Electric Power Systems Research 238 (2025) 111091

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

EV load forecasting using a refined CNN-LSTM-AM

Juan Ran *, Yunbo Gong, Yu Hu, JiaLing Cai

School of Intelligent Computing Engineering, Tianjin Renai College, Tianjin, China

A R T I C L E I N F O

Keywords:
EV load forecasting
MIMO
CNN-LSTM-AM
Multi-step-ahead forecasting

A B S T R A C T

Electric vehicle (EV) load forecasting is becoming increasingly important for power system operation. Accurately multi-step-ahead forecasting EV loads is challenging. The correlation between the series at different time intervals and the key points in forecasting the time series will affect the results of EV load forecasting. Therefore, in this paper, a method is presented for the combination of time series of different length intervals into a hybrid CNN-LSTM-AM model for multi-step-ahead forecasting. The input matrix consists of combining time series of different lengths. A designed CNN network with a one-dimensional convolutional structure is used to extract features. After the convolutional layer, the temporal features remain. Finally, LSTM Encoder-Decoder and Attention Mechanism (AM) are combined to solve the problem of forgetting multi-step-ahead forecasting. Through the validation of the public ACN-data, it is demonstrated that the proposed method achieve accurate prediction results. According to error metrics, MAE, RMSE and R^2 outperform other models with a value of 0.5268, 0.9519 and 0.9138 respectively. The maximum number of multi-step-ahead prediction reaches 96 steps. This provides a reference for longer multi-step predictions in the future. It is also confirmed in the ACN-data that the accuracy of the hybrid model is better than the single model in EV load prediction.

1. Introduction

There are now a number of difficulties as a result of the rising demand for electric vehicle (EV) charging. First, the power grid is strained by the frequent use of public charging stations. Second, 75–80% of China's electricity is generated through coal combustion, and the carbon emissions from grid-charged EVs are still higher than those from conventional fuel vehicles [1]. For charging station managers to efficiently plan power consumption and resource allocation, accurate power load forecasting is therefore essential [2,3].

There are a large number of methods for EV load forecasting in the literature, such as mathematical modeling, statistical techniques, and AI methods [4,5]. Although these models are capable of performing well to some extent for EV load forecasting, they perform poorly for time series forecasting over short time intervals. However, the accuracy of the model is improved by using combined structure consisting of Convolutional Neural Networks (CNN), Attention Mechanism (AM), and Long Short-Term Memory (LSTM) Network [6].

Inspired by the above, this paper uses a hybrid model based on CNN, LSTM and AM for EV load forecasting. The model is based on a dynamic EV charging dataset, which is called ACN-data. ACN-Data was collected from two Adaptive Charging Networks located in California [7]. It has been published in 2019. To our knowledge, this is the only publicly available large-scale fine-grained charging data. For the EV load forecast, a single model has been used on the ACN-dataset [5,8,9]. The paper underlines the hybrid model based on CNN, LSTM and AM, which outperforms the single model for EV load forecasting on ACN-dataset.

On other hand, charging load forecasting is a subfield of time series forecasting and involves predicting future values based on historical data. There are two types of time series forecasting techniques: single-step forecasting [4] and multi-step forecasting [10]. Multistep forecasting can be beneficial for charging load forecasting since studies have demonstrated that it can predict future patterns of change over numerous time steps [4]. Accordingly, the focus of this paper is multi-step forecasting. Recursive, direct, and Multiple-Input Multiple-Output (MIMO) strategy are examples of multi-step prediction approaches that are frequently employed [10,11]. The Multiple-Input Multiple-Output (MIMO) strategy have been introduced and analyzed in the literature [12-14].The LSTM layer is favored for building time series prediction models due to its ability to efficiently capture temporal correlation in data [15]. To our knowledge, multi-step forecasting of time series using LSTM methods includes 5 steps, 6 steps, 10 steps, etc [16-18].

For multi-step forecasting, multivariate forecasting has become the mainstream in load forecasting due to the non-linear behavior of

* Corresponding author.
E-mail address: syan_cn4@163.com (J. Ran).

https://doi.org/10.1016/j.epsr.2024.111091
Received 7 February 2024; Received in revised form 21 August 2024; Accepted 15 September 2024
Available online 23 September 2024
0378-7796/

electrical load patterns [19]. However, its large model size may lead to substantial resource consumption in practical applications. In many cases, adding such variables affects the prediction accuracy slightly due to their small changes over short periods Moreover, additional variables can complicate and slow down the model training process [20]. This adds to the difficulty of choosing variables. Using univariate forecasting ignores the time interval of time series sampling and the accuracy of prediction are closely related [21]. For instance, series collected using 1 min and series collected using 15 min reflect different information. The Time-Aware LSTM (T-LSTM) model is introduced by modifying the internal architecture of the LSTM to handle sequences with irregular intervals [22]. Therefore, the prediction accuracy is significantly influenced by the interval of the input series.

In this work, a hybrid load forecasting model called CNN-LSTM-AM is proposed on the ACN-data. To make multi-step forecasts, including 4-step, 12-step, 24-step, 48-step and 96-step forecasts, the method combines two time series with different intervals. In this model, two sequences with different intervals are first combined by matrix transformation as an input matrix. The aim is to combine information with different temporal granularity. Previous work use look-back load to extract features by CNN while changing the shape of the original timestamp. We design a one-dimensional convolutional structure that remaps the features at each time step, preserving information from previous time points. The features of each time step and the current time step are fed into the LSTM-Encoder-Decoder architecture and the Attention Mechanism is used to solve the problem with forgetting in multi-step prediction.

The main contributions of paper are as follows:

- A multi-step forecasting method combining long and short interval time series data is proposed. This method effectively overcomes the limitation of exclusive reliance on a single predictive attribute from a single time series.
- A hybrid CNN-LSTM-AM load forecasting model is proposed to perform multi-step-ahead forecasting on the public ACN-dataset. This representation outperforms the single model for EV load forecasting on ACN-dataset.
- A one-dimensional convolutional structure is designed. It remaps the features at each time step, preserving information from previous time points.
- For multi-step forecasting, our method performs short-term electricity load forecasting up to 96 steps in the ACN dataset. It provides a reference for performing ultra-long-term forecasting in EV load forecasting in the future.

The paper is organized as follows: Section 2 presents an overview of the relevant literature. In Section 3, the architecture of the proposed model is presented and its details are explained. In Section 4, the CNN-LSTM-AM model is employed on the ACN-data and illustrate its effectiveness by comparison. Section 5 gives the conclusions the work.

2. Related work

The forecasting of EV charging load is a complex task that involves many factors. Currently, two main approaches are used for load forecasting in the charging sector: statistical-based techniques and artificial intelligence-based techniques. Among them, statistical methods include Monte Carlo method [23,24], Autoregressive (AR) model, Autoregressive Moving Average (ARMA) model [25,26], Autoregressive Integrated Moving Average (ARIMA) model [27-29] and Gaussian process [30]. However, statistical models have limits in handling raw data and maintaining stability in time series analysis. Thus, statistical methods forecast non-linear data less well.

In recent years, many scholars have used machine learning to estimate power load. A support vector regression (SVR) charging load prediction model based on past load data is proposed in Ref. [31]. The authors utilize Artificial Neural Network (ANN) through back propagation in combination with Genetic Algorithm model is used aimed at short term load forecasting in Ref. [32]. Ref. [9] considers the use of Random Forest and XGBoost machine learning algorithms for forecasting electrical loads on ACN datasets. The stochasticity of electric car charging duration and state of charge has been addressed to improve forecast accuracy. However, machine learning methods for load prediction in complex nonlinear datasets fail to attain reasonable accuracy.

The emergence of deep learning models is well suited to solving nonlinear relationships in time series and has been well-received by many researchers in electricity load forecasting. The authors propose a nonlinear relationship extraction (NRE) method and design a simple CNN to solve nonlinear relationships for residential load forecasting in Ref. [33]. Ref. [5,34] applies three learning frameworks for the prediction of the EV load on the ACN-data, which include the AR model, the SVR and the LSTM. The results show the superiority of LSTM in forecasting on the ACN-data. However, the LSTM model fails to consider that input vectors at different time steps may contribute differently to the prediction when it treats the input equally [35]. Hybrid models, which integrate the strengths of different models, have received increasing attention compared to single model architectures [6,8,36,37]. Hybrid models comprise various architectures, such as the dual-stage attention architecture combined with Long Short-Term Memory networks (CDA-LSTM) [36], wavelet packet decomposition combined with Long Short-Term Memory networks (WPD-LSTM) [37], and convolutional neural network combined with Long Short-Term Memory (CNN-LSTM) [6].

Multivariate forecasts commonly involve the consideration of multiple elements, including climatic conditions, temporal features, and interdependencies among variables. However, the presence of abundant noise in the data presents difficulties in computational complexity and factor analysis [36], as well as the problems described in Section 3. In [38], Industrial load forecasting was performed using a Quantile Regression Forests (QRFs) model in which exogenous electrical predictor variables were added to capture the physical dependence between electricity quantities. The findings, however, indicate that the inclusion of electrical predictor variables does not produce adequate outcomes. Accordingly, this paper focuses on extracting additional data information using univariate analysis to enhance EV load forecasting.

Univariate forecasting lowers noise and generates accurate forecasts utilizing efficient computation. Many univariate time series prediction studies have focused on dissecting the original series into more complicated components to increase forecast accuracy. In [39], a decomposition strategy is used to generate a sequence of derived variables from the target sequence, and a dual attention mechanism based on the encoder-decoder framework is introduced to extract the instantaneous fluctuation features and time-varying features of the input variables. Li et al. [37] developed a hybrid DL model using wavelet packet decomposition and LSTM networks. This model was created to forecast solar power one hour ahead. However, decomposing or recreating initial sequence data loses important information, making spatio-temporal correlation analysis difficult.

Comment:

Inspired by the above work, this study initially aims to utilize a hybrid model CNN-LSTM-AM on the publicly available ACN-data, demonstrating that the accuracy is better than a single model in EV load forecasting. Secondly, we designed a combination of time series with different length intervals for multi-step prediction to address the univariate multi-step prediction issue. Thirdly, we design a one-dimensional convolutional CNN structure to retain information previous time point. Finally, we aimed to predict the maximum number of multi-step prediction steps possible using our method on the ACN-data.

3. Methodology

In this section, we first formulate the time series forecasting problem, and then discuss the details of the CNN-LSTM-AE model in the following part.

3.1. Problem formulation

In previous studies, given the input sequence x_1, x_2, …, x_T, the sequence y_1, y_2, …, y_n after the moment T is predicted as Eq.(1).

$$[\hat{y}_1, \hat{y}_2 \dots \hat{y}_n] = f(x_1, x_2 \dots x_T) \tag{1}$$

y_1, y_2, …, y_n denote the values of n time points in the predicted future, x_1, x_2, …, x_T denote the time series sampled at fixed time intervals at moment T, and f denotes the network model that can be used for prediction.

To forecast n time points by incorporating long and short time intervals, the concatenated input time series *Xt* is derived from the long interval time series $\{xl_1, xl_2, \dots, xl_T\}$ and the short interval time series $\{xs_1, xs_2, \dots, xs_T\}$. Therefore, multi-step prediction via MIMO strategy can be expressed as Eqs. (2) and (3).

$$X_t = concat(xl, xs) \tag{2}$$

$$[\hat{y}_1, \hat{y}_2 \dots \hat{y}_n] = \hat{f}(X_1, X_2 \dots X_T) \tag{3}$$

The *concat* operation represents connecting matrices to form an input matrix [34]. Where $[\hat{y}_1, \hat{y}_2 \dots \hat{y}_n]$ are the predicted EV charging load values for m steps, and $\hat{f}$ is a vector-valued function. The forecast values are returned in one computation step using the MIMO strategy.

3.2. CNN-LSTM-AM

The framework for EV charging load forecasting using CNN-LSTM-AM is shown in Fig.1. The framework consists of several steps.

3.2.1. Input matrix

Previous studies have demonstrated the correlation between electrical load and time [40]. However, the aspect of time intervals and their durations is often neglected. Therefore, we propose a new method of combining different interval sequences to reconstruct the time series matrix. As shown in Fig. 2. The input long-interval time series and the short-interval time series are concatenated based on Eqs. (4) and (5).

$$X_t = [concat(xl_1, xs_{1:k}), concat(xl_2, xs_{k+1:2*k}) \dots concat(xl_T, xs_{T*k+1:T*k})]^T \tag{4}$$

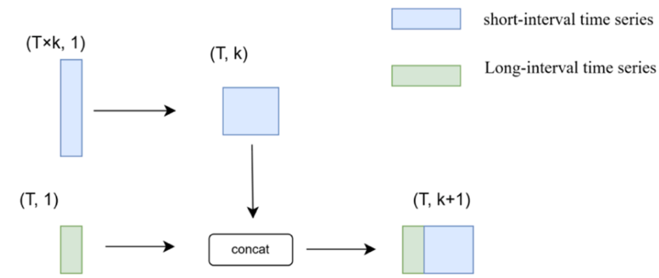

Fig. 2. Input matrix.

$$xl_i = \sum_{(i-1)*k+1}^{i*k} xs_i \tag{5}$$

where *xs* denotes a short-interval time series and *xl* denotes a long-interval time series. $xs_{(i-1)*k+1} \sim xs_{t*k}$ can be understood as the characteristics of *xl* at moment *i*. First, *xl* is transformed to $T * k$, then the *xl* and *xs* genearte a matrix of dimension $T * (k+1)$. The final matrix includes $xs_{(i-1)*k+1} \sim xs_{t*k}$ at T time. This solves the problem of weak performance of single series input and missing time series features. It should be noted that the interval of the two sequences needs to satisfy the integer multiple relationship k when sampling.

3.2.2. Convolutional neural networks

This paper design one-dimensional convolutional structure for extracting interval variation features using CNN networks. The CNN layer comprises three convolutional layers and fully layers, as shown Fig.3.

The RELU activation function and Dropout layer are added to each convolutional layer. Section 3.2.1 produces an input matrix $T * (k+1)$. The matrix is input into the CNN network for convolution. We take short-interval series as the feature variable of *xl*. The input matrix is first input into the first convolutional layer to extract features. After the first convolutional layer, a vector of $C_1 * F_1$ is generated. C_1 denotes the number of channels, and F_1 calculated according to Eq. (6).

$$F_n = num * \left(\frac{r + 2 * \text{padding} - dilation * (\text{kernel} - 1) - 1}{stride} + 1 \right) \tag{6}$$

where F_n is the output dimension, num is the number of convolutional layers, r denotes the length ratio of compact and sparse sequences, padding denotes the padding size of the matrix, kernel denotes the one-dimensional convolution size, dilation refers to the spacing between

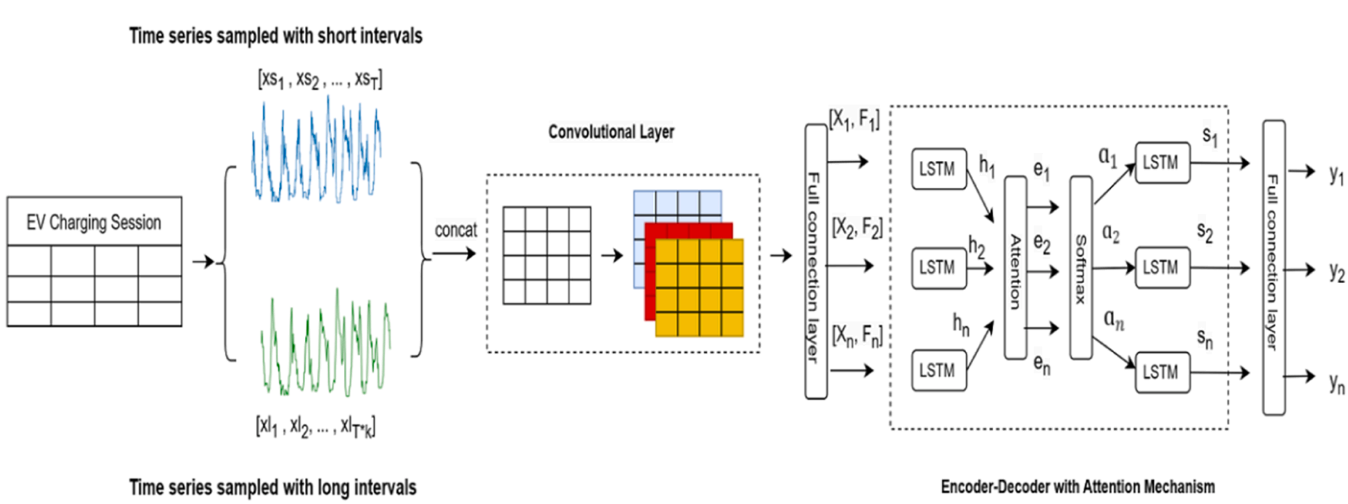

Fig. 1. The overall framework for EV Load Forecasting applying CNN-LSTM-AM.

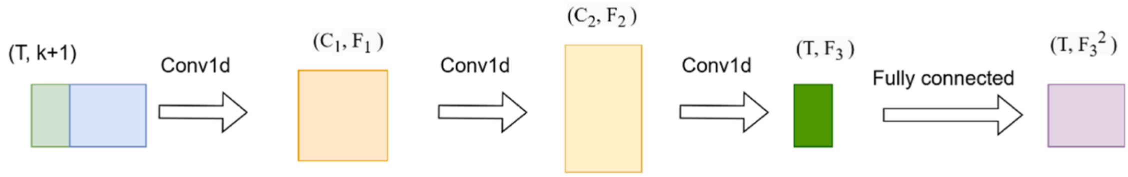

Fig. 3. Convolution layer.

kernel elements, and stride denotes the step size of the convolutional sliding. In this way, the features in the time dimension are expanded. The second convolutional layer still extract in the same way. At the last convolutional layer, the number of channels is mapped to T to keep previous timestamp information. In the fully connection layer, we use the square formula to map F_3 to higher dimensions for more information.

3.2.3. Encoder and decoder

The feature vectors are then dimensionally transformed by a fully connected layer, where $X_1 \sim X_n$ denote the load values of each time step and $F_1 \sim F_n$ denote the feature vectors of each time step. The vectors for each time step are fed into the encoder-decoder architecture as shown in Fig. 4. Finally, the predicted values are output through a fully connected layer.

Encoder: Given an input sequence $X = \{(X_1, F_1), (X_2, F_2) \ldots, (X_n, F_n)\}$. LSTM is applied as an encoder to load input samples sequentially and calculate the corresponding hidden activation $h_t = \{h_1, h_2, \ldots, h_n\}$.

AM: When loading is done, the attention layer generates a context vector c_t for the input of the decoder. The attention layer makes the model focus more on important information and can effectively improve the prediction accuracy of long sequences.

Decoder: The decoder is another LSTM designed to output sequence $Y = \{y_1, y_2, \ \ldots, y_n\}$. It predicts the next step output y_n based on the hidden state s_n.

4. Experiments and discussions

The goal of this work is to utilize historical electric load data to forecast the electric load data before the 4-step, 8-step, 12-step, 24-step and 96-step ahead forecasting, respectively. In this section, we provide an experimental assessment of the performance of our proposed methodology utilizing ACN-data.

4.1. Datasets

This work uses an open-source ACN-data [7]. This consist of data from two EV charging points: Caltech and JPL. The Caltech site is public use charging, while JPL site is representative of workplace charging. The dataset for Caltech's Adaptive Charging Networks (ACN) was collected from 55 charging stations in a parking garage on the Caltech campus. The dataset for JPL's ACN was collected from 52 Electric Vehicle Supply Equipment (EVSE) in a parking garage. The dataset contains over 30,000 charging sessions and is updated daily. A total of 15 features are available in the dataset, including disconnection, connection, charging completeness, kWh provided, etc.

As a result, the ACN-data provides a representative sample of charging usage in a mixed environment, catering to the charging requirements of both workplaces and public locations. A log of 28,397 charging session from 25 April 2018 to 31 January 2021 are used in this work. Fig. 5 shows the power load time series data, with observations recorded at 1-minute intervals for Caltech and JPL. It can be observed that the electric vehicle (EV) load time series from the two datasets are very different. Even within the same season, the power values vary dramatically. This is one of the reasons why these two datasets were chosen to validate the effectiveness and generalization capability of the proposed method.

4.1.1. Data preparation

The work intends to study the evolving trends and specific values of power load in the future period. For this purpose, it is necessary to generate two time series: the long interval time series and the short interval time series. Two datasets were collected from April 2018 to January 2021, with observations recorded at 1-minute intervals and at 15-minute intervals. The Caltech dataset contains a total of 1,018,080 and 67,872 samples, respectively. Similarly, the JPL dataset contains a total of 832,320 and 55,488 samples, respectively.

As shown Table 1, three subsets of data are used in the evaluation: The model is trained using data from April 2018 and August 2019, which includes 758,784 samples. Validation is conducted using data from September 2019 and December 2019, comprising 187,392 samples. Data for January 2020 and April 2020 are allocated for testing the model's performance using 139,776 samples. The distribution of data is plotted, as shown in Fig 6.

4.1.2. Data standardization

In order to assure accuracy and robustness, we use the normalization procedure to every dimension of the data in this study to address this problem. The specific formula is as in (7).

$$x' = \frac{x - x_\mu}{x_\sigma} \tag{7}$$

Where x denotes the original sample, x_μ denotes the sample mean, x_σ denotes the sample variance, and x' denotes the normalized sample.

4.2. Experimental setting

The proposed model is implemented and executed on a PC with an Intel i7–13700H CPU, NVIDIA GeForce GTX 3060 Laptop GPU, and 16GB of RAM. The programming environment is based on PyTorch 1.10.2. Adam optimizer and the Mean Square Error (MSE) loss function are used to train the CNN-LSTM-AM model. The process of grid search is used to determine the hyper-parameters, as shown in Table 2.

By comparing five different deep learning models, the performance of the CNN-LSTM-AM model is verified. Table 3 lists the various deep learning model parameters and their selected values. Each hyper-parameter value was adjusted to the best performance for its model.

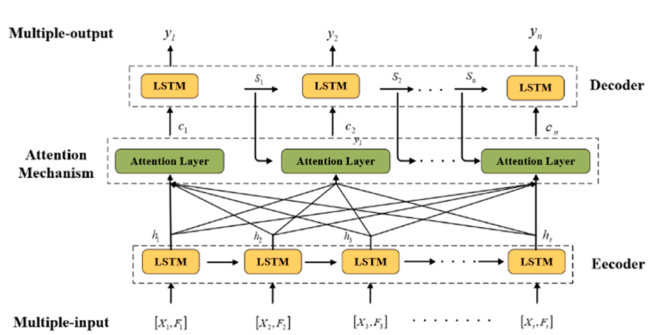

Fig. 4. Attention-based encoder-decoder LSTM networks.

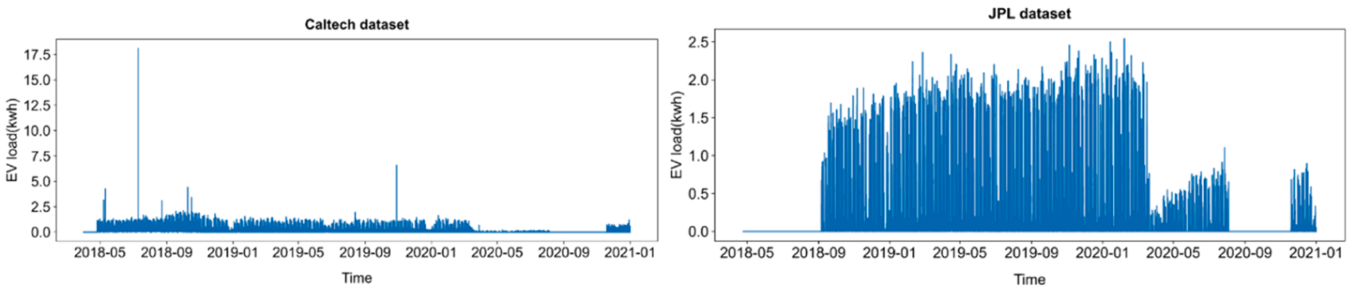

Fig. 5. Caltech (a) and JPL (b) dataset.

Table 1
Dataset division.

Dataset	Train set	Val set	Test set
Caltech	2018.4.25~2019.8.31	2019.9.1~2019.12.31	2020.1.1~2020.3.31
JPL	2018.9.1~2019.8.31	2019.9.1~2019.12.31	2020.1.1~2020.3.31

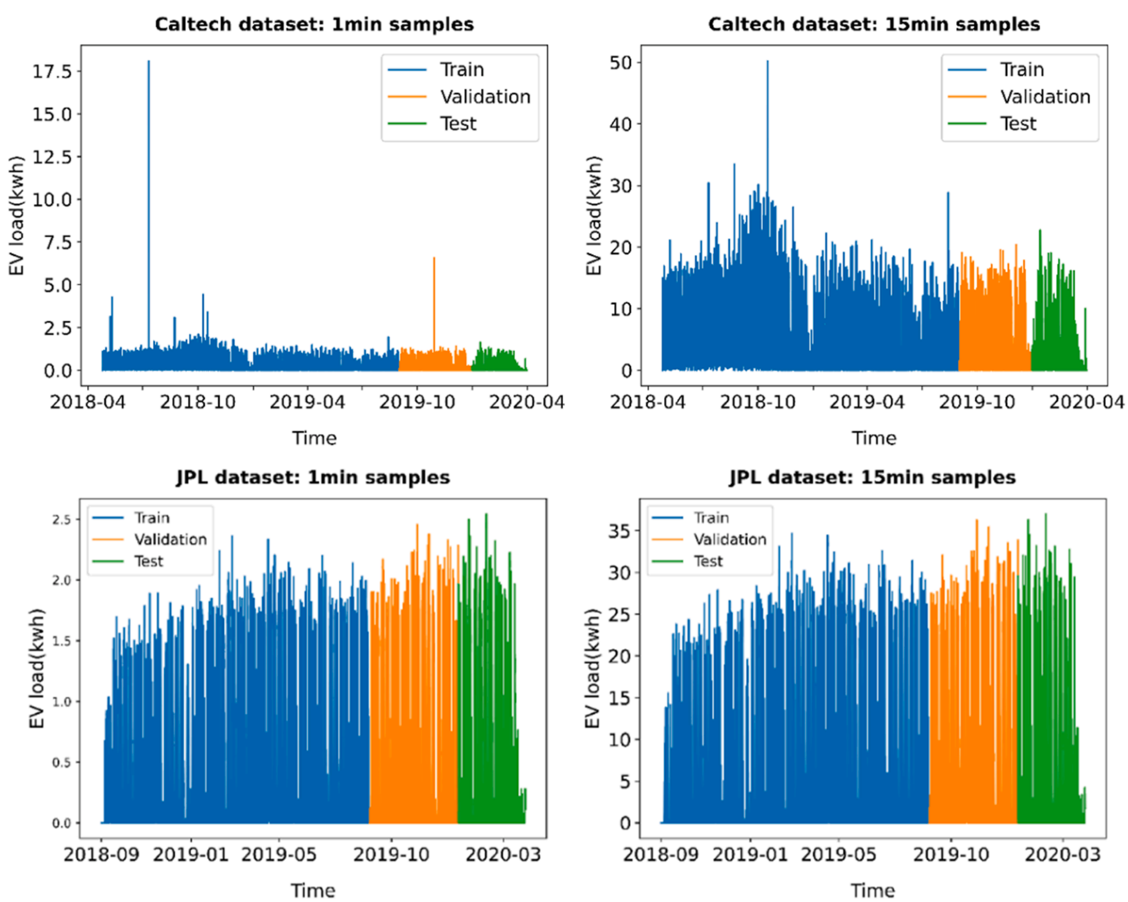

Fig. 6. Dataset divided for training, validation and testing.

4.3. Evaluation metrics

we use the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE) and the coefficient of determination (R^2) to evaluate the performance of the proposed model.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{8}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{9}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}\left(y_i - y_\mu\right)^2} \tag{10}$$

where y_i is the real electrical load value at time t and $\hat{y}_i$ is the forecasted electrical load value at time t. y_μ denotes the average value of the

Table 2
Hyper-parameters of the proposed model.

Hyper-Parameters	Value
Input Sequence Length	96(1 day)
Large interval (Large time series sampling interval)	15min
Small interval (Small time series sampling interval)	1min
Channal of first Conv1d Layer	128
Kenerl size of first Conv1d Layer	(1,3)
Channal of second Conv1d Layer	256
Kenerl size of second Conv1d Layer	(1,3)
Channal of third Conv1d Layer	144
Kenerl size of third Conv1d Layer	(1,3)
Number of units in the encoder layer	64
Number of units in the decoder layer	64
Active function	RELU
Optimizer	adam
Learning rate	0.001
Dropout rate	0.1

Table 3
Hyper-parameters settings for DP models.

#	model	description
1	ANN	number of hidden units=100
2	TCN [41]	number of hidden units=30, kernel size=7, levels=8
3	LSTM	number of hidden units=100, num_layers=3
4	LSTNet [42]	HidSkip=4, CNN hidden units=100, RNN hidden units=100
5	Transformer [43]	d_model=256, nhead=8, num_layers=3

observed output over existing samples.

4.4. Experimental results

4.4.1. Performance with the CNN-LSTM-AM model

The time series with 1-minute and 15-minute intervals were obtained by sampling the dataset of Section 4.1.1, then two series were used to generate the input matrix using the method of Section 3.2.1. Matrix is passed through a 1-dimensional convolutional layer to extract the spatial characteristics of the 1-minute-interval series. The feature matrix generated by the convolution is decomposed into multiple time steps as the multi-input to the LSTM encoder. The attention mechanism computes the attention mechanism score for the hidden vectors at each time step, which is then decoded by the LSTM decoder. Finally, the multi-step load predictions are generated by dense layer. Fig. 7 exhibits the power load forecasting curves in different step lengths. As can be seen, the blue curve in Fig. 7 represents the actual load and the red curve is the prediction of our model on the test set predicted by different step lengths. The 4 steps, 12 steps, 24 steps, 48 steps, and 96 steps, represent 1 h, 3 h, 6 h, 12 h, and 24 h ahead forecasting, respectively. Our method accurately predicts the electrical loads at the Caltech charging station and predicts the local variations at the peaks equally well in time for that test set. The prediction curve deviates from the true value as the prediction step increases, because of multi-step prediction, but our method still predicts the trend of electrical loads across the test set. The difference between the predicted value and the true value for 1 h ahead forecasting is very small, with a RMSE of 0.9519, a MAE of 0.5268, and an R^2 of 0.9138; and with a RMSE of 2.1325, a MAE of 1.3898, and an R^2 of 0.5871 for 24 h ahead forecasting.

Table 4 compares the single and multiple sequences at 10 min intervals and 20 min intervals. The combinations of 1 min, 5 min, 10 min, and 20 min were denoted as "1–10 min," "5–10 min," "1–20 min," and "5–20 min," respectively. As shown, the combined sequence of 1 min and 10 min has a smaller error from the true value than the 10-minute single

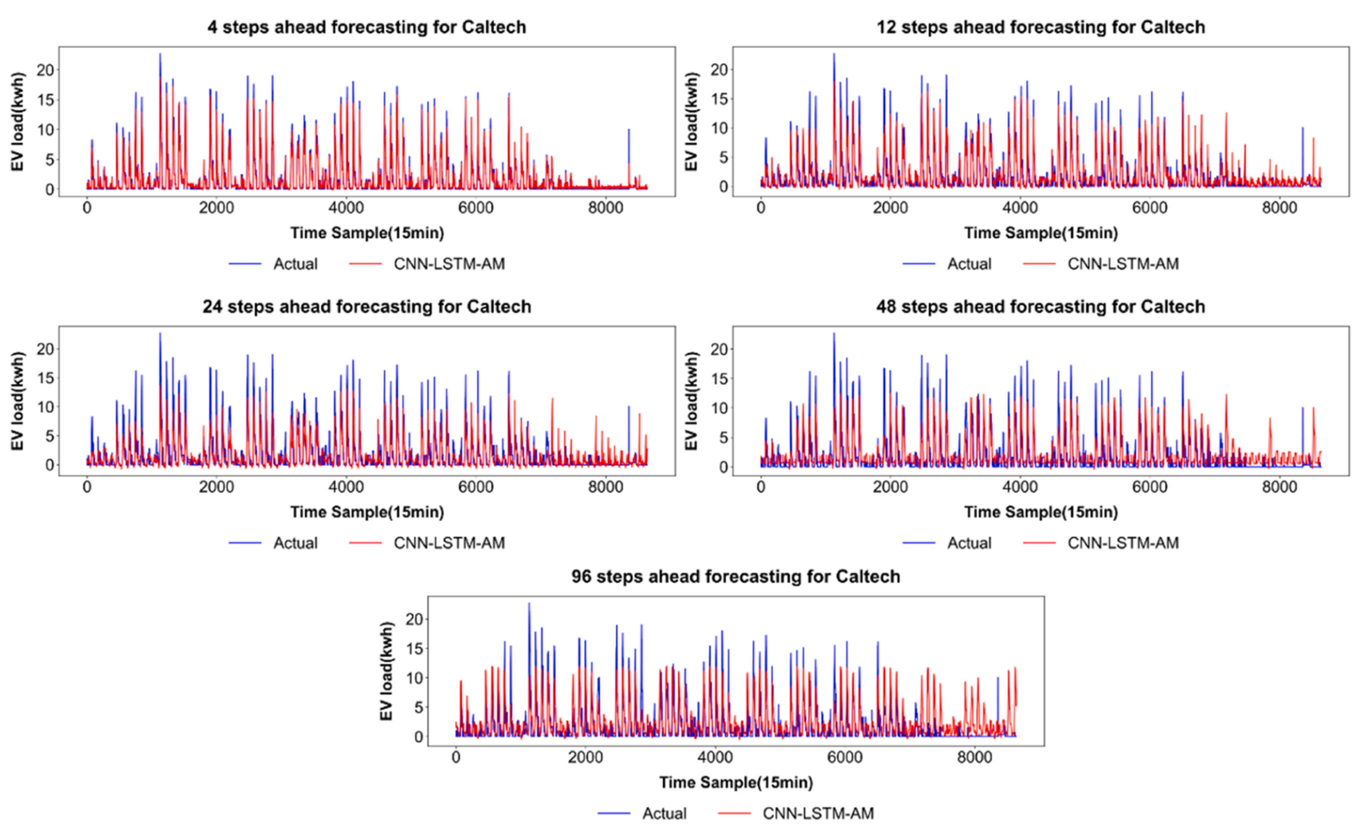

Fig. 7. 4-,12-,24-,48- and 96-step-ahead forecasting of the CNN-LSTM-AM model in the Caltech dataset.

Table 4
Compares the single and multiple sequences at 10 min intervals and 20 min intervals.

Step	Metric	10Min	1–10Min	5–10Min	20Min	1–20Min	5–20Min
4	RMSE	0.5806	**0.5756**	0.5789	1.6854	1.4269	1.4251
	MAE	**0.3026**	0.3284	0.3380	0.9107	0.8546	0.7909
	R^2	0.9286	**0.9294**	0.9288	0.8477	0.8908	0.8910
12	RMSE	1.1363	**0.9343**	1.0249	3.0284	2.3476	2.3723
	MAE	0.5975	**0.5631**	0.5470	1.7516	1.4268	1.4561
	R^2	0.7272	**0.8153**	0.7765	0.5120	0.7093	0.7056
24	RMSE	1.7929	**1.2771**	1.2992	3.2765	2.4075	2.5830
	MAE	1.0496	**0.7530**	0.7829	2.1357	1.5927	1.7431
	R^2	0.3197	**0.6570**	0.6506	0.4369	0.6933	0.6528
48	RMSE	1.8826	**1.3357**	1.4087	3.6093	2.7378	2.7627
	MAE	1.3132	**0.8756**	0.9403	2.5471	1.8137	1.8590
	R^2	0.2749	**0.6175**	0.5914	0.3399	0.6047	0.6078
96	RMSE	1.9140	**1.4411**	1.4472	4.0150	2.9506	3.3374
	MAE	1.3427	**0.9309**	0.9710	3.0523	1.9560	2.2585
	R^2	0.2365	**0.5663**	0.5768	0.2276	0.5457	0.4210

sequence. In the advance 4-step forecasting, the predictions of 1–10 min sequence achieved a RMSE of 0.5736, a MAE of 0.3284 a R^2 of 0.9294. In multi-step-ahead forecasting, our method potentially results in an improvement of up to 28 %. From 24 to 96 steps ahead forecasting, the performance of the combined sequences of 1 min and 20 min has also significantly improved.

Table 6 compares multivariate and univariate sequences for multi-step forecasting. As far as possible, we selected factors related to electricity load as covariates from the dataset for electricity load forecasting, as shown in Table 5. The study results indicate that the univariate time series multistep prediction method proposed in this paper outperforms the multivariate time series when using a 10-minute time interval in Table 6. This illustrates the impact that the inclusion of these variables can have on the prediction results.

4.4.2. Comparison and analysis of model prediction results

The effectiveness of the proposed CNN-LSTM-AM model is compared with different model in forecasting the performance at the 4-step, 12-step, 24-step, 48-step, and 96-step, as shown Table 7 and Figs. 8, 9. The analysis of the results is presented below.

In this experiment, the data from 10 February 2020 to 17 February 2020 are adopted and 672 samples are considered. As shown in Fig. 8, the load prediction curves of the proposed model are closer to the actual power data curve than the other models. From Fig. 8(a), it is clear that all models have a better degree of fit in 4-step ahead forecast. Fig. 8(b) and 8(c) demonstrate that the other models gradually deviate from the actual curves in 12-step,24-step ahead forecasts, whereas the CNN-LSTM-AM model maintains a better degree of fit. In particular, the Transformer, LSTNet, and TCN models begin to overfit when the actual curve reaches its lowest point. Fig. 8(d) and 8(e) demonstrate that the other models gradually deviate from the actual curves after the peaking in 48-step,96-step ahead forecasts. The comparison reveals that the CNN-LSTM-AM has the best effect on predictions than other models.

In Table 7, the CNN-LSTM-AM obtains the smallest error indices than other models. The R^2 of the CNN-LSTM-AM for 4-step,12-step, 24-step,48-step , and 96-step ahead predictions are 0.9318, 0.8013, 0.7043, 0.6303, and 0.5871, respectively. The MAE values are 0.5268, 0.8665, 1.1230, 1.3253,and 1.3898, respectively. The RMSE values are 0.9519, 1.4529, 1.7991, 1.9896, and 2.1325, respectively. Compared to other models, the CNN-LSTM-AM significantly improved the prediction accuracy. The CNN-LSTM-AM model obtains the smallest error indices than other models. As the time step increases, our proposed method performs better with better robustness. In the 96-step ahead prediction, the RMSE value has decreased by 15.2 % by our method compared with Transformer, the MAE value has decreased by 3.1 % and R^2 value has increased by 16.8 %.

The bar charts in Fig. 9 shows RMSE, MAE and R^2 values based on different models. As can be seen, the evaluation metrics are very sensitive to the prediction step size. The difference in evaluation metrics gradually increases as the prediction step increases. Compared with other models, our model got the smallest metrics of RMSE and MAE, and the R^2 metric is the largest.

Table 5
Field Selection.

Field	Description
sessionID	Unique identifier for the session.
siteID	Unique identifier for the site.
spaceID	Unique identifier of the parking space.
stationID	Unique identifier of the EVSE.
timezone	Timezone of the site.

Table 6
Comparison input sequence.

Step	Metric	Multivariate input sequence	Our input sequence
4	RMSE	0.5712	0.5756
	MAE	0.3354	0.3284
	R^2	0.9135	0.9294
12	RMSE	0.9546	0.9343
	MAE	0.5579	0.5631
	R^2	0.7926	0.8153
24	RMSE	1.3637	1.2771
	MAE	0.8735	0.7530
	R^2	0.5824	0.6570
48	RMSE	1.5636	1.3557
	MAE	1.0963	0.8756
	R^2	0.4358	0.6175
96	RMSE	1.6219	1.4411
	MAE	1.2369	0.9309
	R^2	0.3981	0.5663

4.4.3. Generalization ability analysis

To verify the generalizability of this method, the CNN-LSTM-AM model was demonstrated using JPL's ACN-data. The segmentation of the dataset, as referenced in Section 4.1.1, is divided into three parts: the training data consists of 56,064 samples, the testing data comprises 139,776 samples, and the verification data includes 187,392 samples. Fig. 10 shows the actual electric load values alongside the predicted results. As can be seen, the proposed model maintains a better degree of fit in the 4-step, 12-step, 24-step, 48-step, and 96-step. In addition, this section is a comparison of the CNN-LSTM-AM with other models. To illustrate the different performance of ANN, LSTM, TCN, Transformer, LSTNet and CNN-LSTM-AM, Fig. 11 shows the corresponding load prediction curves. As can be been, the CNN-LSTM-AM model achieves a better of fit than other models, capturing changes in the true value

Table 7
Multi-step-ahead Load results on Caltech.

Step	Metric	ANN	LSTM	TCN	LSTNet	Transformer	Ours model
4	RMSE	1.0757	1.0720	1.1060	1.1026	0.9670	**0.9519**
	MAE	0.6371	0.5973	0.5621	0.6109	0.5274	**0.5268**
	R^2	0.8911	0.8918	0.8850	0.8855	0.9120	**0.9138**
12	RMSE	1.8956	1.7425	1.8577	1.8324	1.6270	**1.4529**
	MAE	1.1583	1.0227	0.9953	1.0573	0.9651	**0.8665**
	R^2	0.6966	0.7147	0.6776	0.6843	0.7508	**0.8013**
24	RMSE	2.2976	2.2188	2.2103	2.1805	2.0213	**1.7991**
	MAE	1.5217	1.2824	1.3030	1.3304	1.3361	**1.1230**
	R^2	0.5460	0.5725	0.5454	0.5531	0.6193	**0.7043**
48	RMSE	2.5995	2.5299	2.5145	2.5395	2.4263	**1.9896**
	MAE	1.7442	1.5356	1.5018	1.4671	1.3964	**1.3253**
	R^2	0.4042	0.4064	0.4087	0.3931	0.4469	**0.6303**
96	RMSE	2.7100	2.5815	2.5519	2.5628	2.5145	**2.1325**
	MAE	1.8166	1.6253	1.5211	1.5208	1.4341	**1.3898**
	R^2	0.3389	0.3842	0.3867	0.3827	0.4301	**0.5871**

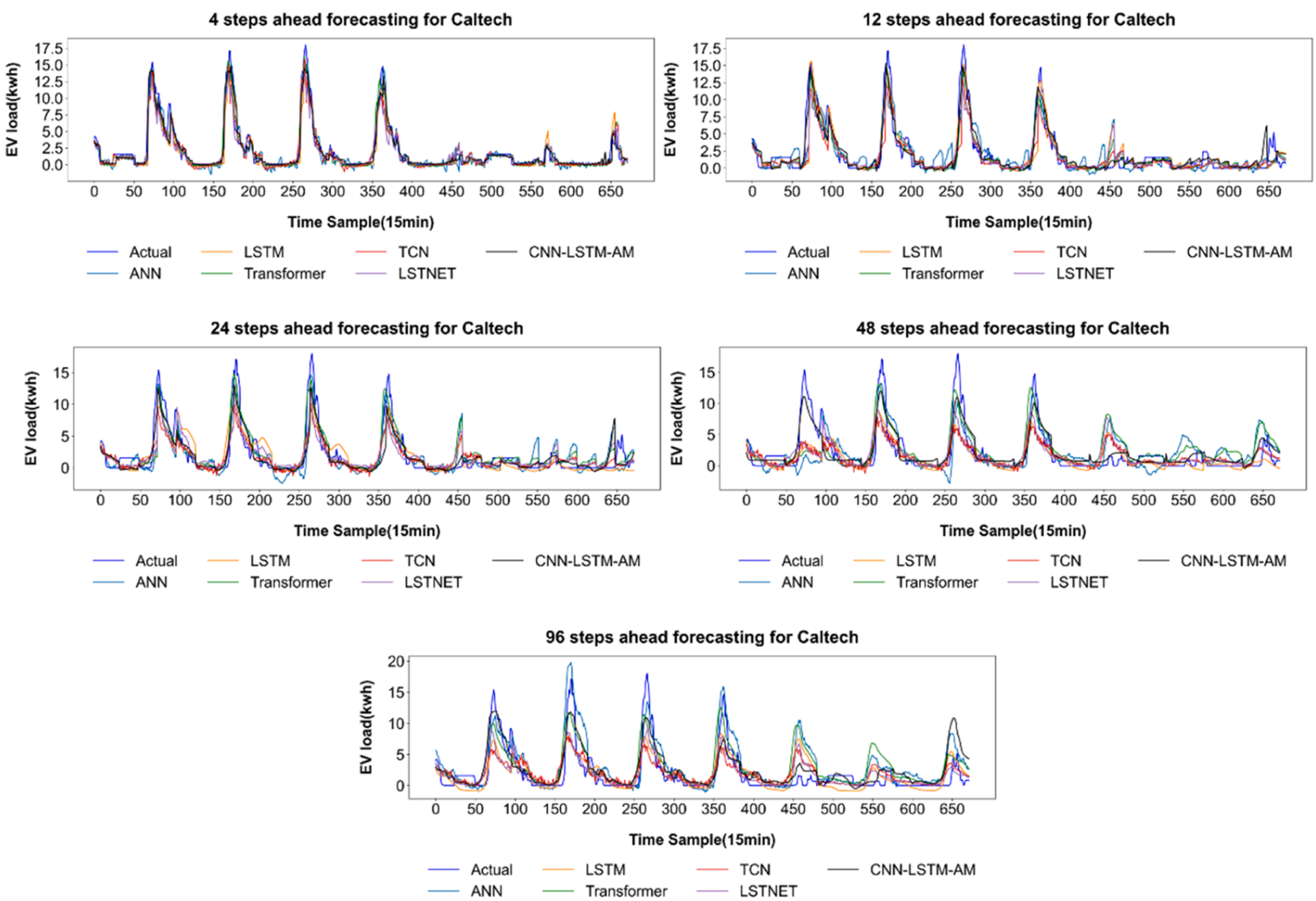

Fig. 8. Load forecasting results of different models for Caltech dataset (a) 4-step ahead forecast (b) 12-step ahead forecast (c) 24-step ahead forecast (d) 48-step ahead forecast (e) 96-step ahead forecast.

during peaks and troughs.

According to the error indices, the CNN-LSTM-AM model is better at predicting than the other models in Table 8. The CNN- LSTM-AM model improves the three metrics by 5.3 %, 9.2 %, and 0.4 % in 12-step ahead prediction compared with LSTM, and by 18.1 %, 31.8 %, and 18.6 % in 96-step ahead prediction, respectively. Therefore, the proposed model has a better forecasting accuracy.

To show experimental improvement and effectiveness, furthermore, we compare proposed approach in this work with other methods based on the same dataset division in Table 9. On a test set of 288 observations, compared with XGBoost [9], our method has a smaller RMSE of 4.9. In the method of [5], some load-related multivariable are added, while we use multiple sequences as the input matrix. However, the results show that our RMSE and MAE metrics are better, with a value of 3.6 and a value of 2.6 respectively. Fig. 12 shows the comparison between the prediction results and the true values for 744 observations in the test set. The results show that our prediction curve is very close to the true value, which proves the effectiveness of CNN-LSTM-AM.

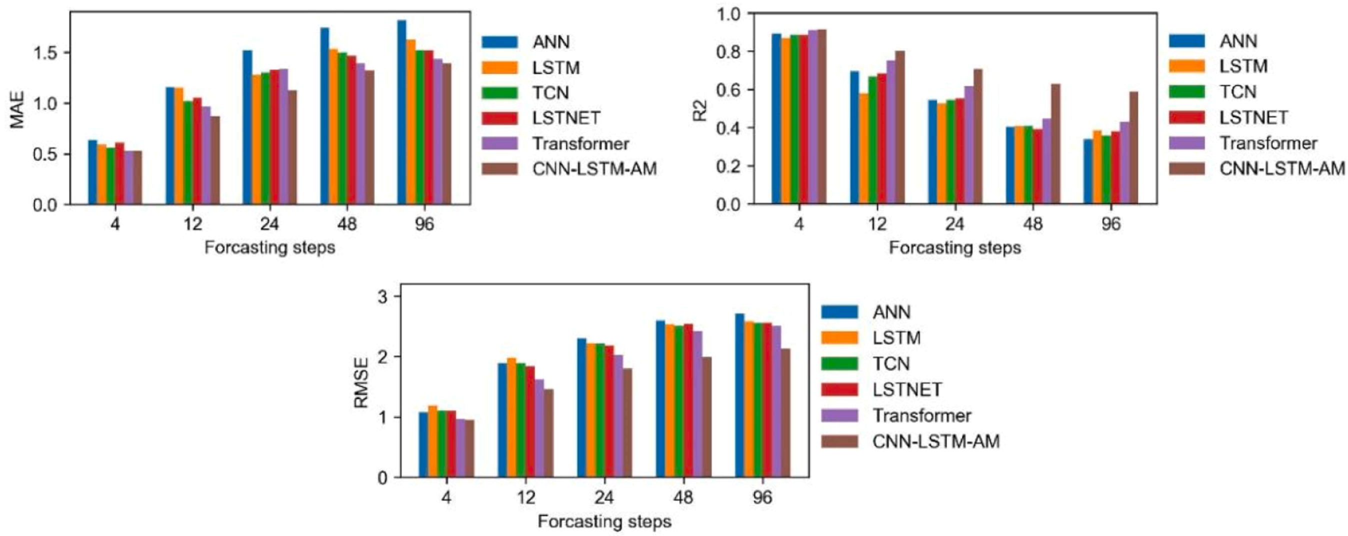

Fig. 9. Error values for Caltech dataset.

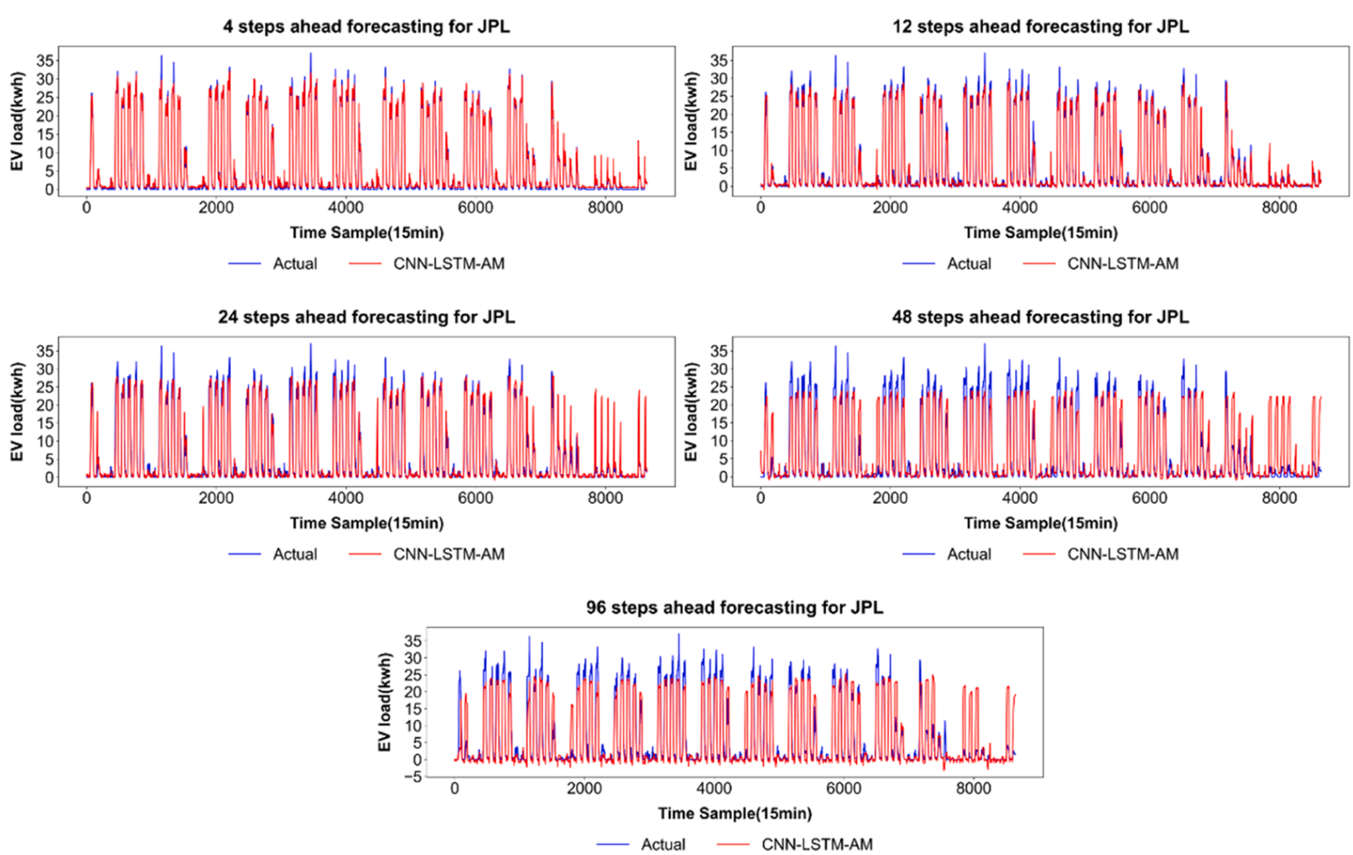

Fig. 10. 4-,12-,24-,48- and 96-step-ahead forecasting of the CNN-LSTM-AM model in the JPL dataset.

5. Conclusion and future work

This paper proposes using two time series with different intervals as inputs to a CNN network for multi-step prediction. This approach solves the limitation of single-series. The hybrid CNN-LSTM-AM model utilizes convolution layers to generate a feature matrix from the time series. Feature matrix is then input into the encoder and the hidden vector produced by the LSTM unit. The attention mechanism (AM) calculate score to each input time step and generates predicted values through the decoder.

We performed simulation experiments on the ACN-data. Comparing to single models, CNN-LSTM-AM obtained the lowest RMSE of 3.6 and MAE of 2.6. Results confirmed that the accuracy of the hybrid model is better than the single model in EV load prediction. In addition, we comparing to existing deep learning models ANN, LSTM, TCN, Transformer, LSTNet. Metrics have demonstrated that hybrid model outperforms in muti-step-ahead forecasting experiment. For 96-steps ahead forecasting, the RMSE value has decreased by 15.2 % by our method compared with Transformer, the MAE value has decreased by 3.1 % and R^2 value has increased by 16.8 %. This provides a reference for longer

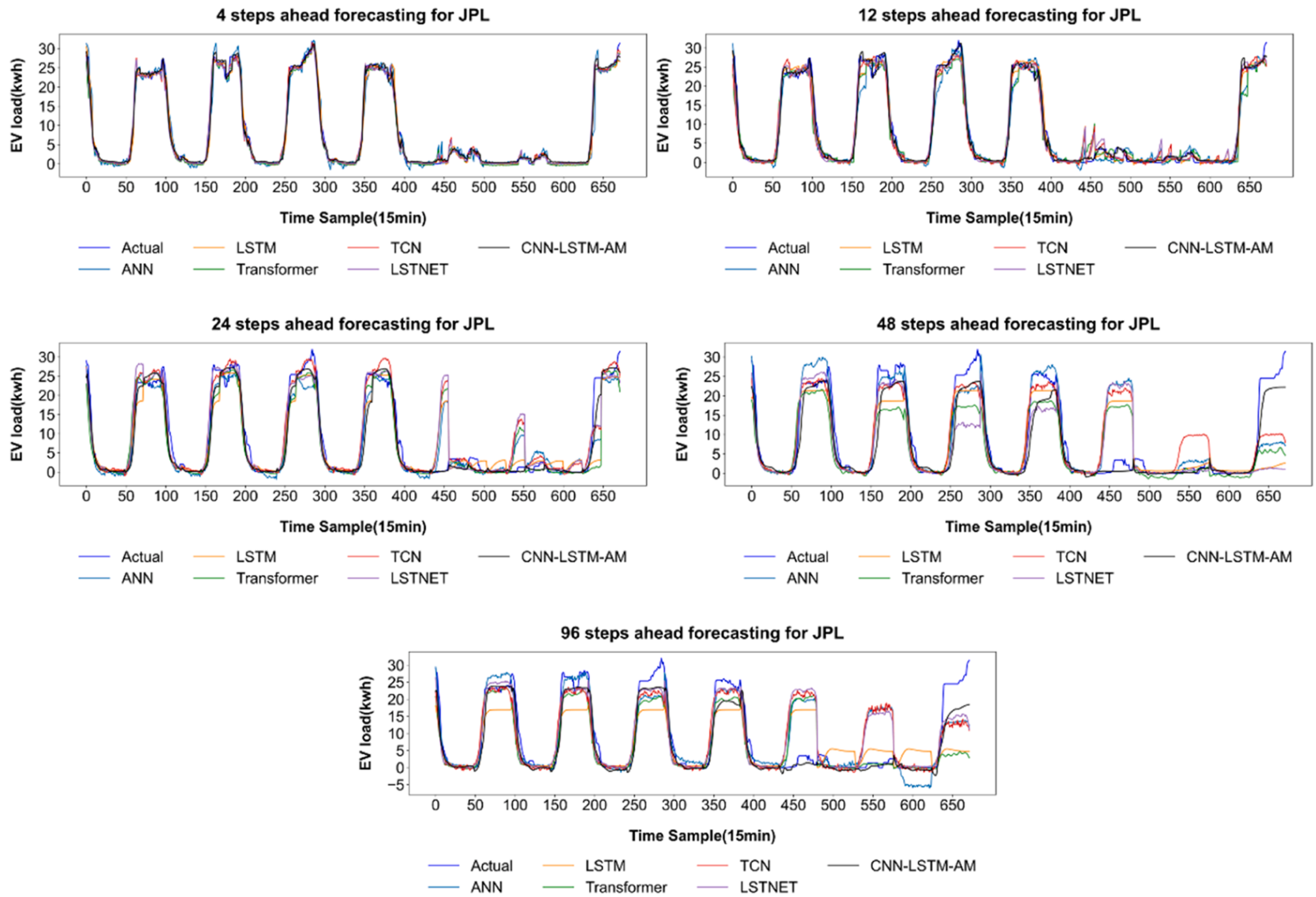

Fig. 11. Load forecasting results of different models for JPL dataset (a) 4-step ahead forecast (b) 12-step ahead forecast (c) 24-step ahead forecast (d) 48-step ahead forecast (e) 96-step ahead forecast.

Table 8
Multi-step-ahead Load results on JPL.

Step	Metric	ANN	LSTM	TCN	LSTNET	Transformer	Ours model
4	RMSE	1.1267	1.1250	1.0764	1.0642	1.1893	**0.9852**
	MAE	0.6697	0.5847	0.6132	0.6267	0.7431	**0.6032**
	R^2	0.9864	0.9864	0.9875	0.9863	0.9848	**0.9879**
12	RMSE	2.2367	1.9629	1.9978	1.9533	2.1266	**1.8604**
	MAE	1.5085	1.1648	1.1242	1.1521	1.2428	**1.0575**
	R^2	0.9464	0.9587	0.9573	0.9591	0.9516	**0.9629**
24	RMSE	4.1169	3.9758	3.9986	3.9345	3.9811	**3.1329**
	MAE	2.3978	2.3045	2.2774	2.1617	2.2120	**1.5658**
	R^2	0.8185	0.8309	0.8287	0.8345	0.8305	**0.8948**
48	RMSE	6.1271	6.0088	5.7747	5.6707	5.6673	**4.6711**
	MAE	4.0749	3.7631	3.5839	3.1120	3.0922	**2.5193**
	R^2	0.6155	0.5988	0.6434	0.6566	0.6599	**0.7665**
96	RMSE	6.5138	6.1602	6.2052	6.1045	6.1600	**5.0243**
	MAE	4.3854	3.6739	3.8107	3.2432	3.4892	**2.5042**
	R^2	0.5642	0.5946	0.5881	0.6033	0.5843	**0.7303**

Table 9
Comparison with other approaches of EV load from ACN-datasets.

Model	Obersevation number of test set	RMSE	MAE
XGBoost [9]	288	5.9	–
Our method	288	4.7	3.2
LSTM [5]	744	5.9	4.2
Our method	744	3.6	2.6

multi-step forecasting in the future. The curves of the actual values and predicted values show that the CNN-LSTM-AM model can effectively capture real-time changes in load peaks and valleys, demonstrating strong generalization in muti-step load forecasting.

It is important to note that the accuracy of short-term EV load forecasting is also influenced by EV owner's behavior and weather conditions. Due to time constraints, this study did not consider other environmental factors in the research process. However, integrating environmental factors into short-term load forecasting will be a key focus of future research.

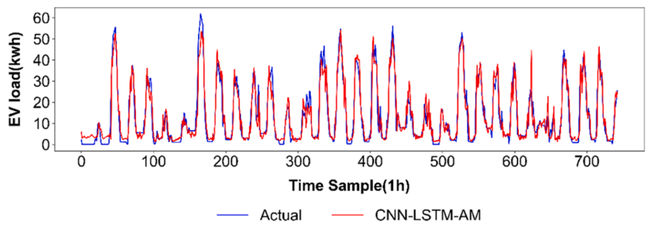

Fig. 12. Load forecasting of the CNN-LSTM-AM model in the Caltech dataset for 744 observations.

CRediT authorship contribution statement

Juan Ran: Writing – review & editing, Software, Methodology, Conceptualization. **Yunbo Gong:** Writing – original draft, Validation, Methodology. **Yu Hu:** Writing – review & editing, Investigation. **JiaLing Cai:** Writing – original draft, Validation, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] J. Dai, M. Dong, R. Ye, A. Ma, W. Yang, A review on electric vehicles and renewable energy synergies in smart grid, in: 2016 China International Conference on Electricity Distribution (CICED), 2016, https://doi.org/10.1109/ciced.2016.7575995.

[2] H. Cai, Q. Chen, Z. Guan, J. Huang, Day-ahead optimal charging/discharging scheduling for electric vehicles in microgrids, Protection and Control of Modern Power Systems 3 (1) (2018), https://doi.org/10.1186/s41601-018-0083-3.

[3] S. Deng, J. Wang, L. Tao, S. Zhang, H. Sun, EV charging load forecasting model mining algorithm based on hybrid intelligence, Comput. Electr. Eng. 112 (2023) 109010, https://doi.org/10.1016/j.compeleceng.2023.109010.

[4] H. Hou, C. Liu, Q. Wang, X. Wu, J. Tang, Y. Shi, et al., Review of load forecasting based on artificial intelligence methodologies, models, and challenges, Electric Power Systems Research 210 (2022), https://doi.org/10.1016/j.epsr.2022.108067.

[5] G. Vishnu, D. Kaliyaperumal, P.B. Pati, et al., Short-term forecasting of electric vehicle load using time series, machine learning, and deep learning techniques, World Electric Vehicle Journal 14 (9) (2023) 266, https://doi.org/10.3390/wevj14090266.

[6] A. Agga, et al., CNN-LSTM: an efficient hybrid deep learning architecture for predicting short-term photovoltaic power production, Electric Power Systems Research 208 (2022), https://doi.org/10.1016/J.EPSR.2022.107908.

[7] Z.J. Lee, T. Li, S.H. Low, ACN-Data: analysis and applications of an open EV charging dataset, in: the Tenth ACM International Conference. ACM, 2019, https://doi.org/10.1145/3307772.3328313.

[8] G. Xue, C. Qi, H. Li, X. Kong, J. Song, Heating load prediction based on attention long short term memory: a case study of Xingtai, Energy 203 (2020) 117846, https://doi.org/10.1016/j.energy.2020.117846.

[9] H. Rathore, H.K. Meena, P. Jain, Prediction of EV energy consumption using random forest and xgboost, in: 2023 International Conference on Power Electronics and Energy (ICPEE), IEEE, 2023, pp. 1–6, https://doi.org/10.1109/ICPEE54198.2023.10060798.

[10] J. Wang, Y. Song, F. Liu, R. Hou, Analysis and application of forecasting models in wind power integration: a review of multi-step-ahead wind speed forecasting models, Renewable Sustainable Energy Rev. 60 (2016) 960–981, https://doi.org/10.1016/j.rser.2016.01.114.

[11] S. Ben Taieb, G. Bontempi, A. Sorjamaa, A. Lendasse, Long-term prediction of time series by combining direct and mimo strategies, in: Proceedings of the 2009 IEEE International Joint Conference on Neural Networks, Atlanta, U.S.A., 2009, pp. 3054–3061.

[12] S. Ben Taieb, G. Bontempi, A.F. Atiya, A. Sorjamaa, A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition, Expert Syst. Appl. 39 (8) (2012) 7067–7083, https://doi.org/10.1016/j.eswa.2012.01.039.

[13] G. Bontempi, Long term time series prediction with multi-input multioutput local learning, in: Proceedings of the 2nd European Symposium on Time Series Prediction (TSP), ESTSP08, 2008.

[14] D.M. Kline, Methods for multi-step time series forecasting with neural networks, in: G.P. Zhang (Ed.), Neural Networks in Business Forecasting, Information Science Publishing, 2004, pp. 226–250.

[15] P. Jiang, Y. Nie, J. Wang, X. Huang, Multivariable short-term electricity price forecasting using artificial intelligence and multi-input multi-output scheme, Energy Econ. 117 (2023) 106471, https://doi.org/10.1016/j.eneco.2022.106471.

[16] C. Zhang, N.N.A. Sjarif, R. Ibrahim, 1D-CapsNet-LSTM: a deep learning-based model for multi-step stock index forecasting, J. King Saud Univ. (2024) 101959, https://doi.org/10.1016/j.jksuci.2024.101959.

[17] H. Luo, D. Wang, J. Cheng, et al., Multi-step-ahead copper price forecasting using a two-phase architecture based on an improved LSTM with novel input strategy and error correction, Resour. Policy 79 (2022) 102962, https://doi.org/10.1016/j.resourpol.2022.102962.

[18] C. Deng, Y. Huang, N. Hasan, et al., Multi-step-ahead stock price index forecasting using long short-term memory model with multivariate empirical mode decomposition, Inf Sci (Ny) 607 (2022) 297–321, https://doi.org/10.1016/j.ins.2022.05.088.

[19] H. Eskandari, M. Imani, M.P Moghaddam, Convolutional and recurrent neural network based model for short-term load forecasting, Electric Power Syst. Rese. 195 (2021) 107173, https://doi.org/10.1016/j.epsr.2021.107173.

[20] Z. Fazlipour, E. Mashhour, M Joorabian, A deep model for short-term load forecasting applying a stacked autoencoder based on LSTM supported by a multi-stage attention mechanism, Appl. Energy 327 (2022) 120063, https://doi.org/10.1016/j.apenergy.2022.120063.

[21] F. Li, Z. Wan, T. Koch, G. Zan, M. Li, Z. Zheng, B. Liang, Improving the accuracy of multi-step prediction of building energy consumption based on EEMD-PSO-Informer and long-time series, Comput. Electrical. Eng. 110 (2023) 108845, https://doi.org/10.1016/j.compeleceng.2023.108845.

[22] I.M. Baytas, C. Xiao, X. Zhang, F. Wang, A.K. Jain, J. Zhou, Patient Subtyping via Time-Aware LSTM Networks, in: 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2017, pp. 65–74, https://doi.org/10.1145/3097983.3097997.

[23] M.S. Abid, H.J. Apon, A. Alavi, M.A. Hossain, F. Abid, Mitigating the effect of electric vehicle integration in distribution grid using slime mould algorithm, Alexandria Engineering Journal 64 (2022) 785–800, https://doi.org/10.1016/j.aej.2022.09.022.

[24] D. Liu, Z. Li, J. Jiang, X. Cheng, G. Wu, Electric vehicle load forecast based on monte carlo algorithm, in: 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), 2020, pp. 1760–1763, https://doi.org/10.1109/itaic49862.2020.9338988.

[25] S.S. Pappas, L. Ekonomou, D.C. Karamousantas, G.E. Chatzarakis, S.K. Katsikas, P. Liatsis, Electricity demand loads modeling using AutoRegressive Moving Average (ARMA) models, Energy 33 (9) (2008) 1353–1360, https://doi.org/10.1016/j.energy.2008.05.008.

[26] D.H. Vu, K.M. Muttaqi, A.P. Agalgaonkar, A Bouzerdoum, Short-term electricity demand forecasting using autoregressive based time varying model incorporating representative data adjustment, Appl. Energy 205 (2017) 790–801, https://doi.org/10.1016/j.apenergy.2017.08.135.

[27] A.M. Nobre, C.A. Severiano, S. Karthik, M. Kubis, L. Zhao, F.R. Martins, T. Reindl, PV power conversion and short-term forecasting in a tropical, densely-built environment in Singapore, Renew Energy 94 (2016) 496–509, https://doi.org/10.1016/j.renene.2016.03.075.

[28] F. Wu, C. Cattani, W. Song, E. Zio, Fractional ARIMA with an improved cuckoo search optimization for the efficient Short-term power load forecasting, Alexandria Eng. J. 59 (2020) 3111–3118, https://doi.org/10.1016/j.aej.2020.06.049.

[29] H. Nguyen, M.A. Naeem, N. Wichitaksorn, R. Pears, A smart system for short-term price prediction using time series models, Comput. Electr. Eng. 76 (2019) 339–352, https://doi.org/10.1016/j.compeleceng.2019.04.013.

[30] B. Manobel, F. Sehnke, J.A. Lazzús, I. Salfate, M. Felder, S. Montecinos, Wind turbine power curve modeling based on gaussian processes and artificial neural networks, Renew Energy 125 (2018) 1015–1020, https://doi.org/10.1016/j.renene.2018.02.081.

[31] Qiming Sun, Jihong Liu, Xiaoxue Rong, Meng Zhang, Xiangqian Song, Zhaohong Bie, Zhaorui Ni, Charging load forecasting of electric vehicle charging station based on support vector regression, in: 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC), 2016, https://doi.org/10.1109/appeec.2016.7779794.

[32] P. Ray, D.P. Mishra, R.K. Lenka, Short term load forecasting by artificial neural network, in: 2016 International Conference on Next Generation Intelligent Systems (ICNGIS), 2016, https://doi.org/10.1109/icngis.2016.7854003.

[33] M. Imani, Electrical load-temperature CNN for residential load forecasting, Energy 227 (2021) 120480, https://doi.org/10.1016/j.energy.2021.120480.

[34] M.A. Gharibi, H.A. Abyaneh, Parking lots Load prediction by LSTM, in: 2022 IEEE 7th International Energy Conference (ENERGYCON), 2022, pp. 1–7, https://doi.org/10.1109/energycon53164.2022.9830378.

[35] H. Zang, R. Xu, L. Cheng, T. Ding, L. Liu, Z. Wei, G. Sun, Residential load forecasting based on LSTM fusing self-attention mechanism with pooling, Energy 229 (2021) 120682, https://doi.org/10.1016/j.energy.2021.120682.

[36] X. Chu, H. Jin, Y. Li, J. Feng, W. Mu, CDA-LSTM: an evolutionary convolution-based dual-attention LSTM for univariate time series prediction, Neural. Comput. Appl. 33 (2021) 16113–16137, https://doi.org/10.1007/s00521-021-06212-2.

[37] L. Pengtao, K. Zhou, X. Lu, S. Yang, A hybrid deep learning model for short-term PV power forecasting, Appl. Energy 259 (2020) 114216, https://doi.org/10.1016/j.apenergy.2019.114216.

[38] A. Bracale, P. De Falco, G. Carpinelli, Comparing univariate and multivariate methods for probabilistic industrial load forecasting, in: 2018 5th International Symposium on Environment-Friendly Energies and Applications (EFEA), 2018, pp. 1–6, https://doi.org/10.1109/EFEA.2018.8617111.

[39] F. Yang, X. Fu, Q. Yang, Z. Chu, Decomposition strategy and attention-based long short-term memory network for multi-step ultra-short-term agricultural power load forecasting, Expert Syst. Appl. 238 (2023) 122226, https://doi.org/10.1016/j.eswa.2023.122226. Vol.

[40] B. Yang, X. Yuan, F. Tang, A novel dispatch adaptation load feature mapping network for multi-timestep load forecast, Energy Reports (2023), https://doi.org/10.1016/j.egyr.2023.09.118.

[41] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, ArXiv (2018), https://doi.org/10.48550/arXiv.1803.01271 abs/1803.01271.

[42] G. Lai, W.-C. Chang, Y. Yang, H. Liu, Modeling long- and short-term temporal patterns with deep neural networks, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval - SIGIR '18, 2018, https://doi.org/10.1145/3209978.3210006.

[43] A. Vaswani, N.M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, NIPS (2017), https://doi.org/10.48550/arXiv.1706.03762.