International Journal of Computational Intelligence Systems (2022) 15:67
https://doi.org/10.1007/s44196-022-00128-y

RESEARCH ARTICLE

Short-Term Electrical Load Forecasting Based on Time Augmented Transformer

Guangqi Zhang[1] · Chuyuan Wei[2] · Changfeng Jing[3] · Yanxue Wang[1]

Received: 26 November 2021 / Accepted: 28 July 2022

Abstract
Electrical load forecasting is of vital importance in intelligent power management and has been a hot spot in industrial Internet application field. Due to the complex patterns and dynamics of the data, accurate short-term load forecasting is still a challenging task. Currently, many tasks use deep neural networks for power load forecasting, and most use recurrent neural network as the basic architecture, including Long Short-Term Memory (LSTM), Sequence to Sequence (Seq2Seq), etc. However, the performance of these models is not as good as expected due to the gradient vanishing problem in recurrent neural network. Transformer is a deep learning model initially designed for natural language processing, it calculates input–output representations and captures long dependencies entirely on attention mechanisms which has great performance for capturing the complex dynamic nonlinear sequence dependence on long sequence input. In this work, we proposed a model Time Augmented Transformer (TAT) for short-term electrical load forecasting. A temporal augmented module in TAT is designed to learn the temporal relationships representation between the input history series to adapt to the short-term power load forecasting task. We evaluate our approach on a real-word dataset for electrical load and extensively compared it to the performance of the existed electrical load forecasting model including statistical approach, traditional machine learning and deep learning methods, the experimental results show that the proposed TAT model results in higher precision and accuracy in short-term load forecasting.

Keywords Short-term load forecast · Transformer · Deep neural network · Time augmentation

Abbreviations

LSTM	Long short-term memory
BiLSTM	Bi-directional long short-term memory
Seq2Seq	Sequence to sequence
ARIMA	Autoregressive integrated moving average
SVR	Support vector regression
TAT	Time augmentation Transformer
RF	Radom forest
GBM	Gradient boosting machines
CNN	Convolutional neural network
RNN	Recurrent neural networks
LSSVM	Least squares support vector machine
MSE	Mean squared error
GRU	Gate recurrent unit
RMSE	Root mean squared error
MAE	Mean absolute error
MAPE	Mean absolute percentage error

1 Introduction

Electricity has become a necessity of daily life in the modern world. Recently, the global demand and usage of electricity has been increasing drastically due to urban development, industrial expansion, climate change, population growth and so on [1–3]. However, the process of power scheduling and transmission is costly and the amount of power is insufficient to meet global demand. As a solution, many studies aim to use various methods to forecast future electricity demand so that the governments and power companies can plan ahead

✉ Chuyuan Wei
weichuyuan@bucea.edu.cn

[1] School of Mechanical-Electronic and Vehicle Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China

[2] School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing 100044, China

[3] School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing 100044, China

Published online: 18 August 2022

effectively and promote energy efficiency among customers [4].

Electrical load forecasting is of vital importance in intelligent power management and has been an interest topic in academic and business domains [5]. The electrical load forecasting is not only a task to reasonably guide power planning, but also an important guarantee for improving the economy of the power system and ensuring the safe operation of the electrical grid. Hence, as an essential function for power management, electrical load forecasting is crucial to the relevant decision-making. However, accurate forecast of electrical load using time series data of historical electric consumption is still a challenging task [6]. Due to the complex patterns and dynamics of the data, it may be affected by various factors, including temperature, seasons, economy and some unpredictable events [7]. How to fit these complex factors affecting power demand into the prediction models needs to be solved urgently [8]. With the different forecasting scale time, the work can be categorized as three types: short-term, medium-term and long-term [9]. Short-term load forecasting can offer strong support for real-time scheduling and operation planning of power system, and reduce the excessive consumption of energy [10]. It has always been a hot spot in power research, with more and more new methods being introduced including statistical methods and machine learning methods. Statistical methods commonly used for power load and network traffic forecasting such as ARIMA [11, 12], etc. can effectively use the input historical data to predict future power load. But with increasing demand for higher forecast accuracy, the predictive power of these models is insufficient, since it's difficult to deal with complex patterns and dynamic electrical demand data for statistical approaches. Machine learning methods, such as Support Vector Regression (SVR) [13], Radom Forest (RF), Gradient Boosting Machines (GBM) [14], etc. are also used for power load forecasting because of their powerful ability on processing and analyzing some nonlinear and complex problems. In recent years, deep learning methods, which gain ground on feature extraction than traditional machine learning methods, have developed rapidly and been able to predict power load more accurately. Many models based on deep learning methods are applied for short-term load forecast such as Convolutional Neural Network (CNN), Recurrent Neural Networks (RNN), LSTM, Bi-directional long short-term memory (Bi-LSTM), Seq2Seq [15, 16], etc.

Transformer is also a deep learning method with a new network architecture initially designed for machine translation[17]. It entirely depends on the attention mechanisms without sequence aligned recurrence and convolutions to calculate input–output representations and captures long sequence dependencies. Transformer model has great performance for capturing the complex dynamic nonlinear sequences dependence on long sequence input to provide a new possibility for power load forecasting. In this work, we focus on the short-term forecasting with multivariate time series data, and propose a new model Time Augmented Transformer (TAT) based on an adaptation of the recent deep self-attention Transformer architecture incorporating a time augmentation method for short-term load forecast. The main contributions and novel findings are the following:

1. A highly accurate electrical short-term load forecasting approach based on Transformer Model was developed. We have modified the original Transformer to adapt the electrical load forecasting to successfully improve the prediction capacity.
2. The new Time Augmented Transformer model is proposed on the basis of an adaptation of the recent deep self-attention Transformer architecture. We extracted additional time features as augmentation encoding to enhance the temporal representation of the historical input sequences. The TAT model further improved the ability of learning the nonlinear relationship between load data and achieved great improvement.
3. We carefully designed experiments to demonstrate that multivariate feature input is more appropriate for the proposed model in the short-term load forecasting task and our approach can use less historical information to make more accurate predictions, which means less memory occupancy and faster calculation speed.

2 Related Work

Previous work on short-term electrical load forecast can be classified into statistical approaches, machine learning and deep learning [4]. Many statistical methods used in electrical load forecasting just like ARIMA [18, 19]. Wei and Zhang [20] proposed an ARIMA model for short-term electrical load forecasting. However, it's difficult to deal with complex patterns and dynamic electrical demand data for statistical approaches, and it has high requirements for the stationarity of the data, so that the accuracy of the prediction results by statistical methods are not enough and the statistical approaches fail to achieve the expected forecasting results.

In recent years, machine learning methods have gradually been investigated for power load forecasting. Artificial intelligence-based methods have accounted for 90 percent of power forecasting research models during 2010 to 2020 [4]. Yi, Niu [21] using a wavelet transform with least squares support vector machine (LSSVM) to predict demand power. The random forest was used for short-term load prediction in one day ahead of one-step in Tunisia [22]. Besides, Zhang, Li [23] compared three

kinds of models, multiple linear regression, RF and gradient boosting, for hourly electricity load forecasting in southern California, the result demonstrated that gradient boosting has the best performance.

Along the rapid development of artificial intelligence, deep learning has widely been used in natural language understanding, image processing, autonomous driving and other fields [24, 25]. Deep learning methods can not only capture the complex dependencies in nonlinear dynamic system, but also achieve remarkable performance in many prediction applications with higher accuracy [26, 27]. Tokgöz and Ünal [28] built a forecasting model based RNN with an ant colony optimization algorithm and improved the prediction accuracy in electrical load forecasting. However, RNN has the problem of gradient vanishing when dealing with long sequence input that the back-propagation error either decays rapidly or grows beyond the limit, and it is difficult to capture the long-distance dependencies between sequences. Long Short Term Memory (LSTM), which is a further developed model based on RNN, realizes the function of forgetting or remembering using "Gates" to control the discarding or adding of information to solve the problem of gradient disappearance of RNN [29]. Peng, Shuai [15] have applied LSTM to improve the forecast accuracy of traditional RNN model. Besides, CNN has also been used in load forecast because of its excellent ability to capture the trend of load data. Wang, Zhao [30] proposed a mothed based on the integration of CNN and LSTM and the results in higher precision in short-term forecasting. Taking into consideration to utilize the global historical information, Gong, An [16] developed a short-term load prediction model based on Seq2Seq, which use encoder–decoder architecture, has exhibiting better performance. However, Seq2Seq model uses a recurrent neural network structure as encoder to encode historical information into an intermediate vector, it will inevitably lose the dynamic dependencies between historical sequences in the encoder vector.

3 The Proposed Approach

3.1 Problem Description

We can convert the power load forecasting to a supervised learning problem, in multi-step ahead electric load forecasting, the input sequence under the rolling forecasting setting with a sliding window, a history time series of historical electrical load and relative features $\boldsymbol{X} = \{x_{t_1}, x_{t_2}, \ldots, x_{t_n} | x_{t_i} \in R^{d_x}\}$ was given, and the output is the prediction of the next m-step electrical load sequence $\boldsymbol{Y} = \left\{x_{t_{n+1}}, x_{t_{n+2}}, \ldots, x_{t_{n+m}} | x_{t_i} \in R^{d_y}\right\}$, where d_x is the number of feature in the input vector and x_{t_i} can be a scaler or a vector that consists of multiple features including historical electrical load, dry bulb temperature, wet bulb temperature, dew point temperature, hours and electricity price, and $d_y = 1$. Figure 1 shows the sliding window for the input electrical load sequence. In this work, for short-term electrical load forecasting, we will make predictions for 30 min, 1 h, 12 h and one day respectively using historical data from the previous 1 day as input, that means $m = 1, 2, 24, 48$ and $n = 48$ while a time step m denotes 30 min.

3.2 Time Augmentation Transformer Model

The Transformer model entirely depends on the attention mechanisms without sequence aligned recurrence and convolutions to calculate input–output representations and captures long sequence dependencies [17]. Furthermore, it does not process data in an ordered sequence manner, but used attention mechanisms to process entire sequence to learn dependencies without regard to their distance from input sequences. Therefore, Transformer-based model has the

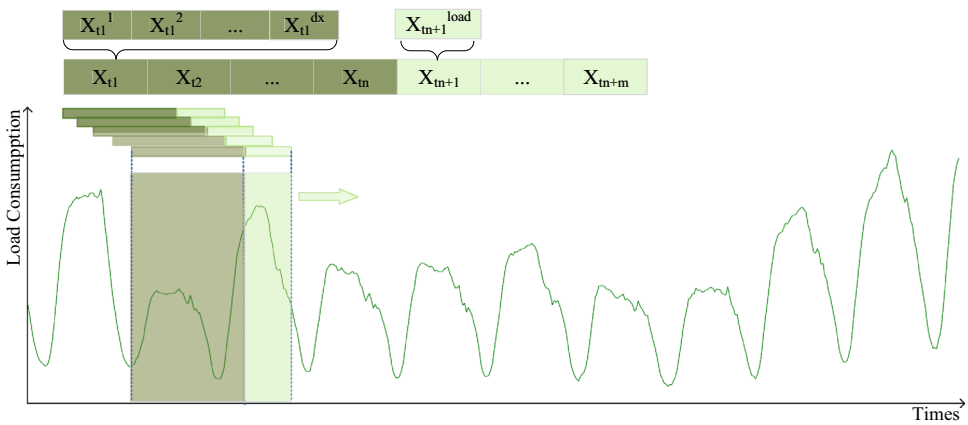

Fig. 1 Sliding windows to construct supervised learning examples for rolling forecasting

potential to model complex dynamics of the electrical load data [31]. Because of the transformer design for machine translation, it cannot be directly used to forecast the electrical load. To this end we have modified the Transformer to adapt our task.

The structure of our Time Augmentation Transformer named TAT is show in Fig. 2. TAT model use encoder–decoder architecture. All the historical load and features are inputted into the encoder to generate a history global information coding result after fusion time information in input layer. The decoder uses the one-position shifted future load data and the historical global attention vector encoded by the encoder as the input to predict the electrical load on next step.

Input Layer: The input layer is composed of a fully connected layer, a position encoding layer and a time augmented encoding layer. The historical electrical load data firstly entry the input layer. Unlike the original Transformer architecture, the historical observation $\boldsymbol{X} \in \mathbb{R}^{n\times d_x}$ is transformed to $\boldsymbol{X} \in \mathbb{R}^{n\times d_{model}}$ that maps the input data to a vector of dimension d_{model} by employing a fully connected layer, where n is the input time step of historical data, d_x is the number of input features for a single time step. Positional-encoding PE was added to above the fully connected layer, it injects relative and absolute position information of the input sequence using sine and cosine functions:

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{model}}\right) \\ PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{model}}\right) \end{aligned} \tag{1}$$

Power load forecasting task is a time-dependent forecasting task. However, inputting the global time information split into "Year, Month, Day, etc." as additional feature with other variables into the Transformer model has resulted in decrease for prediction accuracy, because too many feature inputs will bring more noise to the model. The positional embedding of the basic Transformer can only obtain the sequential representation between the input sequences but failed to effectively represent the relationship of each point in the sequence in the global time. For example, in real-world scenarios, consumers will consume more electricity at night than during the day, and more on weekends than weekdays. It is difficult for the basic Transformer model to effectively utilize the time information in the power load data. To better learn the time relationship between historical data, we proposed a time augmentation layer to enhance the temporal representation of the historical input sequence. For each time step of the input sequence, the basic time feature as input for time augmentation layer T_{t_i} such as "2010/1/1 00:30" used for generation of derived features: Year, Y; Month, M; Day, D; Time-stamp of the day, divided into 30 min interval each, H; Current day of the week, W; Holidays represented by a binary label L. We convert these discrete temporal features to one-hot encoding and concatenate them to a vector $T_i \in \mathbb{R}^{n\times d_t}$, where d_t is total dimension of the one-hot encoding of the temporal features:

$$T_i = \text{Concat}(\text{one} - \text{hot}(Y, M, D, H, W, L) \tag{2}$$

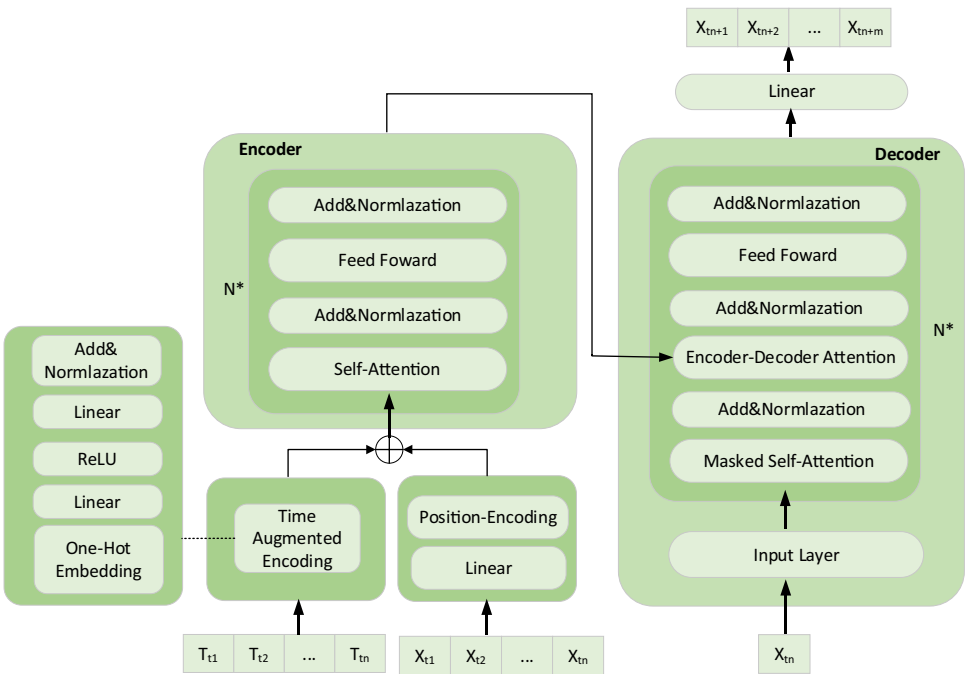

Fig. 2 Structure of Time Augmented Transformer model for load forecasting

Each time step's time encoding $T_i \in \mathbb{R}^{n\times d_t}$ is transformed to $T_i \in \mathbb{R}^{n\times d_{model}}$ employed two fully connected network and ReLU activation function:

$$\mathrm{FFN}(T_i) = \max(0, T_i W_1 + b_1) W_2 + b_2 \tag{3}$$

While W_1, W_2, b_1 and b_2 are the learnable parameter matrices of linear mapping. In addition, to prevent the disappearance of gradients caused by the excessive number of layers in the overall model, we use a residual connection and layer normalization which can be expressed as (4):

$$T_i = \mathrm{LayerNorm}(T_i + FFN(T_i)) \tag{4}$$

Thus, we have the final vector X_i as the final inputs to Encoder and Decoder which contains the original sequence input, absolute position information PE from position encoding and time information T_i from the time augmentation layer:

$$X_i = X_i + T_i + PE \tag{5}$$

Encoder: The encoder is composed of a stack of encoder layers and the number of encoder layers is a free parameter. The vector is fed into the encoder layer after being processed by the input layer. There are a multi-head self-attention sub-layer and a fully connected feed-forward sub-layer in each encoder layer. As the name implies, self-attention is responsible for the calculation of the attention of the input sequence within the encoder. The entered historical sequence uses encoder to encode all historical load information and captures each of their interdependence, and the context vector encoded by encoder will be inputted to the decoder and provide global historical load information for the decoder. Besides, to speed up the training and reduce the disappearance of gradients, a residual connection [32] and layer normalization [33] was employed for each of the two sub-layers.

Decoder: The decoder is also consisted of a stack of decoder layers. In the training phase, the input of the encoder is the sequence shifted one-position offset to the target output we predict and the start token of decoder is the load in last step of encoder's input sequence. In the predicting phase, the input of the decoder is just one data which is the load in last step of encoder's input sequence, and predict load in next time step by step. The input of sequence is transformed into a d_{model} dimensional vector representation through input layer and position encoding, and then feed to a stack of decoder layer. There are three sub-layers in each of decoder layer: an encoder–decoder attention layer, a fully connected feed-forward network layer and a masked multi-head self-attention layer. For self-attention layer in decoder, self-attention is modified to a masked self-attention by setting the sequence after the current prediction step to $-\infty$, since each position can attend to all positions when performing attention calculations in decoder and it will result in the disclosure of future sequence information when decoder makes prediction. To train decoder in batches during the training phase, we use an upper triangular matrix as the masking to prevent the decoder from obtaining future information. Encoder–decoder attention performs multi-head attention over the input of the decoder and the output of the encoder stack. It converts the vector of the encoded historical electrical load feature to generate global attention vector as the input of the decoder by building the relationship between data from each historical time step and every future time step. Finally, the soft-max layer that is used for classifying in original Transformer is also omitted, we use a fully connected layer transformed the $\boldsymbol{Y} \in \mathbb{R}^{m\times d_{model}}$ output vector from decoder to $\boldsymbol{Y} \in \mathbb{R}^{m\times 1}$, where m is the number of forecasting ahead step and we use Mean Squared Error (MSE) loss to measure training loss:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{6}$$

Self-Attention: Attention is an indispensable and complex cognitive function of human beings, which refers to the selective ability of focus on some information while ignoring others [34]. The attention mechanism draws on the human brain to improve the ability of the neural network to process information. When the neural network processes a large amount of input information, the attention mechanism allows the network just to select some key information as inputs.

The calculation of the attention mechanism can be divided into two steps: the first step is to calculate the attention distribution on all input sequences, and the second step is to calculate the weighted average of the input sequences based on the attention distribution [35]. The N group input information is represented by $X = [x_1, \ldots, x_N] \in \mathbb{R}^{D\times N}$, where D-dimension vector $x_n \in \mathbb{R}^D, n \in [1, N]$. The input information can be represented in a query-key-value pair format, for each input x_i, first map it linearly to three different spaces to get the query vector $q_i \in \mathbb{R}^{D_k}$, the key vector $k_i \in \mathbb{R}^{D_k}$ and the value vector $v_i \in \mathbb{R}^{D_k}$. For the entire input sequence X, the linear mapping process can be expressed as (7) (8) (9), while $W_q \in \mathbb{R}^{D_k\times D_x}$, $W_k \in \mathbb{R}^{D_k\times D_x}$ and $W_v \in \mathbb{R}^{D_v\times D_x}$ are the parameter matrices of linear mapping. $Q = [q_1, \ldots, q_N]$, $K = [k_1, \ldots, k_N]$, $V = [v_1, \ldots, v_N]$ are the matrices composed of query vector, key vector and value vector respectively.

$$Q = W_q X \in \mathbb{R}^{D_q\times N} \tag{7}$$

$$K = W_k X \in \mathbb{R}^{D_k\times N} \tag{8}$$

$$V = W_v X \in \mathbb{R}^{D_v\times N} \tag{9}$$

Transformer uses the scaled dot product as the attention scoring function to calculate the attention distribution. When

the dimension D of the input vector is relatively high, the value of the dot product model usually has a large variance, resulting in a small gradient of the soft-max function. Using the scaled dot product can solve this problem well. The formula of the dot product model is as follows:

$$(x, q) = \frac{x^{\intercal} q}{\sqrt{D}} \tag{10}$$

For each query vector $q_n \in Q$, the key-value pairs attention mechanism of formula (11) is used to obtain the output vector:

$$\begin{aligned} h_n &= \mathrm{att}\big((K, V), q_n\big) \\ &= \sum_{j=1}^{N} \alpha_{nj} v_j \\ &= \sum_{j=1}^{N} \mathrm{softmax}\big(s\big(k_j, q_n\big)\big) v_j \end{aligned} \tag{11}$$

where $n, j \in [1, N]$ is the position of the output and input vector sequences, α_{nj} represents the weight of the n-th output focusing on the j-th input. The output vector sequence can be abbreviated as:

$$\begin{aligned} H &= \mathrm{softmax}\left(\frac{QK^{\intercal}}{\sqrt{D_k}}\right) V \\ &= \sum_{n=1}^{N} \frac{\exp\left(\frac{QK^{\intercal}}{\sqrt{D_k}}\right)}{\sum_j \exp\left(\frac{QK^{\intercal}}{\sqrt{D_k}}\right)} V \end{aligned} \tag{12}$$

The self-attention module makes the historical load feature sequence and the future load sequence interrelated, so that the embedding representation of the source sequence and the target sequence will contain more abundant information. The information input from the attention layer to the subsequent FFN also has stronger model representation ability. The self-attention mechanism was shown in Fig. 3.

4 Experiment

4.1 Dataset and Preprocessing

The electrical load data of New South Wales were publicly obtained from the Australian National Electricity Market, where data points are collected every half hour, 5 years from 2006 to 2010. Each data point consists of the target value electrical load and other six features including: hours, dry bulb temperature, wet bulb temperature, dew point temperature, humidity and electricity price.

We use data from the first 5 years as the training set, the first 6 months of the last year as the validation set, and the last 6 months as the test set. All of the data was normalized via the zero-mean method. Then a fixed-length sliding window show in Fig. 1 was applied to construct (X, Y) pairs, in which X are previous n-step feature vector including our target electrical load data and Y are next m-step data as our forecast target.

4.2 Experimental Design

We compared our Time Augmented Transformer model with following forecasting models: ① ARIMA; ② SVR; ③LSTM; ④Bi-LSTM; ⑤CNN-LSTM; ⑥Seq2Seq; ⑦Basic Transformer.

For all methods, the input history data length for model is 48 step and the step of predict length is chosen from {1, 2, 24, 48} that means 30 min, 1 h, 12 h and 1 day. For ARIMA, we choose the parameter as $p = 1, d = 2\ and\ q = 1$ by analyzing the ACF and PACF diagrams produced from dataset. For SVR model, we used a multiple regression strategy to use SVR for multi-step prediction. For LSTM, we set a dense connected network and a stack of LSTM layers. The data input into LSTM-layer for learning historical sequential information, and the final output from the LSTM-layer was feed into dense connected layer to fit the number of steps for target prediction. For seq2seq, we used the Gate Recurrent Unit (GRU) and dense connected network as the basic components. The encoder in

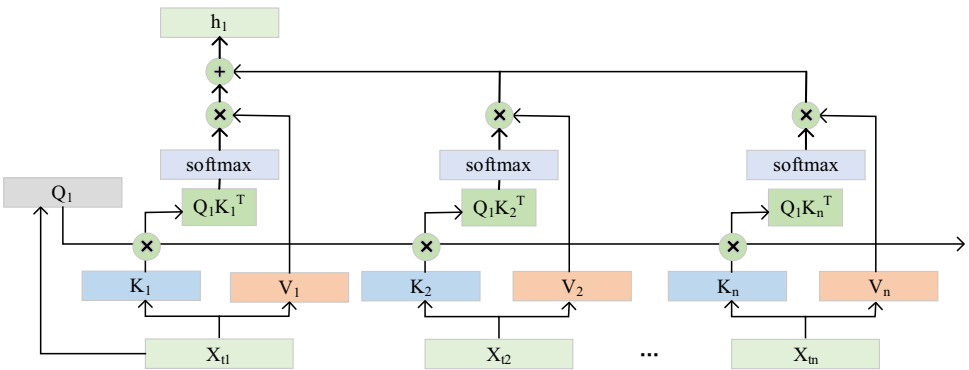

Fig. 3 Self-attention mechanism

Seq2seq receive and process historical input data. The results of the GRU network in decoder was feed into a fully connected feedforward neural network and then predict backwards step by step with autoregressive methods. For LSTM, Bi-LSTM and Seq2Seq, the size of hidden state is chosen from {16, 32, 64, 128, 256} and the number of layers was chosen from {1, 2, 3, 4}. For the model of CNN-LSTM, we choose one dimensional convolution and the number of filters is 64 with kernel size 3, and the size of hidden state is 200.

For basic Transformer and TAT, the head number of multi-head attention was chosen from {8,16}, the layers of encoder and decoder was chosen from {2, 3, 4, 5, 6}, and the dimension of multi-head attention's output was chosen from {16, 32, 64, 128, 256, 512} respectively. We use grid search to select optimal hyper-parameters by observing their performance in the validation set. We set the number of encoder layer to 4, decoder layer to 2, the dimension of model to 64, the number of heads to 8, the number of hidden states in FFC layer to 2048 and the dimension of attention q, k and v to 8. For time augmentation layer, we set the hidden state size to 1024. Our model was optimized with Adam optimizer [36], and we use $1e^{-5}$ as learning rate. For best generalization performance, we use 20 epochs with proper early stopping for all deep learning methods. A mini-batch of size 1024 was used for training. Besides, the dropout of 0.2 was applied for our model.

We computed Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) between the actual data and the predicted value to evaluate the performance for all the methods. The measures of test error RMSE and MAPE are expressed as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2} \tag{13}$$

$$\mathrm{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{14}$$

All the experiments were carried out on a personal server with two Nvidia Tesla V100 (16 GB) GPU.

4.3 Results and Discussion

4.3.1 Multi-step Ahead Forecasting for SW Data

In our first experiment, we compare different methods for 30 min (1 step), 1 h (2 steps), 12 h (24 steps) and one day (48 steps) ahead forecasting on the electrical dataset on New South Wales. For all the predictive models, we used 24 h of historical data (48 historical steps) as input vector. We compared our TAT model's performance with ARIMA, LSTM, Bi-LSTM, CNN-LSTM, Seq2Seq and basic Transformer. Table 1 summarizes the MAE, MSE and MAPE values for each method for multi-step ahead forecasting and our model have the best results in all different time step ahead predictions.

Figure 4. shows the predictions of the four models at prediction steps 1, 12, 24 and 48 ahead in subgraph a, b, c and d respectively. It can be seen that the prediction results of each model are both accurate when predicting 1 step forward. However, our method is more sensitive to local changes in the load curve and can more accurately predict subtle changes in the electrical load over a shorter period of time, as show in local zoomed images in subgraph Fig. 4 (a). As the prediction step increases, the deviation of the prediction curve of ARIMA, machine learning and other deep learning methods from the real curve gradually increases, while the prediction result of our method is closer to the actual value curve than other models, especially at the bottom and top of the power load curve in subgraph Fig. 4 (b), (c), (d), which demonstrates that our model shows significantly better results than other forecasting models.

Table 1 Comparison of different models for forecasting multi-step electrical load

Model	30 min			1 h			12 h			1 day		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
ARIMA	118.39	157.51	1.17%	206.42	306.99	2.08%	734.06	963.28	8.71%	800.57	1015.36	9.43%
SVR	86.55	109.71	1.01%	197.91	238.67	2.37%	374.90	439.61	4.51%	463.41	556.72	5.55%
LSTM	93.23	116.46	0.938%	358.80	475.64	3.98%	375.70	493.92	4.33%	498.85	649.25	5.82%
Seq2Seq	100.40	133.90	0.994%	157.65	223.33	1.59%	310.33	430.05	3.64%	486.88	639.13	5.68%
Bi-LSTM	83.36	106.09	0.858%	117.65	156.66	1.20%	314.21	447.06	3.68%	408.14	562.25	4.67%
CNN-LSTM	83.20	106.55	0.860%	104.59	137.89	1.07%	301.21	398.55	3.40%	370.51	483.21	4.24%
Transformer	78.81	99.91	0.744%	84.19	113.38	0.82%	281.10	376.69	3.27%	339.67	449.32	3.89%
TAT	*47.20*	*62.49*	*0.465%*	*65.99*	*87.59*	*0.618%*	*147.55*	*215.18*	*1.69%*	*230.24*	*325.21*	*2.62%*

Italic values are the best results on traditional and deep neural methods

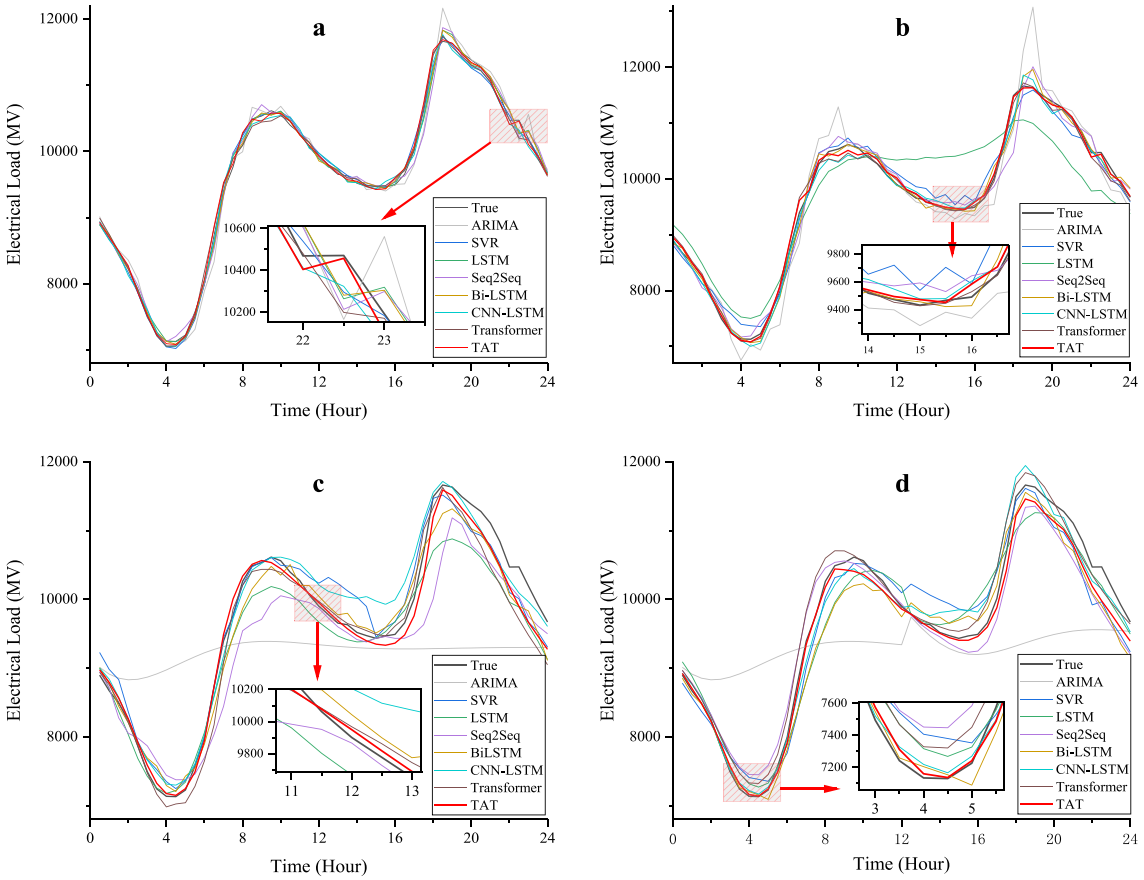

Fig. 4 Result in forecast testing of different-step forecasting for 30 min, 1 h, 12 h, and 1 day ahead respectively using 4 models

4.3.2 Multivariate and Univariate Variable Input with Multi-step Ahead Forecasting

Our model can be used for both univariate and multivariate input predictions by adjusting the input layer of encoder. To solve the prediction problems for univariate inputs, we only use load consumption as a single variable time series and construct supervised learning pairs by sliding windows to use the historical load to predict the subsequent multi-step load. For multivariate variable input, we use electrical load and other six features including hours, dry bulb temperature, wet bulb temperature, dew point temperature, humidity and electricity price as the input data. In this section, we have validated the validity of the univariate model using only historical power load data as a single variable serial data input into our TAT model. Same as aforementioned TAT of multivariate, using historical univariate data as input, we made predictions for 30 min, 1 h, 12 h, and 1 day ahead, respectively, and compared them with the multivariate-TAT model. As shown in Table 2, we can see that multivariable inputs produce better predictions than univariate inputs. Suggesting that the change of electric load is related to many factors, not only depends on its own features, but also is directly interfered by random factors. It is shown that more prior knowledge is beneficial to the improvement of our model's prediction accuracy because the multivariate variable input brings

Table 2 Comparison of performance for multi-step forecasting under the univariate and multivariate variable input

Method	30 min			1 h			12 h			1 day		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
univariate	103.06	136.99	1.04%	146.47	195.13	1.48%	200.00	286.55	2.14%	249.91	348.79	2.83%
multivariate	*47.20*	*62.49*	*0.465%*	*65.99*	*87.59*	*0.618%*	*147.55*	*215.18*	*1.69%*	*230.24*	*325.21*	*2.62%*

Italic values are the best results on traditional and deep neural methods

more dependent features to the model and self-attention mechanism have sufficient capacity to capture complex dynamical patterns in the multivariate variable data.

Besides, we ranked the importance of the multivariate features to explore how each variable feature contributes to the prediction results. By successively removing the variable input to the model, the decline of model's accuracy reflects the contribution of the variable to the prediction result, and the results are shown in Fig. 5 The dry bulb temperature has the greatest effect on the predicted results, and the electricity price has the weakest effect in all the variables.

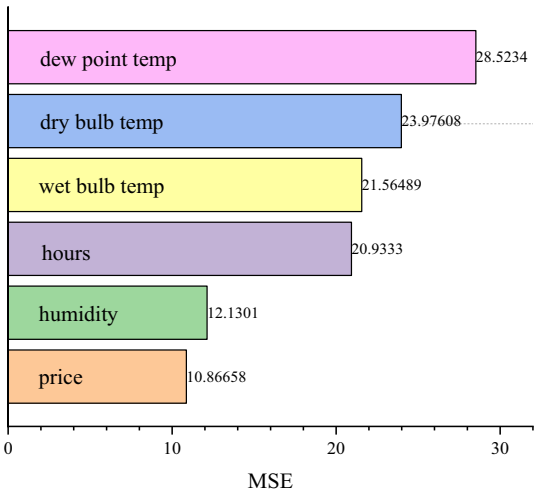

Fig. 5 The rank of the importance of the multivariate features. The value of the horizontal axis represents the decrease of the accuracy of the model after this variable was removed. The greater the decrease, the greater the contribution of which variable for the model prediction

4.3.3 Comparison of Different Input Length for an Hour Ahead Forecasting

During the experiment, we found that the input time step of historical data has great influence on the prediction results of the model, so we tried to use historical sequences of different lengths as input to predict the power load in the next hour, the comparison graph is shown in Fig. 6. Prediction errors of models except LSTM are decreasing with the length of input historical data, because longer historical data may contain more dependencies and provides more historical information for the model. But for LSTM, further increasing causes the RMSE to drop since it cannot effectively capture the dependency and regularity of the history records in the case of longer input sequence. Both Bi-LSTM and CNN-LSTM can improve this defect. In the prediction of 1 h (2 steps) ahead, our TAT model always performs the best regardless of the length of historical information as input. Our experiments show that the model occupies preferable prediction performance and practicability that can use less data to capture the load features and make more accurate predictions. Only 6 steps of historical data needed to achieve the same prediction effect as the basic Transformer with 48 steps input, which means less memory occupancy and faster calculation speed.

5 Conclusion

In this paper, we developed a short-term forecast model TAT for electrical load forecasting, which was tested in the data of electrical load in New South Wales. Compared with other six methods (ARIMA, LSTM, Bi-LSTM, Seq2Seq, CNN-LSTM and basic Transformer), our model has the best

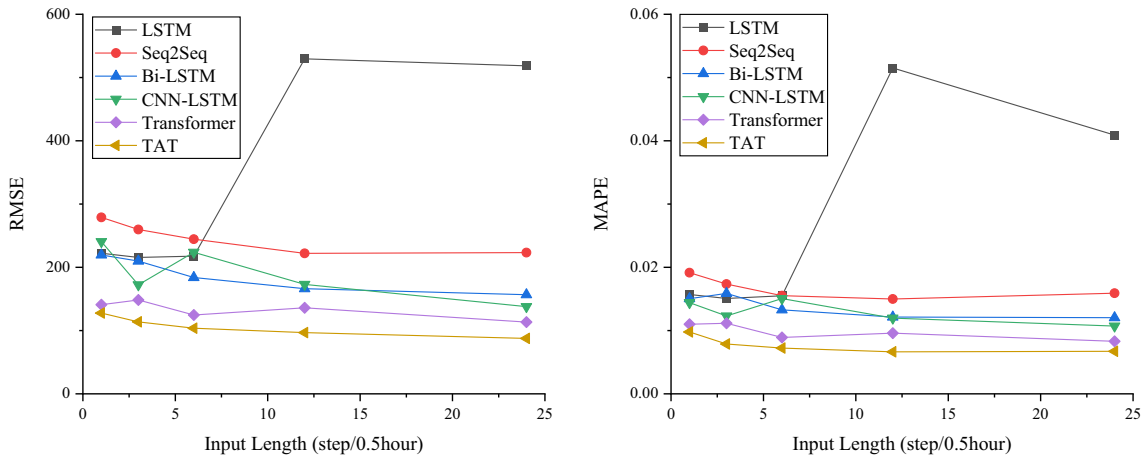

Fig. 6 The RMSE and MAPE of different input for an hour ahead forecast

forecast performance. Moreover, we compare the model with univariate variable input using only historical power load data as a single variable serial to the previous multivariate TAT, and multivariable inputs produce better predictions than univariate, suggesting that the multivariate input brings more dependent features to the model and our approach can better learn the dynamic dependencies in the complex input sequence. In addition, we compare the predictive ability of the model with different input steps, it was found that our approach can rely on less historical data to obtain better prediction results than other models. In summary, it can be concluded that our model is a satisfactory approach in terms of electrical load forecasting. Finally, although our approach has been very effective in short-term electrical load forecasting, with the increase of the prediction step, the prediction accuracy gradually decreases. In future work, we hope to further improve the performance of the model from the perspective of external factors of power load.

Acknowledgements The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the article.

Authors' Contributions CW identified this problem and designed the method. GZ performed the model building and writing the manuscript. CJ and YW performed the experiments. All authors read and approved the final manuscript.

Funding This study was supported by Scientific research project of Beijing Municipal Education Commission—General Project of science and technology plan (Grant Z20018) and Basic scientific research business fee project of municipal colleges and Universities—special subsidy for youth scientific research and innovation (Grantx 18258).

Data Availability The datasets used during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of Interest The authors declare no conflict of interest.

Ethics Approval and Consent to Participate Not applicable.

Consent for Publication The authors consent to this work for publication.

References

1. Reddy, M., Vishali, N.: Load forecasting using linear regression analysis in time series model for RGUKT, R.K. Valley campus HT feeder. Int J Eng Sci. **6**, 624–625 (2017). https://doi.org/10.17577/ijertv6is050443
2. Jahan, I.S., Snasel, V., Misak, S.: Intelligent systems for power load forecasting: a study review. Energies **13**, 6105 (2020). https://doi.org/10.3390/en13226105
3. Memarzadeh, G., Keynia, F.: Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm. Elector Pow Syst Res. **192**, 106995 (2021). https://doi.org/10.1016/j.epsr.2020.106995
4. Nti, I.K., Teimeh, M., Nyarko-Boateng, O., Adekoya, A.F.: Electricity load forecasting: a systematic review. J Electr Syst Inf Technol **7**, 1–19 (2020). https://doi.org/10.1186/s43067-020-00021-8
5. Nespoli, A., Ogliari, E., Pretto, S., Gavazzeni, M., Vigani, S., Paccanelli, F.: Electrical load forecast by means of LSTM: the impact of data quality. Forecasting **3**, 91–101 (2021). https://doi.org/10.3390/forecast3010006
6. Sun, G., Jiang, C., Wang, X., Yang, X.: Short-term building load forecast based on a data-mining feature selection and LSTM-RNN method. Ieej T Electr Electr **15**, 1002–1010 (2020). https://doi.org/10.1002/tee.23144
7. Malek, Y.N., Najib, M., Bakhouya, M., Essaaidi, M.: Multivariate deep learning approach for electric vehicle speed forecasting. Big Data Min Anal **4**, 56–64 (2021). https://doi.org/10.26599/bdma.2020.9020027
8. Mamun, A.A., Sohel, M., Mohammad, N., Sunny, M., Hossain, E.: A comprehensive review of the load forecasting techniques using single and hybrid predictive models. IEEE Access **8**, 134911–134939 (2020). https://doi.org/10.1109/ACCESS.2020.3010702
9. Chen, J., Wu, Y., Lin, Z., Zhao, L., Deng, X.: Review of Load Forecasting Based on Artificial Intelligence Models. 2021 6th Asia Conference on Power and Electrical Engineering 2021, 340–344 (2021). https://doi.org/10.1109/acpee51499.2021.9436916
10. Yang, A., Li, W., Yang, X.: Short-term electricity load forecasting based on feature selection and Least Squares Support Vector Machines. Knowl Based Syst **163**, 159–173 (2019). https://doi.org/10.1016/j.knosys.2018.08.027
11. Lu, J.C., Niu, D.X., Jia, Z.Y.: A study of short-term load forecasting based on ARIMA-ANN. Int Conf Mach Learn Cybernet **5**, 3183–3187 (2005). https://doi.org/10.1109/icmlc.2004.1378583
12. Zhou, D., Chen, S., Dong, S.: Network traffic prediction based on ARIMA model. arXiv preprint arXiv:1302.6324 (2013). https://doi.org/10.48550/arXiv.1302.6324
13. Yang, J.F., Cheng, H.Z.: Application of SVM to power system short-term load forecast. Electric Power Automat Equip **24**(2), 30–32 (2004)
14. Huo,J., Shi,T.T., Chang, J.: Comparison of Random Forest and SVM for Electrical Short-term Load Forecast with Different Data Sources. In: 2016 IEEE 7th International Conference on Software Engineering and Service Science. 2016, 1077–1080 (2016). https://doi.org/10.1109/ICSESS.2016.7883252
15. Peng, L.I., Shuai, H.E., Han, P., Zheng, M., Huang, M., Sun, J.: Short-term load forecasting of smart grid based on long-short-term memory recurrent neural networks in condition of real-time electricity price. Power Syst Technol **42**(12), 4045–4052 (2018). https://doi.org/10.13335/j.1000-3673.pst.2018.0433
16. Gong, G., An, X., Mahato, N.K., Sun, S., Wen, Y.: Research on Short-term load prediction based on Seq2seq model. Energies **12**, 3199 (2019). https://doi.org/10.3390/en12163199

17. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N.: Attention Is All You Need. arXiv preprint arXiv:1706.03762 (2017). https://arxiv.org/abs/1706.03762
18. Pooniwala, N., Sutar, R.: Forecasting Short-Term Electric Load with a Hybrid of ARIMA Model and LSTM Network. In: 2021 International Conference on Computer Communication and Informatics (ICCCI). 2021, 1–6 (2021). https://doi.org/10.1109/ICCCI50826.2021.9402461
19. Marrero, L., García-Santander, L., Carrizo, D., Ulloa, F.: An application of load forecasting based on ARIMA models and particle swarm optimization. In: 2019 11th International Symposium on Advanced Topics in Electrical Engineering (ATEE). 2019, 1–6 (2019). https://doi.org/10.1109/atee.2019.8724891
20. Wei, L., Zhang, Z.G.: Based on time sequence of ARIMA model in the application of short-term electricity load forecasting. Int Conf Res Challenge Comp Sci **2009**, 11–14 (2009). https://doi.org/10.1109/ICRCSS.2009.12
21. Yi, L., Niu, D., Ye, M., Hong, W.C.: Short-term load forecasting based on wavelet transform and least squares support vector machine optimized by improved cuckoo search. Energies **9**, 827 (2016). https://doi.org/10.3390/en9100827
22. Lahouar, A., Slama, J.: Random forests model for one day ahead load forecasting. Renew Energ Congress **2015**, 1–6 (2015). https://doi.org/10.1109/irec.2015.7110975
23. Zhang, N., Li, Z., Zou, X., Quiring, S.M.: Comparison of three short-term load forecast models in Southern California. Energy **189**, 116358 (2019). https://doi.org/10.1016/j.energy.2019.116358
24. Sun, Q.Y., Yang, L.X., Zhang, H.G.: Smart energy — Applications and prospects of artificial intelligence technology in power system. Kongzhi yu Juece/Control Decis **33**, 938–949 (2018). https://doi.org/10.13195/j.kzyjc.2017.1632
25. Dong, S., Wang, P., Abbas, K.: A survey on deep learning and its applications. Comput Sci Rev **40**(1), 100379 (2021). https://doi.org/10.1016/j.cosrev.2021.100379
26. Wang, H., Lei, Z., Zhang, X., Zhou, B., Peng, J.: A review of deep learning for renewable energy forecasting. Energ Convers Managf. **198**, 111799 (2019). https://doi.org/10.1016/j.enconman.2019.111799
27. Mamun, A., Sohel, M., Mohammad, N., Sunny, M.S.H., Dipta, D.R., Hossain, E.: A comprehensive review of the load forecasting techniques using single and hybrid predictive models. IEEE Access **8**, 134911–134939 (2020). https://doi.org/10.1109/ACCESS.2020.3010702
28. Tokgöz, A., Ünal, G.: A RNN based time series approach for forecasting turkish electricity load. 2018 26th Signal Processing and Communications Applications Conference (SIU). 2018, 1–4 (2018): IEEE https://doi.org/10.1109/siu.2018.8404313
29. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
30. Wang, R., Zhao, J.: Deep learning-based short-term load forecasting for transformers in distribution grid. Int J Comput Int Sys **14**, 1–10 (2021). https://doi.org/10.2991/ijcis.d.201027.001
31. Wu, N., Green, B., Xue, B., O'Banion, S.: Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case. arXiv preprint arXiv: 2001.08317 (2020). https://doi.org/10.48550/arXiv.2001.08317
32. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016, 770–780 (2016). https://doi.org/10.1109/CVPR.2016.90
33. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer Normalization. arXiv preprint arXiv:1607.06450. (2016). https://doi.org/10.48550/arXiv.1607.06450
34. Luong, M.T., Pham, H., Manning, C.D.: Effective Approaches to Attention-based Neural Machine Translation. Comput Sci. (2015). https://doi.org/10.18653/v1/D15-1166
35. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate. arXiv preprint arXiv:1409.0473 (2014). https://doi.org/10.48550/arXiv.1409.0473
36. Kingma, D., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014). https://doi.org/10.48550/arXiv.1412.6980