Short-Term Load Forecasting in Power System Using CNN-LSTM Neural Network

Truong Hoang Bao Huy
Department of Future Convergence Technology
Soonchunhyang University
Asan-si, Chuncheongnam-do, South Korea
trhbhuy@sch.ac.kr

Dieu Ngoc Vo
Department of Power Systems
Ho Chi Minh City University of Technology (HCMUT)
Vietnam National University Ho Chi Minh City
Ho Chi Minh City, Vietnam
vndieu@hcmut.edu.vn

Khai Phuc Nguyen
Department of Power Systems
Ho Chi Minh City University of Technology (HCMUT)
Vietnam National University Ho Chi Minh City
Ho Chi Minh City, Vietnam
phuckhai@hcmut.edu.vn

Viet Quoc Huynh
Department of Power Systems
Ho Chi Minh City University of Technology (HCMUT)
Vietnam National University Ho Chi Minh City
Ho Chi Minh City, Vietnam
hqviet@hcmut.edu.vn

Minh Quang Huynh
Department of Power Systems
Ho Chi Minh City University of Technology (HCMUT)
Vietnam National University Ho Chi Minh City
Ho Chi Minh City, Vietnam
hqminh@hcmut.edu.vn

Khoa Hoang Truong
Department of Power Delivery
Ho Chi Minh City University of Technology (HCMUT)
Vietnam National University Ho Chi Minh City
Ho Chi Minh City, Vietnam
Correspoding author: trhkhoa@hcmut.edu.vn

2023 Asia Meeting on Environment and Electrical Engineering (EEE-AM) | 979-8-3503-8106-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/EEE-AM58328.2023.10395221

***Abstract*—The accurate forecasting of short-term load plays a significant role in power systems operation and planning. This paper suggests a short-term load forecasting model combining Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM). The developed CNN-LSTM aims to capture both spatial and temporal dependencies within the load data, leveraging the strengths of both architectures. Simulations are performed using real-world power system load data. Comparative analyses are carried out against standalone CNN and LSTM models. The CNN-LSTM has significantly better forecasting accuracy than other models, showcasing its effectiveness in short-term load forecasting.**

***Index Terms*—Short-term load forecasting, CNN-LSTM, Long Short-Term Memory, Convolutional Neural Networks**

I. Introduction

Short-Term Load Forecasting is essential for operations and planning in power systems, offering numerous benefits to power utilities and grid operators. The accurate prediction of load demand is crucial for maintaining the stability, reliability, and efficiency of the power system. Grid operators rely on load forecasting to effectively balance the electricity supply and demand, avoiding imbalances that can lead to disruptions or blackouts. By anticipating load variations in the near future, operators can make informed decisions regarding the optimal allocation and dispatch of power generation resources, transmission capacities, and distribution networks. This enables efficient resource planning, minimizes operational costs, and improves overall system efficiency. Therefore, it is very important to develop efficient short-term load forecasting models.

Traditional methods of forecasting load are exponential smoothing [1], regression analysis [2], and autoregressive integrated moving normal (ARIMA) [3]. During recent years, machine learning methods have been suggested, such as fuzzy logic [4], support vector machine (SVR) [5], and artificial neural network (ANN) [6]–[8]. In [9], a multi-step recursive method using an echo state network (ESN) was employed for forecasting short-term loads. Long short-term memory (LSTM) was suggested in [10] and [11] to forecast the short-term aggregated load. To forecast day-ahead load profiles, a shallow ANN and deep neural network (DNN) were combined in [12]. Based on weather forecast variables and historical load data, the authors in [13] applied a wavelet neural network (WNN) forecasting system. In [14], short-term load forecasts were presented using a multi-stage ANN model based on forecasted temperatures.

This paper suggests a short-term electricity load forecasting model using Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM). By leveraging CNN and LSTM strengths, the CNN-LSTM efficiently predicts electrical loads.

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.

The CNN-LSTM model is applied to make one-step and multi-step forecasting. This study considers real-world power system load data of Ho Chi Minh City. New input features are also generated based on rolling time-index series, including hour, weekday/weekend, and month indexes. The results from the suggested model are also compared with those from CNN and LSTM models, which indicate its superior performance.

II. CNN-LSTM Model

A. CNN

CNN is a feed-forward neural network inspired by biological visual cognition. CNN has the capability to enhance the features of limited data and thoroughly explore the latent information of time series data. Hence, CNN is used to capture the local features of time series data. Its structure contains convolutional, pooling, and fully connected layers. The quantity of convolution kernels can determine the level of feature extraction abstraction. The number of convolution kernels can define the level of abstraction of feature extraction. Additionally, the convolution kernel size can be modified to match the fixed length of the input sequence data. The pooling layer is utilized to discard less significant features. Common pooling techniques include average pooling and maximum pooling. These pooling operations frequently make use of the ReLU activation function. The primary function of the fully connected layer is to connect the neurons from the pooling layer into one-dimensional vectors. The one-dimensional convolutional layer is utilized to discover potential features in the fixed-length input sequence. Following the convolutional and pooling layers, the corresponding feature vectors are produced as output. The convolution operation can be defined in the following equation:

$$y_i = \sigma \sum_{j=1}^{k} (w_j \cdot x_{i-j+k} + b) \tag{1}$$

where x_{i-j+k} represents the input series, w_j represents the weight matrix of the convolution kernel, σ represents the activation function, k represents the number of convolution kernels, b represents the deviation value, and y_i represents the output value of the convolutional layer.

B. LSTM

LSTM is an RNN architecture designed to capture long-term dependencies within sequential data, overcoming the drawbacks of conventional RNNs. LSTM can address the challenge of the vanishing gradient problem, which occurs when gradients exponentially diminish during backward propagation through time in conventional RNNs. This problem hinders the ability of RNNs to capture long-range dependencies effectively. The LSTM architecture is depicted in Fig. 1. At the core of the LSTM architecture are memory cells, which can store information over long periods of time. These memory cells have a unique structure that allows them to selectively learn and forget information based on the context of the input sequence. This selective memory management is facilitated by

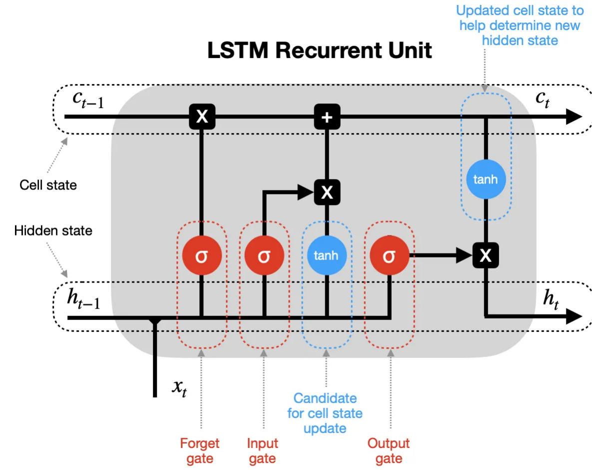

Fig. 1. The architecture of LSTM [15].

three main components: the input gate, the forget gate, and the output gate. The mathematical operation of the LSTM can be given in the following equations:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \tag{2}$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \tag{3}$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \tag{4}$$

$$\tilde{c}_t = \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \tag{5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{6}$$

$$h_t = o_t \odot \tanh(C_t) \tag{7}$$

where i_t, f_t, and o_t denote the input gate, forget gate, and output gate, respectively; W and U represent weight matrices; b represent biased values of different gates; x_t denotes the input vector at the current time-step; h_{t-1} and h_t denote the hidden state at the previous time-step and current hidden state, respectively; c_{t-1}, c_t, and $\tilde{c}_t$ denote the cell state at the previous time-step, current cell state, and candidate cell state, respectively; $\odot$ denotes an element-wise operator; σ_g and σ_c represent the non-linear activation functions.

C. CNN-LSTM

CNN-LSTM is proposed specifically for time series forecasting tasks, where the input data has a spatial structure or grid-like format. The key idea behind CNN-LSTM is to replace the matrix-vector multiplication operations in the LSTM memory cell and gating mechanisms with convolutional operations. This enables the CNN-LSTM to process input sequences of arbitrary length while preserving the spatial structure of the data.

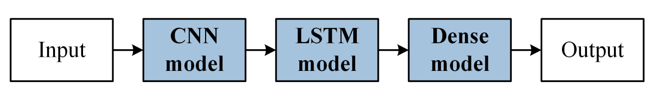

Fig. 2. A typical CNN-LSTM model.

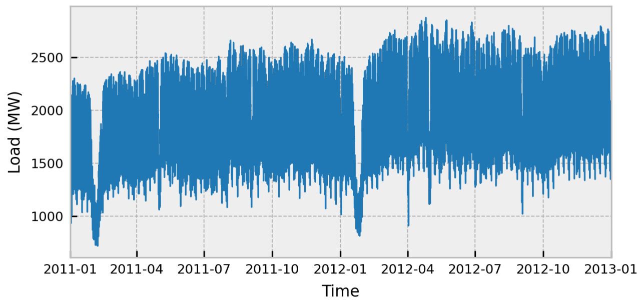

Fig. 3. Data on electricity load demand in Ho Chi Minh City.

Fig. 2 depicts CNN-LSTM architecture. The CNN model captures the local pattern of the data. As input to the LSTM layer, a single one-dimensional vector is created by flattening the samples. The LSTM model is specifically created for learning the long-term dependencies in the data. The final forecasting is generated by passing through dense layers. By incorporating convolutional operations into the LSTM architecture, time series data are effectively captured in terms of both short-term and long-term dependencies while leveraging spatial structure in the CNN-LSTM model.

III. CNN-LSTM-Based Forecasting Model

A. Data collection

Time series data on load consumption is collected from historical data, and then null values are examined to confirm reliability and accuracy.

B. Feature extraction

New features, which include the month indexes $\{M = 1, 2, ..., 12\}$, weekday indexes $\{W = 0, 1, 2\}$, and hour indexes $\{H = 0, 1, ..., 23\}$, are created.

C. Normalization

The features can be normalized using the min-max technique as follows:

$$x_{\text{norm}} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{8}$$

D. Time Series Construction

A length T is determined for the input sequence. Following the normalization of input features, the sequences can be structured in the following manner: the hourly load consumption of previous T time-steps; the month indexes of previous T time-steps; the weekday indexes of previous T time-steps; the hour indexes of previous T time-steps. After normalization, the input features are converted to shape: [samples, time-steps, features].

TABLE I
Results of single-step forecasting under various lengths of input sequences

Time-step	RMSE (MW)	MAE (MW)	MAPE (%)
2	46.1851	30.8555	1.5434
4	44.6730	30.2609	1.5247
8	43.6801	29.4679	1.4820
12	42.8693	28.3057	1.4264
16	42.7247	28.3743	1.4157
20	42.3591	28.4095	1.4270
24	42.4666	28.0615	1.4125
48	42.4933	27.7398	1.4048
72	43.8714	30.2230	1.5045
120	43.0970	28.3001	1.4321
144	42.5879	28.8703	1.4513
168	43.1719	29.2877	1.4597

TABLE II
Results of single-step forecasting for different months

Time-step	RMSE (MW)	MAE (MW)	MAPE (%)
September	42.1169	29.9549	1.5853
October	51.2590	28.1783	1.4374
November	35.1319	25.2258	1.2258
December	39.5973	27.5938	1.3712

E. Forecasting Model

The CNN-LSTM model uses a single one-dimensional convolutional layer with a convolutional kernel size of 2. LSTM contains a single layer with 100 neurons. A flatten layer is attached to output one-dimensional vectors. Two Dense layers incorporate the output into one-dimensional data. The initial Dense layer consists of 50 neurons, while the other layer is the output layer. In CNN-LSTM, the mean square error (MSE) loss function is minimized using the Adam optimizer.

- Single-step forecasts create one output value $Y_t = \{d_{t+1}\}$ for the following time step.
- Multi-step forecasts create the output sequence $Y_t = \{d_{t+1}, d_{t+2}, ..., d_{t+N}\}$ for the following N time steps.

IV. Simulation Results

A. Simulation Setup

Fig. 3 presents the electricity load demand, which is taken from historical data for 2011-2012 in Ho Chi Minh City. With an hourly sampling interval, the dataset contains 17544 samples. The training data is from January 2011 to April 2012, while the validation data is from May 2012 to August 2012. Data from September - December 2012 are employed to test the performance of the developed forecasting model.

B. Performance Indicators

This study uses the following performance metrics to evaluate forecasting results:

$$\text{RMSE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}(y_k - \hat{y}_k)^2} \tag{9}$$

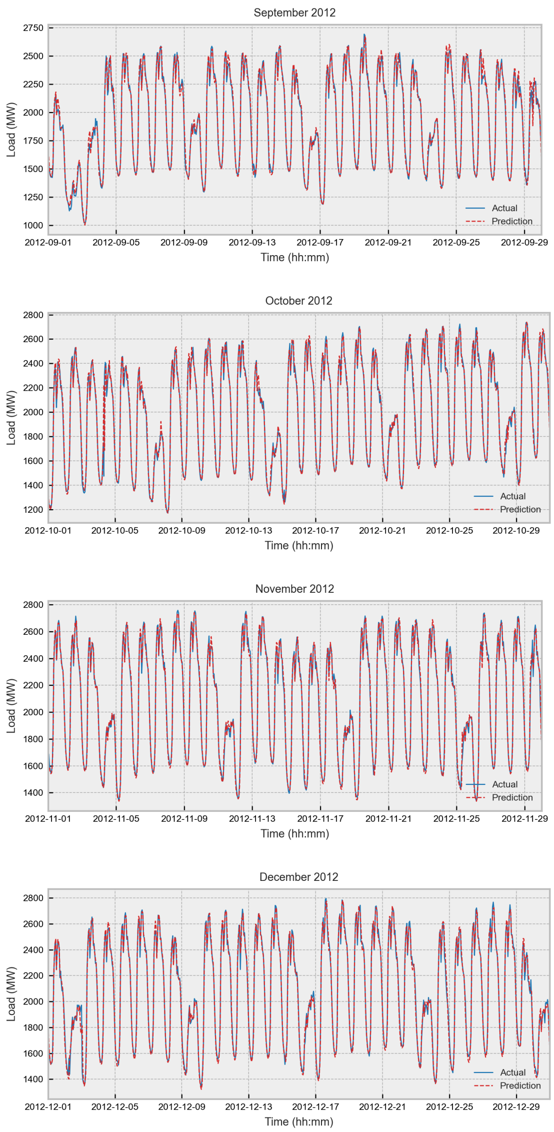

Fig. 4. Results of single-step forecasting for different months.

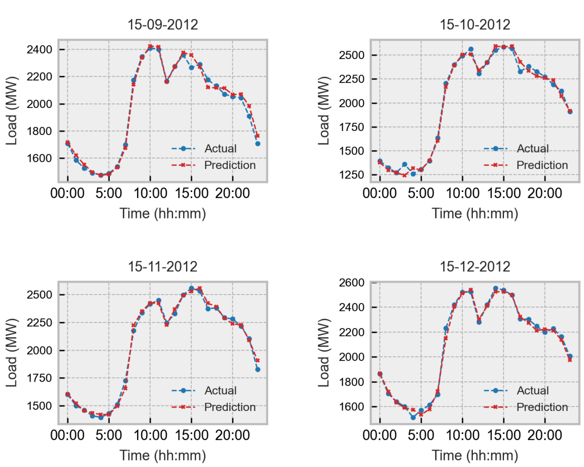

Fig. 5. Results of single-step forecasting for different typical days.

TABLE III
RESULTS OF MULTI-STEP FORECASTING FORECASTING UNDER VARIOUS LENGTHS OF INPUT SEQUENCES

Time-step	RMSE (MW)	MAE (MW)	MAPE (%)
24	86.6166	61.5528	3.1143
48	92.6941	67.6267	3.3957
72	93.9456	66.3116	3.3136
120	94.2306	65.9103	3.3780
144	90.4473	63.1566	3.2580
168	96.6155	69.4435	3.4947

TABLE IV
RESULTS OF MULTI-STEP FORECASTING FOR DIFFERENT MONTHS

Time-step	RMSE (MW)	MAE (MW)	MAPE (%)
September	100.8959	74.2654	4.0733
October	74.0861	53.4525	2.7356
November	79.4760	58.7400	2.7548
December	89.8640	60.0917	2.9146

$$\mathrm{MAE} = \frac{1}{K}\sum_{k=1}^{K}|y_k - \hat{y}_k| \tag{10}$$

$$\mathrm{MAPE} = \frac{1}{K}\sum_{k=1}^{K}\left|\frac{y_k - \hat{y}_k}{y_k}\right| \tag{11}$$

where K denotes the size of the data samples, $\hat{y}_k$ denotes the predicted value, and y_k denotes the actual value.

C. Single-step forecast

The CNN-LSTM undergoes testing with different lengths of input sequences to determine an appropriate input sequence length for a single-step-ahead forecast. As shown in Table I, the CNN-LSTM model obtains the best results with the lowest values of MAE, MAPE, and RMSE metrics by selecting 48-step sequences. This indicates that a look-back window spanning 48 time steps can provide the model with the most relevant historical information for accurate forecasting.

Table II displays the accuracy of forecasts from September 2012 to December 2012. Notably, the best forecasting results are obtained in November 2012. This implies that the model works remarkably well in capturing and predicting patterns for that particular month. Figs. 4 and 5 illustrate the one-step forecast results for the final four months of 2012 as well as the typical days of these months. From these figures, predicted values can be observed alongside the actual values for each month and each typical day. This analysis provides valuable insights into the predictive performance of the model during this specific period. It can be seen that forecasted load values closely align with the actual data, showcasing the efficacy of the forecasting model to forecast one-step loads accurately.

D. Multi-step forecast

In this section, the CNN-LSTM is employed to forecast the 24-step ahead. Table III presents performance comparisons of the CNN-LSTM model for multi-step forecasting considering different input sequence lengths. The analysis of different

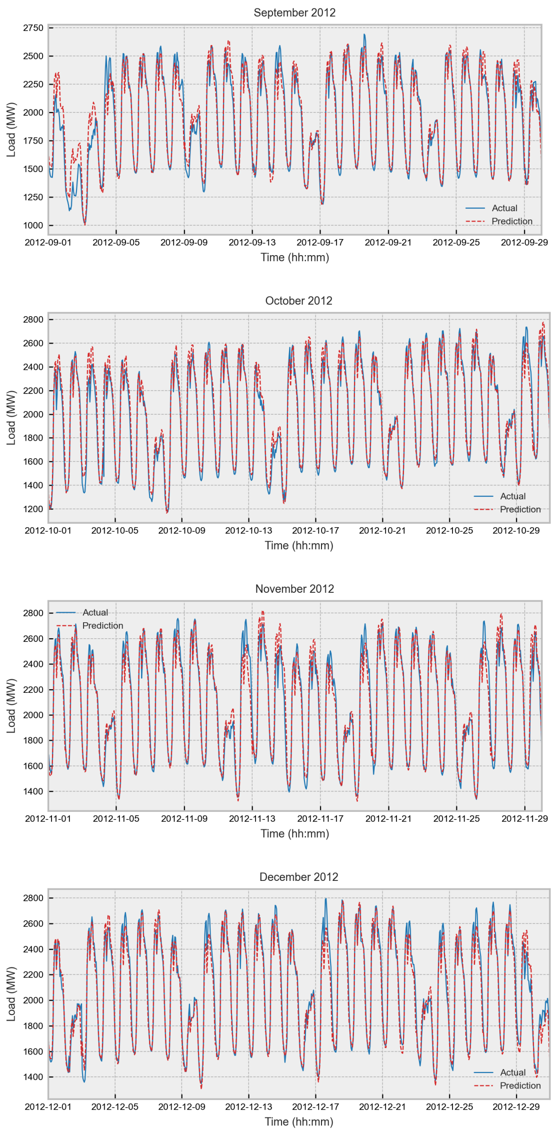

Fig. 6. Results of multi-step forecasting for different months.

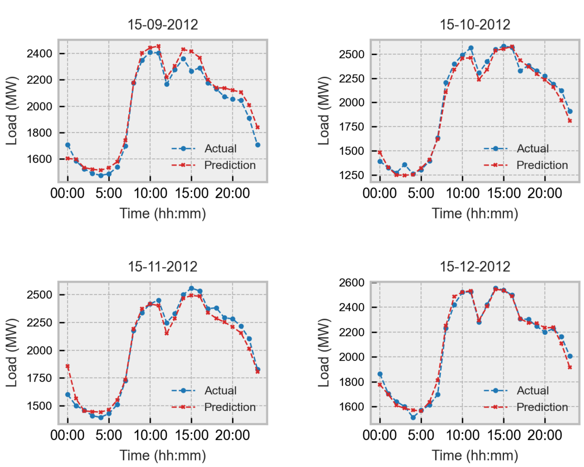

Fig. 7. Results of multi-step forecasting for different typical days.

TABLE V
COMPARISON OF FORECASTING PERFORMANCE FOR DIFFERENT MODELS

Methods	Single-step forecast			Multi-step forecast		
	RMSE (MW)	***MAE (MW)***	***MAPE (%)***	***RMSE (MW)***	***MAE (MW)***	***MAPE (%)***
CNN-LSTM	42.49	27.73	1.40	86.61	61.55	3.11
CNN	46.24	32.49	1.62	90.12	64.73	3.32
LSTM	43.54	28.61	1.45	93.33	64.16	3.23

input sequence lengths highlights the importance of selecting an appropriate look-back window size, as it directly impacts the efficacy of the proposed model. From Table III, the highest accuracy in multi-step forecasting is achieved by utilizing 24-step input sequences to forecast 24-step horizons.

The precision of multi-step forecasts is assessed over different months, and the results are tabulated in Table IV. The observed smallest forecasting error in October suggests that the model is successful in capturing the specific dynamics and trends present during that time of the year. Figs. 6 and 7 illustrate the results of the multi-step forecast for the final four months of 2012 as well as the typical days of these months. As shown in Figs. 6 and 7, the forecasted load values exhibit a considerable degree of consistency with the actual data. Hence, the CNN-LSTM proves its effectiveness in multi-step forecasting as well.

E. Performance comparison

Table V compares the forecast results obtained from CNN-LSTM, CNN, and LSTM models. It can be observed that the CNN-LSTM model yields the smallest values for MAE, MAPE, and RMSE among the three models. The CNN-LSTM model combines the advantages of both architectures, wherein relevant features are extracted from time series data using convolutional layers of CNNs and then passed to LSTM layers for sequence modeling and forecasting. By capturing both local and long-term dependencies in the data, this architecture improves forecasting performance. The fact that the CNN-LSTM model obtains the best forecast errors indicates that it outperforms both other models in this dataset.

V. CONCLUSION

This paper suggested an efficient forecasting model for short-term load forecasting using specialized CNN and LSTM structures. The CNN initially captures the distinctive information from the load sequence, which is then passed as one-dimensional vectors to the LSTM. The forecasting performance of the CNN-LSTM was validated using the load demand dataset from the power system in Ho Chi Minh City (Vietnam). The forecasting results indicate that the CNN-LSTM model has successfully captured the underlying patterns and dynamics of the dataset of hourly load demand. Moreover, comparisons show that the developed model obtained the best MAE, MAPE, and RMSE results compared to other models. Therefore, the proposed CNN-LSTM model is a potential method for accurate short-term electricity load forecasting.

REFERENCES

[1] W. Christiaanse, "Short-term load forecasting using general exponential smoothing," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-90, no. 2, pp. 900–911, 1971.

[2] W. Charytoniuk, M. Chen, and P. V. Olinda, "Nonparametric regression based short-term load forecasting," *IEEE Transactions on Power Systems*, vol. 13, no. 3, pp. 725–730, 1998.

[3] C.-M. Lee and C.-N. Ko, "Short-term load forecasting using lifting scheme and arima models," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5902–5911, 2011.

[4] H. H. Çevik and M. Çunkaş, "Short-term load forecasting using fuzzy logic and anfis," *Neural Computing and Applications*, vol. 26, no. 6, pp. 1355–1367, 2015.

[5] Y. Chen, P. Xu, Y. Chu, W. Li, Y. Wu, L. Ni, Y. Bao, and K. Wang, "Short-term electrical load forecasting using the support vector regression (svr) model to calculate the demand response baseline for office buildings," *Applied Energy*, vol. 195, pp. 659–670, 2017.

[6] D. Liu, L. Zeng, C. Li, K. Ma, Y. Chen, and Y. Cao, "A distributed short-term load forecasting method based on local weather information," *IEEE Systems Journal*, vol. 12, no. 1, pp. 208–215, 2018.

[7] J. Xie, Y. Chen, T. Hong, and T. D. Laing, "Relative humidity for load forecasting models," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 191–198, 2018.

[8] Y. Wang, Q. Chen, M. Sun, C. Kang, and Q. Xia, "An ensemble forecasting method for the aggregated load with subprofiles," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3906–3908, 2018.

[9] B. A. Hoverstad, A. Tidemann, H. Langseth, and P. Ozturk, "Short-term load forecasting with seasonal decomposition using evolution for parameter tuning," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1904–1913, 2015.

[10] K. Park, S. Yoon, and E. Hwang, "Hybrid load forecasting for mixed-use complex based on the characteristic load decomposition by pilot signals," *IEEE Access*, vol. 7, pp. 12 297–12 306, 2019.

[11] M. S. Hossain and H. Mahmood, "Short-term load forecasting using an lstm neural network," in *2020 IEEE Power and Energy Conference at Illinois (PECI)*, Feb 2020, pp. 1–6.

[12] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-term load forecasting with deep residual networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3943–3952, 2019.

[13] X. Sun, P. B. Luh, K. W. Cheung, W. Guan, L. D. Michel, S. S. Venkata, and M. T. Miller, "An efficient approach to short-term load forecasting at the distribution level," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 2526–2537, 2016.

[14] K. Methaprayoon, W. J. Lee, S. Rasmiddatta, J. R. Liao, and R. J. Ross, "Multistage artificial neural network short-term load forecasting engine with front-end weather forecast," *IEEE Transactions on Industry Applications*, vol. 43, no. 6, pp. 1410–1416, 2007.

[15] S. Dobilas, "Lstm recurrent neural networks — how to teach a network to remember the past," 2022. [Online]. Available: https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e