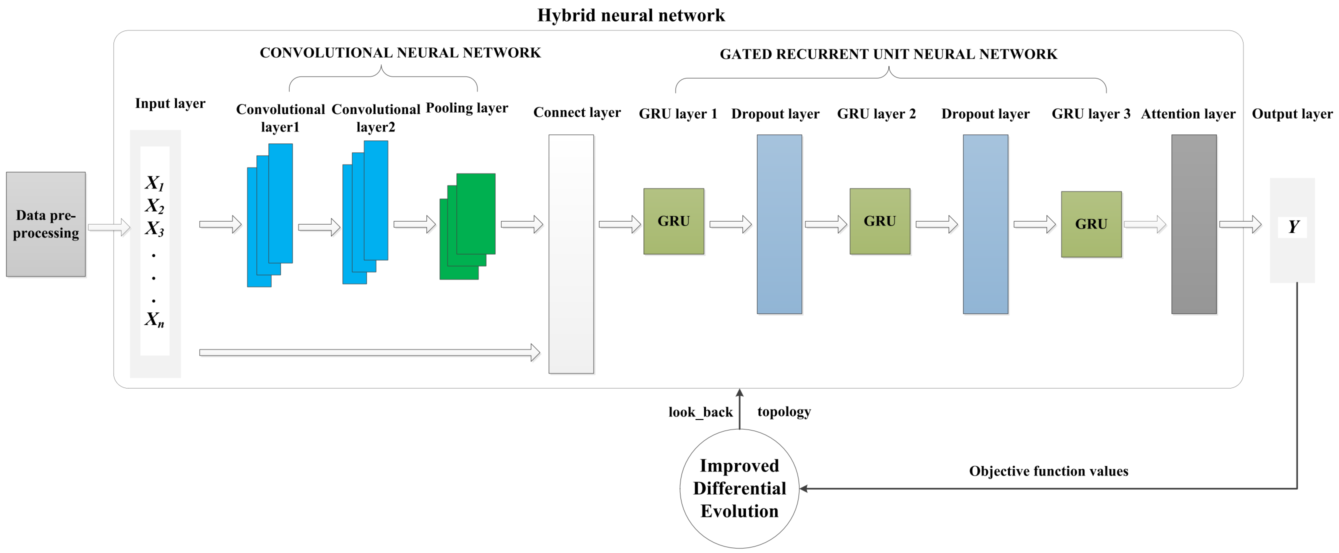
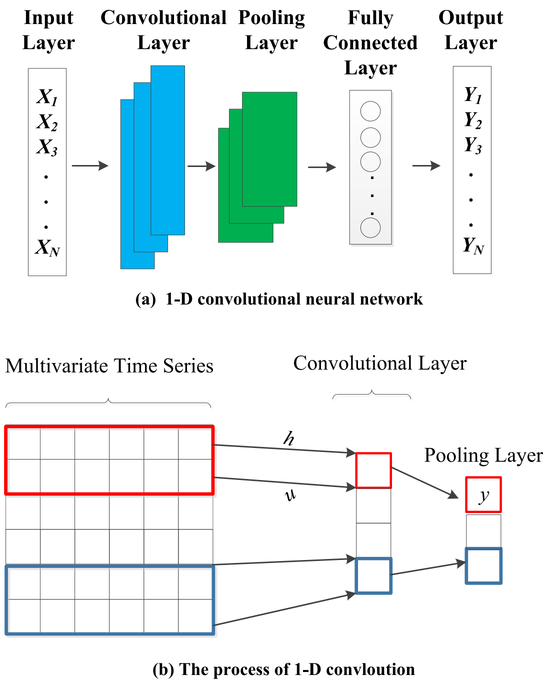


FIGURE 1. The structure of the proposed prediction model.



(a) 1-D convolutional neural network

(b) The process of 1-D convloution

FIGURE 2. 1-D CNN.

the following formula:

$$y(k) = h(k) \cdot u(k) = \sum_{i=0}^{N} h(k-i)u(i) \tag{1}$$

where y, h, u are series, as shown in Fig.2(b), h and u are a row of a multivariate time series, they are convoluted from top to bottom. k represents the times of convolution, the length of u is N.

B. GATED RECURRENT UNIT

As the extension of the feed-forward neural network, the re-current neural network(RNN) can handle variable-length sequence data via hidden state units [42]. However, the vanishing and the exploding gradient problems put a limit on training RNN [43]. Long short-term memory and gated recurrent unit were proposed to solve the vanishing and the exploding gradient problems. LSTM and GRU are gated recurrent neural networks, which use various gates to capture long-term dependencies of a sequence data.

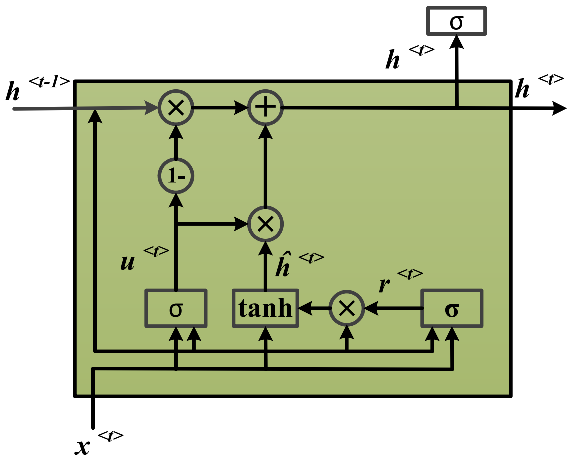


FIGURE 3. GRU cell.

LSTM has three gates, including an input gate, an output gate, and a forget gate. Unlike LSTM, GRU has two gates. As shown in Fig.3, GRU uses an update gate u to control the forgetting factor and the decision to update the state unit simultaneously. Besides, a reset gate r can control how much historical information to forget, and the update equations are the following:

$$u^{<t>} = \sigma(W_u[h^{<t-1>}, x^{<t>}] + b_u) \tag{2}$$

$$r^{<t>} = \sigma(W_r[h^{<t-1>}, x^{<t>}] + b_r) \tag{3}$$

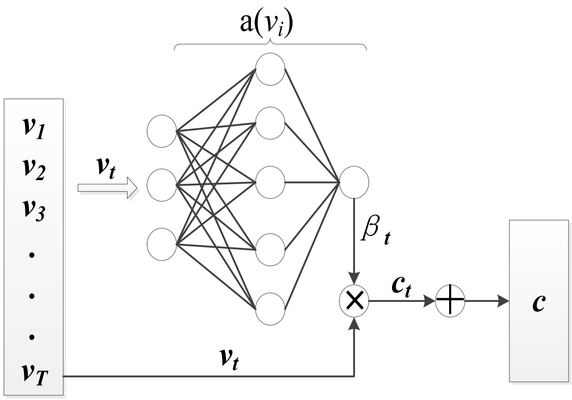


FIGURE 4. Attention model.

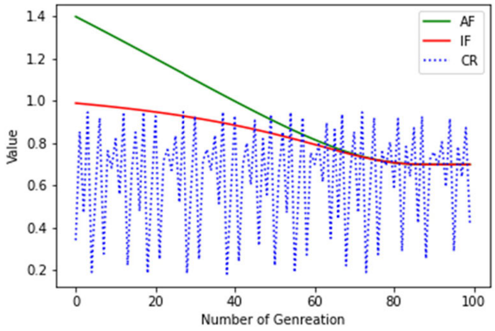


FIGURE 5. Mutation and crossover operator ($F_0 = 0.7, CR_0 = 0.1$).

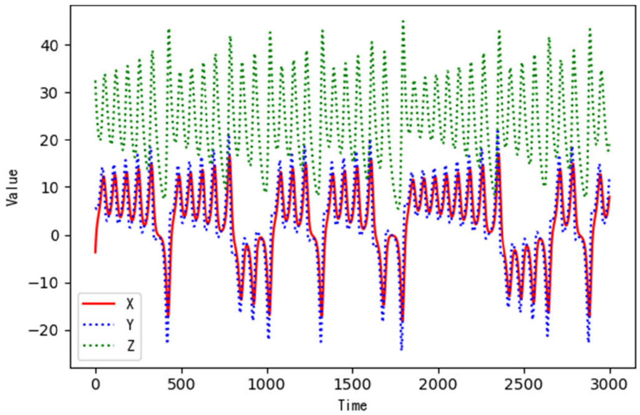


FIGURE 6. Lorenz Chaotic Time Series.

$$\hat{h}^{<t>} = \tanh(W_h[r^{<t>} * h^{<t-1>}, x^{<t>}] + b_h) \qquad (4)$$

$$h^{<t>} = (1 - u^{<t>}) * h^{<t-1>} + u^{<t>} * \hat{h}^{<t>} \qquad (5)$$

where W and b stand for weights and biases, σ is a sigmoid function, *tanh* represents activation function, $h^{<t>}$ is the output of the GRU cell, t represents the current time state.

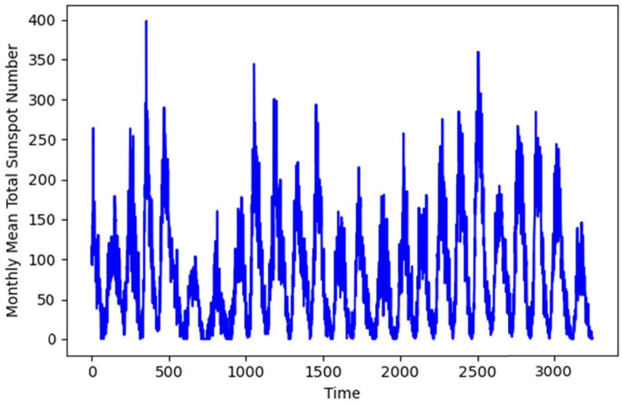


FIGURE 7. The curve of monthly mean total sunspot number.

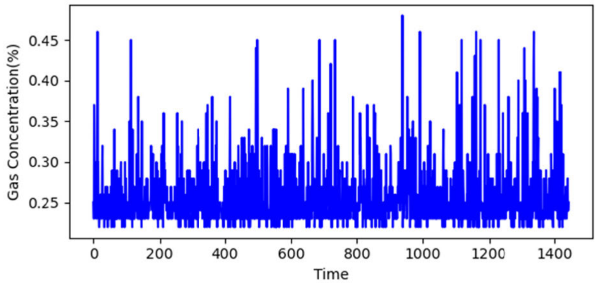


FIGURE 8. The curve of coal-mine gas concentration.

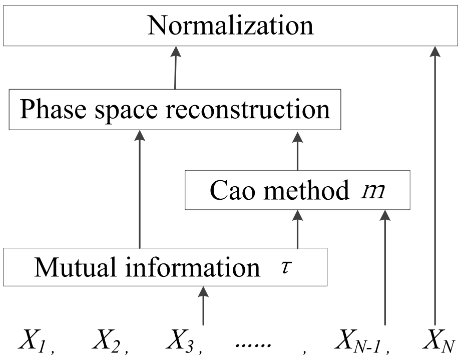

Normalization
Phase space reconstruction
Cao method m
Mutual information τ
X_1, X_2, X_3, ... , X_{N-1}, X_N

FIGURE 9. Data preprocessing.

C. ATTENTION MECHANISM

It tends to focus on certain parts of the things when the human brain observes something, and these focused parts are the key to acquire information form things, which are very useful for us to recognize similar things. The attention mechanism is a unique method that mimics this cognitive process [44]. Attention mechanism has been applied to the computer version and natural language processing [29], [30], and we apply attention mechanisms to the analysis of chaotic time series.

In CTSP, we use CNN to extract spatial features from the reconstructed phase space of the chaotic time series, and then use GRU to extract spatio-temporal features based on spatial features. However, the prediction accuracy is affected by too many or non-critical features. Thus, we apply the attention mechanism to extract the key features from the hybrid CNN-GRU model. As shown in Fig.4, the attention

TABLE 1. The performance of the neuroevolution-DHNN.

Datasets	Evolution algorithm	CNN neurons	GRU neurons	Time Steps	RMSE	MAPE	Average error	Running times (s)
Lorenz	**IDE**	**[64,19]**	**[26,49,23]**	**19**	**0.0756**	**0.0135**	**0.0587**	**2623**
	ADE	[42,34]	[23,36,50]	9	0.1137	0.0322	0.0954	3563
	DE	[48,29]	[34,50,42]	19	0.0936	0.0225	0.0727	5819
	ES	[39,64]	[38,50,21]	2	0.1605	0.0948	0.1268	3823
	GA	[30,47]	[47,34,30]	26	0.1334	0.0256	0.1008	7136
SunSpots	**IDE**	**[44,63]**	**[35,50,14]**	**14**	**3.3834**	**0.0419**	**2.4868**	**5421**
	ADE	[30,42]	[47,48,15]	28	6.4325	0.0497	4.3808	5774
	DE	[31,48]	[35,34,30]	3	6.6567	0.0499	4.5041	6935
	ES	[30,49]	[7,30,24]	29	6.8555	0.0513	4.5994	4273
	GA	[44,16]	[20,33,30]	26	6.6037	0.0504	4.4675	7285
Coal-mine Gas concentration	**IDE**	**[50,48]**	**[10,8,25]**	**26**	**0.0493**	**0.1094**	**0.0332**	**2871**
	ADE	[33,48]	[8,14,37]	19	0.0556	0.1365	0.0382	3575
	DE	[30,23]	[14,4,44]	19	0.0521	0.1301	0.0362	4252
	ES	[40,64]	[50,39,48]	23	0.0524	0.1330	0.0371	3215
	GA	[30,47]	[47,48,15]	9	0.0538	0.1216	0.0348	3271

mechanism is a crucial feature extractor, and it performs a weighted sum operation. It will give high weight to important features and weaken useless features, the vector c is the extracted key features, and its formula is as follows:

$$c = \sum_{i=1}^{m} \beta_i v_i \tag{6}$$

where m is the sum of input time-steps of the GRU, v is the feature vector output by the GRU, and β represents the weight of the vector v.

In order to obtain β, we add a small neural network a(v) with softmax activation function to the attention model, the formula is as follows:

$$\beta_i = \frac{\exp(e_i)}{\sum_{k=1}^{m} \exp(e_k)} \tag{7}$$

where $e_i = a(v_i)$.

D. DIFFERENTIAL EVOLUTION AND ITS IMPROVEMENT

Differential evolution algorithm is a stochastic heuristic algorithm that is simple to use and has strong robustness and global excellence seeking ability[45]. Rainer Storn and Kenneth Price proposed the original and a few variants of the differential evolution [31], [38], [39], [46], [47], defining notations DE/x/y/z, where x specifies the mutation method, y represents the number of difference vectors, and z is the cross method.

1) STANDARD DE/BEST/1/BIN

In this paper, we use standard DE/best/1/bin [46] as the underlying algorithm template, in which the mutation method uses the best population individual to generate vectors, and the bin represents DE obtains the experimental population using the binomial distribution crossover method. In DE, we set population size, the number of generation, mutation and crossover operator as *NP*, *G*, *F* and *CR*, respectively. For each D-dimensional target vector, the calculate equations are the following:

$$x_{i,G}(i = 1, 2, \cdots, NP) \tag{8}$$

$$v_{i,G+1} = x_{best,G} + F \cdot (x_{r1,G} - x_{r2,G}) \tag{9}$$

$$u_{i,G+1} = (u_{1i,G+1}, u_{2i,G+1}, \cdots, u_{Di,G+1}) \tag{10}$$

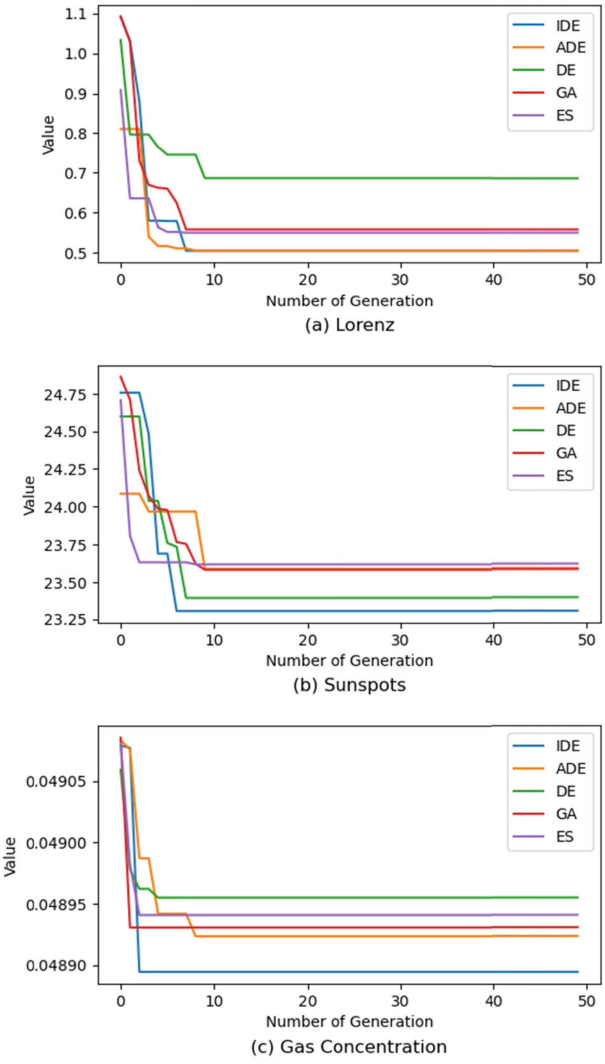
$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & if\ randb(j) \le CR\ or\ j = rnbr(i) \\ x_{ji,G} & if\ randb(j) > CR\ or\ j \ne rnbr(i) \end{cases}$$
$$i = 1, 2, \cdots, NP; j = 1, 2, \cdots, D \tag{11}$$

where $x_{i,G}$ is ith population individual of generation G, a mutant vector $v_{i,G+1}$ is generated according to (9), $x_{best,G}$ is the best individual of generation G, $r1$ and $r2$ are random indexes in range $\{1, 2, \cdots, NP\}$, F is an invariant operator $\in [0, 2]$, which determines the magnification ratio of the differential vector. As shown in (10) and (11), the trial vector $u_{i,G+1}$ is selected form the mutation vector $v_{ji,G+1}$ and original vector $x_{ji,G}$. CR is a constant operator $\in [0, 1]$, $randb(j)$ represents the jth estimated value of random number generator with the outcome $[0, 1]$, $rnbr(i)$ is a casually selected index $\in 1, 2, \cdots, D$.

2) IMPROVED DIFFERENTIAL EVOLUTION ALGORITHM

In DE/best/1/bin, F and CR are a real and constant factor, which are difficult to choose during the search process. Adaptive DE (ADE) [39], [48], [49] provides a way to solve this problem, which uses adaptive strategies with generation to choose F, the equation is expressed as [49]:

$$\lambda = \exp(1 - \frac{G_m}{G_m + 1 - G}), \quad F = F_0 \cdot 2^{\lambda} \tag{12}$$



(a) Lorenz

(b) Sunspots

(c) Gas Concentration

FIGURE 10. Convergence curve of evolution algorithms.

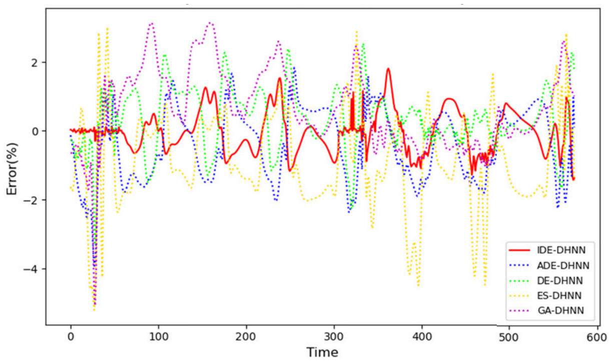
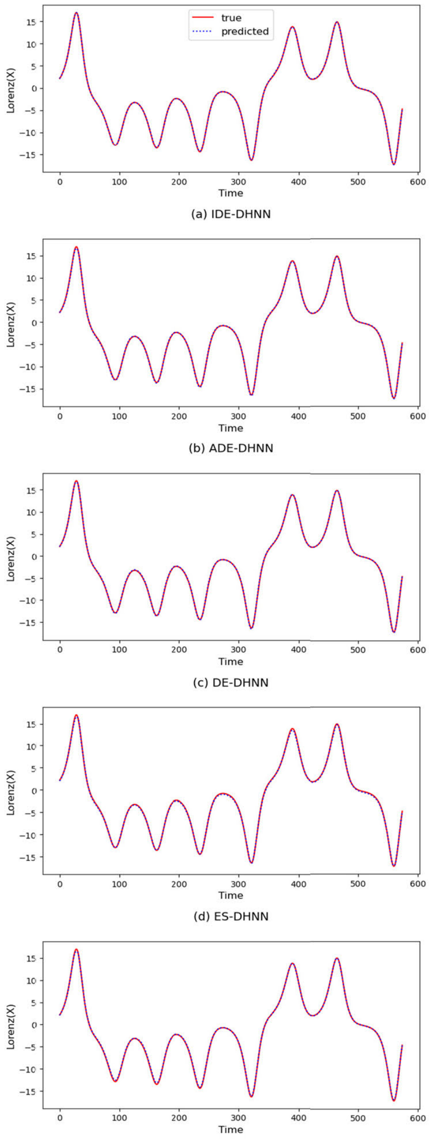


FIGURE 11. The prediction error Curve of Neuroevolution-DHNN on Lorenz datasets.



(a) IDE-DHNN

(b) ADE-DHNN

(c) DE-DHNN

(d) ES-DHNN

(e) GA-DHNN

FIGURE 12. Comparison of true values and predicted values of different neuroevolution-DHNN (Lorenz datasets).

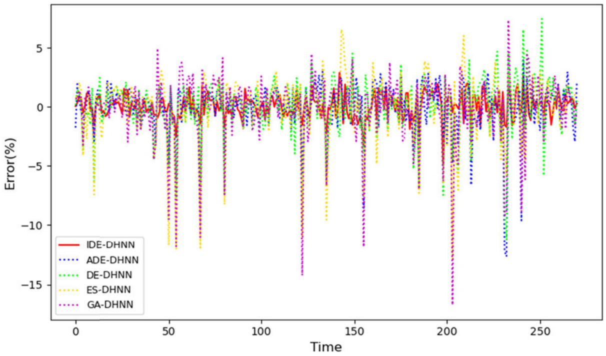


FIGURE 13. The prediction error curve of neuroevolution-DHNN on gas concentration datasets.

where F_0 is original mutation operator, G_m is the max generation, F trends and eventually equals F_0, G represents the current generation. and $1 \leq \mathrm{G} \leq \mathrm{G}_m$.

It is efficient to choose a proper mutation operator in the implementation. However, we found that F tend to lager factor while F_0 is large, and this affected the efficiency of

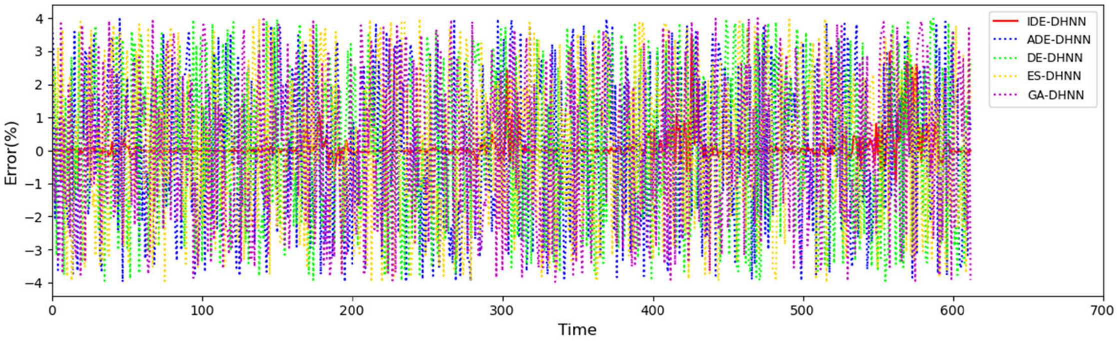


FIGURE 14. The prediction error curve of neuroevolution-DHNN on Sunspts datasets.

the search process. Thus, we improved (12) as follows:

$$\alpha = \frac{G_m}{G_m + 1 - G}, \lambda = \frac{1}{1+\exp(-\alpha)}, \quad F = F_0 \cdot 2^{\lambda} \tag{13}$$

where F_0 is original mutation operator, G_m is the max generation, F trends and eventually equals F_0, G represents the current generation, and $1 \leq \mathrm{G} \leq \mathrm{G}_m$.

As expressed in Fig.5, AF is the operator described by Eq. (12) and IF represents the mutation operator computed via Eq. (13). It is apparent that AF varies over a wide range, while IF varies over a smaller range, which can not only maintain the population diversity in the initial stage but also ensures the search efficiency.

In this paper, we use the Logistic chaotic mapping equation to compute CR. Chaotic disturbance not only allows CR to control the crossover probability and diversify the population but also accelerates the convergence. Chaotic CR calculated from the following formula:

$$CR_{G+1} = \mu \cdot CR_G \cdot (1 - CR_G) \tag{14}$$

where μ is a parameter, Eq.(14) is chaos when $3 \leq \mu \leq 4$, in the literature, $\mu = 4$. The change curve of CR is shown in Fig.5.

3) OPTIMIZATION OF HYBRID NEURAL NETWORK USING IMPROVED DIFFERENTIAL EVOLUTION

In this paper, we use improved differential evolution algorithm to optimize the topologies and time-steps on the hybrid neural network. In the optimization, the mean square error (MSE) is used as the evaluation criterion to select the best individual. It means that MSE is the fitness function, which computed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{15}$$

where y_i and $\hat{y}_i$ stand for raw and predictive values, n is the number of predicted points.

Algorithm 1 describes the process of IDE optimizing the hybrid neural network.

Algorithm 1 Improved Differential Evolution Optimizes Hybrid Neural Network

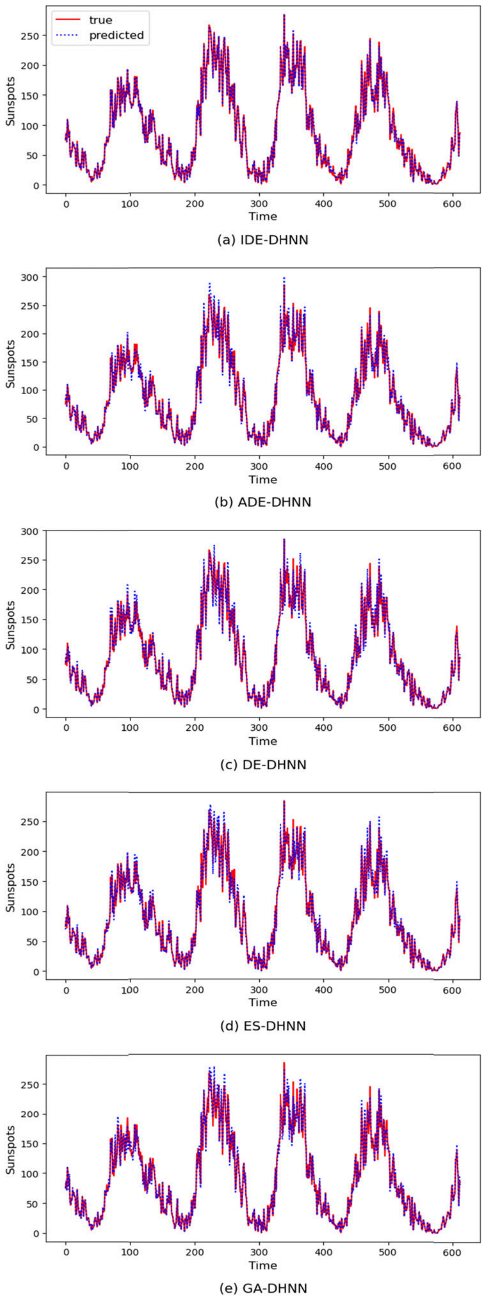
```
Step 1: Set control parameters: mutation factor $F_0$, crossover operator $CR_0$, population size $NP$ and max generation $MAX\_G$
Step 2: Randomly initialize a population of $NP$ individuals $x_{i,0} = (c1, c2, g1, g2, g3, l)$, where $c1$ and $c2$ is the size of the CNN filter, $g1$, $g2$, and $g3$ is the number of neurons of the GRU layers, and $l$ is the time-steps for forecasting. Set the generation number $G = 1$
Step 3: while the stopping criterion is not satisfied
  for $i = 1$ to $NP$
    Step 2.1Mutation:
    compute $F$ by Eq. (13)
    generate a mutant vector by Eq. (9)
    Step 2.2Crossover:
    compute $CR$ via Eq. (14)
    generate a trial vector by Eq. (11)
    Step 2.3Selection:
    set up a hybrid neural network according to each individual
    train prediction model
    compute the MSE on the validation set by Eq. (15), which has smaller value will be selected
  end for
  $G = G + 1$
  end while
```

III. EXPERIMENTS

In this section, we introduce the detail of data access, data preprocessing, and evaluation criteria.

A. DATA ACCESS

In this paper, we use two data sets to verify the predictive performance of the proposed model, including theoretical Lorenz datasets and a coal-mine gas concentration datasets.



(a) IDE-DHNN

(b) ADE-DHNN

(c) DE-DHNN

(d) ES-DHNN

(e) GA-DHNN

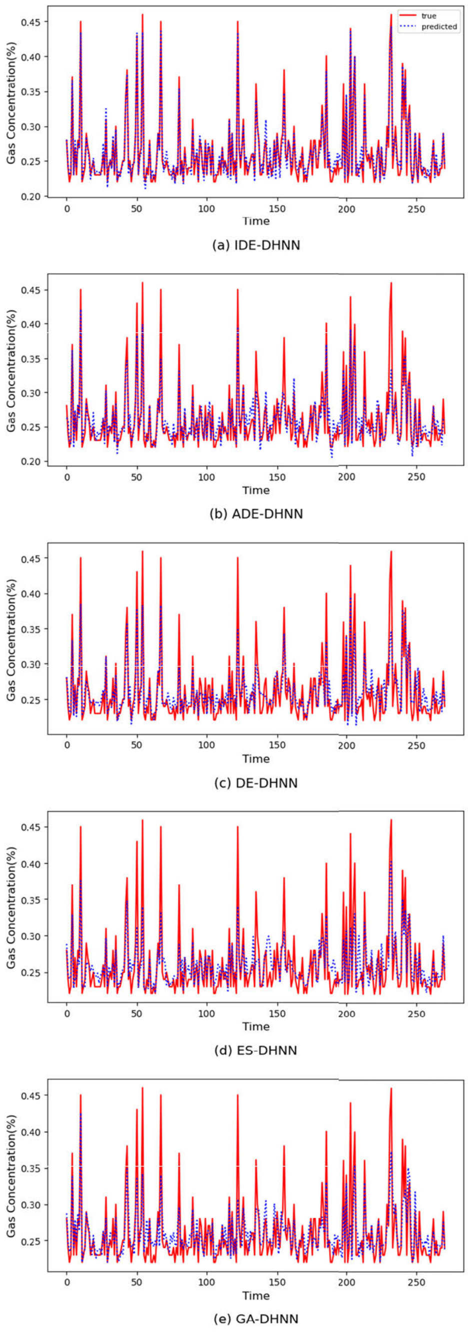
FIGURE 15. Comparison of true values and predicted values of different neuroevolution-DHNN (Sunspot datasets).

1) LORENZ CHAOTIC TIME SERIES

The equation of Lorenz chaotic mapping is:

$$\begin{aligned} \frac{dx}{dt} &= -a(x-y) \\ \frac{dy}{dt} &= -xz + \text{cx-y} \\ \frac{dz}{dt} &= xy - bz \end{aligned} \tag{16}$$

The initial value of equation selected as x = y = z =1, the parameters a =10, b =8/3, c =28. to ensure chaos, discarding the first 10,000 samples and select the last 3,000 samples as experiment data. Fig.6 is an example of the Lorenz time series, and we use the X variable of Lorenz to train and test the proposed deep hybrid model.



(a) IDE-DHNN

(b) ADE-DHNN

(c) DE-DHNN

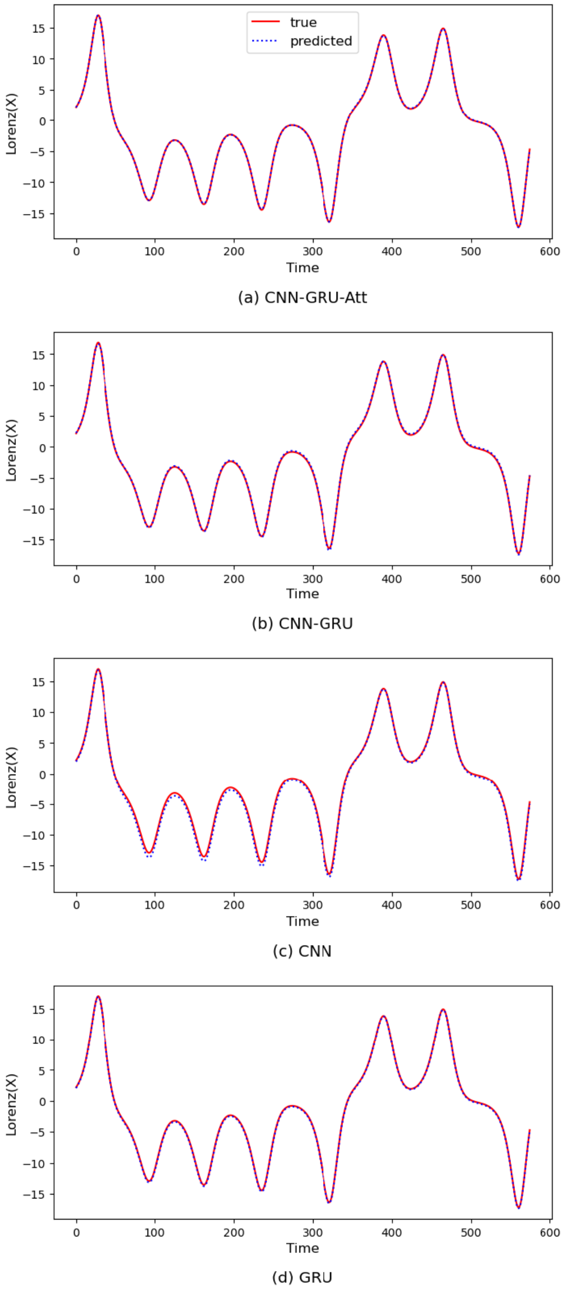
(d) ES-DHNN

(e) GA-DHNN

FIGURE 16. Comparison of true values and predicted values of different neuroevolution-DHNN (Gas concentration datasets).

2) MONTHLY MEAN TOTAL SUNSPOT NUMBER

Sunspots are common phenomena on the sun's photosphere that appear as spots darker than the surrounding areas. We collected monthly mean total sunspot numbers



(a) CNN-GRU-Att

(b) CNN-GRU

(c) CNN

(d) GRU

FIGURE 17. **Comparison of actual values and predicted values of different models (Lorenz datasets).**

form1749 to 2019, and 3240. records are valid and used in this paper. Fig.7 expresses the curve of monthly mean total sunspot number.

3) CHAOTIC COAL-MINE GAS CONCENTRATION TIME SERIES

In this paper, we also use a chaotic coal-mine gas concentration series to the proposed model. We captured the actual data of a mining face in Xingtai Mine, and 1464 records are valid and used in this paper. The Fig.7 shows the curve of gas concentration datasets.

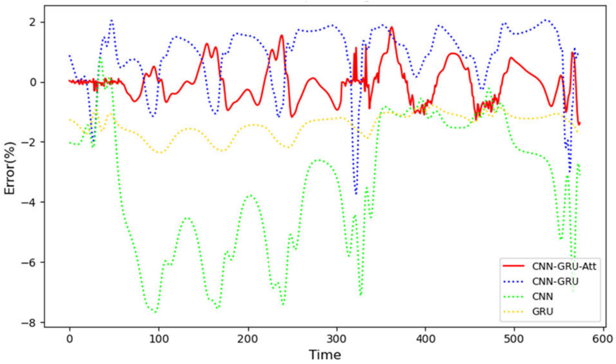


FIGURE 18. **The prediction error curve of different models (Lorenz datasets).**

B. DATA PREPROCESSING

1) PHASE SPACE RECONSTRUCTION

The emergence of the theory of phase space reconstruction provides a theoretical basis for forecasting chaotic time series. In the basic idea of phase space reconstruction, any variable in the system is determined by other variables interacting with each other. Therefore, any variable's development and change contain information on the development and change of other variables [50]. Packer *et al.* proposed that the phase space can be reconstructed by using the delayed coordinates of a variable in the dynamical system [50]. Takens Floris demonstrated that the dimensions of the original dynamical system could be recovered with the appropriate embedding dimension [51]. In this paper, we use the mutual information method [52] and Cao method [53] to determine the delay time τ and embedding dimension. Time-series lists as $\{x_1, x_2, \cdots, x_N\}$, the delay time is τ and embedding dimension is m. Phase space reconstruction $\bar{D} = \{X(t), Y(t)\}, t = 1, 2, \cdots, M$, where $M = N - (m - 1)\tau, X(t) = [X_t, X_{t+\tau}, \cdots, X_{t+(m-1)\tau}]$, $Y(t) = [X_{t+1}]$, The matrix is represented as follows:

$$X = \begin{bmatrix} x_1 & x_{1+\tau} & \cdots & x_{1+(m-1)\tau} \\ x_2 & x_{2+\tau} & \cdots & x_{2+(m-1)\tau} \\ \vdots & & \cdots & \\ x_{M-1} & x_{M-1+\tau} & \cdots & x_{M-1+(m-1)\tau} \end{bmatrix}, Y = \begin{bmatrix} x_2 \\ x_3 \\ \vdots \\ x_M \end{bmatrix}$$

2) DATA NORMALIZATION

It is necessary to normalize datasets in deep learning, which not only eliminates the magnitude and unify the data to the same scale but also enhance the convergence speed and prediction accuracy of the model. In this paper, we use normalization to unify phase space reconstruction datasets and original chaotic time series to a range between (0,1), and the normalization function can be expressed as:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{17}$$

The Fig.8 shows the process of data preprocessing.

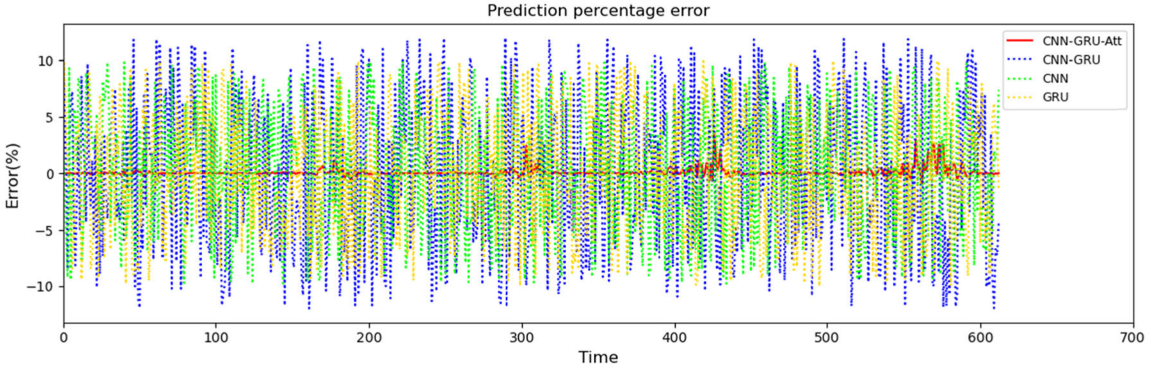


FIGURE 19. The prediction error curve of different models (Sunspots datasets).

C. EVALUATION CRITERIA

In the literature, we use two kinds of criteria to evaluate the performance of the prediction model, and there are root mean square error (RMSE) and mean absolute percentage error (MAPE), the calculation equations are the following:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{18}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{19}$$

where y_i and $\hat{y}_i$ stand for raw and predictive values, n is the number of predicted points.

D. TRAINING OF PREDICTION MODEL

In this paper, we choose the first 80% of datasets as the tra-ining datasets and the rest of 20% as the testing datasets. In the training phase of DHNN experiments, we use the impr-oved differential evolution algorithm to infer optimal topo-logies and time-steps for the proposed model. We use Keras to code experimental programs and implement the ES, GA, and DE using *The genetic and evolutionary algorithm tool-box with high performance in Python python(geatpy)* [54] in Python library. We also implement improved DE quickly by using geatpy. As the loss function, the mean square error is applied to compute the quantity that a model should seek to minimize during training. We also use *Adam* [55] to optimize the gradient of the stochastic objective function.

IV. ANALYSIS AND DISCUSSION

In this section, we analyze and discuss the prediction accuracy of the proposed hybrid model through two experiments. One of them is to optimize the hybrid neural network through different evolutionary algorithms, and the other is to compare the proposed model with other variant models.

A. VARIOUS EVOLUTION ALGORITHMS FOR HYBRID NEURAL NETWORK

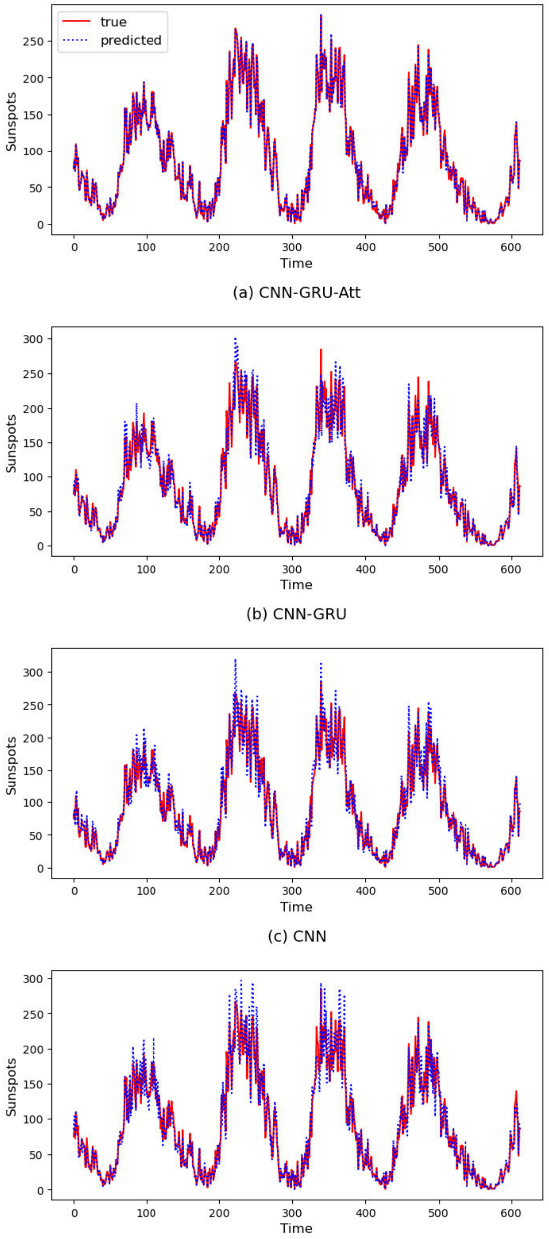
In this part, we analyze and discuss the predictive performance of the hybrid neural network, which optimized by different evolution algorithms. We use the differential evolution (IDE) algorithm, adaptive differential evolution (ADE) algorithm, standard differential evolution (DE) algorithm, evolution strategy (ES) [56], and genetic algorithm (GA) [57] to infer optimal topologies and time-steps for the hybrid neural network.

As shown in Fig.9, the IDE not only has a faster converg-ence speed but also achieves the lowest target value, which compares with other algorithms. From Fig.10 (a), it can be seen that the convergence value and convergence rate of IDE and ADE are similar, but the convergence rate of IDE is faster than ADE, and both of them are better than DE, ES, and GA. From Fig.10 (b) and (c), it is clear that the IDE has the smallest convergence value. From Table 1, it is obvious that IDE runs faster than DE, ADE, ES, and GA. Thus, it is proved that the IDE proposed in the literature can improve the convergence speed and reduce the optimization time.

In Table 1, we can notice that the RMSE, MAPE, and average prediction error of IDE-DHNN model are the lowest. The RMSE of IDE-DHNN model is obviously lower than other models, and the MAPE values of IDE-DHNN models are slightly higher than others. From Fig.12, it is clear that the values predicted by IDE-DHNN are close to actual values. And Fig.11 shows that the max percent-age error of IDE-DHNN is 1.5%, which is lower than ADE, DE, ES, and GA-DHNN.

TABLE 2. The prediction errors of experimental results.

Datasets	Model	RMSE	MAPE	Average error
Lorenz	**CNN-GRU-Att**	**0.0756**	**0.0335**	**0.0587**
	CNN-GRU	0.1338	0.1006	0.1203
	CNN	0.4085	0.1555	0.3454
	GRU	0.1492	0.0813	0.1493
Sunspots	**CNN-GRU-Att**	**3.3834**	**0.0419**	**2.4868**
	CNN-GRU	10.3999	0.0800	7.0994
	CNN	12.6620	0.0995	8.5569
	GRU	12.5915	0.0719	8.2467
Gas concentration	**CNN-GRU-Att**	**0.0493**	**0.1094**	**0.0332**
	CNN-GRU	0.0548	0.1276	0.0362
	CNN	0.0681	0.1856	0.0506
	GRU	0.0653	0.1688	0.0461



(a) CNN-GRU-Att

(b) CNN-GRU

(c) CNN

(d) GRU

FIGURE 20. Comparison of actual values and predicted values of different models (Sunspots concentration datasets).

On the other hand, IDE-HNN also has higher forecasting accuracy on the sunspot datasets. Table1 shows that the RMSE of IDE-DNHH is the lowest with a value of 3.3834. The MAPE value of IDE-DHNN is 0.0419, which is the minimum. Fig.14 and Fig.15 show that the accuracy of IDE-DHNN is higher than other models. Fig.13 and Fig.16 respectively show the curve of actual-predicted values and prediction error on gas concentration datasets. We notice that IDE-DHNN performance well, and the max error is 5%, which is far lower than other forecasting models. Varieties of evaluation criteria all verify that IDE-HNN has excellent forecasting performance, and it is the right choice for chaotic time series prediction.

B. COMPARISON OF VARIANT PREDICTION MODELS

In the literature, the hybrid neural network includes three parts, which are CNN, GRU, and attention model. We use three various forecasting variant models to verify the fore-casting accuracy of the proposed hybrid neural network, which are hybrid CNN-GRU model, single CNN model, and single GRU model. Table 2 shows the RMSE, MAPE, and average error of various prediction models.

As described in Table 2, the RMSE, MAPE, and the average error of the CNN-GRU-Att prediction model are far lower than CNN-GRU, CNN, and GRU model. The CNN-GRU-Att has the lowest RMSE, MAPE, and the average error on three different datasets.

On the Lorenz datasets, the RMSE of the various prediction models is quite diverse. The RMSE of CNN-GRU-Att is the lowest with a value of 0.0756. Besides, the MAPE of CNN-GRU-Att is obviously lower than that of other models. It is clear that CNN-GRU performs well too, which is better than CNN and GRU. From Fig.17 and Fig.18, it is evident that CNN-GRU-Att has higher prediction accuracy, and the forecasting error is controlled within 1.5%.

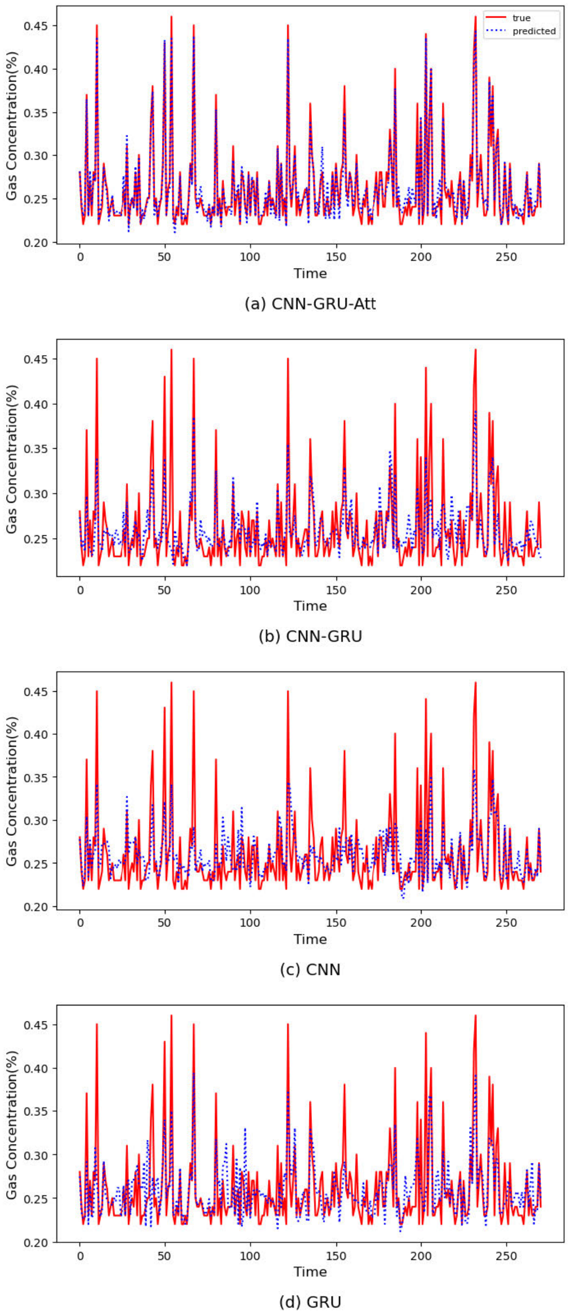
Fig.19 and Fig.20 respectively show the curve of perdic-tion error and true-predicted values of four models. As shown in Fig.20 (a), the values forecasted by CNN-GRU-Att are close to actual values. The prediction error curve represents that the proposed CNN-GRU-Att model has higher prediction accuracy. Table 1 shows that the RMSE and MAPE of CNN-GRU-Att are the lowest with a value of 3.3834 and 0.0419, respectively.

From Table 2, Fig.21, and Fig.22, it is evident that the hybrid model with CNN, GRU, and attention model performs very well on the gas concentration datasets. As expressed in Fig.22, the prediction accuracy of CNN-GRU-Att is higher than the others, and the prediction error is controlled within 5%. It is also clear that the prediction error of CNN-GRU, CNN, and GRU models are higher than the CNN-GRU-Att model. It is worth noting that the gas concentration dataset is significantly less than the other two datasets, but the proposed hybrid neural network still has high predictive accuracy.

C. DISCUSSION

Based on previous trial results, we can summarize the following findings:

(1). Neuro-evolution is an excellent choice for designing topologies and searching for some hyperparameters for neural networks. The improved differential evolution proposed in this paper performs well in optimizing hybrid neural network. We can see that the IDE can improve convergence speed and reduce running times simultaneously. The most important thing is that the optimization result of IDE for hybrid neural network achieves better prediction performance and higher prediction accuracy.



(a) CNN-GRU-Att

(b) CNN-GRU

(c) CNN

(d) GRU

FIGURE 21. Comparison of actual values and predicted values of different models (Gas concentration datasets).

(2). The hybrid neural network plays an essential role in the prediction model, and the prediction accuracy is low while using the single CNN and GRU model, and hybrid neural network without attention model. It is because that chaotic time series expands with high dimensions in the phase space reconstruction, and in this condition, the chaotic system contains rich spatial information. At the same time, the original chaotic time series also contains rich temporal characteristics. It is difficult to fully extract the temporal or spatial characteristics from the chaotic system with a single CNN or GRU model. Besides, even if the hybrid CNN-GRU model can extract temporal and spatial features, but the prediction accuracy is affected by lots of non-key features. At this point, the attention model plays a crucial role in extract temporal-spatial features. It gives high weights to critical features while weakening non-critical features. From the previous trial results, it is evident that the hybrid neural network optimized by IDE and added attention mechanism has high prediction accuracy.

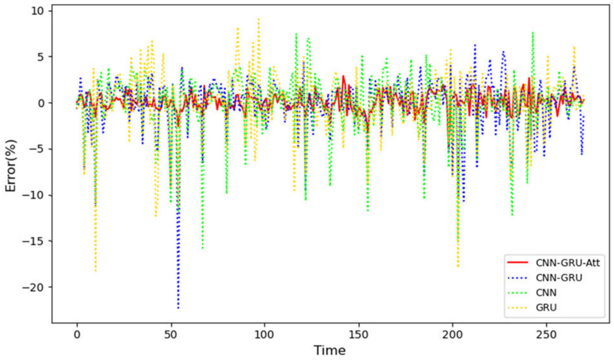


FIGURE 22. The prediction error curve of different models (Gas concentration datasets).

V. CONCLUSION

In this paper, we propose a hybrid model to forecast the chaotic time series, which includes convolutional neural network, gated recurrent unit, attention mechanism, and improved differential evolution algorithm. The proposed hybrid model can be summarized as two parts, one is the deep hybrid neural network, and the other is neuroevolution based on IDE. In the deep hybrid neural network, we use CNN and GRU to extract spatial and temporal features from phase space reconstruction and time series, respectively. The attention model can extract critical spatio-temporal features, which can improve prediction accuracy. In the neuroevolution, we first develop the IDE with an adaptive mutation operator and dynamic chaos crossover operator, which can improve convergence speed and reduce optimization time. Then, we use IDE to infer appropriate topologies and time-steps for the deep neural network. Simulation experiment results show that IDE can improve convergence speed and reduce optimization time. Furthermore, it also can acquire a lower prediction error. Thus, the deep hybrid neural network is an excellent choice for chaotic time series prediction.

REFERENCES

[1] R. Carrasco, M. Vargas, I. Soto, G. Fuertes, and M. Alfaro, "Copper metal price using chaotic time series forecating," *IEEE Latin Amer. Trans.*, vol. 13, no. 6, pp. 1961–1965, Jun. 2015.

[2] T. Chaodong, F. Gang, L. Ping, P. Zhenhua, Y. Ruogu, and L. Jingjia, "Prediction model of unit consumption for oilfield water injection based on the grain of association rule and chaotic time series," in *Proc. IEEE Int. Conf. Adv. Manuf. (ICAM)*, Nov. 2018, pp. 148–151.

[3] N. Safari, C. Y. Chung, and G. C. D. Price, "Novel multi-step short-term wind power prediction framework based on chaotic time series analysis and singular spectrum analysis," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 590–601, Jan. 2018, doi: 10.1109/tpwrs.2017.2694705.

[4] C. Rodriguez Rivero, J. Pucheta, A. Orjuela Canon, L. Franco, Y. Tupac Valdivia, P. Otano, and V. Sauchelli, "Noisy chaotic time series forecast approximated by combining Reny's entropy with energy associated to series method: Application to rainfall series," *IEEE Latin Amer. Trans.*, vol. 15, no. 7, pp. 1318–1325, Jun. 2017.

[5] N. Li, J. Tuo, and Y. Wang, "Chaotic time series analysis approach for prediction blood glucose concentration based on echo state networks," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 2017–2022.

[6] A. D. Pano-Azucena, E. Tlelo-Cuautle, and S. Tan, "Electronic system for chaotic time series prediction associated to human disease," in *Proc. IEEE Int. Conf. Healthcare Informat. (ICHI)*, Jun. 2018, pp. 323–327.

[7] S. Iqbal, X. Zang, Y. Zhu, H. M. A. A. Saad, and J. Zhao, "Nonlinear time-series analysis of different human walking gaits," in *Proc. IEEE Int. Conf. Electro/Inf. Technol. (EIT)*, May 2015, pp. 025–030.

[8] Y. Bao, C. Yi, J. Jiang, Y. Xue, and Y. Dong, "Implementation of chaotic analysis on retweet time series," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining ASONAM*, Aug. 2015, pp. 1225–1231.

[9] M. Baosong and S. Chenguang, "Prediction models for network multi-source dissemination of information based on multivariate chaotic time series," in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, Dec. 2017, pp. 767–771.

[10] S. M. Tabatabaie Nezhad, M. Nazari, and E. A. Gharavol, "A novel DoS and DDoS attacks detection algorithm using ARIMA time series model and chaotic system in computer networks," *IEEE Commun. Lett.*, vol. 20, no. 4, pp. 700–703, Apr. 2016.

[11] S. J. Abdulkadir and S.-P. Yong, "Lorenz time-series analysis using a scaled hybrid model," in *Proc. Int. Symp. Math. Sci. Comput. Res. (iSMSC)*, May 2015, pp. 373–378.

[12] A. Verikas, Z. Kalsyte, M. Bacauskiene, and A. Gelzinis, "Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: A survey," *Soft Comput.*, vol. 14, no. 9, pp. 995–1010, Jul. 2010.

[13] Z. Shi and M. Han, "Support vector echo-state machine for chaotic time-series prediction," *IEEE Trans. Neural Netw.*, vol. 18, no. 2, pp. 359–372, Mar. 2007.

[14] M. Ardalani-Farsa and S. Zolfaghari, "Chaotic time series prediction with residual analysis method using hybrid Elman–NARX neural networks," *Neurocomputing*, vol. 73, nos. 13–15, pp. 2540–2553, Aug. 2010.

[15] M. F. C. Nhabangue, G. N. Pillai, and M. L. Sharma, "Chaotic time series prediction with functional link extreme learning ANFIS (FL-ELANFIS)," in *Proc. Int. Conf. Power, Instrum., Control Comput. (PICC)*, Jan. 2018, pp. 1–6.

[16] M. Xu, M. Han, T. Qiu, and H. Lin, "Hybrid regularized echo state network for multivariate chaotic time series prediction," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2305–2315, Jun. 2019.

[17] Y. Li, R. W. Liu, Z. Liu, and J. Liu, "Similarity grouping-guided neural network modeling for maritime time series prediction," *IEEE Access*, vol. 7, pp. 72647–72659, 2019.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[20] M. Sangiorgio and F. Dercole, "Robustness of LSTM neural networks for multi-step forecasting of chaotic time series," *Chaos, Solitons Fractals*, vol. 139, Oct. 2020, Art. no. 110045.

[21] N. Boullé, V. Dallas, Y. Nakatsukasa, and D. Samaddar, "Classification of chaotic time series with deep learning," *Phys. D, Nonlinear Phenomena*, vol. 403, Feb. 2020, Art. no. 132261.

[22] I. E. Livieris, E. Pintelas, and P. Pintelas, "A CNN–LSTM model for gold price time-series forecasting," *Neural Comput. Appl.*, pp. 1–10, 2020, doi: 10.1007/s00521-020-04867-x.

[23] Y. Li, R. W. Liu, Q. Ma, and J. Liu, "EMD-based recurrent neural network with adaptive regrouping for port cargo throughput prediction," in *Proc. ICONIP*, Siem Reap, Cambodia, Dec. 2018, pp. 499–510.

[24] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," *Nature Mach. Intell.*, vol. 1, no. 1, pp. 24–35, Jan. 2019.

[25] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, Jun. 2002.

[26] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley, and J. Clune, "Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning," 2017, *arXiv:1712.06567*. [Online]. Available: http://arxiv.org/abs/1712.06567

[27] E. Conti, V. Madhavan, F. P. Such, J. Lehman, K. O. Stanley, and J. Clune, "Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents," in *Proc. NIPS*, Montreal, QC, Canada, 2018, pp. 5032–5043.

[28] J. Lehman and R. Miikkulainen, "Neuroevolution," *Scholarpedia*, vol. 8, no. 6, 2013, Art. no. 030977.

[29] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "ABCNN: Attention-based convolutional neural network for modeling sentence pairs," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 259–272, Dec. 2016.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, Long Beach, CA, USA, 2017, pp. 6000–6010.

[31] C. Raffel and D. P. W. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," 2015, *arXiv:1512.08756*. [Online]. Available: http://arxiv.org/abs/1512.08756

[32] Y. Li, Z. Zhu, D. Kong, H. Han, and Y. Zhao, "EA-LSTM: Evolutionary attention-based LSTM for time series prediction," *Knowl.-Based Syst.*, vol. 181, Oct. 2019, Art. no. 104785.

[33] J. Hu and W. Zheng, "Multistage attention network for multivariate time series prediction," *Neurocomputing*, vol. 383, pp. 122–137, Mar. 2020, doi: 10.1016/j.neucom.2019.11.060.

[34] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," 2017, *arXiv:1704.02971*. [Online]. Available: http://arxiv.org/abs/1704.02971

[35] Y. Liu, C. Gong, L. Yang, and Y. Chen, "DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction," *Expert Syst. Appl.*, vol. 143, Apr. 2020, Art. no. 113082, doi: 10.1016/j.eswa.2019.113082.

[36] J. Hu and W. Zheng, "A deep learning model to effectively capture mutation information in multivariate time series prediction," *Knowl.-Based Syst.*, vol. 203, Sep. 2020, Art. no. 106139.

[37] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*. [Online]. Available: http://arxiv.org/abs/1409.1259

[38] R. Storn and K. Price, "Minimizing the real functions of the ICEC'96 contest by differential evolution," in *Proc. IEEE Int. Conf. Evol. Comput.*, May 1996, pp. 842–844.

[39] H. A. Abbass, "The self-adaptive Pareto differential evolution algorithm," in *Proc. Congr. Evol. Comput., CEC*, May 2002, pp. 831–836.

[40] Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, p. 326.

[41] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv:1408.5882*. [Online]. Available: http://arxiv.org/abs/1408.5882

[42] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[43] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. ICML*, Atlanta, GA, USA, 2013, pp. 1310–1318.

[44] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Proc. NIPS*, Montreal, QC, Canada, 2014, pp. 2204–2212.

[45] S. Das and P. N. Suganthan, "Differential evolution: A survey of the State-of-the-Art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[46] K. V. Price, "Differential evolution: A fast and simple numerical optimizer," in *Proc. North Amer. Fuzzy Inf. Process.*, Jun. 1996, pp. 524–527.

[47] R. Storn, "On the usage of differential evolution for function optimization," in *Proc. North Amer. Fuzzy Inf. Process.*, Jun. 1996, pp. 519–523.

[48] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2005, pp. 1785–1791.

[49] Y. Xue-feng, Y. Juan, Q. Feng, and D. Jun-wei, "Kinetic parameter estimation of oxidation in supercritical water based on modified differential evolution," *J. East China Univ. Sci. Technol.*, vol. 32, no. 1, pp. 94–97, 2006.

[50] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, "Geometry from a time series," *Phys. Rev. Lett.*, vol. 45, no. 9, p. 712, Sep. 1980.

[51] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical Systems and Turbulence*. Berlin, Germany: Warwick, 1981, pp. 366–381.

[52] J. M. Martinerie, A. M. Albano, A. I. Mees, and P. E. Rapp, "Mutual information, strange attractors, and the optimal estimation of dimension," *Phys. Rev. A, Gen. Phys.*, vol. 45, no. 10, pp. 7058–7064, May 1992.

[53] L. Cao, "Practical method for determining the minimum embedding dimension of a scalar time series," *Phys. D, Nonlinear Phenomena*, vol. 110, nos. 1–2, pp. 43–50, Dec. 1997.

[54] Jazzbin. (2020). *Geatpy: The Genetic and evolutionary Algorithm Toolbox With High Performance in Python*. [Online]. Available: http://www.geatpy.com/

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[56] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies—A comprehensive introduction," *Natural Comput.*, vol. 1, no. 1, pp. 3–52, 2002.

[57] J. Holland, "Adaptation in natural and artificial systems," Ph.D. dissertation, Dept. Eng., Michigan Univ., Detroit, MI, USA, 1975.

WEIJIAN HUANG received the Ph.D. degree from Tianjin University, China. He is currently a Professor with the School of Information and Electrical Engineering, Hebei University of Engineering. His research interests include cloud computing, big data, and machine learning.

YONGTAO LI received the B.S. degree from Anhui Xinhua University, China, in 2018. He is currently pursuing the M.S. degree with the Hebei University of Engineering. His research interest includes deep learning.

YUAN HUANG was born in Hebei, China, in 1987. He received the B.S. and M.S. degrees from the Hebei University of Engineering, in 2010 and 2013, respectively, and the Ph.D. degree from Yanshan University, in 2017. Since 2017, he has been working as a Teacher with the School of Information and Electrical Engineering, Hebei University of Engineering. He has published 11 articles. His research interests include data mining and machine learning.