An Edge-AI Based Forecasting Approach for Improving Smart Microgrid Efficiency

Lingling Lv, Zongyu Wu, Lei Zhang, Brij B. Gupta, *Senior Member, IEEE*, and Zhihong Tian, *Senior Member, IEEE*

***Abstract*—Smart Grid 2.0 is the energy Internet based on advanced metering infrastructure and distributed systems that require an instantaneous two-way flow of energy information. Edge computing benefits from its proximity to the servers and edge nodes of the smart grid distributed systems, which can provide efficient and low latency information transmission to the smart grid. With the massive number of Internet of Things being used, the amount of real-time power usage information generated by that represents a huge challenge for edge computing. To improve the efficiency of information transmission and processing in power systems, this article combines different deep learning algorithms with edge computing to analyze and process distributed renewable energy generation and consumer power data in smart microgrid. Experiments on two real-world datasets from China and Belgium show that the proposed framework can obtain satisfactory prediction accuracy compared to existing approaches.**

***Index Terms*—Deep learning, edge-AI, energy efficiency, smart microgrid.**

Manuscript received 13 December 2021; revised 6 March 2022; accepted 25 March 2022. Date of publication 29 March 2022; date of current version 9 September 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB2012402, in part by the key project of the National Natural Science Foundation of China under Grant U20B2046, in part by the Scientific and Technological Innovation Team of Colleges and Universities in Henan Province under Grant 22IRTSTHN011, in part by the Excellent Youth Fund of Henan Natural Science Foundation under Grant 212300410058, in part by the Henan High Level Talent Support Plan under Grant ZYQR201810138, and in part by the Guangdong Province Universities and Colleges Pearl River Scholar Funded Scheme (2019). Paper no. TII-21-5525. *(Corresponding authors: Lei Zhang; Brij B. Gupta.)*

Lingling Lv and Zongyu Wu are with the Institute of Electric Power, North China University of Water Resources and Electric Power, Zhengzhou 450011, China (e-mail: lingling_lv@163.com; zongyu_wu@yeah.net).

Lei Zhang is with the Henan Key Laboratory of Big Data Analysis and Processing, Institute of Data and Knowledge Engineering, School of Computer and Information Engineering, Henan University, Kaifeng 475004, China, and also with the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China (e-mail: zhanglei@henu.edu.cn).

Brij B. Gupta is with the Department of Computer Science and Information Engineering, Asia University, Taichung 413, China, with the Research and Innovation Department, Skyline University College, Sharjah 1797, UAE, and also with the Staffordshire University, ST4 2DE Stoke-on-Trent, U.K. (e-mail: gupta.brij@gmail.com).

Zhihong Tian is with the Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China (e-mail: tianzhihong@gzhu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2022.3163137.

Digital Object Identifier 10.1109/TII.2022.3163137

I. Introduction

With the development of renewable energy sources (RESs) and the growth of consumer demand for electricity, the challenge of generating and supplying power from smart microgrids is enormous. Nowadays, distributed RESs (DRES) mainly include wind power, photovoltaic power, and biomass power, among which wind power is most widely used in smart microgrids [1]. Besides, consumer electricity demand contains transportation, medical, buildings, and factories, and they are also distributed [2]. The massive distributed systems make energy management less efficient and increase economic losses. Edge computing can place servers near distributed systems, which can increase bandwidth to improve the speed of information transfer [3]. Therefore, edge computing makes sense to improve the efficiency of smart microgrid energy management systems.

In recent years, edge computing has received sufficient attention from academics and researchers. Edge computing is used to process and analyze data from data sources at the edge nodes, where the edge nodes are nodes with computational and network resources in between the data sources and cloud services [4]. The privacy challenges associated with IoT sensors being applied to smart microgrids will be addressed by edge computing, which enables data to be processed and analyzed by nearby edge servers, avoiding data transmission over public networks and reducing the risk of privacy leaks [5]. The application of edge computing can reduce the time consumed by data transmission from the root, speed up data processing, reduce the amount of computing in the cloud, and, thus, improve the efficiency of the system [6]. Multiaccess edge computing is proposed to handle the problem of bandwidth explosion and central network computational overload caused by the rapid growth of Internet of Things (IoT) smart devices [7]. Power data has complex nonlinear relationships; so it is difficult to analyze and process power data with edge computing.

The digital grid is the future direction of the electric power grid. In 2020, about 102.9 million smart meters were installed in USA, which record and transmit huge electricity data to database [8]. This real-time electricity data provides a rich source of information about electricity demand, which can be used to improve the efficiency of the smart grid [9]. Therefore, a lot

of artificial intelligence algorithms are applied to the smart grid for predicting future electric data information [10].

In fact, achieving high accuracy forecasting of renewable energy generation and electric load is still a problem. RESs such as wind and solar power are susceptible to weather factors [11]. In addition, the accuracy of the forecast is easily affected by the obvious seasonality of the power load data [12]. The authors of [13] presented a hybrid forecasting model based on long short-term memory (LSTM) convolutional neural network (CNN) and considered weather factors such as temperature, precipitation, relative humidity, and cloud cover. The experimental results showed that the forecast skill score of the hybrid model is 37%–45% higher than few standalone models. In [14], outliers in the supervisory control and data acquisition (SCADA) data were detected and removed by an isolated forest (IF), and then the SCADA data were fed into the LSTM gated recurrent unit (GRU) model to predict wind power generation in the next few days, where the SCADA data contains 12 features. A hybrid method based LSTM and variational mode decomposition (VMD) considering seasonal factors and error correction was proposed in [15]. A combined wind speed forecasting model based improved complete ensemble empirical mode decomposition (iCEEMDAN) with adaptive noise and artificial neural networks was proposed in [16]. The methods mentioned in this paragraph provide support for considering weather factors. However, the prediction accuracy of these methods cannot meet the current demand of smart grid optimization (SMO). Therefore, prediction models need to be designed to meet SMO based on the respective characteristics of RESs generation and electric load.

Smart Grid 2.0 is the next generation of the power grid, which includes distributed renewable energy generation systems, bi-directional energy and information flow networks, and energy management systems. To realize Smart Grid 2.0, there should be a smart microgrid that enables fully automated regulation of distributed renewable energy generation and consumption power supply. Therefore, empowering artificial intelligence technology to edge computing (edge-AI) at the edge nodes, it plays a key role in realizing the smart microgrid. In this article, a hybrid framework based on edge-AI to analyze the electricity data from smart meters to optimize the efficiency of smart microgrids is proposed. The first algorithm for edge-AI is used to analyze the distributed wind power data to predict short-term electricity generation. The second algorithm for edge-AI is used to analyze the data from smart meters to predict short-term consumption of electricity. The main concentrations of this article include the following.

1) An edge-AI based forecasting framework for improving smart microgrid efficiency is proposed. A novel short-term wind generation forecasting algorithm based on 1-D CNN and GRU is applied in DRES edge devices, and a novel short-term electric load prediction algorithm with CEEMDAN and GRU is applied in end-user edge devices.
2) The weather data recorded by smart sensors such as temperature, humidity, air pressure, and air density are considered in the algorithm of DRES edge devices. The attention mechanism is introduced to give enough weights to useful data and reduce the weights of useless data in the all edge-AI algorithms.

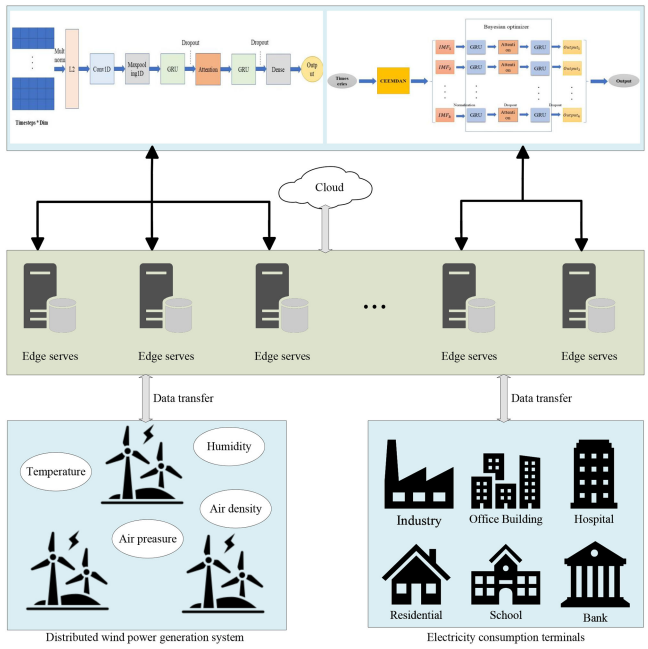

Fig. 1. System framework.

3) Bayesian optimization based on kerastuner is used to tune the hyperparameters and weights of the short-term forecasting methods. A novel strategy for controlling the learning rate of the Adam optimizer is applied to model training.
4) Experimental results on real-world data show that the model proposed in this article improves the mean absolute percentage error (MAPE) of wind power prediction by at least 4% and the MAPE of electric load by at least 0.93% compared to other models.

The rest of this article is organized as follows. Section II introduces the system framework of the smart microgrid. Section III clearly describes the proposed methods including CNN, GRU, attention mechanism, and CEEMDAN. In Section IV, real-world datasets from China and Belgian are used to test the performance of the proposed framework and contrast methods. Finally, Section V concludes the whole work of this article.

II. System Framework

The smart microgrid with autonomous control based on edge computing and AI proposed in this article is shown in Fig. 1, which contains distributed wind electricity generation system, edge devices, electricity consumption terminal, and a cloud-based power management system. Its main principles can be summarized as follows.

1) Distributed wind power generation system contains plenty of smart sensors that constantly record data that affects wind power generation, such as air density, humidity, air pressure, and temperature. All devices in the distributed system are IoT devices, and the information sent by these devices are received by the nearby edge servers. To instantly analyze and process the multidimensional data received by the edge servers, CNN and GRU algorithm with attention mechanism are deployed to edge-AI.

Therefore, edge-AI assumes part of the computing functions of the cloud server in the smart microgrid and sends the forecasting power generation of the distributed wind power generation system to the cloud server to improve the transfer efficiency.

2) Electricity consumption terminals encompass all fields of people's lives, while the application of IoT devices also generates enormous electricity consumption information. These seasonal discrete electricity consumption data are correlated with people's lives. To efficiently analyze and process these electricity consumption data, CEEMDAN and GRU algorithms with attention mechanism are deployed to edge-AI. Therefore, the edge-AI analyzes and processes the electricity consumption data and sends the predicted electricity demand to the cloud server.
3) The cloud-based energy management system receives forecasts from edge-AI devices and issues control commands to the power system based on future generation and power load demand to ensure safe and stable grid-connected operation of the power system and improve the microgrid and its optimal dispatch.

III. Theoretical Foundation

A. Convolutional Neural Network

CNN is widely used to implicit features from the feeding data and reduce the time consumed in computation. 1-D CNN contains a convolution layer, pooling layer, and fully connected layer. The convolutional layer contains multifilters and activation functions. The multifilters are used to implicit information from the input data and the activation functions are used to filter the useless information. After the data is processed through the convolution layer, the pooling layer performs feature extraction based on the feature matrix to reduce the data dimensionality. Finally, the data matrix obtained from the pooling layer optimization is passed through the fully connected layer to obtain the output vector. For a time-series dataset x_i, the calculation procedures of 1-D CNN are shown as follows:

$$c_i = x_i \otimes w_i + \alpha_i \tag{1}$$

$$f(x) = \max(0, x) \tag{2}$$

$$\gamma(c_{i-1}, c_i) = \max(f(c_{i-1}), f(c_i)) \tag{3}$$

$$\delta_i = \gamma(c_{i-1}, c_i) + \beta_i \tag{4}$$

$$y_i = f(\delta_i \otimes k_i + \varsigma_i) \tag{5}$$

where x_i denotes the input of convolution, c_i denotes the output of the ith feature map, the $f(\cdot)$ function denotes the relu activation function, the $\gamma(\cdot)$ function denotes the maximum pooling subsampling function, w_i and δ_i denote the weight matrices of the convolution and fully connected layers, respectively, and α_i, β_i, and ς_i denote the bias of the convolution, pooling, and fully connected layers, respectively.

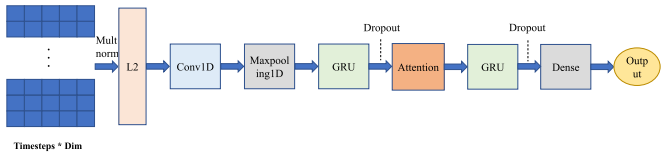

Fig. 2. Network structure of CNN-based GRU with attention mechanism.

B. Gated Recurrent Unit

The GRU model replaces the forgetting and input gates of the LSTM with update gates and adds cell states and hidden layers. The GRU model maintains the LSTM effect while making the structure simpler and less computationally intensive. The specific calculation procedure for each cell of GRU model is shown as follows.

1) *Reset gate:* It is used to calculate how much of the previous moment's state H_{t-1} information is ignored

$$R_t = \sigma(U_{xr} \cdot x_t + V_{hr} \cdot H_{t-1} + b_r) \tag{6}$$

where U_{xr} and V_{hr} denote the weights of the reset gate for the input x_t and the previous moment state H_t, respectively, and b_r denotes the bias term.

2) *Update gate:* It is used to calculate how much of the previous moment's state H_{t-1} information is taken into account in the current state H_t information

$$Z_t = \sigma(U_{xz} \cdot x_t + V_{hz} \cdot H_{t-1} + b_z) \tag{7}$$

where U_{xz} and V_{hz} denote the weights of the update gate for the input x_t and the previous moment state H_t, respectively, and b_z denotes the bias term.

3) *Candidate hidden layer:* The candidate hidden layer is proposed to control how much of the current input x_t and the information ignored by R_t is taken into account in the current cell computation

$$\tilde{H}_t = \tanh(U_{xh} \cdot x_t + V_{hh} \cdot (R_t * H_{t-1}) + b_h) \tag{8}$$

where U_{xh} and V_{hh} denote the weights of the update gate for the input x_t and the previous moment state H_t, respectively, $\tanh(\cdot)$ denotes the tanh activation function, and b_h denotes the bias term.

4) *Hidden state:* Finally, the calculation of the update gate Z_t for the previous moment hidden state H_{t-1} and the current moment candidate hidden state is implemented

$$H_t = Z_t * H_{t-1} + (1 - Z_t) * \tilde{H}_t. \tag{9}$$

C. Wind Energy Hybrid Model

Fig. 2 shows the detail of the architecture of the wind energy hybrid model, which contains L2-norm, Conv1D, Maxpooling1D, attention layer, and GRU. L2 regularization is used to constrain the weights of the neural network to avoid overfitting. The loss function of the model after L2 regularization is shown as follows:

$$L(w) = \arg\min_{w} \sum_{i=1}^{N} (T(x_i) - w_i x_i)^2 + \lambda \sum_{i=1}^{N} {w_i}^2 \tag{10}$$

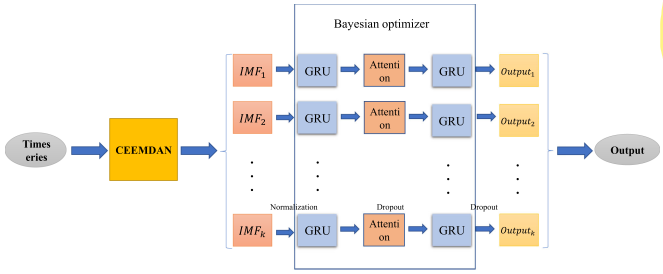

Fig. 3. Network structure of CEEMDAN-based GRU with Bayesian operation.

where $T(\cdot)$ function denotes the objective function, w_i denotes the weight matrix, and x_i denotes the input.

After the above steps, the time series data is fed to the Conv1D layer. The output information needs to be treated by the rectified linear activation function (relu). But the relu activation function can only pass numbers greater than zero, which will cause information loss and reduce the prediction accuracy. Therefore, we chose the exponential linear unit (elu) activation function instead of the relu activation function in the traditional CNN model. The mathematical expression of the elu activation function is shown as follows:

$$E(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \le 0 \end{cases} \tag{11}$$

where α is the constant. It can be clearly seen that the elu activation function has a negative saturation region compared to the relu activation function; so it has some robustness to noise.

As shown in Fig. 2, the GRU is used to filter the information of the input information x. We use the attention mechanism to filter x to give enough attention to the useful information before the last layer of the GRU. The useless information will be suppressed, and the accuracy of the hybrid model will be enhanced. The output after the first GRU layer is $Y = [y_1, y_2, y_3, \ldots, y_m]$. The attention layer first calculates the weights of the input information as shown in the following:

$$e_k = \tanh(y_k) \qquad k = 1, 2, \ldots, m \tag{12}$$

$$a_k = \frac{\exp(e_k)}{\sum_{i=1}^{n} \exp(e_i)} \qquad k = 1, 2, \ldots, m \tag{13}$$

where a_k is the attention weight measuring the importance of the kth input of Y. Then, we process the input information according to the weight of attention

$$\widetilde{Y} = (a_1 y_1, a_2 y_2, ..., a_n y_n)^{\mathrm{T}} \tag{14}$$

where $\widetilde{Y}$ denotes the output of the attention layer, which is also the input of the next GRU layer.

D. Electric Load Forecasting Hybrid Model

Electric load data is different from wind energy data in that it changes with obvious periodicity; so this article designs a composite model based on CEEMDAN and GRU with Bayesian optimizer for electricity load forecasting. The structure of the power load forecasting model is shown in Fig. 3, where the

Algorithm 1: Pseudocode of the proposed short-term electric load forecasting hybrid model (CEEMDAN-Attention -GRU).

Input: Electric load dataset
Output: The forecasting demand of the electric load
Begin algorithm: Use the CEEMDAN to decompose the dataset into K IMFs;
for *each IMF* **do**
  Normalize and split IMF into training set and test set;
  Random generate a set of hyperparameters of the model;
  Training the model and predict the result;
  Calculate the MSE of the training set;
  Bayesian optimization is used to obtain a new set of hyperparameters;
  Forecasting the electric load by the trained model using test set;
end
Inverse normalization the forecasting result. Summing the prediction results of all IMFs;
End algorithm

time series data are first decomposed by CEEMDAN. Empirical mode decomposition (EMD) was proposed by the authors of [17], which can be used to decompose nonlinear data. In fact, the mode mixing problem is always encouraged in the application of EMD. To overcome the problem of very similar patterns or amplitude anomalies in EMD, ensemble empirical mode decomposition (EEMD) with Gaussian white noise added before EMD decomposition is proposed. However, EEMD causes new problems, such as reconstructed signals containing residual noise and too large uncorrelated signals causing abnormal variations in the number of models. In [18], the CEEMDAN model is proposed, which reduces the error during the decomposition while speeding up the decomposition; the calculation process of time series data X_t is shown as follows.

1) Reconstruct the original time series data $X(t)$ as follows:

$$X^n(t) = X(t) + \varepsilon_i \omega^n(t) \qquad n = 1, 2, \ldots, k \tag{15}$$

where $X^n(t)$ is the reconstructed data, ε_i is the ratio of the additional noise to the data, and $\omega^i(t)$ is the Gaussian white noise series.

2) EMD is used to decompose the reconstructed signal X^n, and then the first intrinsic mode function (IMF) $\widetilde{\mathrm{IMF}_1}(t)$ is calculated by averaging the components of EMD

$$\widetilde{\mathrm{IMF}_1}(t) = \frac{1}{I} \sum_{k=1}^{I} \mathrm{IMF}_k(t). \tag{16}$$

3) The first residual $R_1(t)$ and the IMF $\widetilde{\mathrm{IMF}_2}(n)$ can be calculated as follows:

$$R_1(t) = X(t) - \widetilde{\mathrm{IMF}_1}(t). \tag{17}$$

$$\widetilde{\mathrm{IMF}_2}(t) = \frac{1}{I} \sum_{k=1}^{I} \mathrm{EMD}_1(R_1(t) + \varepsilon_1 \mathrm{EMD}_1(\omega^i(t))) \tag{18}$$

where $\mathrm{EMD}_k(\cdot)$ denotes the $k-th$ IMF mode calculated by EMD.

4) Then the k-th residual and the $(k+1)$-th component can be calculated in the following equations:

$$R_k(t) = R_{k-1}(t) - \widetilde{IMF_k}(t) \tag{19}$$

$$\widetilde{IMF_{k+1}}(t) = \frac{1}{I}\sum_{k=1}^{I} MED_1(r_k(t) + \varepsilon_k \mathrm{EMD}_k(\omega^i(n))) \tag{20}$$

where $\mathrm{EMD}_k(t)$ is the k-th mode of EMD, ε_i is the ratio of the additional noise to the data in the $k+1$ stage of CEEMDAN.

5) Finally, the original time series data X_t can be decomposed as following:

$$X_t = \sum_{m=1}^{M} \widetilde{IMF_m}(t) + R(t) \tag{21}$$

where $R(t)$ is the final residual.

The data characteristics are different between the IMFs obtained from the decomposition of the power load data by CEEMDAN. Therefore, each IMF is fed into a deep learning model based attention mechanism and GRU with different parameters. Bayesian optimizer is used to tune the hyperparameters of the model, such as units of GRU, learning rate, and dropout rate. The detailed procedure of the proposed method is represented in Algorithm 1.

E. Modification of Parameter Optimization Algorithm

The Adam optimizer is a currently widely used optimization algorithm in deep learning models, which applies the first-order momentum and second-order momentum in the stochastic gradient down to the weight update.

First-order momentum is defined as follows:

$$m_t = \eta_1 \cdot m_{t-1} + (1-\eta_1)\cdot g_t. \tag{22}$$

The second-order momentum is defined as follows:

$$V_t = \eta_2 \cdot V_{t-1} + (1-\eta_2)\cdot {g_t}^2. \tag{23}$$

The Adam parameter optimization algorithm is defined as follows:

$$\begin{aligned}\varpi_{t+1} &= \varpi_t - \mathrm{lr}.m_t/\sqrt{V_t} \\ &= \varpi_t - \frac{\mathrm{lr}.(\eta_1 \cdot m_{t-1} + (1-\eta_1)\cdot g_t)}{\sqrt{\eta_2 \cdot V_{t-1} + (1-\eta_2)\cdot {g_t}^2}}\end{aligned} \tag{24}$$

where ϖ_t denotes the current weights, g_t denotes the gradient of the current parameter, lr is the learning rate, and η_1 and η_2 are momentum coefficients and are close to 1.

During the training of many deep learning models in this article, the Adam optimizer enables the training time of the models to be reduced. By the time the experiment is in the several tens of epochs, the learning rate is already too small and the change in loss is very small. Therefore, it is hard to reach the optimal result. From (24), we can find that the learning rate is mainly controlled by the second-order momentum. To solve the above problem, we introduce a new control strategy as follows:

$$\varpi_{t+1} = \varpi_t - \frac{\mathrm{lr}.e^{V_{t-1}-V_t}.(\eta_1 \cdot m_{t-1} + (1-\eta_1)\cdot g_t)}{\sqrt{\eta_2 \cdot V_{t-1} + (1-\eta_2)\cdot {g_t}^2}}. \tag{25}$$

When the learning rate decays too much (little) due to the last iteration update, this time, the learning rate will be increased (decreased) to avoid overfitting during the pretraining period.

IV. Experimental Evaluation

A. Experimental Environment and Evaluation Metrics

The load framework proposed in this article includes a wind prediction model and an electric load model. In this section, wind generation prediction experiments and electric load prediction experiments are designed to verify the effectiveness of the proposed framework. Four indexes are selected to assess the results of the hybrid models: mean absolute error (MAE), root mean square error (RMSE), *R*-square (R^2), and MAPE. These indexes are defined as follows.

1) MAE is defined as follows:

$$\mathrm{MAE} = \frac{1}{M}\sum_{i=0}^{M}|y_i - \hat{y}_i|. \tag{26}$$

2) RMSE is defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{M}\sum\nolimits_{i=0}^{M}|y_i - \hat{y}_i|^2}. \tag{27}$$

3) R^2 is defined as follows:

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=0}^{M}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{M}(\overline{y_i} - \hat{y}_i)^2}. \tag{28}$$

4) MAPE is defined as follows:

$$\mathrm{MAPE} = \frac{100}{M}\sum_{i=0}^{M}\left|\frac{y_i - \hat{y}_i}{\hat{y}_i}\right|. \tag{29}$$

Here, M represents the number of data series, y_i and $\hat{y}_i$ denote the prediction and real values, and $\overline{y_i}$ is the mean of y_i.

B. Case Study on Wind Power Generation Data From Liaoning Province, China

1) Data Description and Experimental Parameters Setting: In order to evaluate the effectiveness and practicality of the proposed model, the dataset is provided by a wind power plant in Liaoning Province, China. The original wind generation data are sampled every 15 min for a total of 35 710 (from October 1, 2019 to October 31, 2020), including weather information like humidity, temperature, air pressure, and air density. Due to the cold weather in Liaoning Province, some wind generation data are lost. Figs. 4 and 5 illustrate the actual wind generation and the weather variables of the whole dataset. The statistical information for the dataset including maximum value (Max), minimum value (Min), and mean value (Mean) is listed in Table I.

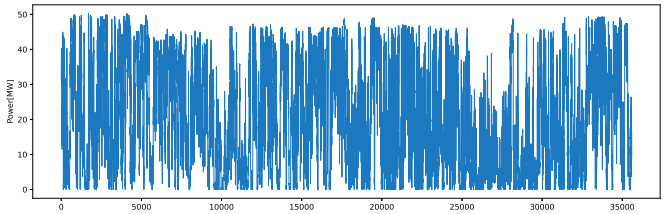

Fig. 4. Wind generation: October 1, 2019 to October 31, 2020.

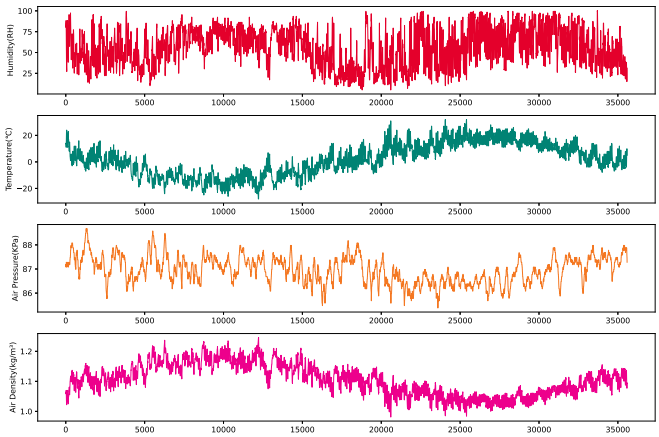

Fig. 5. Actual values of the four weather input variables.

TABLE I
STATISTICAL INFORMATION FOR THE WIND GENERATION AND WEATHER VARIABLES

Data	Max	Min	Mean
Generation(MW)	50.23	0.0	20.23
Humidity(RH)	100.5	5.00	56.27
Temperature(°C)	32.00	-28.0	1.52
Air pressure(KPa)	88.68	85.38	86.96
Air density(kg/m^3)	1.246	0.981	1.106

Scatter plots are often used to reflect relationships between variables, but hexbin plots are more informative than scatter plots. Therefore, we use the hexbin plot to visualize the relationship between weather variables and wind power generation. Fig. 6 shows the hexbin plot of weather variables and wind power generation. It can be seen from the figure that the weather variables and wind generation have a certain correlation, exceptional temperature and air density. It may be that there is a calculated relationship between temperature and air pressure and air volume.

Each weather variable contains 35710 time-series data points, which are denoted as $\{x_t{}^{(1)}\}_{t=1}^{t=35710}$, $\{x_t{}^{(2)}\}_{t=1}^{t=35710}$, $\{x_t{}^{(3)}\}_{t=1}^{t=35710}$, $\{x_t{}^{(4)}\}_{t=1}^{t=35710}$, respectively. On the other hand, the wind generation is denoted as $\{x_t{}^{(5)}\}_{t=1}^{t=35710}$. Here, the sample is set as the following equation:

$$f(\{x_i{}^{(n)}\}_{t=i}^{t=i+\text{timestep}}) \rightarrow \{y_i = x_{i+\text{timestep}+1}{}^{(5)}\} \tag{30}$$

where $i \in [1, 2, 3, \ldots, 35710 - \text{timestep}]$, $n \in [1, 2, 3, 4, 5]$, $f(\cdot)$ denotes the model, y_t denotes the forecasting result, and timestep denotes the number of moments of data before to predict the next moment.

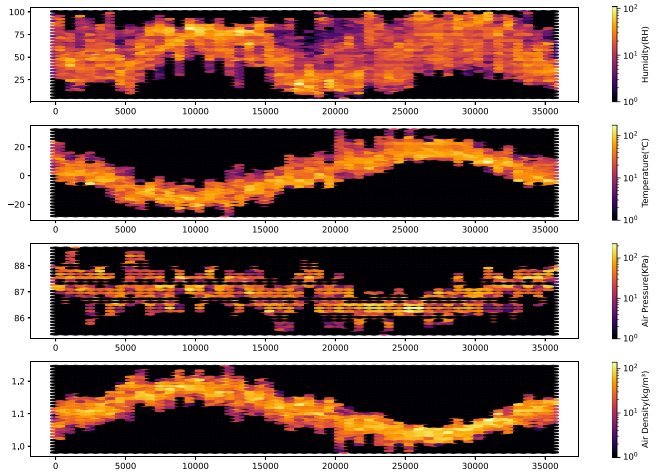

Fig. 6. Hexbin plot of the four weather variables and wind generation.

Using the same data for model generation has a high risk of overfitting [19]. However, the model with too little data may not be able to tune the best hyperparameters. Therefore, we introduce the *K*-fold cross-validation method by randomly dividing the dataset into four and randomly selecting three as the training set and the other as the test set. The total size of the actual input data used as training model is less than 35710 because there is continuous loss of wind power generation data. Although there are missing data cases, the input weather variables and wind power data are consistent in dimensionality after processing by us. Therefore, the results of the experiment do not show any anomalies due to the missing data.

2) Comparison and Analysis of Experimental Results: In this section, the proposed method is compared with LSTM-CNN proposed in [20] (Method 1), stacked GRU-RNN proposed in [21] (Method 2), EA-LSTM proposed in [22] (Method 3), CNN-LSTM proposed in [23] (Method 4), and CNN-GRU without attention mechanism (Method 5), with the proposed method CNN-Attention-GRU (Method 6) to evaluate the effectiveness. The structure of the proposed model is shown in Fig. 2, and the specific structure of the deep learning neural network is "Dense-CNN-MaxPooling1D-GRU-Dropout-Attention-GRU-Dropout-Dense," where the number of units in each layer is "128-128-120-140-1," respectively. The first dense layer contains the L2 regularization function, and the activation function chosen in the model is the elu activation function. The initial learning rate and epochs are 0.0004 and 200, respectively. The pooling size is 2 and the dropout rate is 0.3. In conclusion, all hyperparameters are trained by kerastuner many times, and we have also tried to use double-layer CNN function and the prediction results are too poor.

In order to have a fair comparison, the initialized hyperparameters of the methods are based on their corresponding papers in which the authors reported their best obtained results. Furthermore, consider that the different training data need to correspond to different hyperparameters of the methods. In particular, configuring the CNN, LSTM, and GRU networks,

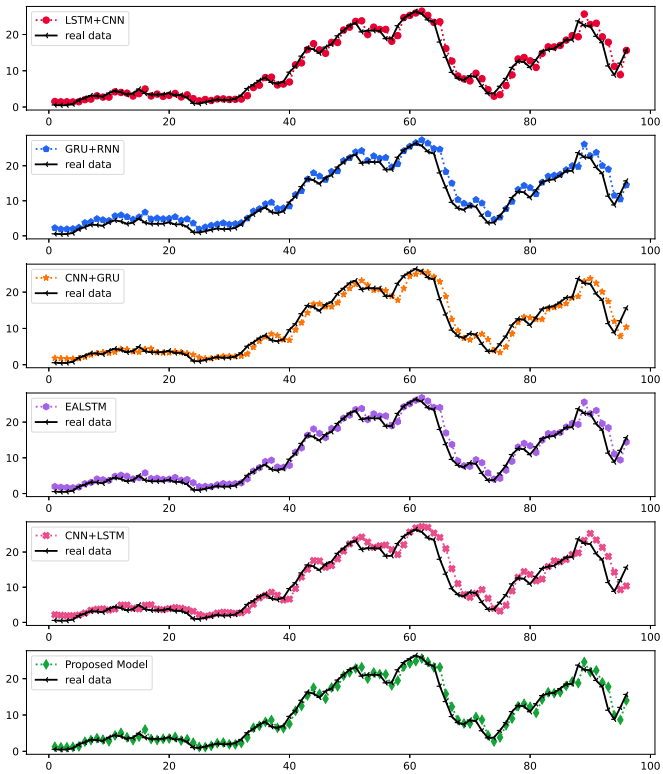

Fig. 7. Predicted results and actual data. The black line represents the actual values.

for the setting considered in their original hyperparameters, we have also trained them with grid search method and the same structure used in the proposed methods. The structure of Method 1 is "LSTM-LSTM-Dense-Conv1D-Conv1D-Dropout-Dense," where the number of neurons is "64-128-128-128-64-1." The activation function of Method 1 is the relu activation function, the learning rate is 0.002, the dropout rate is 0.2, and the epochs are 200. The structure of Method 2 is "GRU-Dropout-GRU-Dense," where the number of neurons is "60-80-1." The learning rate is 0.005, the dropout rate is 0.3, and the epochs are 200. The structure of Method 3 (evolutionary attention-based LSTM) is "LSTM-Dropout-EA-LSTM-Dropout-Dense," where the number of neurons is "64-EA-80-1." The structure of Method 4 is "Dense-Conv1D-Maxpooling1D-LSTM-Dropout-LSTM-Dropout-Dense," where the number of neurons is "128-128-80-100-1." The structure of Method 5 is the same with the proposed method just without attention mechanism. Finally, the best experimental results are obtained after extensive training with the $timestep$ of 100. Therefore, the $timestep$ of all methods is set as 100.

Fig. 7 depicts the prediction results of wind power generation for the next three days using the proposed method and the other five methods, respectively. In Table II, the bolded values are the evaluated values of the proposed method, which indicates the good predictive performance of the proposed model. The experimental results demonstrate that the forecasting accuracy of the proposed method for wind generation is better than the other methods. In addition, we can find that the prediction accuracy of Method 3 is improved by 16% based on MAE, and

TABLE II
PERFORMANCE EVALUATIONS OF DIFFERENT METHODS FOR THE WIND POWER FORECASTING EXPERIMENT

Methods	RMSE	MAE	R^2	MAPE(%)	Time/s
Method 1	1.65	1.18	0.9586	21.82	129.25
Method 2	2.08	1.58	0.9288	37.17	84.69
Method 3	1.77	1.26	0.9499	26.83	74.48
Method 4	2.3	1.67	0.9196	32.54	88.96
Method 5	1.97	1.43	0.9349	26.09	177.74
Method 6	**1.57**	**1.09**	**0.9597**	**17.99**	**171.66**

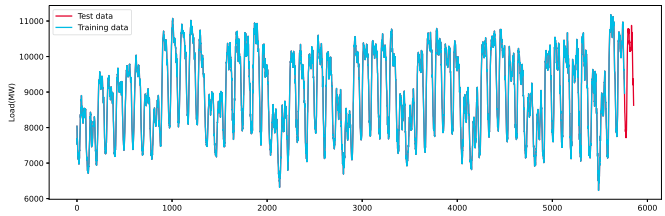

Fig. 8. Actual electric load of Belgain from August 1, 2020 to September 30, 2020.

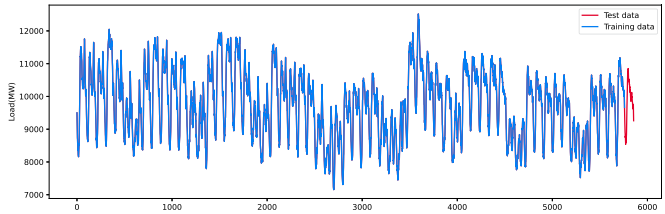

Fig. 9. Actual electric load of Belgain from March 1, 2021 to April 30, 2021.

the reason is that the attention mechanism gives enough weight to the useful information.

C. Case Study on Electric Load Data From Belgain

In this section, the proposed novel CEEMDAN-Attention-GRU is used for short-term load prediction. The dataset used are from January 1, 2020 to May 1, 2021 in Belgain [24]. In order to demonstrate the effectiveness and accuracy of the proposed method, we select recorded data of two different periods. The first one is from August 1, 2020 to September 30, 2020 (total 5856 points), as shown in Fig. 8. The second one is from March 1, 2021 to April 30, 2021 (total 5856 points), as shown in Fig. 9. The last day (96 points) of the two datasets are used to verify the accuracy of the methods, and the rest are used to tune the hyperparameters of the model.

In this section, Method 1, Method 2, Method 4, the VMD-LSTM in [25] (Method 7), and CEEMDAN-GRU (Method 6) without attention mechanism are selected to compare with CEEMDAN-Attention-GRU (Method 7). As shown in Fig. 10, different IMFs have different characteristics of variation. It can be clearly found that the IMFs have a small range of values for the complex part of the variation, while the data with the main characteristics of the original data vary simply. Therefore, the error arising from prediction on this basis is less. And the timestep of each IMF prediction model is set as a different value

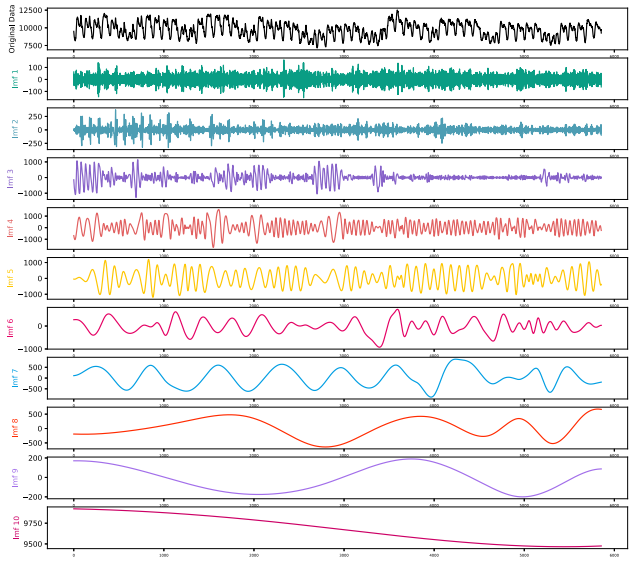

Fig. 10. IMFs of the first dataset decomposed by CEEMDAN.

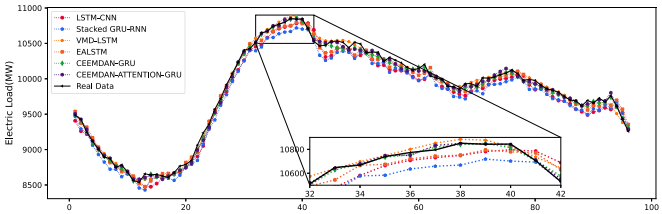

Fig. 11. Prediction results of different methods on the first electric load dataset.

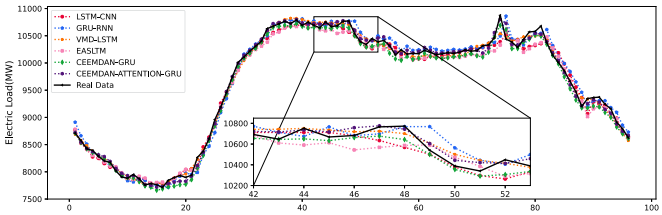

Fig. 12. Prediction results of different methods on the second electric load dataset.

TABLE III
PERFORMANCE EVALUATIONS OF DIFFERENT METHODS FOR THE FIRST ELECTRIC LOAD DATASET

Methods	RMSE	MAE	R	MAPE	Time/s
Method 1	104.46	87.37	0.9731	8.84	36.73
Method 2	123.22	107.18	0.9614	10.83	84.69
Method 4	70.85	57.24	0.9875	5.83	29.679
Method 7	45.69	35.12	0.9951	3.59	846.8
Method 8	40.25	32.06	0.9961	3.26	256.43
Method 9	**34.59**	**28.19**	**0.9972**	**2.88**	**535.1**

in order to achieve high prediction accuracy. After extensive experiments, Methods 1 and 2 have the highest prediction accuracy when the timestep is set to 60.

Figs. 11 and 12 show the prediction results of the five methods used for the first and second test datasets, respectively. Tables III and IV list the values of the four error indicators including RMSE, MAE, R^2 and MAPE, where the bold values mean the best performance of all models. The prediction error of the proposed model is the smallest in the two experiments, where R^2 indicates the correlation between the prediction results and the real data. The comparison results demonstrate the superiority of the proposed method for short-term electricity load.

TABLE IV
PERFORMANCE EVALUATIONS OF DIFFERENT METHODS FOR THE SECOND ELECTRIC LOAD DATASET

Methods	RMSE	MAE	R	MAPE	Time/s
Method 1	146.36	113.37	0.9754	11.85	86.43
Method 2	120.45	91.78	0.9867	9.54	84.69
Method 4	131.77	101.03	0.9819	10.26	33.65
Method 7	74.3	56.36	0.9947	5.79	800
Method 8	78	59.99	0.9942	6.3	261.79
Method 9	**61.83**	**46.48**	**0.9965**	**4.86**	**402.96**

V. CONCLUSION

In this article, a novel system framework based on edge-AI was proposed to balance power supply and demand to improve the overall efficiency of the smart microgrid supply chain. In the section on wind power generation prediction, the weather index was considered as the relevant factors, which was fed to the CNN-Attention-GRU model as an input variation. Attention mechanism was used in this work and it improved the accuracy of the prediction results. In the section on electric load prediction, CEEMDAN model was used to decompose the time series data which reduced the difficulty in prediction. Bayesian optimizer based kerastuner was used to tune the hyperparameters of the CNN-Attention-GRU which improved the prediction accuracy of the model. The results on real-world wind generation and electric load prediction indicated the superiority of the proposed framework in improving the accuracy. Future work could focus on how to improve the robustness of energy management systems and apply them to larger power networks.

REFERENCES

[1] V. A. Memos, K. E. Psannis, Y. Ishibashi, B.-G. Kim, and B. Gupta, "An efficient algorithm for media-based surveillance system (eamsus) in IoT smart city framework," *Future Gener. Comput. Syst.*, vol. 83, pp. 619–628, 2018.

[2] H. Fatemidokht, M. K. Rafsanjani, B. B. Gupta, and C.-H. Hsu, "Efficient and secure routing protocol based on artificial intelligence algorithms with UAV-assisted for vehicular ad hoc networks in intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4757–4769, Jul. 2021.

[3] A. Mishra, N. Gupta, and B. Gupta, "Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller," *Telecommun. Syst.*, vol. 77, no. 1, pp. 47–62, 2021.

[4] Z. Zhou *et al.*, "Coverless information hiding based on probability graph learning for secure communication in IoT environment," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2021.3103779.

[5] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proc. IEEE*, vol. 107, no. 8, pp. 1655–1674, Aug. 2019.

[6] Z. Tian *et al.*, "Real-time lateral movement detection based on evidence reasoning network for edge computing environment," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4285–4294, Jul. 2019.

[7] Z. Tian, Y. Wang, Y. Sun, and J. Qiu, "Location privacy challenges in mobile edge computing: Classification and exploration," *IEEE Netw.*, vol. 34, no. 2, pp. 52–56, Mar./Apr. 2020.
[8] A. Tewari and B. B. Gupta, "Secure timestamp-based mutual authentication protocol for IoT devices using RFID tags," *Int. J. Semantic Web Inf. Syst.*, vol. 16, no. 3, pp. 20–34, 2020.
[9] S. R. Sahoo and B. Gupta, "Multiple features based approach for automatic fake news detection on social networks using deep learning," *Appl. Soft Comput.*, vol. 100, 2021, Art. no. 106983.
[10] R. Jiao, S. Wang, T. Zhang, H. Lu, H. He, and B. B. Gupta, "Adaptive feature selection and construction for day-ahead load forecasting use deep learning method," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 4, pp. 4019–4029, Dec. 2021.
[11] Y. Du, Y. Liu, X. Wang, J. Fang, G. Sheng, and X. Jiang, "Predicting weather-related failure risk in distribution systems using Bayesian neural network," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 350–360, Jan. 2021.
[12] P. Jiang, R. Li, N. Liu, and Y. Gao, "A novel composite electricity demand forecasting framework by data processing and optimized support vector machine," *Appl. Energy*, vol. 260, 2020, Art. no. 114243.
[13] P. Kumari and D. Toshniwal, "Long short term memory convolutional neural network based deep hybrid approach for solar irradiance forecasting," *Appl. Energy*, vol. 295, 2021, Art. no. 117061.
[14] A. Kisvari, Z. Lin, and X. Liu, "Wind power forecasting—A data-driven method along with gated recurrent neural network," *Renewable Energy*, vol. 163, pp. 1895–1909, 2021.
[15] L. Lv, Z. Wu, J. Zhang, Z. Tan, L. Zhang, and Z. Tian, "A VMD and LSTM based hybrid model of load forecasting for power grid security," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2021.3130237.
[16] Z. Liu, P. Jiang, L. Zhang, and X. Niu, "A combined forecasting model for time series: Application to short-term wind speed forecasting," *Appl. Energy*, vol. 259, 2020, Art. no. 114137.
[17] P. Flandrin, G. Rilling, and P. Goncalves, "Empirical mode decomposition as a filter bank," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 112–114, Feb. 2004.
[18] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, "A complete ensemble empirical mode decomposition with adaptive noise," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2011, pp. 4144–4147.
[19] S. Zhang, Y. Wang, Y. Zhang, D. Wang, and N. Zhang, "Load probability density forecasting by transforming and combining quantile forecasts," *Appl. Energy*, vol. 277, 2020, Art. no. 115600.
[20] K. Wang, X. Qi, and H. Liu, "Photovoltaic power forecasting based LSTM-convolutional network," *Energy*, vol. 189, 2019, Art. no. 116225.
[21] M. Xia, H. Shao, X. Ma, and C. W. de Silva, "A stacked GRU-RNN-based approach for predicting renewable energy and electricity load for smart grid operation," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 7050–7059, Oct. 2021.
[22] Y. Li, Z. Zhu, D. Kong, H. Han, and Y. Zhao, "EA-LSTM: Evolutionary attention-based LSTM for time series prediction," *Knowl. Based Syst.*, vol. 181, 2019, Art. no. 104785.
[23] K. Wang, X. Qi, and H. Liu, "A comparison of day-ahead photovoltaic power forecasting models based on deep learning neural network," *Appl. Energy*, vol. 251, 2019, Art. no. 113315.
[24] "Belgian open power system data." Accessed: Aug. 17, 2020. [Online]. Available: https://www.elia.be/en/grid-data/data-download-page
[25] F. He, J. Zhou, Z.-K. Feng, G. Liu, and Y. Yang, "A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm," *Appl. Energy*, vol. 237, pp. 103–116, 2019.