Hindawi
Complexity
Volume 2021, Article ID 1125630, 10 pages
https://doi.org/10.1155/2021/1125630

Research Article

DWNet: Dual-Window Deep Neural Network for Time Series Prediction

Jin Fan, Yipan Huang, Ke Zhang, Sen Wang, Jinhua Chen, and Baiping Chen

Hangzhou Dianzi University, Hangzhou, China

Correspondence should be addressed to Baiping Chen; chenbp@hdu.edu.cn

Received 22 July 2021; Accepted 25 September 2021; Published 13 October 2021

Academic Editor: Fei Xiong

Multivariate time series prediction is a very important task, which plays a huge role in climate, economy, and other fields. We usually use an Attention-based Encoder-Decoder network to deal with multivariate time series prediction because the attention mechanism makes it easier for the model to focus on the really important attributes. However, the Encoder-Decoder network has the problem that the longer the length of the sequence is, the worse the prediction accuracy is, which means that the Encoder-Decoder network cannot process long series and therefore cannot obtain detailed historical information. In this paper, we propose a dual-window deep neural network (DWNet) to predict time series. The dual-window mechanism allows the model to mine multigranularity dependencies of time series, such as local information obtained from a short sequence and global information obtained from a long sequence. Our model outperforms nine baseline methods in four different datasets.

1. Introduction

In the age of big data, sequence data is everywhere in life [1, 2]. Time series prediction algorithms are becoming more and more important in many areas, such as financial market prediction [3], passenger demand forecasting [4], and heart signal prediction [5]. In most cases, time series data is multivariate. The key to multivariate time series prediction is to obtain the spatial and temporal relationships between different attributes at different times [6]. As a widely used traditional time series prediction algorithm, ARIMA [7] has shown its effectiveness in many areas. However, ARIMA cannot model nonlinear relationships and can only be applied to stationary time series [8–10]. Recurrent neural network (RNN) [11] has achieved great success in sequence modeling. But RNN has the problem of vanishing gradients, and it is difficult to capture the long-term dependence of time series [12]. Long Short-Term memory (LSTM) [13] and gated recurrent unit (GRU) [14, 15] alleviate the problem of RNN's vanishing gradients and have developed many models for time series prediction, such as Encoder-Decoder networks [15, 16]. Encoder-Decoder networks are excellent in time series prediction tasks, especially Attention-based Encoder-Decoder networks [17]. Attention-based Encoder-Decoder network can not only find the spatial-temporal correlation between different series but also find important information in raw data and increase its weight [17]. Among them, dual-stage attention-based recurrent neural network (DARNN) is one of the state-of-the-art methods, creatively using a two-stage attention mechanism [18].

Although DARNN can capture spatial correlations between different attributes at the same time and the temporal correlations between different times in the same attribute, when the length of the sequence is too long, the prediction effect will be worse [18]. This problem is common to all Encoder-Decoder networks. A long sequence means more historical information, so better results should be obtained. However, due to the limitations of Encoder-Decoders, the information of the long sequence is not effectively used, even interfere with the prediction results. This is because LSTM does not solve the problem of vanishing gradient, and when the length of the time series is too long, the previous information will be covered by the latter. Therefore, Encoder-Decoders generally use a small window size to ensure the accuracy of prediction. Dual-stage two-phase attention-based recurrent neural network (DSTP) [19] has made

improvements to this problem of DARNN and optimized the prediction effect of long sequences. However, DSTP still does not make effective use of long sequences.

When the time window size is small, the series is very close to the prediction point. Such data has the closest relationship with the prediction point. For instance, if the values before the prediction point are gradually increasing, then the value at the prediction point is also likely to increase. When the time window size is large, series contain more time steps. It is difficult for other models to extract recent information, such as trends, in such a long series, so it cannot get good prediction results. However, more information brought by more time steps is very important for time series prediction. It is key to how to make good use of the different characteristics of short sequence and long sequence.

To solve this problem, we propose a dual-window deep neural network (DWNet). DWNet consists of two parts. The first part captures spatial and temporal correlations from the short sequence and is responsible for providing recent details, based on Encoder-Decoder [15]. The second part obtains long-term dependencies such as periodicity and seasonality from the long sequence, based on TCN. Temporal convolutional network (TCN) [20] is an emerging CNN-based model. With the parallelism of convolution operation and large receptive field, it has gained everyone's expectations in the areas of sequence modeling. Short-term time series generally contain only one or two periods. However, long-term time series are the opposite, including enough time steps. The setting of two different time window sizes for long sequence and short sequence makes it possible to mine multigranularity dependencies.

The main contributions of our work are as follows:

(i) We propose a dual-window mechanism that can extract multigranularity information from sequences of different lengths.

(ii) We propose the DWNet approach, which includes the advantages of Encoder-Decoder networks and TCN at the same time. Encoder-Decoder networks have a strong ability to mine dependence from the short sequence. Meanwhile, TCN's large receptive field and fast training speed are more suitable for long sequences.

(iii) DWNet can be applied to time series prediction tasks in many domains, and there is no requirement for input data. To justify the effectiveness of the DWNet, we compare it with nine baseline methods using the Human Sports dataset, SML 2010 dataset, Appliances Energy dataset, and EEG dataset. The experiment showed the effectiveness and robustness of DWNet.

2. Related Work

For the time series prediction task, there are various approaches from traditional methods to deep learning methods. As the most famous traditional method, ARIMA can effectively obtain the long-term dependence of target series [7]. However, ARIMA does not consider the spatial correlation between exogenous series [18], can only be used to deal with stationary series [7], and cannot model nonlinear relationships [8]. ARIMA is not suitable for the increasingly complex time series data analysis. As a deep neural network dedicated to machine learning and data mining applications [21–23], RNN can model nonlinear relationships [24] and has achieved great success in time series prediction. However, the gradient vanishing of RNN makes it difficult to obtain long-term dependence from time series. LSTM [13] and GRU [15] add a gating mechanism based on RNN and process the addition and deletion of timing information through the gating mechanism, which alleviates gradient vanishing of RNN. Based on LSTM and GRU, many influential deep neural networks have been proposed, such as the Encoder-Decoder network that has received great attention in the area of natural language processing [17]. Encoder-Decoder networks convert input series into context vector through Encoder and then convert context vector into output through Decoder. Encoder-Decoder networks have a problem. When the length of the sequence increases, the performance of Encoder-Decoder will first become better and then worse [17]. Attention-based Encoder-Decoder network can automatically select important information, thereby effectively alleviating the shortcoming of performance degradation when the length of the sequence increases.

Many attention-based models emerge endlessly. And DARNN [18], GeoMAN [25], and DSTP [19] are models that are improved based on the Attention-based Encoder-Decoder and used for time series prediction. Inspired by some theories of human attention [26], DARNN uses a dual-stage attention mechanism. The first stage uses a spatial attention mechanism to assign different weights to exogenous series to the hidden state of Encoder at the previous time step. The second stage uses a temporal attention mechanism to select the most relevant Decoder hidden states in all time steps. After DARNN was proposed, it has always been one of the state-of-the-art methods in time series prediction. Multilevel Attention Network (GeoMAN) is specially used to predict geo-sensor time series data. Many time series data are collected by sensors distributed in many locations. Such data is called geo-sensor time series data. If each series in the geo-sensor time series is simply treated as a normal attribute, it will lose the connection between different locations. GeoMAN adds local spatial attention and global attention mechanisms to Encoder and adds external factor information to Decoder to solve this problem. DSTP adds a new spatial attention mechanism to Encoder to obtain a spatial correlation between target series and exogenous series so that DSTP achieved better results in the long time series prediction.

While the Attention-based Encoder-Decoder network has attracted much attention, TCN has also shown strong sequence modeling ability [20]. TCN is based on CNN and includes causal convolution, dilated convolution [27, 28], and residual block [29]. To apply to series data, TCN is specially adjusted for different data formats of series and image. TCN has advantages that RNNs do not have. (1) TCN

can process series in parallel and does not need to be processed sequentially like RNN or LSTM. This means that there is no possibility that the information of the previous time step will be overwritten and it also means that there is a faster training speed. (2) TCN's receptive field varies with the number of layers, kernel size, and dilation rate and can be flexibly changed according to a different situation. (3) Compared with LSTM, TCN rarely has the problem of gradient vanishing. Due to the flexible receptive field, fewer parameters than LSTM, and parallel processing, TCN can not only reduce the training time of long sequence but also ensure that the information of the previous time step will not be covered. Therefore, TCN has a strong ability to obtain information from long sequences and is suitable for long sequence modeling.

Long- and short-term time series network (LSTNet) [30] is based on CNN and RNN and realizes that time series have two different dependencies, short-term and long-term. Therefore, LSTNet uses a recurrent-skip mechanism to obtain short-term dependence and then uses RNN to obtain long-term dependence from previous results. But it does not consider that the closer to the prediction point, the more important the information. Therefore, LSTNet will lose some recent information in the time series prediction.

3. Dual-Window Deep Neural Network

3.1. Notation and Problem Statement. In our work, there are two different window sizes, T_l and T_s. Given n exogenous series, that is, $\mathbf{X} = \mathbf{X}_1 = (\mathbf{x}_l^1, \mathbf{x}_l^2, \ldots, \mathbf{x}_l^n)^\top = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{T_l}) \in \mathbb{R}^{n \times T_l}$, we segmented a short series like this $\mathbf{X}_2 = (\mathbf{x}_s^1, \mathbf{x}_s^2, \ldots, \mathbf{x}_s^n)^\top = (\mathbf{x}_{T_l - T_s + 1}, \mathbf{x}_{T_l - T_s + 2}, \ldots, \mathbf{x}_{T_l}) \in \mathbb{R}^{n \times T_s}$. We use $\mathbf{x}_l^i = (x_1^i, x_2^i, \ldots, x_{T_l}^i)^\top \in \mathbb{R}^{T_l}$ to represent the i-th long exogenous series, use $\mathbf{x}_s^i = (x_{T_l - T_s + 1}^i, x_{T_l - T_s + 2}^i, \ldots, x_{T_l}^i)^\top \in \mathbb{R}^{T_s}$ to represent the i-th short exogenous series, and use $\mathbf{x}_t = (x_t^1, x_t^2, \ldots, x_t^n)^\top \in \mathbb{R}^n$ to denote a vector of n exogenous series at time t. We use $\mathbf{Y} = (y_1, y_2, \ldots, y_{T_l})^\top \in \mathbb{R}^{T_l}$ to represent target series, which has the long window size T_l.

Given previous values of target series and exogenous series, that is, $(y_1, y_2, \ldots, y_{T_l})$ with $y_t \in \mathbb{R}$ and $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{T_l})$ with $\mathbf{x}_t \in \mathbb{R}^n$, we aim to predict the next time step value of target series y_{T+1}:

$$\widehat{y}_{T_l+1} = F\left(y_1, \ldots, y_{T_l}, \mathbf{x}_1, \ldots, \mathbf{x}_{T_l}\right), \tag{1}$$

where $F(.)$ is a nonlinear mapping function we aim to learn.

3.2. Model. Figure 1 presents the framework of our method. The input of DWNet is divided into two parts, long series with window size T_l and short series with window size T_s. Short series is a part of long series and is located at the end of the long series (Figure 1 shows the relationship between the two series). Long series is processed by TCN [20] and used to obtain more detailed historical information than short series. The short series is processed by Encoder-Decoder to capture local information. Finally, the output of the two parts is combined to get the predicted value of the target series at time T_{l+1}.

3.2.1. Capture Short-Term Dependence. First of all, we introduce the short series processing module. This part is based on Encoder-Decoder and uses spatial attention and temporal attention mechanism [18] to emphasize key information in short series. Encoder is based on LSTM, the input data of Encoder is short series $\mathbf{X}_2 = (\mathbf{x}_{T_l - T_s + 1}, \mathbf{x}_{T_l - T_s + 2}, \ldots, \mathbf{x}_{T_l}) \in \mathbb{R}^{n \times T_s}$. Given i-th short exogenous series $\mathbf{x}_s^i = (x_{T_l - T_s + 1}^i, x_{T_l - T_s + 2}^i, \ldots, x_{T_l}^i)^\top \in \mathbb{R}^{T_s}$, we use the spatial attention module to adaptively obtain the spatial correlation between exogenous series:

$$e_t^i = \mathbf{v}_e^\top \tanh\left(\mathbf{W}_e\left[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}\right] + \mathbf{U}_e \mathbf{x}^i + \mathbf{b}_e\right), \tag{2}$$

$$\alpha_t^i = \frac{\exp\left(e_t^i\right)}{\sum_{j=1}^n \exp\left(e_t^j\right)}, \tag{3}$$

where $\mathbf{v}_e \in \mathbb{R}^{T_s}$, $\mathbf{W}_e \in \mathbb{R}^{T_s \times 2p}$, $\mathbf{U}_e \in \mathbb{R}^{T_s \times T_s}$, and $\mathbf{b}_e \in \mathbb{R}^{T_s}$ are parameters to learn. Here, p is the hidden size of Encoder and $\mathbf{h}_{t-1} \in \mathbb{R}^p$ and $\mathbf{s}_{t-1} \in \mathbb{R}^p$ are the hidden state and cell state of LSTM unit in the Encoder at time $t-1$. α_t^i is the attention weight measuring the importance of i-th exogenous series at time t. After we get the attention weight, we can adaptively extract exogenous series with

$$\widetilde{x}_t = \left(\alpha_t^1 x_t^1, \alpha_t^2 x_t^2, \ldots, \alpha_t^n x_t^n\right)^\top. \tag{4}$$

Thus, the hidden state at time t can be updated as

$$\mathbf{h}_t = f_e\left(\mathbf{h}_{t-1}, \widetilde{x}_t\right), \tag{5}$$

where f_e is an LSTM unit in the Encoder. The spatial attention module calculates the weight of each exogenous series through equations (2) and (3) at time t and uses $\widetilde{x}_t$ to adjust the hidden state at time t.

The input of Decoder is the previous target series and the output of the Encoder, which is the hidden state of Encoder. Decoder aims to predict $\widehat{y}_{T_l+1}$. To get accurate prediction results, we need to capture the temporal correlation between each series. So, we add a temporal attention module to the Decoder. The same as Encoder, the attention weight of Encoder hidden state at time t is calculated based upon the previous Decoder hidden state and cell state of LSTM unit with

$$d_t^i = \mathbf{v}_d^\top \tanh\left(\mathbf{W}_d\left[\mathbf{h}'_{t-1}; \mathbf{s}'_{t-1}\right] + \mathbf{U}_d \mathbf{h}_i + \mathbf{b}_d\right) \tag{6}$$

$$\beta_t^i = \frac{\exp\left(d_t^i\right)}{\sum_{j=1}^{T_s} \exp\left(d_t^j\right)}, \tag{7}$$

where $\mathbf{v}_d^\top \in \mathbb{R}^p$, $\mathbf{W}_d \in \mathbb{R}^{p \times 2q}$, $\mathbf{U}_d \in \mathbb{R}^{p \times p}$, and $\mathbf{b}_d \in \mathbb{R}^p$ are parameters to learn. q is the hidden size of Decoder, and $\mathbf{h}'_{t-1} \in \mathbb{R}^n$ and $\mathbf{s}'_{t-1} \in \mathbb{R}^n$ are the hidden state and cell state of LSTM unit in the Decoder at time $t-1$. β_t^i is the attention weight and can show the importance of i-th Decoder hidden state at time $t-1$. And, we can get context vector with

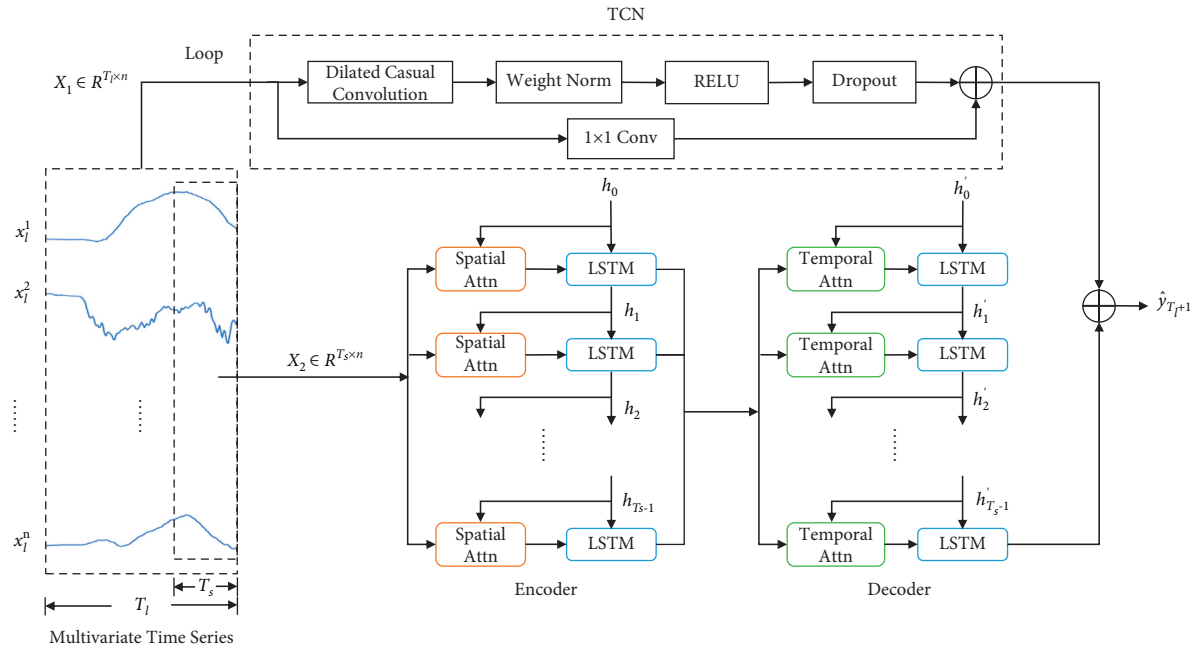

FIGURE 1: Framework of our method. T_l: the window size of long series. T_s: the window size of short series. n: the number of exogenous series. $\mathbf{x}_l^i$: i-th long exogenous series. $\mathbf{h}_t$: hidden state in the encoder at time t. $\mathbf{h}_t'$: hidden state in the decoder at time t. Spatial Attn: spatial attention module. Temporal Attn: temporal attention module. $\hat{y}_{T_l+1}$: the predicting value at time $T_l + 1$.

$$\mathbf{c}_t = \sum_{i=1}^{T_s} \beta_t^i \mathbf{h}_i. \tag{8}$$

Context vector $\mathbf{c}_t$ is the sum of all weighted encoder hidden states at time t. Then, we combine context vector $\mathbf{c}_t$ and target series to update the Decoder hidden state $\mathbf{h}_t'$:

$$\mathbf{h}_t' = f_d\left(h_{t-1}', [\mathbf{c}_t : y_t]\right), \tag{9}$$

where f_d is an LSTM unit in the Decoder.

3.2.2. Capture Long-Term Dependence. We obtain long-term dependence through TCN [20], because TCN can process time series data in parallel and have much fewer parameters than LSTM. Therefore, TCN can quickly handle long series and improve time efficiency. And TCN does not have the problem of the previous information being covered. When window sizes are too large, the integrity of the information can be guaranteed. In our model, the input of the TCN part is long series from time 1 to T_l. In time series analysis, we cannot allow leakage from the future into the past. A high layer element at time t is obtained by convolution of elements from time t and earlier in the previous layer. To avoid information leakage, TCN uses casual convolution. To expand the receptive field, TCN uses dilated convolution [27, 28]. For long exogenous series $\mathbf{X}_1 = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{T_l}) \in \mathbb{R}^{n \times T_l}$ and filter $\mathbf{g}$: $(\mathbf{g}_0, \mathbf{g}_1, \ldots, \mathbf{g}_{k-1})$, the element at time t is

$$O_t = \left(\mathbf{x} * d_g\right)(t) = \sum_{i=0}^{k-1} \mathbf{g}_i \mathbf{x}_{t-d \cdot i}, \tag{10}$$

where d is the dilation factor, k is the filter size, and O is dilated convolution operation. d will increase exponentially with the number of layers to expand the receptive field. A deep neural network is so easy to have the problem of gradient exploding and gradient vanishing, so TCN uses residual block [29]. The residual connection enables the network to transfer information in a cross layer and improve the efficiency of feature extraction.

3.2.3. Training. Figure 1 shows that the predicted value is determined by two parts. We combine the output of Decoder $\mathbf{h}_{T_s}'$ and TCN O_{T_l} to predict $\hat{y}_{T_{l+1}}$:

$$\begin{aligned} \hat{y}_{T_l+1} &= F\left(y_1, \ldots, y_{T_l}, \mathbf{x}_1, \ldots, \mathbf{x}_{T_l}\right) \\ &= \mathbf{v}_y^\top \left(\mathbf{W}_y \left[\mathbf{h}_{T_s}' : O_{T_l}\right] + \mathbf{b}_w\right) + b_v, \end{aligned} \tag{11}$$

where $\mathbf{v}_y \in \mathbb{R}^q, \mathbf{W}_y \in \mathbb{R}^{q \times (q+m)}, \mathbf{b}_w \in \mathbb{R}^q$, and $b_v \in \mathbb{R}$ are parameters to learn. Here, m is the number of hidden units per layer, and $[\mathbf{h}_{T_s}' : O_{T_l}] \in \mathbb{R}^{q+m}$. We use the back-propagation algorithm to train DWNet. We use the Adam optimizer [31] to minimize the mean squared error (MSE) between the predicted value $\hat{y}_{T_{l+1}}$ and the ground truth $y_{T_{l+1}}$:

$$L(\theta) = \left\| \hat{y}_{T_{l+1}} - y_{T_{l+1}} \right\|_2^2, \tag{12}$$

where θ are all parameters to learn in DWNet.

4. Experiment

Our model and all baseline methods are implemented on the PyTorch framework [32]. In this section, we first introduce four different datasets used in the experiment. Then, we

introduce nine baseline methods. Next, we introduce the model evaluation methods and parameters. Finally, experiment results show the effectiveness of DWNet.

4.1. Datasets. We use four datasets to verify the effect of our model. They are in the field of sports, energy, climate, and medicine. We divide datasets into training sets and testing sets according to the ratio of 4 : 1.

4.1.1. Human Sports [33]. Human Sports data is collected by 10 volunteers of different genders, heights, and weights who performed sports including squat, walking, jumping jacks, and high knee. Four sensors worn on the arms and thighs record data every 50 milliseconds, including acceleration and angular acceleration of the x-axis, y-axis, and z-axis. In our experiment, we take the resultant acceleration as the target series and others as exogenous series. We only use the squat data of one volunteer and use the first 8796 data points as the training set and the remaining 2197 data points as the testing set.

4.1.2. SML 2010 [34]. SML 2010 is a public dataset for indoor temperature prediction. SML 2010 contains nearly 40 days of data, which is collected by the monitoring system. The data were sampled every minute, computing and uploading it smoothed with 15-minute means. In our experiment, we take the weather temperature as target series and select fifteen exogenous series. We use the first 1971 data points as the training set and the remaining 493 data points as testing set.

4.1.3. Appliances Energy [35]. Appliances energy is a public dataset used for home appliance energy consumption prediction. This dataset is at 10 minutes for about 4.5 months. Room temperature and humidity conditions were monitored with a wireless sensor network. The energy data is recorded with m-bus energy meters every 10 minutes. Weather data was downloaded from the nearest airport weather station. In our experiment, we take energy use as target series and others as exogenous series. We use the first 15548 data points as a training set and the remaining 3887 as a testing set.

4.1.4. EEG [36]. EEG is a public dataset for classification and regression. This database consists of 30 subjects performing Brain–Computer Interface for Steady-State Visual Evoked Potentials. In our experiment, we only use the data from one of those subjects. We take the electrode O1 attribute as the target series and others as exogenous series. We use the first 7542 data points as a training set and the remaining 1886 as a testing set.

4.2. Baseline

4.2.1. ARIMA [8]. It is one of the well-known statistical algorithms for time series prediction.

4.2.2. LSTM [13]. LSTM is improved by RNN, through the gating mechanism to control the adding and deletion of information, alleviating the gradient vanishing.

4.2.3. Encoder-Decoder [16]. It is widely used in machine translation. However, Encoder-Decoder has the disadvantage of losing information.

4.2.4. Input-Attn-RNN [18]. It adds a spatial attention module on the basis of Encoder-Decoder to the Encoder to obtain the spatial correlation of raw data.

4.2.5. Temp-Attn-RNN [19]. It adds a temporal attention module on the basis of Encoder-Decoder to the Decoder to obtain the temporal correlation of Encoder hidden state.

4.2.6. TCN [20]. It is an emerging sequence modeling model that has attracted much attention, including casual convolution, dilated convolution, and residual blocks.

4.2.7. LSTNet [30]. It combines CNN and RNN to obtain short-term and long-term dependencies in sequence.

4.2.8. DARNN [18]. As one of the state-of-the-art methods, inspired by the human attention system, DARNN uses both spatial attention and temporal attention to extract spatial-temporal correlation.

4.2.9. DSTP-RNN [19]. It improves DARNN and adds an attention module to Encoder. In the Encoder, more stationary weights can be obtained. DSTP-RNN is good at long time series prediction.

4.3. Evaluation Metrics. We employ root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and symmetric mean absolute percentage error (SMAPE) to evaluate our model and baseline methods. These four evaluation metrics are scale-independent and widely used in time series prediction. RMSE has a strong feedback ability for predicted results that deviate too much from the ground truth. MAE treats all results equally. MAPE is able to compare forecast accuracy among differently scaled time series data because relative errors do not depend on the scale of the dependent variable. However, when truth value y_t is small, different $\widehat{y}_t$ will have a huge difference in MAPE value. And SMAPE can solve this problem. Assuming $\widehat{y}_t$ is predicted value at time t and y_t is the ground truth, RMSE is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(\widehat{y}_t - y_t\right)^2}. \tag{13}$$

MAE is defined as follows:

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{N}\left|\widehat{y}_t - y_t\right|. \tag{14}$$

MAPE is defined as follows:

$$\text{MAPE} = \frac{100\%}{N}\sum_{t=1}^{N}\left|\frac{\widehat{y}_t - y_t}{y_t}\right|. \tag{15}$$

SMAPE is defined as follows:

$$\text{SMAPE} = \frac{100\%}{N}\sum_{t=1}^{N}\frac{\left|\widehat{y}_t - y_t\right|}{\left(\left|\widehat{y}_t\right| + \left|y_t\right|\right)/2}. \tag{16}$$

4.4. Parameters Settings. Most time series prediction models have chosen a small window size in their experiment. For example, DARNN set the window size to 10 [18], and GeoMAN set the window size to 6 [25]. To show the influence of window size on prediction, we select the window size $T = \{2, 4, 8, 16, 32, 128\}$. For DWNet, we set $T_l = 128$ and $T_s = 16$. For baseline methods, we conducted experiments on $T = 16$ and $T = 128$, respectively. In training, we set the batch size to 128 and learning rate to 0.001. In our model, there are also parameters such as the hidden size of Encoder p, the hidden size of Decoder q, kernel size, and levels of TCN. For simplicity, we use the same hidden size at Encoder and Decoder, that is, $p = q$, and conducted a grid search over $\{16, 32, 64, 128, 256\}$. For TCN level and kernel size, we also conducted a grid search. The setting in which $p = q = 128$, level $= 8$, kernelsize $= 7$ outperforms the others in the testing set. And we fixed these parameters in all experiments.

5. Results and Discussion

In this section, we first compare our model with baseline methods on the four datasets. Then, we conduct a grid search to show the performance of our model in different long time steps and short time steps combinations. Next, we investigate ablation experiments and study the time efficiency of our model.

5.1. Model Comparison. To show the effectiveness of DWNet, we compare DWNet with 9 different methods, including the state-of-the-art methods and emerging methods. For the sake of fairness, we use two different window sizes for baseline methods so that we can compare the baselines' results of long window size and short window size with DWNet. The prediction results of DWNet and baseline methods are shown in Tables 1 and 2 .

Table 1 shows that DWNet achieves the best RMSE and MAE across four datasets. Table 2 shows that DWNet also achieves the best MAPE and SMAPE in four datasets. This is because DWNet obtains not only the short-term dependency in the short sequence but also the long-term dependency in the long sequence. ARIMA performs worse than other models for ARIMA cannot capture linear relationships and does not consider the spatial correlation between exogenous series [7]. Encoder-Decoder network performs better than normal LSTM in four datasets, which means Encoder-Decoder is easier to obtain dependency from raw data [16]. Attention-based Encoder-Decoder networks, that is, Input-Attn-RNN and Temp-Attn-RNN, are better than normal Encoder-Decoder networks in four datasets because the attention mechanism pays more attention to more important features in raw data. DARNN and DSTP combine spatial attention and temporal attention mechanism and have good performance in four datasets. The performance of TCN is very unstable, and its performance in Human Sports is better than DSTP, but it is far worse than DARNN and DSTP in other datasets, especially EEG. LSTNet's performance is also unstable. And it performs very well in Human Sports, but it performs poorly in the other three datasets. Meanwhile, we can also find that LSTM-based networks perform better than long sequences in short sequences.

5.2. Time Step Study. In this section, we study the impact of long window size T_l and short window size T_s on prediction. When we vary T_l and T_s, we keep other parameters fixed. We plot the RMSE versus different long window size ($T_l \in \{64, 128, 256, 512\}$) and short window size ($T_s \in \{4, 8, 16, 32\}$) in Figure 2.

It is easily observed that the performance of DWNet is simultaneously affected by two parameters T_l and T_s. When T_l is fixed, the performance of DWNet will be worse when T_s is too large or too small and vice versa. And we notice that DWNet achieves the best performance when $T_l = 128$ and $T_s = 16$.

5.3. Ablation Experiment. To further investigate the effectiveness of each model component, we compare DWNet with Input-Attn-RNN, Temp-Attn-RNN, DARNN, and other variants in Human Sports and EEG datasets. In this experiment, we set window size T of Input-Attn-RNN, Temp-Attn-RNN, and DARNN to 16 and set $T_l = 128$ and $T_s = 16$. The variants of DWNet are as follows:

(i) DWNet-ni: there is no spatial attention module in the Encoder part.

(ii) DWNet-nt: there is no temporal attention module in the Decoder part.

The experiment results are shown in Figure 3. Input-Attn-RNN performs better than Temp-Attn-RNN in the EEG dataset but performs worse than Temp-Attn-RNN in the Human Sports dataset. However, DARNN achieves better RMSE and MAE than Input-Attn-RNN and Temp-Attn-RNN in both two datasets. Apparently, the model based on a two-stage attention mechanism is better than the single attention model. And that is why DWNet is superior to DWNet-ni and DWNet-nt. It is easily observed that DWNet achieves the best RMSE in Human Sports and EEG, which shows that the information in the long sequence is valuable for the time prediction task. Without the long

Table 1: RMSE and MAE performance comparison among different methods and datasets (best result is displayed in **boldface**).

Models	SML 2010		Human Sports		EEG		Energy	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
ARIMA (16)	0.2786	0.2219	0.1371	0.0617	0.5694	0.4724	0.8640	0.5740
LSTM (16)	0.1905	0.1489	0.0831	0.0325	0.2244	0.1724	0.6907	0.3663
LSTM (128)	0.2099	0.1671	0.0983	0.0437	0.3033	0.2283	0.8017	0.4376
Encoder-Decoder (16)	0.1438	0.0907	0.0774	0.0296	0.2499	0.1401	0.5983	0.2839
Encoder-Decoder (128)	0.1648	0.1012	0.0831	0.0303	0.4650	0.3036	0.6524	0.3117
Input-Attn-RNN (16)	0.1296	0.0762	0.0680	0.0282	0.2055	0.1447	0.5452	0.2564
Input-Attn-RNN (128)	0.1008	0.0897	0.0766	0.0362	0.4217	0.2881	0.5782	0.2502
Temp-Attn-RNN (16)	0.1097	0.0692	0.0646	0.0311	0.2220	0.1500	0.5414	0.2507
Temp-Attn-RNN (128)	0.1105	0.0770	0.0740	0.0334	0.3943	0.2998	0.5488	0.2563
TCN (16)	0.1156	0.0817	0.0628	0.0270	1.1845	0.9545	0.8279	0.5186
TCN (128)	0.1473	0.1136	0.0727	0.0329	1.1050	0.8696	0.8126	0.4567
LSTNet (16)	0.1277	0.0957	0.0582	0.0269	0.2322	0.1807	0.5733	0.2762
LSTNet (128)	0.1352	0.1020	0.0642	0.0312	0.2384	0.1868	0.6078	0.3296
DARNN (16)	0.0977	0.0644	0.0643	0.0232	0.1804	0.1442	0.5270	0.2439
DARNN (128)	0.1093	0.0778	0.0733	0.0435	0.3483	0.3250	0.5556	0.2525
DSTP (16)	0.0932	0.0614	0.0641	0.0227	0.1805	0.1414	0.5320	0.2459
DSTP (128)	0.0954	0.0670	0.0641	0.0235	0.1754	0.1384	0.5456	0.2525
DWNet	**0.0891**	**0.0565**	**0.0575**	**0.0217**	**0.1702**	**0.1371**	**0.5015**	**0.2362**

The window size of baseline methods is set to 16 and 128, and the short window size and long window size of DWNet are set to 16 and 128, respectively.

Table 2: MAPE and SMAPE performance comparison among different methods and datasets (best result is displayed in **boldface**).

Models	SML 2010		Human Sports		EEG		Energy	
	MAPE (%)	SMAPE (%)	MAPE (%)	SMAPE (%)	MAPE (%)	SMAPE (%)	MAPE (%)	SMAPE (%)
ARIMA (16)	123.0993	62.0098	22.3507	18.4392	159.6348	83.7905	178.4365	77.4287
LSTM (16)	78.9095	45.0082	17.4439	13.9803	80.3427	56.9819	163.9849	65.8066
LSTM (128)	83.1021	49.5439	17.9035	12.0033	87.5609	63.5271	176.3415	69.5442
Encoder-Decoder (16)	70.7142	43.0760	13.3326	9.5610	66.4635	38.5606	170.0863	69.3110
Encoder-Decoder (128)	78.9981	50.5022	15.7987	10.0923	79.3445	40.2327	181.7583	76.1764
Input-Attn-RNN (16)	61.1121	30.0997	11.7831	7.9462	41.8856	32.4032	145.8688	67.5386
Input-Attn-RNN (128)	68.9089	35.3459	12.0034	7.9897	41.4628	29.7608	152.3287	64.7685
Temp-Attn-RNN (16)	54.3435	32.8703	11.2627	7.2110	40.8683	29.0871	140.6838	56.0774
Temp-Attn-RNN (128)	57.8065	31.9911	11.4980	7.1153	45.8705	30.0085	128.0644	59.3527
TCN (16)	83.2350	49.5797	18.5920	11.0097	675.9030	131.4202	258.1707	93.9882
TCN (128)	85.4479	67.3689	14.6141	10.7326	995.0083	133.4580	265.3023	100.9782
LSTNet (16)	50.5956	29.6186	13.0975	8.6524	46.9208	34.3753	135.8396	68.8810
LSTNet (128)	83.2999	46.3060	13.3192	9.3698	50.0573	41.5482	140.2021	72.9974
DARNN (16)	43.1275	28.9558	11.9568	8.0686	36.4658	26.6514	123.0556	59.8798
DARNN (128)	45.2110	33.1612	11.8951	7.4177	33.8550	27.1255	139.0686	64.3326
DSTP (16)	40.6946	24.5261	11.7359	7.1343	34.5063	22.7594	138.8959	59.8884
DSTP (128)	36.2600	24.8048	11.7928	7.3406	35.1179	24.0093	142.8744	56.9903
DWNet	**31.5764**	**23.0888**	**10.3833**	**7.0483**	**31.3287**	**20.6706**	**82.1119**	**52.5880**

The window size of baseline methods is set to 16 and 128, and the short window size and long window size of DWNet are set to 16 and 128, respectively.

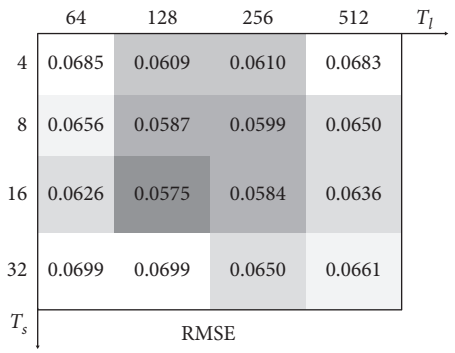

Figure 2: Performance of DWNet in Human Sports based on different short window size T_s and long window size T_l. We use different colors to indicate the prediction effect. The better the prediction, the darker the color.

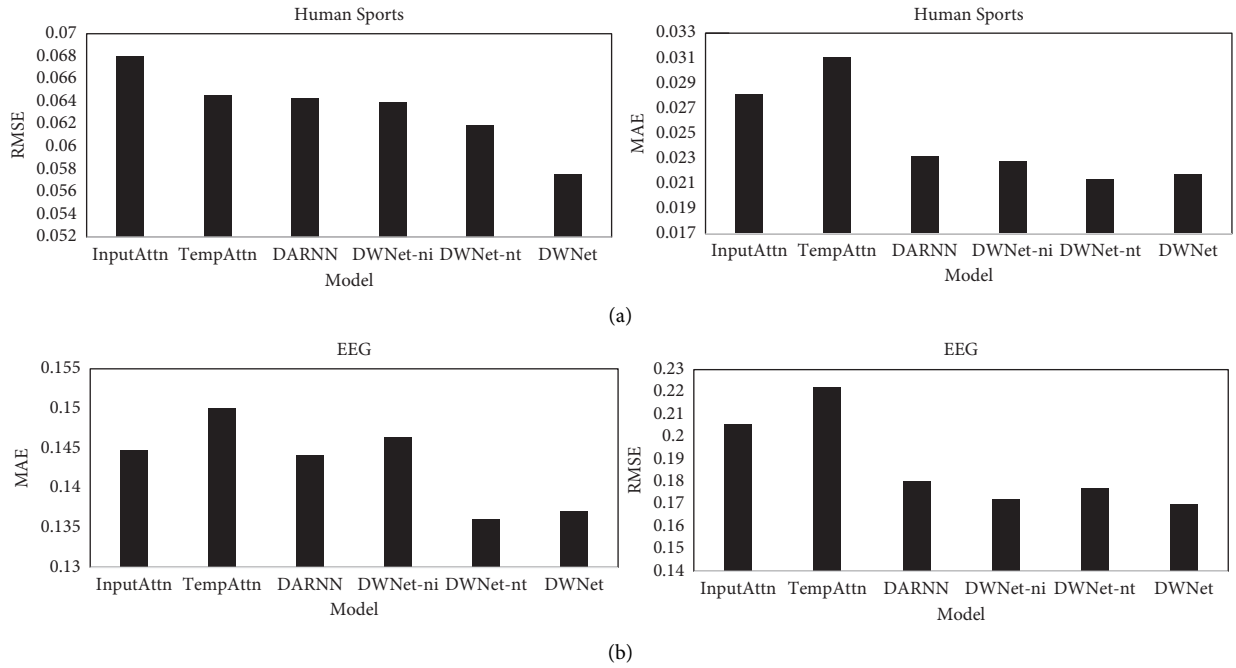

Figure 3: Performance of different methods in Human Sports and EEG. (a) RMSE and MAE versus different methods over Human Sports. (b) RMSE and MAE versus different methods over EEG.

sequence processing module, it is impossible to outperform the state-of-the-art methods in time series prediction.

5.4. Time Complexity. The time efficiency of deep neural networks is also an evaluation metric that needs to be considered. In this section, we compare the time efficiency of DWNet and baseline methods. In this experiment, we set T to 16, T_l to 128, T_s to 16, and fixed other parameters. We experimented on Human Sports and EEG datasets and recorded the time (in seconds) spent in 10 epochs. The results are shown in Figure 4. We can observe that, with more attention modules, the time spent by the model gradually increases. Input-Attn-RNN and Temp-Attn-RNN have only one attention module: one is spatial attention and the other is temporal attention, but the amount of computation is essentially the same. Temp-Attn-RNN's training time is slightly longer than Input-Attn-RNN, but it is far less than the DARNN that both attention modules have. DSTP has two attention modules in the Encoder part and one attention module in the Decoder part, so the training time spent is longer than DARNN. TCN is superior to fewer parameters and the characteristics of parallel processing and has a very large advantage in time spent. It takes the least time in both two datasets. In DWNet, there are two attention modules and a long sequence processing module (implemented by TCN). Therefore, DWNet is inferior to DARNN in terms of time efficiency and even worse than TCN. However, DWNet has stronger time series forecasting capabilities than DARNN and TCN and is more suitable for situations that require high accuracy rather than low time consumption.

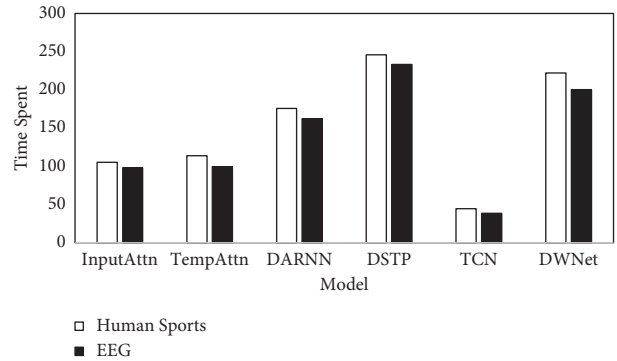

Figure 4: Time spent per 10 epoch in different models and datasets.

6. Conclusion

In this paper, we propose a dual-window deep neural network (DWNet) to make good use of the long sequence for time series prediction. The dual-window mechanism splits the end of a sequence as a short sequence and treats this sequence as a long sequence. The long sequence processing module in DWNet can extract historical information from long time series, and the short sequence processing module obtains recent information from short time series. These allow the model to learn both long-term dependence and short-term of the sequence. Our model outperforms the state-of-the-art methods in four datasets. In the future, we are going to perform model compression and reduce the model running time. Moreover, we will improve the long sequence processing module and enhance its stability, thereby enhancing the performance of DWNet.

Data Availability

The Human Sports dataset is available from Hangzhou Dianzi University's fitness club. Due to personal privacy, data cannot be made publicly available. The remaining datasets analyzed during the current study were derived from the following public domain resources: https://archive.ics.uci.edu/ml/datasets/SML2010 https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction https://archive.ics.uci.edu/ml/datasets/EEG+Steady-State+Visual+Evoked+Potential+Signals.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by a grant from the National Natural Science Foundation of China (no. U1609211) and National Key Research and Development Project (2019YFB1705102).

References

[1] Q. Zhang, J. Wu, H. Yang, Y. Tian, and C. Zhang, "Unsupervised feature learning from time series," in *Proceedings of the IJCAI*, pp. 2322–2328, New York, NY, USA, July 2016.

[2] H. Wang, Q. Zhang, J. Wu, S. Pan, and Y. Chen, "Time series feature learning with labeled and unlabeled data," *Pattern Recognition*, vol. 89, pp. 55–66, 2019.

[3] B. Moews, J. M. Herrmann, and G. Ibikunle, "Lagged correlation-based deep learning for directional trend change prediction in financial time series," *Expert Systems with Applications*, vol. 120, pp. 197–206, 2019.

[4] L. Bai, L. Yao, S. Kanhere, X. Wang, and Q. Z. Sheng, "Stg2seq: spatial-temporal graph to sequence model for multi-step passenger demand forecasting," 2019, https://arxiv.org/pdf/2108.05940.pdf.

[5] S. Fraga, M. A. Aceves-Fernandez, J. C. Pedraza-Ortega, and J. M. Ramos-Arreguin, "Screen task experiments for eeg signals based on ssvep brain computer interface," *International Journal of Advanced Research*, vol. 6, no. 2, pp. 1718–1732, 2018.

[6] Q. Zhang, J. Wu, P. Zhang, G. Long, and C. Zhang, "Salient subsequence learning for time series clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2193–2207, 2018.

[7] D. Asteriou and S. G. Hall, "Arima models and the box–jenkins methodology," *Applied Econometrics*, vol. 2, no. 2, pp. 265–286, 2011.

[8] A. Geetha and G. M. Nasira, "Time-series modelling and forecasting: modelling of rainfall prediction using arima model," *International Journal of Society Systems Science*, vol. 8, no. 4, pp. 361–372, 2016.

[9] L. Yan, A. Elgamal, and G. W. Cottrell, "Substructure vibration narx neural network approach for statistical damage inference," *Journal of Engineering Mechanics*, vol. 139, no. 6, pp. 737–747, 2013.

[10] P. J. Brockwell, R. A. Davis, and S. E. Fienberg, *Time Series: Theory and Methods: Theory and Methods*, Springer Science & Business Media, Berlin, Germany, 1991.

[11] J. L. Elman, "Distributed representations, simple recurrent networks, and grammatical structure," *Machine Learning*, vol. 7, no. 2-3, pp. 195–225, 1991.

[12] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

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

[14] J. Chung, C. Gulcehre, K. H. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, https://arxiv.org/abs/1412.3555.

[15] K. Cho, B. Van Merriënboer, C. Gulcehre et al., "Learning phrase representations using rnn encoder-decoder for statistical machine translation," 2014, https://arxiv.org/abs/1406.1078.

[16] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: encoder-decoder approaches," 2014, https://arxiv.org/abs/1409.1259.

[17] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, https://arxiv.org/abs/1409.0473.

[18] Q. Yao, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," 2017, https://arxiv.org/abs/1704.02971.

[19] 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 Systems with Applications*, vol. 143, p. 113082, 2020.

[20] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, https://arxiv.org/abs/1803.01271.

[21] X. Su, S. Xue, F. Liu et al., "A comprehensive survey on community detection with deep learning," 2021, https://arxiv.org/abs/2105.12584.

[22] F. Liu, S. Xue, J. Wu et al., "Deep learning for community detection: progress, challenges and opportunities," 2020, https://arxiv.org/abs/2005.08225.

[23] X. Ma, J. Wu, S. Xue et al., "A comprehensive survey on graph anomaly detection with deep learning," 2021, https://arxiv.org/abs/2106.07178.

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

[25] Y. Liang, K. Songyu, J. Zhang, X. Yi, and Y. Zheng, "Geoman: multi-level attention networks for geo-sensory time series prediction," in *Proceedings of the IJCAI*, pp. 3428–3434, Stockholm, Schweden, July 2018.

[26] R. Hübner, M. Steinhauser, and C. Lehle, "A dual-stage two-phase model of selective attention," *Psychological Review*, vol. 117, no. 3, pp. 759–784, 2010.

[27] A. Van Den Oord, D. Sander, H. Zen et al., "Wavenet: a generative model for raw audio," 2016, https://arxiv.org/abs/1609.03499.

[28] Y. Fisher and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, https://arxiv.org/abs/1511.07122.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[30] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *Proceedings of the 41st International ACM SIGIR*

Conference on Research & Development in Information Retrieval, pp. 95–104, Ann Arbor, MI, USA, July 2018.

[31] D. P. Kingma and B. Jimmy, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.

[32] P. Adam, S. Gross, S. Chintala et al., Automatic differentiation in pytorch, 2017.

[33] J. Fan, H. Wang, Y. Huang, K. Zhang, and B. Zhao, "Aedmts: an attention-based encoder-decoder framework for multi-sensory time series analytic," *IEEE Access*, vol. 8, pp. 37406–37415, 2020.

[34] F. Zamora-Martínez, P. Romeu, P. Botella-Rocamora, and J. Pardo, "On-line learning of indoor temperature forecasting models towards energy efficiency," *Energy and Buildings*, vol. 83, pp. 162–172, 2014.

[35] L. M. Candanedo, V. Feldheim, and D. Deramaix, "Data driven prediction models of energy use of appliances in a low-energy house," *Energy and Buildings*, vol. 140, pp. 81–97, 2017.

[36] S. M. Fernandez-Fraga, M. A. Aceves-Fernandez, J. C. Pedraza-Ortega, and S. Tovar-Arriaga, "Feature extraction of eeg signal upon bci systems based on steady-state visual evoked potentials using the ant colony optimization algorithm," *Discrete Dynamics in Nature and Society*, vol. 2018, Article ID 2143873, 2018.