Grid Load Forecasting Based on Dual Attention BiGRU and DILATE Loss Function

XIANG WANG1,2,3, ZHANXIA WU2,3, JUNXIONG GE3, ZHANHAO ZHANG4, LINCHENG HAN5, SHUNJIANG WANH6 AND XIUYU ZHANG5

1Beijing Smart-Chip Microelectronics Technology Co., Ltd., Beijing 102200, China
2Shenzhen Smart-Chip Microelectronics Technology Co., Ltd., Shenzhen 518048, China
3Shenzhen Guodian Technology Communication Co., Ltd., Shenzhen 518000, China
4Laajin (Tianjin) Auto Parts Co., Ltd., Tianjin 300000, China
5Department of Automation Engineering, Northeast Electric Power University, Jilin 132012, China
6State Grid Liaoning Electric Power Co., Ltd., Shenyang 110006, China.

Corresponding author: LINCHENG HAN (e-mail: hanlincheng2008@hotmail.com).

Project Supported by the Computer Software Commissioned Development Project of Shenzhen Guodian Technology Communication Co. Ltd. (No. SGJITZXSZWXX2101391) and National Science Foundation of China (No. 61673101).

ABSTRACT To learn the deep relationship implicit in load data and improve the accuracy of load prediction, this paper presents a new Seq2seq framework based on dual attention and a bidirectional gated recurrent unit (BiGRU). The authors also introduce the Distortion Loss including Shape and Time (DILATE) loss function. Firstly, a dual attention mechanism is added to the Seq2seq architecture. The first layer of the attention mechanism enables the encoder to output multiple intermediate states, making the decoder more targeted when calculating the predicted values at different times. The second layer is an automatic attention mechanism, which reduces the possibility of errors accumulated during a long-time span prediction by calculating the internal correlation of the decoder output sequence data. Secondly, the DILATE loss function is introduced to improve the prediction lag problem caused by the mean square error (MSE) loss function. Finally, the proposed model is tested using power load data from a region in northern China. The simulation results show that the method proposed in this paper has a better prediction effect than LSTM, GRU and LS-SVR.

INDEX TERMS Load Forecasting; BiGRU; Dual Attention; Seq2seq; DILATE

I. INTRODUCTION

As an important part of energy management, power load forecasting plays an important role in the power system [1,2]. Accurate load forecasting can economically and reasonably arrange the start and stop of generators inside the power grid, maintain the safety and stability of power grid operation, and reduce unnecessary rotating reserve capacity. The level of power load forecasting can measure whether a power enterprise’s management is modern and scientific, which is why it is essential to power system theoretical research.

Most early power load forecasting methods were based on mathematical statistics. Related models in the theory [3-5] have higher requirements on the stationarity of the original time series. When grid conditions are normal and the climate and other factors are fairly stable, the prediction results are better. However, when the random factors vary greatly or bad data interfere, the prediction results are not ideal [6,7].

With the widespread installation of data sensors in power systems, neural network algorithms supported by big data are widely used in power systems [8-10]. Among them, the recurrent neural network (RNN), designed for time series research, is the most widely used in load forecasting. In [11], a load prediction method is proposed based on isolation forest (iForest) and long short-term memory (LSTM)[12] network. In [13], a load hybrid prediction method is introduced, combining convolutional neural networks (CNN) and LSTM. In [14], the Seq2seq [15] architecture is used to predict short-term loads. In [16], the authors propose an LSTM-based Seq2seq-LSTM load prediction model. However, for RNN-based prediction models, whether using LSTM or Gated Recurrent Unit (GRU)[17], setting too long a sequence length may affect the stability[18]. In [19], we find that the effective con-text size of language models using LSTM is around 200 units on average, but only the closer 50 tokens can be clearly distinguished, suggesting that even
LSTM struggles to capture long-term dependencies. Obviously, long-term dependencies in time series data are especially important for data prediction. The attention mechanism [20] allows the model to access any part of the historical sequence, making it more suitable for grasping the dependencies of long-term sequences. In [21], an interpretable self-attention mechanism is implemented to learn the dependencies of long-term series data, which makes it possible to predict sales data. The authors in [22] introduced an attention mechanism to predict ultra-short-term power generation and gave different weights to the hidden states of the bidirectional long-short-term memory (BiLSTM) network, so as to selectively obtain more effective information. In [23], feature space and the time series dual attention mechanism are based on LSTM, which enabled the creation of an interpretable offshore wind power output prediction model. The authors in [24] proposed a two-stage time attention mechanism (TAM) in the short-term load forecasting of power grids, which enhanced the model's memory for long-time series information. The article[25] forecasts three kinds of data, namely cold, heat and electricity, and designs three attention modules, but only one module is used for power load prediction; the article[26] introduces a model based on Attention's BiGRU and convolutional layers act as feature extractors to explicitly model channel features using SENet commonly used in image recognition. When designing the time-related attention layer, although the above articles have added attention mechanisms to the models, the attention mechanisms in them all capture the relationship between the data in the input sequence or the relationship between the input sequence data and the output sequence, but the output sequence itself also contains information, which needs to be fully excavated to improve the prediction accuracy.

The loss functions for training and evaluating models in deep learning are generally mean absolute error (MAE), mean square error (MSE), as well as root mean square error (RMSE). Most loss functions are calculated by comparing pairs of time series and their predicted values, thereby looking at the "vertical" Euclidean distance between the two series. However, when one sequence is misaligned in time relative to another, then a "horizontal" distance or "time" distance becomes necessary; otherwise, a certain time lag will result. Therefore, a method is needed that allows elastic movement of the timeline to accommodate similar but out-of-phase sequences. The DILATE (Distortion Loss including Shape and Time) is a loss function [27], and it is used in an evaluation algorithm that integrates shape error and time error. Taking shape error and temporal distortion as loss values, the temporal offset and distortion losses can be obtained more efficiently. Combined with the DILATE loss function, this paper proposes a Seq2Seq architecture based on a dual attention mechanism and Bidirectional Gated Recurrent Unit (BiGRU). The contributions of this research are:

1) A dual attention mechanism is applied. The first layer of attention mechanism considers the relationship between input and output sequences, strengthens the weight of key data that affects load prediction, and weakens the weight of data with low correlation. The weighted data provides more information to the decoder. The second layer of attention mechanism is a self-attention mechanism, which excavates the internal relationship of the hidden state output by BiGRU in the decoding layer, which further improves the accuracy of the prediction results.

2) The DILATE loss function is introduced to solve the lag problem caused by the Mean Squared Error (MSE) loss function in time series forecasting tasks.

II. SEQ2SEQ MODEL BASED ON DUAL ATTENTION MECHANISM

Given input series, i.e., \( X = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n \), which represents historical load data, and where \( m \) is the length of input data. \( Y = [y_1, y_2, \ldots, y_n] \in \mathbb{R}^n \) is the target series, which represents the load data to be predicted, and where \( n \) is the length of output data. The forecasting model aims to learn a nonlinear mapping function \( F(\cdot) \) to the target series \( Y \):

\[
Y = F(X)
\]

In the model proposed in this paper, BiGRU neural network is applied to the encoder and decoder. The attention mechanism module and the self-attention mechanism module are embedded in the encoder and decoder respectively. The model structure is shown in Figure 1.

**FIGURE 1.** Seq2seq model based on Attention mechanism and self-Attention mechanism.

A. **BIGRU**

GRU is a simplified version of LSTM, which keeps the effect similar to LSTM while making the structure simpler. GRU merges the forget gate and the input gate into an "update gate" that merges the unit state and the hidden state. When calculating the hidden state at the current moment, a candidate state \( \tilde{h}_t \) is first calculated, and when calculating the candidate state, the value of the reset gate \( r_t \) is considered:

\[
r_t = \sigma(W_r[\tilde{h}_{t-1}, x_t])
\]

We used the sigmoid function to limit the value of the gate to \([0,1]\). If the reset gate is close to 0, the current candidate value \( \tilde{h}_t \) ignores the previous hidden state \( h_{t-1} \) and is computed with the current input \( x_t \). This effectively lets the hidden state throw away any irrelevant information found in the future:
\[ \hat{h}_t = \tanh(W_h [r_t * h_{t-1}, x_t]) \] (3)

After calculating the candidate value \( \hat{h}_t \), the update gate is used to control how much information from the previous hidden state can be transmitted to the current hidden state. The update gate \( z_t \) is:

\[ z_t = \sigma(W_z [h_{t-1}, x_t]) \] (4)

The last hidden state \( h_t \) at the current moment and can be calculated as:

\[ h_t = (1 - z_t) * h_{t-1} + z_t * \hat{h}_t \] (5)

The GRU neural network uses a cyclic structure to store and retrieve information, but the neural network only considers the information at the last moment of the prediction point and cannot consider the state at the future moment; thus, the prediction accuracy cannot be further improved. The bi-directional GRU (BiGRU) network has a future layer that allows the data sequence to be predicted in the opposite direction to overcome this problem. The BiGRU principle is shown in Figure 2:

**FIGURE 2. BiGRU Schematic**

The network uses two hidden layers to extract information from the past and future, and both are connected in the same output layer. In BiGRU:

\[ \tilde{h}_t = GRU(x_t, \tilde{h}_{t-1}) \] (6)

\[ \tilde{h}_t = GRU(x_t, \tilde{h}_{t-1}) \] (7)

\[ h_t = u_t \tilde{h}_t + v_t h_{t-1} + h_t \] (8)

Among the hidden layers, the function represents the nonlinear transformation of the input data. \( u_t \) and \( v_t \) are the weights corresponding to the forward hidden layer, while state \( \tilde{h}_t \) and the reverse hidden layer state \( \tilde{h}_t \) correspond respectively. Moreover, \( h_t \) represents the hidden layer state at time \( t \) of the corresponding offset.

**B. ENCODER WITH ATTENTION MECHANISM**

After the input data \( X \) is calculated by the BiGRU module in the encoder, the hidden state \( H = f_1(X) = [h_t, h_{t+1}, ..., h_n] \in \mathbb{R}^{k \times n} \) is obtained, where \( k \) is the hidden layer dimension. \( f_1 \) is the BiGRU network unit in the decoder. The output value of the BiGRU module of the decoder is the hidden state \( Y' = [y'_1, y'_2, ..., y'_n] \in \mathbb{R}^{k \times n} \). The relationship between \( X \) and \( Y' \) is represented by calculating the degree of attention between \( H \) and \( Y' \) in the attention module. The calculation process of attention mechanism is shown in Figure 3.

**FIGURE 3. The calculation process of attention mechanism**

Input the hidden state \( H \) into the attention mechanism, and combine the BiGRU output \( y'_{i-1} \) at time \( i \) to calculate the attention weight corresponding to each \( h_t \) at the current time. \( y'_i \) is obtained from the last hidden state \( h_n \) in the encoder. The weight calculation is as follows:

\[ e_y = V^T \tanh(W_h + U y'_{i-1} + \text{bias}) \]

where \( V \in \mathbb{R}^T, W \in \mathbb{R}^{T \times d} \) and \( U \in \mathbb{R}^{d \times d} \) are learnable weight matrices.

Use the Softmax function to normalize \( e_y \) according to Equation 10.

\[ a_y = \frac{\exp(e_y)}{\sum_{j=1}^{T} \exp(e_y)} \] (10)

The feature correlation coefficient \( a_y \) is multiplied by the corresponding hidden state value \( h_t \) to obtain a weighted cell state \( C \in \mathbb{R}^{T \times d} \) that considers different feature contribution rates.

\[ C = [c_1, c_2, ..., c_T] = [(a_y_1, a_y_2, ..., a_y_T)] \otimes [h_1, h_2, ..., h_T] \] (11)

The hidden state \( Y' \) is updated according to Equation 9:

\[ y'_i = f_2(c_i, y'_{i-1}) \] (12)

\( f_2 \) is the BiGRU network unit in the decoder, and the input is no longer the original cell state, but the weighted data considering the size of the influence. Through attention mechanism, the encoder part can adaptively extracts the contribution rate of each input data to improve the prediction accuracy. Algorithm 1 is the algorithm pseudocode of the attention mechanism in this paper.

**Algorithm 1 Attention Mechanism Pseudocode**

1: Hidden = encoder.bigru.output
2: \( y'_0 = h_t \)
3: for \( i = 1 \) to TargetLength do
4:   for \( j = 1 \) to InputLength do
5:     \( e_y = \text{MLP}(y'_{i-1}, h_t) \)
6:   end for
7:   \( A_i = \frac{\exp(e_y)}{\sum_{j=1}^{T} \exp(e_y)} \)
8:   \( c_i = A_i \odot \text{Hidden} \)
9: \( y'_i = f_z(c_i, y'_{i-1}) \)
10: end for
11: return \( Y' \)

C. DECODER WITH SELF-ATTENTION MECHANISM

In order to capture the interrelationships among \( Y' \), we introduce a self-attention mechanism when designing the decoder part. The calculation process of self-attention mechanism is shown in Figure 4.

The calculation process of self-attention mechanism

The self-attention mechanism does not employ the recursive mechanism of the recurrent neural network, so the sequence data does not contain any position or order information. This paper adopts the same absolute position embedding (APE) as the transformer to solve this problem. Let \( t \) be the expected position in the input sentence, \( \tilde{p}_t \in \mathbb{R}^d \) is its corresponding code, where \( d \) is the coding dimension, and \( f: \mathbb{N} \rightarrow \mathbb{R}^d \) is the function \( p_t \) that generates the output vector. It is defined as follows:

\[
\tilde{p}_t^{(m)} = f(t)^{(m)} = \begin{cases} 
\sin \left( \frac{1}{10000^{2k/d}} t \right), & \text{if } m = 2k \\
\cos \left( \frac{1}{10000^{2k+1/d}} t \right), & \text{if } m = 2k + 1 
\end{cases} 
\]  

(13)

The positional embedding \( \tilde{p}_t \) can also be understood as a vector containing pairs of sine and cosine for each frequency. \( d \) is divisible by 2.

The self-attention layer takes position-encoded data \( Y'' = (Y' \oplus \tilde{P}) \in \mathbb{R}^{n \times (k + d)} \) as input. By capturing the relationships and interactions between the output time series, the model can grasp the dependencies of the time series, which can further modify the output of the decoder and improve the prediction ability. The calculation process of self-attention mechanism is:

1) The position-encoded data \( Y'' \) is multiplied by the learnable weight matrices \( W_O, W_K, W_V \in \mathbb{R}^{n \times (k + d)} \) to obtain three vectors,

\[ Q = W_O Y'', K = W_K Y'', V = W_V Y'' \]

which corresponds to query, key, and value, respectively;

2) Calculate the similarity between each Query( \( Q \) ) and each Key( \( K \) ) by dot product to get the weight

\[ E = [e_1, e_2, ..., e_m]^T \]

of each Key( \( K \) ) corresponding to Value( \( V \) );

3) Use the softmax function to normalize the weights \( E \) and thereby obtain the weight coefficient

\[ A = \left[ [a_{11}, a_{12}, ..., a_{1m}], [a_{21}, a_{22}, ..., a_{2m}], ..., [a_{n1}, a_{n2}, ..., a_{nm}] \right] \]

4) Divide \( A \) by \( \sqrt{d_k} \) to scale it so that it fits a standard Gaussian distribution, and then multiply it by \( V \) to get the weighted value \( Y'' \). The scaled dot-product attention is the attention calculation method adopted by the transformer, and its expression is as follows:

\[
\text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V 
\]

(14)

5) Combine the positional encoded value \( Y'' \) and the weighted value \( Y'' \) to the linear layer to obtain the final predicted result \( \hat{Y} \).

The self-attention mechanism reduces the dependence on external information, can link information at different positions in the input sequence, and is better at capturing the internal correlation of data or features. Algorithm II is the pseudocode of Self-Attention.

Algorithm II Self-Attention pseudocode

1: hidden = decoder.bigru.output
2: hidden = hidden + positional codes
3: for sq=1 to Target.length do
4:     for sk=1 to Target.length do
5:         \( E = W_q \text{hidden} \cdot (W_r \text{hidden})^T \)
6:         \( A = \text{softmax} \left( \frac{E}{\sqrt{d_k}} \right) \)
7:         \( \hat{Y} = \text{Linear}(A \odot \text{hidden} + \text{hidden}) \)
8: end for
9: end for
10: return \( \hat{Y} \)

III. DILATE LOSS FUNCTION

In the process of training a predictive model, a loss function needs to be used to update the model parameters. In the past, most of the time series forecasting used the loss function based on Euclidean distance. It calculates the loss between the predicted value and the true value according to a strict one-to-one mapping of each data point, ignoring information such as the shape and time of the time series. As shown in Figure 5, the dashed box identifies a relatively obvious time dislocation. While the model predicted correct sequence data, it was not accurate on time. Clearly, a method is needed that allows elastic movement of the timeline to accommodate similar but out-of-phase sequences.
The DILATE loss function is a sequence prediction evaluation algorithm that integrates shape error and temporal error. DILATE takes shape error and temporal distortion as loss values, which minimizes the effects of temporal offset and distortion. As shown in Figure 6, the prediction effect of Figure b is better than that of Figure a, however the Euclidean distance loss values in the two figures are exactly the same, because the Euclidean distance loss function ignores the time shape information contained in the curve, the DILATE loss function perfectly captures this information and changes the corresponding loss value. The insensitivity of the MSE loss function to time information can lead to lag problems in time series forecasting. The main purpose of load forecasting is to apply to power generation planning, equipment maintenance planning, real-time economic dispatch and long-term power grid planning[28]. The existence of lag will cause the time difference between the predicted condition and the actual load condition, which will have an adverse effect on the operation of the power grid.

\[
\hat{y}_i = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n) \in R^{n \times 1} \quad \text{and} \quad \tilde{y}_i = (\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n) \in R^{n \times 1},
\]

where \( \delta(\hat{y}_i, \tilde{y}_i) \) is the Euclidean distance between the two sequence points.

Step 2: Find the optimal cost path. Based on the obtained cost matrix, the optimal cost path is obtained by finding the minimum value of each step, which minimizes the overall cost.

To improve the convergence speed of the algorithm and find the optimal dynamic path, three constraints must be satisfied in the path planning.

1) Boundary conditions: Since the arrangement order of the compared two sequences' elements is inherent, and the elements of the two sequences have a certain correspondence, the dynamic path rounding must begin with the start element \((x_1, y_1)\) and end with element \((x_n, y_n)\).

2) Monotonicity: When comparing two sequences, a monotonic comparison in one direction is required, and repeated comparisons are not allowed.
3) Continuity: The distance between two points in the dynamic programming path must be adjacent to two points, and no cross-comparison can occur across points. According to the above steps and corresponding constraints, calculate the Euclidean distance between all points of the two curves and fill in the corresponding square to obtain an \( n \times n \) -dimensional cost matrix. The starting point \((x_1, y_1)\) and the ending point \((x_n, y_n)\) of the optimal curve are obtained under the constraints of the boundary conditions in the constraints. Under the constraint of monotonicity, it progresses from the starting point to the direction close to the ending point, and the continuity guarantees that there will be no breakpoints. In all the curves obtained, find the curve with the minimum Euclidean distance in each of the \( n \) steps forward, which is the optimal path. The resulting cost matrix and optimal path are shown in Figure 7.

![Figure 7. Cost Matrix and Optimal Path](image)

To identify the optimal path to the cost matrix, define binary matrices \( A^* \) \( A \in R^{n \times n} \). If the optimal curve between \( \hat{y}_h \) and \( \hat{y}_i \) in the cost matrix is in the path, the value of \( a_{hi} \) in matrix \( A \) is 1; otherwise, the value of \( a_{hi} \) is 0. The definition \( A_{n \times n} \) is the set of all paths that can go from \((1, 1)\) to \((n, n)\) under constraints. Since DTW is discrete and non-differentiable, a smooth operator is introduced:

\[
\min_{i,j} (a_1, a_2 \cdots a_n) = -\gamma \ln (\sum_{i=2}^{n} \exp (-\Delta A_{ij} / \gamma))
\]  

From the differential \( \gamma > 0 \), the shape loss function can be obtained as:

\[
\Gamma_{shape}(\hat{y}_i, \hat{y}_j) = DTW'_\gamma(\hat{y}_i, \hat{y}_j) = -\gamma \log (\sum_{\Delta \in A_{n \times n}} \exp (-\Delta A(\hat{y}_i, \hat{y}_j) / \gamma))
\]

The purpose of the temporal distortion function is to penalize the time delay between \( \hat{y}_i \) and \( \hat{y}_j \). Define \( A = \arg \min_{\Delta \in A_{n \times n}} \{ A, D(\hat{y}_i, \hat{y}_j) \} \) to represent the best path between the predicted value \( \hat{y}_i \) and the true value \( \hat{y}_j \). Define the penalty square \( \Omega = \left[ w_{h,j} \right] \) and \( \Omega \in R^{n \times n} \) as:

\[
TDI(\hat{y}_i, \hat{y}_j) = \langle A', \Omega \rangle
\]

\[
= \left\{ \arg \min_{\Delta \in A_{n \times n}} \{ A, D(\hat{y}_i, \hat{y}_j) \}, \Omega \right\}
\]

Where \( w_{h,j} = (h - j)^2 / n^2 \). From this, the temporal distortions index (TDI) can be defined.

TDI imposes a delay penalty on the best path \( A' \). The area between the optimal path and the first diagonal of the cost matrix represents the time error due to time distortion when selecting the optimal path for DTW. Since TDI contains two non-differentiable matrices, \( \Omega \) and \( A' \), it cannot be differentiable by introducing a smooth operator. But because of \( A' = \nabla_\Delta DTW_\gamma(\hat{y}_i, \hat{y}_j) \), a smooth approximation \( A' \) can be defined, and then a smooth temporal distortion function can be obtained based on \( A' \).

\[
A' = \nabla_\Delta DTW_\gamma(\hat{y}_i, \hat{y}_j) = \frac{\partial dw_\gamma(\hat{y}_i, \hat{y}_j)}{\partial \Delta(\hat{y}_i, \hat{y}_j)}
\]

\[
\Gamma_{mw}(\hat{y}_i, \hat{y}_j) := \langle A', \Omega \rangle
\]

IV. MODEL OVERALL STRUCTURE

Based on the forecasting model proposed above, combined with data preparation and model testing, the complete load forecasting process can be obtained, as shown in Figure 8. The algorithm processing flow follows:

1) Data preprocessing

The purpose of data preprocessing is to ensure the quality of the dataset when cleaning historical load data. Check for missing values and outliers in the dataset to prevent their impact on prediction accuracy. Remove outliers and treat them as missing values. Missing values are repaired from historical loading data. This paper uses linear interpolation to deal with:

\[
y = y_0 + \frac{y_1 - y_0}{x_1 - x_0} (x - x_0)
\]

where \( y \) represents the missing value, and \( x \) represents the abscissa of the missing value. \( y_0 \) and \( x_0 \) represent the horizontal and vertical coordinates of the first selected point, respectively. Moreover, \( y_1 \) and \( x_1 \) represent the horizontal and vertical coordinates of the second selected point.

2) Data normalization

Divide the preprocessed data into a training set and test set; then, perform the data normalization processing, starting with the training set and ending with the test set. Normalization is the operation of converting data into the range of 0–1, which can better extract data features, and at the same time, speed up the convergence speed of neural network training and improve the prediction accuracy. Data normalization can be ex-pressed as:

\[
y = y' (y_{\text{max}} - y_{\text{min}}) + y_{\text{min}}
\]

where \( y_{\text{max}} \) represents the maximum value in the data set; \( y_{\text{min}} \) represents the minimum value in the data set; \( y' \)
represents the uninitialized actual value, and \( y \) represents the initialized value.

Step 2: Load Forecasting

Input the processed training set data sequence into the prediction model. The input data is processed by BiGRU in the decoder to generate the hidden state sequence \( H = (h_1, h_2, ..., h_m) \). The hidden state sequence \( H \) and the last moment hidden state \( y_{t-1}' \) output by the decoder calculate the attention value. \( H \) is combined with the attention value to predict \( y_t' \) at the current moment. After all the prediction sequences \( Y' \) are obtained, which are sent to the self-attention mechanism module, and the prediction accuracy is further improved by mining the internal sequence relationship between them.

Take the DILATE loss function to compare the predicted sequence with the actual value to get the loss value, where the actual value is the actual load sequence. The DILATE loss function splits the loss value into shape error and temporal distortion. The model parameters are updated according to the comparison results, and the training of the model is completed after several iterations. Finally, the test set is input into the trained model, and the final prediction result is output.

Step 3: Model Validation

To evaluate the accuracy of the established load forecasting model, the fitting degree and forecasting accuracy of the model are evaluated by selecting evaluation indicators.
Figure 9 shows the optimization process of the epoch. It can be seen that when the model starts training, the error is large, around 14%. With the increase of training period, MAPE gradually decreases. After the training cycle reaches 120 times, the objective function MAPE is the smallest, and then slowly rises. At this time, it can be considered that the model is overfitting. Therefore, this paper determines the optimal training period of the model to be 120.

![Figure 9. The Bayesian optimization process of Epoch](image)

### C. EVALUATION INDICATORS

This paper considers four different evaluation metrics, namely MSE, RMSE, MAE and DILATE. Among them, RMSE and MAE are two scale-dependent measures. The RMSE is the square root of the MSE, making the prediction results more intuitive by orders of magnitude. The DILATE is split into two indicators, DTW and TDI, so that the shape and time errors of the prediction results can be examined separately.

The expressions of MAE, RSME, and mean absolute percentage error (MAPE) are shown as:

\[
\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |F_i - R_i|
\]  

(21)

\[
\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{R_i - F_i}{R_i} \right|
\]  

(22)

\[
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (F_i - R_i)^2}
\]  

(23)

Where \( n \) represents the number of data points, \( F_i \) represents the predicted value, and \( R_i \) represents the true value.

### D. MODEL COMPARATIVE EXPERIMENT

In this section, the model proposed in this paper is compared with other baseline methods. The chosen baseline method is as follows:

LSTM is an artificial neural network used in the field of artificial intelligence and deep learning, which is specially designed to solve the long-term dependence problem of general RNN.

GRU is like LSTM with forget gate, but has fewer parameters than LSTM because it has no output gate. GRU is found to perform similarly to LSTM on some tasks.

LS-SVR[29] is an improvement of Support Vector Regression (SVR). It changes inequality constraints into equality constraints, and uses the error sum of squares (SSE) loss function as the empirical loss of the training set, which is widely used in sequence prediction.

In order to verify the validity and stability of the model, 192 hours of load data were predicted, and the length of the input data was 96 hours. The quantitative evaluation results of each model are shown in Table 1. From the perspective of the average value of the prediction results, the prediction accuracy of the method in this paper is the highest compared with the other seven methods, which can accurately predict the periodicity and long-term variation. In addition, Figure 10 shows the training convergence process of three neural network-based models. GRU and LSTM can reach convergence after training for 60-80Epochs. The method proposed in this paper adds a dual attention mechanism on the basis of BiGRU, which has high model complexity and slow convergence speed. Convergence can be reached after training for 120 Epochs.

| Algorithm        | MSE      | RMSE     | MAE     | MAPE    | DTW     | TDI     | Epoch |
|------------------|----------|----------|---------|---------|---------|---------|-------|
| GRU              | 38799.30 | 196.97   | 184.68  | 14.70%  | 1581.72 | 0.46    | 66    |
| LSTM             | 30508.35 | 174.67   | 137.35  | 11.20%  | 1136.04 | 1.27    | 80    |
| LS-SVR           | 11594.50 | 107.68   | 87.14   | 7.44%   | 988.47  | 0.66    | –     |
| Dual Attention   | 9950.14  | 99.75    | 78.61   | 6.45%   | 851.73  | 0.08    | 118   |
| with DILATE      |          |          |         |         |         |         |       |
As shown in Figure 11, the proposed model achieves the best performance among all metrics. Due to the multi-step output 192 setting, the output span is long, and the results of LSTM and GRU are not ideal compared to single-step prediction. In terms of shape and time indicators, it is very different from the original data. It can be seen that the RNN network will lose a lot of information in long-term sequence prediction, cannot effectively capture sequence features, and the accuracy of the prediction results is very low. LSSVR is a regression algorithm that aims to find the shortest distance between the regression curve and the target sample point. It can be seen from Figure 11(c) that the results predicted by LSSVR can basically reflect the fluctuation trend of the sequence, and the fluctuation time points are basically correct. But the volatility gap is very large. The model proposed in this paper can still have a good grasp of the trend of the data in the long span prediction. The introduction of the dual attention mechanism makes the magnitude and trend of the predicted curve basically correct. This shows that the attention mechanism can still learn the dependencies between long-term series data well. And because of the use of the DILATE loss function, the prediction results are ideal in terms of time indicators, and there is basically no lag. This means that the present model retains better long-term robustness, which has important implications for real-world practical applications such as weather warning and long-term energy consumption planning.

**E. ABLATION STUDY**

In order to verify the effect of the DILATE loss function, the MSE and DILATE functions were used to conduct comparative experiments for the prediction algorithms LSTM and Dual Attention. In order to clearly show the effect of the DILATE function, a single-step test was performed. After predicting one time step based on each time window of length 12, the window is filled with an actual value. As can be seen from Figure 12, the LSTM with MSE...
predictions are unclear, under-performing in the presence of data dips or spikes, and with significant time lags. Dual Attention with MSE produces sharper shape predictions, but there is a large temporal misalignment. In contrast, LSTM with DILATE and Dual Attention with DILATE predict series that have both correct shapes and temporal localization. Among them, the model proposed in this paper has the best prediction results. The prediction evaluation is shown in Table II.

TABLE II
COMPARISON OF EVALUATION INDEXES OF MSE AND DILATE

| Algorithm                  | MSE   | RMSE  | MAE   | MAPE  | DTW   | TDI  |
|----------------------------|-------|-------|-------|-------|-------|------|
| LSTM with MSE              | 15477.64 | 124.41 | 88.79 | 7.96% | 357.60 | 0.27 |
| LSTM with DILATE           | 4797.10  | 69.26  | 50.15 | 4.24% | 270.19 | 0.06 |
| Dual Attention with MSE    | 5190.02  | 72.86  | 53.58 | 4.64% | 311.28 | 0.12 |
| Dual Attention with DILATE | 2090.68  | 45.72  | 38.04 | 3.24% | 184.54 | 0.02 |

VI. CONCLUSION

In this paper, we propose a seq2seq framework based on dual attention mechanism and BiGRU for long-term power load prediction, replacing the loss function commonly used in deep learning with DILATE. The advantages are as follows:

1) BiGRU can simultaneously consider the information and state of the past and future moments of the prediction point. Compared with ordinary GRU neural networks, BiGRU can more effectively capture the relationship between past and future data in long-term sequences.

2) The first-layer attention mechanism can obtain the relationship between the input data and the data to be predicted, providing more targeted input information for the decoding layer. The second layer uses the self-attention mechanism to fully consider the time series characteristics of the output data. By exploring the internal relationship between the sequences, the prediction error is corrected, and the accuracy of the prediction result is further improved.

3) The DILATE loss function considers the two indicators of deformation and time variation, which has advantages compared with the prediction algorithm trained by MSE, and can effectively improve the lag problem of prediction.

In the follow-up work, the impact of different factors on the load, such as the integration of renewable energy, smart
appliances, and extreme weather will be considered. It is foreseeable that by adding these factors, the accuracy of load forecasting will be further improved.

REFERENCES

[1] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, “Short-Term Load Forecasting With Deep Residual Networks,” IEEE Trans. Smart Grid, vol. 10, no. 4, pp. 3943–3952, Jul. 2019, doi: 10.1109/TSG.2018.2844430.

[2] ZHAO Pei and DAI Yeming, “Power Load Forecasting of SVM Based on Real-time Price and Weighted Grey Relational Projection Algorithm,” Power System Technology, vol. 44, no. 4, pp. 1325-1332, April 2020, doi: 10.13335/j.0258-6999.2019.0942.

[3] LONG Yong, SU Zhenyu, GAI Xiaoping, “Monthly Load Forecasting Using Component Decomposition Method,” Proceedings of the CSEE/EPRI, vol. 29, no. 4, pp. 35-50, May 2017, doi: 10.3969/j.issn.1003-8930.2017.05.006.

[4] D. H. Vu, K. M. Muttaqi, A. P. Agalgaonkar, and A. Bouzeraoud, “Intra-hour and hourly demand forecasting using selective order autoregressive model,” 2016. Accessed: Feb. 26, 2022. [Online]. Available: http://ieeexplore.ieee.org/document/7754014/citations.

[5] K. G. Boroojeni, M. H. Amini, S. Bahrami, S. S. Iyengar, A. I. Sarwat, and O. Karabasoglu, “A novel multi-time-scale modeling for electric power demand forecasting: From short-term to medium-term horizon,” Electric Power Systems Research, vol. 142, no. JAN., pp. 58–73, 2017, doi: 10.1016/j.epsr.2016.08.031.

[6] WANG Zengping, ZHAO Bing, JI Weijia, et al., “Short-term Load Forecasting Method Based on GRU-NN Model,” Automation of Electric Power Systems, vol. 43, no. 5, pp. 53-58, March 2019, doi: 10.7500/AEPS2018062003.

[7] GE Lianjiao, ZHAO Kang, SUN Yonghui, et al., “Short-term Load Forecasting of Distribution Network Based on Combination of Siamese Network and Long Short-Term Memory Network,” Automation of Electric Power Systems, vol. 45, no. 23, pp. 41-50, September 2021, doi: 10.7500/AEPS20210522003.

[8] H. Shi, M. Xu, and R. Li, “Deep Learning for Household Load Forecasting – A Novel Pooling Deep RNN,” IEEE Transactions on Smart Grid, pp. 1–1, 2017, doi: 10.1109/TSG.2017.2686012.

[9] H. Jia, Y. Deng, X. Qiu, X. Tong, and F. Li, “Line Loss Rate Prediction Method Based on Deep Learning with Long Short-Term Memory,” 2019. Accessed: Feb. 26, 2022. [Online]. Available: http://www.researchgate.net/publication/335863505_Loss_Rate_Prediction_Method_Based_on_Deep_Learning_with_Long_Short_Term_Memory.

[10] LÜ Haican, WANG Weifeng, ZHAO Bing, ZHANG Yi, GUO Qiting, HU Wen, “Short-term Substation Load Forecast Based on Wide & Deep-LSTM Model,” Power System Technology, vol. 44, no. 2, pp. 428-436, February 2020, doi: 1000-3673 (2020) 02-0428-09.

[11] Y. Ma, Q. Zhang, J. Ding, Q. Wang, and J. Ma, “Short Term Load Forecasting Based on iForest-LSTM,” in 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), Xi’an, China, Jun. 2019, pp. 2278–2282. doi: 10.1109/ICIEA.2019.8833755.

[12] H. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[13] LIU Yuhui, ZHAO Qian, “Ultra-short-term Power Load Forecasting Based on Cluster Empirical Mode Decomposition of CNN-LSTM,” Power System Technology, vol. 45, no. 11, pp. 4444-4451, November 2021, doi: 1000-3673 (2021) 11-4444-08.

[14] G. Gong, X. An, N. K. Mahato, S. Sun, and Y. Yen, “Research on Short-term Load Prediction Based on Seq2seq Model,” Energies, vol. 12, no. 16, pp. 3199, 2019, doi: 10.3390/en12163199.

[15] J. Xu, K. Wang, C. Lin, L. Xiao, and Y. Zhang, “FM-GRU: A Time Series Prediction Method for Water Quality Based on seq2seq Framework,” Water, vol. 13, no. 8, pp. 1031, 2021, doi: 10.3390/w13081031.

[16] Y. Cui, B. Yin, R. Li, Z. Du, and M. Ding, “Short-time Series Load Forecasting By Seq2seq-LSTM Model,” 2020. Accessed: Feb. 26, 2022. [Online]. Available: http://www.researchgate.net/publication/349020193_Short-time_Series_Load_Forecasting_By_Seq2seq-LSTM_Model.

[17] Yang et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” Computer Science, 2014, doi: 10.3115/v1/D14-1179.

[18] YANG Jingxian, ZHANG Shuai, LIU Jichun, et al., “Short-term Photovoltaic Power Prediction Based on Variational Mode Decomposition and Long Short-term Memory with Dual-stage Attention Mechanism,” Automation of Electric Power Systems, vol. 45, no. 3, pp. 174-182, October 2020, doi: 10.7500/AEPS2020026011.

[19] U. Khandelwal, H. He, P. Qi, and D. Jurafsky, “Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context,” 2018, doi: 10.18653/v1/P18-1027.

[20] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” Computer Science, 2014, doi: 10.48550/arXiv.1409.0473.

[21] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, “Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting,” arXiv:1912.09363 [cs, stat], Dec. 2019, Accessed: Feb. 26, 2022. [Online]. Available: http://arxiv.org/abs/1912.09363.

[22] LI Bing, GAO Feng, “Short-term Load Forecasting for Regional Power Grid Based on Virtual Similar Days and Dual-stage Attention-based Long and Short Time Pattern Network,” Automation of Electric Power Systems, vol. 45, no. 22, pp. 55-64, September 2021, doi: 10.7500/AEPS20200718002.

[23] SU Xianguang, ZHOU Wenxin, LI Chaojie, MI Yang, FU Yang, and DONG Chaoyang, “W-BiLSTM Based Ultra-short-term Generation Power Prediction Method of Renewable Energy,” Automation of Electric Power Systems, pp. 1-16, 2021, doi: 10.7500/AEPS20210427004.

[24] J. Qi and GAO Feng, “Short-term Load Forecasting for Regional Power Grid Based on Virtual Similar Days and Dual-stage Attention-based Long and Short Time Pattern Network,” Automation of Electric Power Systems, vol. 45, no. 22, pp. 55-64, 2021, doi: 10.7500/AEPS20210423011.

[25] B. Dua, Y. Min, B. Lsa, G. Tian, and C. Kw, “Short-term multi-energy load forecasting for integrated energy systems based on CNN-BiGRU optimized by attention mechanism,” Applied Energy, vol. 313, pp. 118801-2022, doi: 10.1016/j.apenergy.2022.118801.

[26] Y. Xu and X. Jiang, “Short-term power load forecasting based on BiGRU-Attention-SENet model,” Accessed: May 08, 2022. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/15567036.2022.205322.

[27] Vincent Le Guen, Nicolas Thorne, “Shape and Time Distortion Loss for Training Deep Time Series Forecasting Models,” in 33rd Conference on Neural Information Processing Systems (NeurIPS), Vancouver, Canada, 2019, 4189-4201, doi:10.5555/3454287.3454664.

[28] Hammad, Mahmoud A., Jereb, Borut, Rosi, Bojan and Dragan, Dejan. “Methods and Models for Electric Load Forecasting: A Comprehensive Review.” Logistics, Supply Chain, Sustainability and Global Challenges, vol.11, no.1, 2020, pp.51-76. doi:10.2478/jlst-2020-0004.

[29] A Hybrid Model of Least Squares Support Vector Regression Optimized by Particle Swarm Optimization for Electricity Demand Prediction | Procedings of the 2019 11th International Conference on Machine Learning and Computing,” vol. 43, no. 1, pp. 1-6, 2022. This article has been accepted for publication in IEEE Access. This is the author’s version which has not been fully edited and content may change prior to final publication. Citation information: DOI:10.1109/ACCESS.2022.3182334.

VOLUME XX, 2017
XIANG WANG received the B.S. degrees in international economy and trade from Peking University, Beijing, China, in 2003 and the M.S. in business administration from Beijing Institute of Technology, Beijing, China, in 2009. He is currently an intermediate economist in Beijing Smart-Chip Microelectronics Technology Co., Ltd.. His current research interests include electricity market analysis and power system load forecasting.

ZHANXIA WU received the M.S. in mechanical practice and theory from Dalian Polytechnic University, Dalian, China, in 2007. He is currently a senior engineer in Shenzhen Smart-Chip Microelectronics Technology Co., Ltd.. His current research interests include power system modeling and power system load forecasting.

JUNXIONG GE received the B.S. degrees in enterprise management from Deakin University, Melbourne, Australia, in 2017 and the M.S. in business administration from University of Tasmania, Hobart, Australia, in 2017. He is currently an engineer in Shenzhen Guodian Technology Communication Co., Ltd.. His current research interests include electricity market analysis and power system load forecasting.

ZHANHAO ZHANG received the B.S. degrees in finance from Deakin University, Melbourne, Australia, in 2017. He is currently an engineer in Laijin (Tianjin) Auto Parts Co., Ltd.. His current research interests include power system modeling and power system load forecasting.

LINCHENG HAN was born in ShiJiaZhuang City, China. He received his master's degree from the Army Armored Corps Academy in 2010. He is currently pursuing the Ph.D degree with Northeast Electric Power University, Jilin City. His current research interests is Grid Load Forecasting.

SHUNJIANG WANG was born in Yuhuan City, China. He received the Ph.D. degree in computer application technology from the University of Chinese Academy of Sciences, Shenyang, China, in 2019. He is currently a professorate senior engineer in State Grid Liaoning Electric Power Co., Ltd.. His current research interests include distribution system analysis and power system analysis.

XIUYU ZHANG was born in Jilin City, China. He received the B.S. and M.S. degrees in control theory and application from Northeast Dianli University, Jilin, China, in 2003 and 2006, respectively, and the Ph.D. degree in control theory and application from Beijing University of Aeronautics and Astronautics, Beijing, China, in 2012. He is currently a Professor with the School of Automation Engineering, Northeast Dianli University, Jilin, China. His research interests include robust and adaptive control for nonlinear systems with smartmaterial based actuators.