Short-Term Load Forecasting Based on a Hybrid Neural Network and Phase Space Reconstruction

YUAN HUANG\textsuperscript{1}, RUIXIAO ZHAO\textsuperscript{1}, QIANYU ZHOU\textsuperscript{2}, AND YUXING XIANG\textsuperscript{1}

\textsuperscript{1}School of Information and Electrical Engineering, Hebei University of Engineering, Handan, Hebei 056038, China
\textsuperscript{2}School of Earth Science and Engineering, Hebei University of Engineering, Handan, Hebei 056038, China

Corresponding author: Qianyu Zhou (zhouqianyu@hebeu.edu.cn)

This work was supported in part by the Natural Science Foundation of China under Grant 61772451, and in part by the Youth Foundation of Hebei Province under Grant D2021402043.

ABSTRACT Most current short-term load forecasting models have difficulty in simultaneously taking into account the time-series nature of load data, the non-linear characteristics, and the ineffectiveness of extracting potential high-dimensional features from historical series. To solve this problem, we propose a hybrid neural algorithm model (DPCL). In the DPCL, we use convolutional neural networks to obtain the high-dimensional spatial features of the phase space reconstruction of the load time series. Then, we combine the obtained high-dimensional spatial features with the external influence features extracted in the Pearson correlation analysis. Long and short-term memory networks retrieve Spatio-temporal features through the connection layer and obtain prediction results. In addition, there are problems of network gradient degradation and overfitting during the training process. We use an improved differential evolutionary algorithm to optimize the topology and time step of the hybrid neural network. We use the public dataset of a European utility and the loaded dataset of a Chinese mathematical competition as practical arithmetic examples. Experiments have higher prediction accuracy and faster prediction speed compared with other traditional algorithms.

INDEX TERMS Phase space reconstruction, convolutional neural network, long short-term memory, improved differential evolution.

I. INTRODUCTION

With the development of the economy, the power demand is constantly expanding [1]. Power dispatching is a crucial link for the safe operation of power grids, reliable external power supply, and orderly production of all kinds of electric power [2]. Load forecasting is the basis of power dispatching [3], which provides information and basis for the planning and construction of power grids and power sources as well as the operational decisions of power grid enterprises and power grid users, and is especially important for the economic and safe operation of power systems [4].

The electrical load system is a complex dynamic system that has internal periodicity and load impact correlation indicating some measurable pattern [5]. The main research methods in the field of short-term load forecasting are traditional statistical methods [6], [7] and machine learning methods [8]. Traditional statistical methods include linear regression [9], the gray model method [10], the fuzzy prediction method [11], and the autoregressive integrated sliding average model [12]. These methods are simple and easy to implement, but they require a high level of raw data processing and time series stability. However, short-term load forecasting has characteristics such as complexity and nonlinearity; in addition, there are many factors affecting load forecasting, which cannot present a stable and orderly state. In the field of machine learning research, random forests [13], support vector machines [14], expert systems [15], artificial neural network forecasting methods [16], and deep learning methods [17] are commonly used for short-term load forecasting. These methods offer a fast convergence of the algorithmic model through a simple internal structure, but they can destroy the temporal integrity of the load data, and therefore the prediction accuracy is frequently not ideal.

With the development of deep learning, researchers obtained new ideas, and deep learning algorithms such as convolutional neural networks [18], recurrent neural networks [19], and deep belief networks [20] were gradually applied to electric load forecasting. Among them, CNN uses convolutional pooling to extract data features, which can...
reduce the error caused by human extraction of features and is widely cited in the fields of image and speech [21]. In the literature [22], by using CNN models to extract features of input data and capture seasonal cycles for load forecasting, it was found to be more accurate for learning highly nonlinear sequences in load data; however, a single CNN model is difficult to learn load dynamics better when load data are more volatile and unstable. RNN compared to other networks can model dynamic time-series data. In 2019, Fan et al. [23] proposed to predict the output power of PV systems using long-short memory neural networks (LSTM), which was evaluated using hourly datasets from different locations over a year, and the LSTM further reduced the prediction error and demonstrated better prediction results [24]. However, when selecting the input quantity, they either did not select any more suitable influencing factor as the input quantity based on the correlation between the influencing factor and the power load data, or the processing of the input quantity was too cumbersome. The literature [25], [26] proposes load forecasting based on the LSTM algorithm, combining CNN extracted features from different perspectives while using deep learning in both spatial and temporal domains to predict short-term load forecasts. Although these researches have achieved good results, the models they used, especially the artificial neural network models, with excessive input data, implied layers, and implied layer nodes are likely to lead to overfitting, gradient disappearance, and the explosion of the network training, and the single neural network models generally exhibit shortcomings such as insufficient prediction accuracy.

In recent years, neuro-evolution is a branch in the field of artificial intelligence and machine learning that attempts to design and build neural networks through evolutionary algorithms [27]–[29]. Genetic algorithms [30] and evolutionary strategies [31] have achieved excellent results on the topology and weights of neural networks, we can determine the structure of the appropriate network for a specific problem and obtain excellent prediction performance. We consider a differential evolutionary algorithm to design the topology and weights of the neural network model, and to denote the appropriate prediction time step. However, the standard differential evolution algorithm and adaptive differential evolution algorithm have slow narrow convergence speed and longer running time [32]–[35]. In this way, we improve the DE algorithm by improving the variational operator and the crossoverb operator.

To address the problems of existing prediction methods, we propose a combined prediction model (DPCL) which combines convolutional neural network (CNN), long short-term memory (LSTM), and an improved differential evolutionary algorithm. Firstly, the load time series are phase space reconstructed and data normalized; Then convolutional neural network (CNN) performs spatial feature extraction on the reconstructed phase space of load time series and obtains high-dimensional features of complex dynamic changes of load; Meanwhile, Pearson correlation analysis is applied to the time series composed of original multidimensional input variables for correlation analysis, and select the influencing factors with greater correlation with the power load data as input quantities, so that the original data can achieve the reduction and optimization. After that, the spatial feature sequences extracted with CNN are fused into new input features, and a long-term short-term memory network (LSTM) extracts temporal features. In addition, too many neural network nodes are likely to lead to problems such as overfitting and gradient explosion in network training, thereby improving the differential evolution algorithm is utilized to optimize the topology and time step of the hybrid neural network.

The paper is organized as follows. Section II describes the data processing. In section III, we introduce CNN, LSTM, and improved ADE. In section IV, we analyze and discuss the experimental results. Finally, section V presents the conclusions and future work directions.

II. DATA PREPROCESSING

The research data in this thesis is drawn from two main sources: an electrical mathematical modeling competition and a European load dataset, which influencing factors are temperatures, humidity, and holidays. $X = \{x_1, x_2, \ldots, x_n\}$ is an M-dimensional multivariate time series, $x_1$ is the short-term load time series, and $x_2, x_3, \ldots, x_n$ is the time series of influencing factors.

The chaotic time series $x_1$ reconstructs in phase space, and $X$ needs to be normalized and de-normalized to the range of (0,1), then transfer the normalized data into the model. The calculation formula is as follows:

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

All these factors can impact the efficiency and effectiveness of Load, so we selected the Pearson correlation coefficients to correlate the time series $F = \{x_1', x_2', \ldots, x_n'\}$. Figure 1 shows the process of data preprocessing.

A. RECONSTRING PHASE SPACE

Reconstructing the phase space is the theoretical basis of chaotic systems, which was first articulated by Packard et al. [36], and usually uses the delayed coordinate method to reconstruct the phase space. According to Takens’s [37] theorem, determining the embedding dimension $m$ and the delay time $\tau$, the univariate load values can
follows short-term load and influencing factors. The formula is as correlation coefficient analyzes the correlation between influence short-term load forecasting $X$ and $Y$. The structure of CNN model.

$$X = \begin{pmatrix} x_1 & x_1+\tau & \cdots & x_1+(m-1)\tau \\ x_2 & x_2+\tau & \cdots & x_2+(m-1)\tau \\ \vdots & \vdots & \ddots & \vdots \\ x_N & x_N+\tau & \cdots & x_N+(m-1)\tau \end{pmatrix}$$

B. PEARSON CORRELATION COEFFICIENT

To establish whether weather and other factors $Y$ influence short-term load forecasting $X$, the Pearson correlation coefficient analyzes the correlation between short-term load and influencing factors. The formula is as follows:

$$\rho_{x,y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - (E(X))^2}\sqrt{E(Y^2) - (E(Y))^2}}$$

The correlation coefficient of the two variables $\rho_{x,y}$ is reduced to the range of $[-1,1]$, and the mathematical expectation $E$. Table 1 shows the correlation between $X$ and $Y$.

As shown in Table 2, the results of the Pearson correlation coefficient analysis. The correlation of temperature is the strongest, and the correlation of maximum temperature, minimum temperature, and average temperature with load is strong, and the correlation of working days and relative humidity with load data is also certain. The above analysis shows that the inputs to the model are mainly the historical load and three characteristic factors of working days, temperature, and humidity.

III. DPCL ALGORITHM

A. CNN MODEL

In this paper, we use a 1D convolutional neural network, to extract the internal features of the electric load data, dig out the interrelationship between the data, combine them into high-level features, and transform the structure of the fully connected layer [40]. So that the extracted features can be input to the LSTM to better learn the cycle change law of the load data and the demand law. LSTM can learn better the cycle change law of the load data and the demand law from the extracted features expression of the output feature vector of the CNN layer is:

$$x_t = f(w \otimes x^t + b) \cdot u(x^t) = \sum_{j=0}^{N} f(i-j)u(j)$$

where $w$ is the weights and $b$ is the biases, and $f$ is convolution functions, and $u$ is pooling functions.

B. LONG SHORT-TERM Memory(LSTM)

Recurrent Neural Network [41] (RNN) has problems such as gradient disappearance or gradient explosion as time passes and the number of layers of the network increases [42]. To solve the long-term dependency problem in neural networks, a special recurrent neural network-LSTM is generated [43], where each hidden layer is no longer a single neural network.

The structure of the LSTM memory module is shown in Figure 3. The LSTM consists of input gate, forgetting gate, and output gate. The input gate $i_t$ is the information input at the moment $t$. The forgetting gate $f_t$ determines whether the memory unit stores or forgets information. Suppose the current moment is $t$, then $c_{t-1}$ is the state of the memory cell at the previous moment, and $\tilde{c}_t$ is the current input state information. $h_t$ and $o_t$ are the output values of the current LSTM. The process of state update and information output by the memory module is as follows:

$$i_t = \delta(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$f_t = \delta(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

$$o_t = \delta(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \odot \tanh(c_t)$$

where $W$ and $b$ stand for weights and biases, $\delta$ is a sigmoid function, $\tanh$ represents activation function, $h_t$ is the output of the LSTM cell, $t$ represents the current time state.
TABLE 2. Correlation analysis of influencing factors.

| Working days | Maximum temperature °C | Minimum temperature °C | Average temperature °C | Amount of rainfall (mm) | Relative humidity (Average) |
|--------------|--------------------------|-------------------------|------------------------|------------------------|---------------------------|
| Load         | 0.170778                 | 0.607370                | 0.638024               | 0.638352               | 0.072843                  | 0.116554                  |

C. DIFFERENTIAL EVOLUTION AND ITS IMPROVEMENT

1) STANDARD DIFFERENTIAL EVOLUTIONARY ALGORITHM
The standard Differential Evolution (DE) algorithm is an optimization algorithm using real vector encoding that searches randomly in continuous space to find the optimal solution [44], and the most commonly used strategy is DE/rand/1/bin. The specific steps of the standard DE algorithm are as follows.

a: INITIALIZING THE POPULATION
Initial population \( \left[ x_{i,j}^{\text{min}}, x_{i,j}^{\text{max}} \right] \), \( i = 1, 2, \ldots, NP; j = 1, 2, \ldots, N \), where NP is the initialized population size as the 0th generation population and N is the number of spatial dimensions. The expressions are:

\[
x_{i,j}^0 = x_{i,j}^{\text{min}} + \text{rand}(0,1)(x_{i,j}^{\text{max}} - x_{i,j}^{\text{min}})
\]

where \( \text{rand}(0,1) \) denotes a random number that follows a uniform distribution in the (0,1) interval.

b: VARIANT OPERATION
The variation operation uses a differential policy to generate variant individuals, the expression is:

\[
v_{i,j}^g = x_{i,j}^g + F \cdot (x_{i,j}^g - x_{r_1,j}^g), \quad i \neq r_1 \neq r_2 \neq r_3
\]

where \( v_{i,j}^g \) is the individual of the \( i \)th variant, \( x_{r_1,j}^g, x_{r_2,j}^g, x_{r_3,j}^g \) for three mutually dissimilar parent individuals, \( i = 1, 2, \ldots, NP \), \( r_1, r_2, r_3 \) are three different Paternal individuals with each other, and \( F \) is the general value of the variance factor [0,2].

c: CROSSOVER OPERATION
To increase the diversity of the population, the various individuals and the target individuals are randomly generated as crossover individuals. The specific calculation process is shown in formula (1).

\[
u_{j,i}^{g+1} = \begin{cases} v_{j,i}^g, & \text{if } r_j \leq C_r or j = j_{rand} \\ x_{j,i}^g, & \text{otherwise} \end{cases}
\]

Among them, \( r_j \in [0,1] \), \( c_r \) is the crossover factor, which generally takes values in the range of [0,1], \( j_{rand} \in [1, 2, \ldots, NP] \) is a random integer.

d: SELECT OPERATION
The selection operation is a competition rule through the elimination of superiority in the target values of parents and children, the Selection of individuals more suitable for the populations to reproduce, and thus the orientation of the population to want the optimal solution. the expression is:

\[
x_{i,j}^{g+1} = \begin{cases} u_{i,j}^g, & \text{if } f(u_{i,j}^{g+1}) \leq f(x_{i,j}^g) \\ x_{i,j}^g, & \text{otherwise} \end{cases}
\]

2) ADE ALGORITHM
The standard differential evolution algorithm mainly includes three control parameters with fixed values of crossover factor \( CR \), variation factor \( F \), and population size, in terms of specific problems, the values of the control parameters vary for different optimization results, and the appropriate values can make the optimization results better. \( F \) and \( CR \) are critical operators that affect the search efficiency, and it is difficult to determine the constant real operator in the search process. When the value of \( F \) is too large, the efficiency of DE search becomes lower and the accuracy of its search solution is lower; when \( F \) is too small, the population diversity of the algorithm will be declined, and the phenomenon of “premature” may occur. This problem is
solved in the adaptive differential evolution algorithm by the adaptive variation factor $F$ proposed in the literature [41]:

$$\lambda = \exp(1 - \frac{G_m}{G_m + 1 - G}), \quad F = F_0 \cdot 2^\lambda$$  \hspace{1cm} (14)

In the equation, $F_0$ is the initialized scaling factor, $G_m$ is the maximum number of iterations, $G$ is the current number of iterations, $F$ is gradually approaching $F_0$, and $G$ takes values in the range $[1, G_m]$. It is possible to make the adaptive variation factor gradually gravitate to a good direction, but it will become less effective due to the change of $w$. We improve the variation factor in this paper as follows to avoid this situation.

$$\alpha = \frac{G_m}{G_m + 1 - G}, \quad \lambda = -\exp(-\alpha)$$

$$F = F_0 \cdot (2^\lambda + 1)$$  \hspace{1cm} (15)

where $F_0$ is the initialized scaling factor, $G_m$ is the maximum number of iterations, $G$ is the current number of iterations, $F$ is gradually approaching $F_0$, and $G$ takes values in the range $[1, G_m]$. Meanwhile, the crossover factor is improved in the adaptive differential evolution algorithm, and to diversify the population, the crossover factor is improved as follows:

$$CR = 0.5 \cdot (1 + \frac{G}{G_m})$$  \hspace{1cm} (16)

where $G_m$ is the maximum number of iterations, $G$ is the current number of iterations, and $G$ takes values in the range $[1, G_m]$. As shown in Figure 5, in contrast to the adaptive algorithm, the improved differential evolution algorithm has fewer iterations, obtains the optimal value faster, and is closer to the optimal value. The gradual increment of the adaptive crossover operator maintains the diversity of the population at the initial stage, improves the local search ability of the algorithm at the later stage.

3) HYBRID MODEL

This section mainly introduces the framework structure of the DPCL hybrid model, as shown in Figure 5. The specific details are as follows:

1) Input layer: input historical electric load data is a matrix $X$ of order $M$: $X = [x_1, x_2, \ldots, x_M]^T, X_{train} \in R^{M \times 1}$.

2) CNN layer: The output shows the feature vector at time $t$ as: $H_t = f(x_{train,t} \times W + b) = Sigmoid(x_{train,t} \times W + b)$, $W$ and $b$ denote the weights and bias values.

3) Combination layer: Layer 3D includes the output matrix of the CNN layer, and also includes new features $F_{train}$,
TABLE 3. The performance of the neuroevolution-DPCL.

| Datasets                  | Evolution algorithm | CNN neurons | LSTM neurons | Time Steps | RMSE    | MAPE    | Running times (s) |
|---------------------------|---------------------|-------------|--------------|------------|---------|---------|------------------|
| the Electrician's         | ADE                 | [64,3]      | [12,34,49]   | 12         | **0.0074** | **0.0063** | **2220**         |
| Mathematical Modeling     | DE                  | [47,64]     | [46,2,49]    | 16         | 0.0257  | 0.0239  | 2323             |
| Contest                   | PSO                 | [30,20]     | [31,26,42]   | 30         | 0.0296  | 0.0262  | 2570             |
|                           | ES                  | [33,35]     | [24,25,43]   | 28         | 0.0262  | 0.0259  | 2413             |
|                           | GA                  | [30,47]     | [20,34,49]   | 14         | 0.0253  | 0.0234  | 2370             |
| European regions          | ADE                 | [57,44]     | [16,2,49]    | 16         | **0.0070** | **0.0059** | **2422**         |
|                           | DE                  | [44,48]     | [42,34,30]   | 26         | 0.0115  | 0.0095  | 3477             |
|                           | PSO                 | [57,60]     | [12,34,34]   | 26         | 0.0221  | 0.0192  | 3454             |
|                           | ES                  | [18,53]     | [47,48,15]   | 18         | 0.0271  | 0.0245  | 3957             |
|                           | GA                  | [44,48]     | [38,33,30]   | 26         | 0.0265  | 0.0233  | 4692             |

FIGURE 7. Convergence curves of basic evolutionary algorithms.

FIGURE 8. Convergence curve of evolution algorithms.
FIGURE 9. Comparison of actual values and predicted values of different models (European regions).
The output $H$ at step $t$ is built to fully learn the proposed features and to capture their internal patterns of variation. The output layer calculates the output with the Concat function into an output matrix for correlation. All input features of layer 3D combine weather and holidays selected based on correlation analysis regions. The sampling period of the dataset is 1 h. Training the LSTM layers, and the CNN filter,

4)LSTM layer: The input matrix is $X_t = \text{concat}(h_{c}, F_{\text{train}}) = [h_c, F_{\text{train}}]$. The LSTM structure is built to fully learn the proposed features and to capture their internal patterns of variation. The output $H$ at step $t$ is $h_t = \text{LSTM}(H_{x', t-1}, H_{x', t}), t \in [1, i]$. The prediction formula can be expressed as $y_t = \text{Sigmoid}(w_0 h_t + b_0)$, where $y_t$ denotes the predicted output value at time $t$; $w_0$ and $b_0$ are the weight matrix and the deviation vector, and the Sigmoid function is chosen as the activation function of the Dense layer in this paper.

4) OPTIMIZATION OF HYBRID NEURAL NETWORK USING IMPROVED DIFFERENTIAL EVOLUTION

In this paper, we use an improved differential evolution algorithm to optimize the topologies and time-steps on the hybrid neural network. In the optimization, we use the mean square error (MSE) to select the best individual. The calculation formula of the MES is as follows:

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

(17)

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

The flow chart of the short-term load forecasting model based on DPCL is shown in Figure 5. The forecasting steps as follows:

IV. EXPERIMENTAL ANALYSIS

A. TYPES OF DATASETS

The proposed algorithm model is varied on two public datasets. The first dataset is the load prediction dataset of electricians and the Electrician’s Mathematical Modeling Contest, the data set contains historical load, daily average temperature, humidity, and holiday data, and the load data sampling period is 1 day. The second dataset is Public data sets provided by European regions. The sampling period of the dataset is 1 h. Training and test set in the ratio of 8:2. We use Adam [46] to optimize the gradient of the stochastic objective function.

B. EVALUATION CRITERIA

The mean absolute percentage error (Mean Absolute Percentage Error, MAPE) and root mean square percentage error (RMSPE) expressions were selected in the model, and the calculation formula is as follows:

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

(18)

$$RMSPE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left( \frac{y_i - \hat{y}_i}{y_i} \right)^2}$$

(19)

where: $\hat{y}_i$ is the predicted value of the neural network at the $i$ time and the actual value of the $y$ neural network at the $i$ time.

C. ANALYSIS AND DISCUSSION

In this section, we analyze and discuss the prediction accuracy of the proposed hybrid model through two experiments.

Algorithm 1 Improved Differential Evolution Optimizes Hybrid Neural Network

Step 1: Parameters initialization: mutation factor $F_0$, crossover operator $CR_0$, population size $NP$, and max generation $MAX_{G}$

Step 2: Population initialization: $x_{i, 0} = (c1, c2, l1, l2, l3, b)$, where $c1$ and $c2$ is the size of the CNN filter, $l1$, $l2$, and $l3$ is the number of neurons of the LSTM layers, and $b$ 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 3.1 Mutation: compute $F$ by Eq. (15)

generate a mutant vector by Eq. (11)

Step 3.2 Crossover: compute $CR$ using Eq. (16)

generate a trial vector by Eq. (12)

Step 3.3 Selection: set up a hybrid neural network according to each individual train prediction model compute the MSE on the validation set by Eq. (17), which has a smaller value will be selected end for

$G = G + 1$

end while

the electrician’s mathematical modeling contest, the data set contains historical load, daily average temperature, humidity, and holiday data, and the load data sampling period is 1 day. The second dataset is Public data sets provided by European regions. The sampling period of the dataset is 1 h. Training and test set in the ratio of 8:2. We use Adam [46] to optimize the gradient of the stochastic objective function.
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.

1) EVOLUTIONARY ALGORITHM COMPARISON EXPERIMENT
In the comparison experiments among different forms of the basic algorithm, it can be seen from Figure 7 that among...
2) MODELS COMPARISON EXPERIMENT

After obtaining the optimization-seeking prediction model, single LSTM, single CNN, hybrid CNN-LSTM, hybrid PSR-CNN-GRU, and DPCL models to verify that the model proposed in this paper can achieve high prediction accuracy.

As described in Table 4, the RMSE and the MAPE of the DPCL prediction model are far lower than PSR-CNN-LSTM, CNN-LSTM, CNN, and LSTM models. The DPCL has the lowest RMSE, MAPE on two different datasets.

On the European regions datasets, we perform load forecasting for one of these days, and the accuracy of the 24-hour forecast. The RMSPE of DPCL is 0.0070, and the MAPE of DPCL is 0.059. It is clear that DPCL is lower than that of other models. PSR-CNN-LSTM does as well, which is better than CNN-LSTM, CNN, and LSTM. From Fig.9 and Fig.10, the DPCL is most fitting. The prediction value of PSR-CNN-GRU is better compared with other methods, but the prediction effect of peak load is advanced or delayed, while the prediction effect of CNN_LSTM is more different and the judgment of load direction is more wrong. In contrast, the predicted values of LSTM and ell with the actual values and can be close to the direction of the load to a large extent, but still cannot accurately judge the change of the load, so relatively speaking, the predicted values in this paper are closer to the actual values.

From Table 4, Fig.11, and Fig.12, it is evident that the DPCL performs very well on the Electrician’s Mathematical Modeling Contest datasets. This paragraph aims to predict the next month of load data, we predict the total daily load amount used in a month. As expressed in Fig.11, the prediction accuracy of DPCL is higher than the others, and the prediction error is controlled within 1.5%. It is also clear that the MAPE values of DPCL compared to PSR-CNN-LSTM, CNN, and LSTM models have improved by 0.88%, 1.07%, 1.36%, 1.82%. It is worth noting that the DPCL has high predictive accuracy in two datasets.

V. CONCLUSION

It is important to further improve the accuracy of short-term load forecasting, to ensure the balance between load demand and power supply and demand, we propose an improved differential evolutionary algorithm (ADE) based model for optimizing CNN-LSTM hybrid neural networks. The experimental analysis by data processing and algorithmic modeling shows that:

1) The adoption of normalization, phase space reconstruction, and correlation analysis reduces the dimensionality of the data and the computational cost and helps to prepare for improving the accuracy of the model later by retaining the data features with high correlation;

2) A hybrid of CNN and LSTM is equivalent to combining the advantages of both neural networks in spatial feature
extraction and temporal feature extraction. It can further improve the model accuracy than a single neural network model.

3) The hybrid neural network model obtained from the training of ADE-optimized CNN and LSTM improves the training drawback that neurons and step size cannot be determined. It can further improve the model accuracy than other neural network models.

In terms of prediction performance and operation time comparison, the DPCL model is valuable for short-term load forecasting systems and overall power plant operation and maintenance.

REFERENCES

[1] A. S. Khwaja, X. Zhang, A. Anpalagan, and B. Venkatesh, “Boosted neural networks for improved short-term electric load forecasting,” Electr. Power Syst. Res., vol. 143, pp. 431–437, Feb. 2017, doi: 10.1016/j.epsr.2016.10.067.

[2] M. Massoudi, S. S. Refaat, I. Chihi, M. Trabelsi, F. S. Oueslati, and H. Abu-Rub, “A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for short-term load forecasting,” Energy, vol. 214, Jan. 2021, Art. no. 118874, doi: 10.1016/j.energy.2020.118874.

[3] Y. Liang, D. Niu, and W.-C. Hong, “Short term load forecasting based on feature extraction and improved general regression neural network model,” Energy, vol. 166, pp. 653–663, Jan. 2019, doi: 10.1016/j.energy.2018.10.119.

[4] H. G. Bergsteinsson, J. K. Møller, P. Nystrup, Ó. P. Pálsson, and D. Guericke, and H. Madsen, “Heat load forecasting using adaptive online learning,” Appl. Energy, vol. 292, Jun. 2021, Art. no. 118672, doi:10.1016/j.apenergy.2021.118672.

[5] Y. Wang, N. Zhang, Y. Tan, T. Hong, D. S. Kirkschen, and C. Kang, “Combining probabilistic load forecasts,” IEEE Trans. Smart Grid, vol. 10, no. 4, pp. 3664–3674, Jul. 2019, doi: 10.1109/TSG.2018.2833869.

[6] M. Sobhani, A. Campbell, S. Sangamwar, C. Li, and T. Hong, “Combining weather stations for electric load forecasting,” Energies, vol. 12, no. 8, p. 1510, Apr. 2019, doi: 10.3390/en12081510.

[7] Y. Alvarez, S. Mazuelas, and J. A. Lozano, “Probabilistic load forecasting based on adaptive online learning,” IEEE Trans. Power Syst., vol. 36, no. 4, pp. 3668–3680, Jul. 2021, doi: 10.1109/tpwrs.2021.3050837.

[8] C. Tian and Y. Hao, “A novel nonlinear combined forecasting system for short-term load forecasting,” Energies, vol. 11, no. 4, p. 712, Mar. 2018, doi: 10.3390/en10040712.

[9] N. Li, L. Wang, X. Q. Li, and Q. Zhu, “An effective deep learning neural network model for short-term load forecasting,” Concurrency Comput., Pract. Exper., vol. 32, no. 7, p. e5595, Apr. 2020, doi: 10.1002/cpe.5595.

[10] M.-W. Li, J. Geng, W.-C. Hong, and Y. Zhang, “Hybridizing chaotic and quantum mechanisms and fruit fly optimization algorithm with least squares support vector regression model in electric load forecasting,” Energies, vol. 11, no. 4, p. 712, Mar. 2018, doi: 10.3390/en10040712.

[11] L. Wen, K. Zhou, and S. Yang, “Load demand forecasting of residential buildings using a deep learning model,” Electr. Power Syst. Res., vol. 179, Feb. 2020, Art. no. 106073, doi: 10.1016/j.epsr.2019.106073.

[12] L. Wen, K. Zhou, S. Yang, and X. Lu, “Optimal load dispatch of community microgrid with deep learning based solar power and load forecasting,” Energy, vol. 171, pp. 1053–1065, Mar. 2019, doi: 10.1016/j.energy.2019.01.075.

[13] H. R. Alirezaei, A. Salami, and M. Mohammadinooshoushan, “A study of hybrid data selection method for a wavelet SVR mid-term load forecasting model,” Neural Comput. Appl., vol. 31, no. 7, pp. 2131–2141, Jul. 2019, doi: 10.1007/s00521-017-3171-9.

[14] Y. M. Liu, S. L. Lei, C. X. Sun, Q. Zhou, and H. J. Ren, “A multivariate forecasting method for short-term load using chaotic features and RBF neural network,” Eur. Trans. Elect. Power, vol. 21, no. 3, pp. 1376–1391, 2011, doi: 10.1002/etep.502.

[15] Y. N. Peng and W. L. Xiang, “Short-term traffic volume prediction using GA-BP based on wavelet denoising and phase space reconstruction,” Phys. A, Stat. Mech. Appl., vol. 549, Jul. 2020, Art no. 123913, doi: 10.1016/j.physa.2019.123913.
[35] Y. Zhang, J. Zhang, L. Luo, and X. Gao, “Optimization of LMBP high-speed railway wheel size prediction algorithm based on improved adaptive differential evolution algorithm,” Int. J. Distrib. Sensor Netw., vol. 15, no. 10, Oct. 2019, Art. no. 1550147719881348, doi: 10.1177/1550147719881348.

[36] 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.

[37] F. Takens, “Detecting strange attractors in turbulence,” in Dynamical Systems and Turbulence, Warwick 1980, 1981, pp. 366–381.

[38] 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.

[39] 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.

[40] A. Haidar and B. Verma, “Monthly rainfall forecasting using one-dimensional deep convolutional neural network,” IEEE Access, vol. 6, pp. 69053–69063, 2018, doi: 10.1109/ACCESS.2018.2880044.

[41] H. Shi, M. Xu, and R. Li, “Deep learning for household load forecasting—A novel pooling deep RNN,” IEEE Trans. Smart Grid, vol. 9, no. 5, pp. 5271–5280, Sep. 2018.

[42] A. Sherstinsky, “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network,” Phys. D: Nonlinear Phenomena, vol. 404, Mar. 2020, Art. no. 132306.

[43] W.-J. Huang, Y.-T. Li, and Y. Huang, “Prediction of chaotic time series using hybrid neural network and attention mechanism,” Acta Phys. Sinica, vol. 70, no. 1, 2021, Art. no. 010501.

[44] Y. Huang et al.: Short-Term Load Forecasting Based on Hybrid Neural Network and Phase Space Reconstruction

[45] X. F. Yan, J. Yu, F. Qian, and J. W. Ding, “Kinetic parameter estimation of oxidation in supercritical water based on modified differential evolution,” J. East China Univ. Sci. Technol. (Natural Sci. Ed.), vol. 32, no. 1, pp. 93–94, 2006.

[46] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” CoRR, vol. abs/1412.6980, 2015.

YUAN HUANG was born in Hebei, China, in 1987. He received the bachelor’s and master’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.

RUIXIAO ZHAO was born in Henan, China, in 1987. She received the bachelor’s degree from the Xinxiang University of Science and Technology, in 2018. She is currently pursuing the master’s degree with the School of Information and Electrical Engineering, Hebei University of Engineering. Her research interests include data mining and machine learning.

QIANYU ZHOU was born in Hebei, China, in 1987. She received the bachelor’s and master’s degrees from the Hebei University of Engineering, in 2010 and 2013, respectively, and the Ph.D. degree from the China University of Mining and Technology, Beijing, in 2016. Since 2016, she has been working as a Teacher with the School of Earth Science and Engineering, Hebei University of Engineering. She has published four articles. Her research interests include data analysis and environmental geology.

YUXING XIANG was born in Chongqing, China, in 1997. He received the bachelor’s degree from Chongqing Technology and Business University, in 2019. Since 2019, he has been a Student with the School of Information and Electrical Engineering, Hebei University of Engineering. His research interests include data mining and machine learning.