IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 15, 2020, accepted November 2, 2020, date of publication November 9, 2020, date of current version November 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3036885

Missing-Insensitive Short-Term Load Forecasting Leveraging Autoencoder and LSTM

KYUNGNAM PARK[1], JAEIK JEONG[1], DONGJOO KIM[2], AND HONGSEOK KIM[1], (Senior Member, IEEE)

[1]Department of Electronic Engineering, Sogang University, Seoul 04107, South Korea
[2]Smart Power Distribution Laboratory, Korea Power Electric Corporation (KEPCO) Research Institute, Daejeon 34056, South Korea

Corresponding author: Hongseok Kim (hongseok@sogang.ac.kr)

This work was supported in part by the Smart City Research and Development project of the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure, and Transport under Grant 19NSPS-B152996-02, in part by the Korea Institute of Energy Technology Evaluation and Planning (KETEP), and in part by the Ministry of Trade, Industry and Energy (MOTIE), South Korea, under Grant 20192010107290.

ABSTRACT In most deep learning-based load forecasting, an intact dataset is required. Since many real-world datasets contain missing values for various reasons, missing imputation using deep learning is actively studied. However, missing imputation and load forecasting have been considered independently so far. In this article, we provide a deep learning framework that jointly considers missing imputation and load forecasting. We consider a family of autoencoder/long short-term memory (LSTM) combined models for missing-insensitive load forecasting. Specifically, autoencoder (AE), denoising autoencoder (DAE), convolutional autoencoder (CAE), and denoising convolutional autoencoder (DCAE) are considered for extracting features, of which the encoded outputs are fed into the input of LSTM. Our experiments show that the proposed DCAE/LSTM combined model significantly improves forecasting accuracy no matter what missing rate or type (random missing, consecutive block missing) occurs compared to the baseline LSTM.

INDEX TERMS Deep learning, short-term load forecasting, missing data imputation, feature extraction.

I. INTRODUCTION

Load forecasting is essential to balance power supply and demand. Long-term load forecasting supports power system infrastructure planning while medium-term and short-term load forecastings are used for power system operation. Unlike system-level electrical load, power consumption of a single user is highly volatile, and thus it is challenging to accurately predict the electrical load of a single user. Highly accurate load forecasting can be flexibly combined with intelligent demand response or energy storage system to achieve peak load shaving.

Short-term load forecasting can be categorized into three cases: statistical methods, similar day-based methods, and artificial intelligence-based methods. Statistical methods are mostly based on finding a linear relationship between inputs and outputs; multiple linear regression [1], exponential smoothing [2], autoregressive integrated moving average (ARIMA) [3], and Kalman filtering [4] fall into this category. Similar day methods are based on searching for daily historical data similar to the day of forecasting considering weather conditions, but this method alone does not provide high forecasting accuracy and is used in combination with artificial intelligence model [5].

Artificial intelligence techniques are widely used for short-term load forecasting, such as artificial neural network (ANN) [6], support vector machine (SVM) [7], extreme learning machine [8], Bayesian neural network [9], deep neural network [10] and recurrent neural network (RNN) [11]. In [12], long short-term memory (LSTM) is used to solve the vanishing gradient problem of RNN and outperforms other neural network methods. In addition, combining ResNet and LSTM was proposed for short-term load forecasting [13]. However, in practice, data can be lost because of communications error, mechanical failure or loss of power [14], and missing imputation has become essential. x So far the methods for dealing with missing data can be categorized by a) deletion, b) full analysis, and c) imputation [15].

The associate editor coordinating the review of this manuscript and approving it for publication was Thomas Canhao Xu.

Deletion is to simply discard missing values from the dataset. However, this method reduces the size of valid data and cannot be used for typical forecasting models that require a complete intact data set. The full analysis method is ''do not impute (DNI)''. As the name suggests, all the missing data remain unreplaced, so the networks must use their default missing values. However, DNI is not a preferred preprocessing method for load forecasting since missing can be recovered by capturing the relationship of load data. Hence we are interested in the third one, missing imputation. The imputation method is most often used in the missing process, and the missing values are recovered by the best possible estimates.

The methods of missing imputation can be categorized by a) linear interpolation, b) historical average, c) deep learning-based [16]. First, the linear interpolation method replaces missing values with an average of measured values, which occur before and after missing. However, if missing occurs consecutively for a long time, linear interpolation may not be effective. Second, the historical average method replaces missing values with hour-ahead, day-ahead, or week-ahead metering data [18]. However, this method requires additional clustering or classification algorithms to find similar historical patterns. Furthermore, customer-level electrical loads are typically random and volatile, which makes missing imputation as difficult as forecasting future loads.

Finally, deep learning-based imputation method has been studied for various fields such as medical data [19], biological data [20], and traffic data [21] using multi-layer perceptron (MLP) [22], K-nearest neighbours (KNN) [23], self organising maps (SOM) [24] and autoencoder (AE) [25]. In the case of smart grid applications, denoising autoencoder is utilized for missing imputation of power system monitoring data [26]. In this case, learning models are created by separating the missing imputation part and the application part that takes the imputed data as an input, such as forecasting or clustering. However, in case missing imputation is performed separately, it can be cumbersome because it has to be processed once in the data itself and may deviate from the characteristics of the original data.

In this article, we propose a novel forecasting method that does not require explicit missing imputation. Our model aims at guaranteeing high forecasting accuracy even with random or block missing data. In doing this, we leverage the unsupervised learning capability of autonencoder and the feature extraction of convolutional neural network. Our intuition is that an autoencoder extracts important attributes, which can be used as an input of a forecasting model. The decoder trained from the autoencoder is discarded, and the encoder and the forecasting model are combined to act as one model. Thus, even though there are some missing values in time domain, the encoded features extracted by the autoencoder can be *insensitive* to missing values. We consider a family of autoencoders such as vanilla autoencoder (AE), denoising autoencoder (DAE), convolutional autoencoder (CAE), and denoising convolutional autoencoder (DCAE) for feature extraction.

We summarize our key contributions as follows. First, missing imputation and forecasting need not be performed separately. Instead, we propose a unified forecasting technique that is insensitive to missing values. The proposed method achieves accurate forecasting in the presence of severe missing rate, e.g., up to 25%, either random or block missing. Second, using two-dimensional image data and two-dimensional convolution, we extract the features of data even if severe missing occurs. By using two-dimensional convolution, similar time information of different days can be obtained by efficient use of the receptive field. As missing rate increases, performance improvement over the conventional methods becomes more significant. Third, we perform extensive experiments with DNN, LSTM, AE, DAE, CAE, and DCAE. We confirm that the proposed DCAE/LSTM combined model outperforms the conventional models that separately process missing imputation and forecasting.

The rest of this article is organized as follows. Section II describes the overall framework, data preprocessing, and statistical information of missing value. We present the proposed models in Section III and the experimental results in Section IV, followed by the conclusion in Section V.

II. OVERALL FRAMEWORK

A. OVERALL STRUCTURE

Fig. 1 shows the overall process of the proposed load forecasting, divided into three main steps: data preprocessing, training, and test. In the first step, data cleansing is performed, and one-dimensional load time series data undergo min-max normalization to make data in the range [0, 1]. In order to evaluate the performance when missing occurs, we intentionally make missing data. In the cases of AE and DAE, we use one-dimensional time series load data. In the cases of CAE and DCAE, we convert one-dimensional time series load data into two-dimensional load image data. The dataset is then partitioned into training set, validation set, and test set. In the second step, training set is used to train a forecasting model, and validation set is used to determine the hyperparameters of each model or each customer. We perform feature extraction using various autoencoders and use the feature data as input to the forecasting model. In the training and validation sets, the training is carried out with intact data only and the test set contains data with missing pattern similar to real-world. We train intact data only once for each customer and do not train multiple times, no matter what missing patterns (random/block) or various missing rates appear in the test set. In the final step, we evaluate the performance with test set to demonstrate the feasibility of the proposed model.

B. DATA PREPROCESSING

The data used in our work is demand-side load data with 15-minute interval and is provided by Korea Electric Power Corporation (KEPCO). There are industrial customers in seven sectors (mining support service, education service,

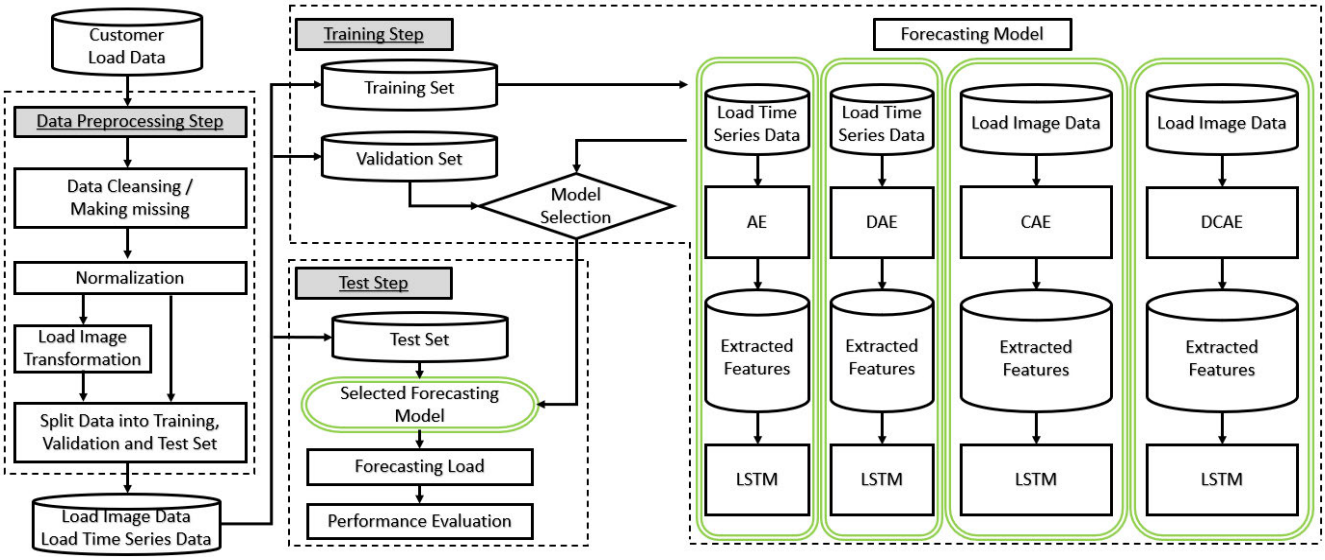


FIGURE 1. Overall process of missing-insensitive load forecasting.

TABLE 1. Customers From Various Industries.

Industry	Customer ID	Peak load (kW)	Average load (kW)
Mining support service	1	564.12	246.17
Education service	2	1,792.80	731.88
Water supply business	3	12,342.40	8,558.95
Water supply business	4	2,933.70	1,843.63
Water supply business	5	1,702.56	1,187.80
Paper products manufacturing	6	164.64	70.20
Information service	7	272.64	165.43
Information service	8	151.68	35.71
Insurance and pensions	9	192.24	35.98
Wooden products manufacturing	10	33.00	3.48

water supply business, paper products manufacturing, information service, insurance and pensions, and wooden products manufacturing), each with 600 days of power usage data. The peak loads of the customers span from 33 kW to 12,342 kW. The detailed information of each customer is shown in Table 1.

Data preprocessing is performed in the order of data cleansing, normalization, and conversion to two-dimensional load image data, if necessary. Before using load data set as experimental data, abnormal values and missing values are replaced by the average of highly correlated data to serve as the *ground-truth* data. After data cleansing is done, we conduct data normalization because each customer has different scales. We use min-max normalization as follows:

$$\bar{l}_t^c = \frac{l_t{}^c - \min(l^c)}{\max(l^c) - \min(l^c)} \tag{1}$$

where $l_t{}^c$ is the load of customer c at timestep t. When presenting the final estimated load results, denormalization is performed. During the test step, the minimum and maximum values of training data are saved in advance and used for denormalization. Because the data also include weekends, we utilize 7 days as input and forecast the next day. We transform one-dimensional time series vector of size 7×96 into two-dimensional load image matrix of size (7, 96) as illustrated in Fig. 2 (a) and (b) when CAE or DCAE is used for feature extraction. For experiment we intentionally make 5%–25% missing occur randomly or block-wise from load data. All points of random missing data and starting points of block missing data are selected uniformly randomly.

C. STATISTICAL INFORMATION OF MISSING VALUES

It is desirable to use data with missing values in real-world, but, in that case, we cannot know the ground-truth value and therefore cannot test the missing imputation performance. Hence we analyze the missing patterns of the real meter data and then synthesize the experimental missing data by referring to the missing patterns and missing rates that actually occur rather than artificially generating the missing data.

Since missing occurs either point-wise or block-wise, we consider the concept of missing point and missing sequence as shown in Fig. 3. The length of missing sequence (l) is the number of consecutive missing points in the sequence ($l > 1$). If $l = 1$, it means one missing point. For example, Fig. 3 has 6 missing points and 3 missing sequences, respectively.

To analyze missing patterns, we observe 1,445 residential customer data in South Korea. The load data set includes 360 days with 15 minute interval, so there are 49,939,200 ($360 \times 96 \times 1445$) points. As shown in Table 2, there are 420k (420,253) missing points (approximately 1%) and 72k (72,412) missing sequences. In the case of missing sequence, the class I ($l = 1$) occupies the second largest portion (47.14%), and the class II ($1 < l \leq 24$) occupies the largest portion in terms of missing sequences (48.61%). Classes I and

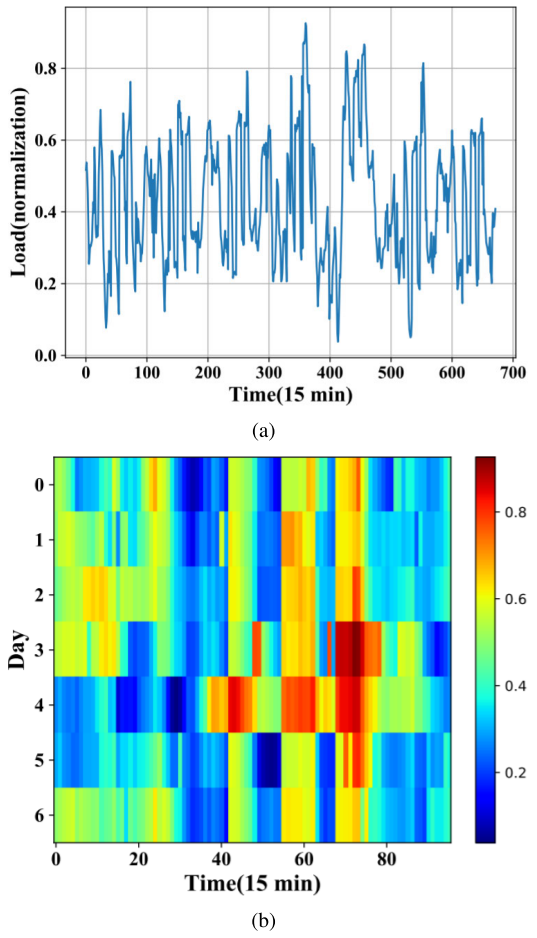


FIGURE 2. Transform of time series data into two-dimensional load image data for the input of CAE and DCAE. (a) time series data (b) image data.

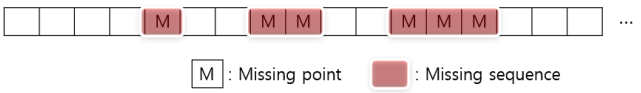


FIGURE 3. Missing points and sequences in time series data.

TABLE 2. Statistics of Missing Values.

Class	l	Missing points		Missing sequences	
		Number	Percent(%)	Number	Percent(%)
I	1	34k	8.12	34k	47.14
II	(1, 24]	185k	44.11	35k	48.61
III	(24, 48]	64k	15.32	2k	2.65
IV	(48, 96]	50k	11.97	760	1.05
V	(96,)	86k	20.48	401	0.55
Total		420k	100	72k	100

II account for most of the missing points and sequences, so we focus on the classes I, II ($1 \leq l \leq 24$) in our experiments, which corresponds to up to 25% of missing in a day.

III. PROPOSED MODELS

In this section we show the structure of each model determined by model selection. Then, we describe four AE-based models: AE/LSTM, DAE/LSTM, CAE/LSTM, and DCAE/LSTM combined models.

A. MODEL SELECTION

Data set of 600 days is split into training set for 420 days (70%), validation set for 90 days (15%), and test set for 90 days (15%). In overall, the hyperparameters include learning rate, the number of iterations, layer configuration, the presence and degree of noise. In the case of (D)CAE, additional hyperparameters are kernel size, the number of strides, dropout ratio, type of pooling, the number of filters in each layer, and encoder output size. In LSTM, we determine the size of the hidden unit vector, sequence length, and the number of LSTM cells. We select the hyperparameters through validation set. All customers have their own hyperparameters separately.

To determine the hyperparameters of each model, mean absolute percentage error (MAPE) is used where accuracy is defined as percentage as follows:

$$MAPE(\%) = \frac{100}{N} \sum_{t=1}^{N} \frac{|y_t - \hat{y}_t|}{y_t} \tag{2}$$

where y_t is the desired value, $\hat{y}_t$ is the forecasted value and N is the number of samples. MAPE is used to solve scale-dependent errors of mean absolute error (MAE) or mean squared error (MSE), and the performance in load forecasting can be verified by comparing MAPE [27]. All frameworks use tensorflow [28] and adaptive moment estimation (Adam) [29] for optimizer.

The selection process of four considered AE-based models, i.e., (D)AE/LSTM layer configuration and (D)CAE/LSTM layer configuration are shown in Table 3 and Table 4 for customer 1 as an example. Hyperparameters for other customers are determined in a similar way. In the layer configuration, convolutional section is the number of filters @ filter size, and if filter size is n, it actually means $n \times n$. In Table 3 and Table 4, if noise is zero, it is AE/LSTM or CAE/LSTM, and adding noise to the input is DAE/LSTM or DCAE/LSTM. All hyperparameters are determined by comparing validation MAPE.

B. AUTOENCODER/LSTM COMBINED MODELS

We first consider a model that jointly exploits the feature extraction of AE and the forecasting of LSTM. AE is one of the unsupervised learning methods for neural networks where only the input is learned to identify the features of the data. AE does not simply copy the input directly to the output but controls to learn how to efficiently represent the data; the encoder network learns the compressed representation of the input, from which the decoder network reconstructs the input. By making primary training with AE (only real data is included in the objective function), it can go beyond the limits of the objective function, which is configured to extract features only in a direction that reduces the difference between the value being predicted and the real value. The

TABLE 3. Validation Error (MAPE) in Determining the (D)AE/LSTM Layer Configuration. (Customer 1).

Layer configuration	Noise	Model complexity	Average MAPE (%)
64	0	43,072	37.64
64	0.3	43,072	38.83
64	0.5	43,072	47.13
100	0	67,300	36.76
100	0.3	67,300	38.52
100	0.5	67,300	40.98
200, 100	0	154,700	36.45
200, 100	0.3	154,700	34.47
200, 100	0.5	154,700	34.72
300, 100	0	232,000	35.07
300, 100	0.3	232,000	33.44
300, 100	0.5	232,000	34.60
300, 200, 100	0	282,200	34.77
300, 200, 100	0.3	282,200	33.77
300, 200, 100	0.5	282,200	33.63
500, 300, 100	0	516,900	34.15
500, 300, 100	**0.3**	**516,900**	**32.31**
500, 300, 100	0.5	516,900	34.04

TABLE 4. Validation Error (MAPE) in Determining the (D)CAE/LSTM Layer Configuration. (Customer 1).

Layout	Layer configuration		Noise	Model complexity	Average MAPE (%)
	Convolutional	Fully-connected			
1/1	4@3	100	0	76,940	37.66
1/1	4@3	100	0.3	76,940	36.51
1/1	4@3	100	0.5	76,940	37.28
1/1	5@3	100	0	96,150	34.53
1/1	5@3	100	0.3	96,150	34.20
1/1	5@3	100	0.5	96,150	34.98
2/1	4@3 16@3	100	0	77,532	46.76
2/1	4@3 16@3	100	0.3	77,532	44.55
2/1	4@3 16@3	100	0.5	77,532	45.41
2/1	5@3 25@3	100	0	121,300	32.09
2/1	5@3 25@3	100	0.3	121,300	31.77
2/1	5@3 25@3	100	0.5	121,300	31.73
2/2	4@3 16@3	300, 100	0	261,432	38.27
2/2	4@3 16@3	300, 100	0.3	261,432	37.65
2/2	4@3 16@3	300, 100	0.5	261,432	37.94
2/2	5@3 25@3	300, 100	0	391,600	38.06
2/2	5@3 25@3	300, 100	0.3	391,600	37.25
2/2	5@3 25@3	300, 100	0.5	391,600	37.54
3/1	4@3 16@3 64@3	100	0	86,812	32.42
3/1	4@3 16@3 64@3	100	0.3	86,812	32.30
3/1	4@3 16@3 64@3	100	0.5	86,812	32.89
3/1	5@3 25@3 125@3	100	0	179,550	32.63
3/1	5@3 25@3 125@3	100	0.3	179,550	31.87
3/1	5@3 25@3 125@3	100	0.5	179,550	32.10
3/2	4@3 16@3 64@3	500, 100	0	444,512	33.57
3/2	4@3 16@3 64@3	500, 100	0.3	444,512	31.85
3/2	4@3 16@3 64@3	500, 100	0.5	444,512	32.44
3/2	5@3 25@3 125@3	500, 100	0	830,050	31.48
3/2	**5@3 25@3 125@3**	**500, 100**	**0.3**	**830,050**	**30.90**
3/2	5@3 25@3 125@3	500, 100	0.5	830,050	31.25

structure of the AE/LSTM combined model is determined through model selection as follows. The encoder consists of three layers: 7 days one-dimensional load (1×672) data are converted to 500 one-dimensional data in the first layer and 300 one-dimensional data in the second layer and 100 one-dimensional data in the third layer. Then, the output is reshaped by (4, 25) to be applied to the input of LSTM. We construct LSTM using four cells to forecast the next one day (1, 96). In the case of denoising, DAE is an autoencoder with denoising capabilities and takes a partially corrupted input instead of the original input, usually by Gaussian noise. Thus, DAE achieves robustness to partial destruction of input by learning common latent representations of the original and corrupted data. In addition, the network is trained to restore correct data from missing state values. In our DAE model, Gaussian noise is added to the input for denoising training, and the remaining structure is the same as AE.

C. CONVOLUTIONAL AUTOENCODER/LSTM COMBINED MODELS

Next, we describe why we use CAE, not AE, and propose the (D)CAE/LSTM models. We transform the time series data into image data and make it possible to learn the features of the time by using information of similar time zones on different days using two-dimensional convolution. In addition, the signal entering the input of CAE is visualized because it is two-dimensional stacking by 7 days. In experimental results, we will see that distribution learning about the time axis is well done when the time series data is put into the CAE as an image. The structure of the (D)CAE/LSTM combined model is determined through model selection and is as follows. As shown in Fig. 4, Gaussian noise is added to the input for denoising training. In the case of CAE/LSTM, the structure is the same as DCAE/LSTM, but it does not have the denoising part. The (D)CAE/LSTM combined model has the encoder consisting of three layers of convolution (conv1, conv2, conv3) and three layers of pooling (pooling1, pooling2, pooling3). The filters in the convolution layers use gradually increasing structures to 5 filters, 25 filters, 125 filters, and use stride of 1. The activation function uses exponential linear unit (ELU), and the model is optimized using the Adam optimizer with a learning rate of 0.001. Since the test set is carried out by zero-filled missing values, max pooling is used rather than average pooling because it is possible to pool the characteristics considering the missing values in average pooling. After the last pooling layer, the feature map unfolds, leading to the fully connected layer. Thus, 7 days load image data (7×96) are converted from the encoder output to 100 one-dimensional data. It is reshaped and applied as an input of (4×25) to the LSTM, which consists of four cells to forecast the next one day (1×96).

IV. EXPERIMENTAL RESULTS

To demonstrate the performance of the proposed DCAE/LSTM, the experimental results are presented in three aspects. First, forecasting results are compared with various missing imputation methods for the cases of random missing and block missing. Second, we compare the DCAE/LSTM with three AE-based models: AE/LSTM, DAE/LSTM, and CAE/LSTM. We also compare the results with the popular forecasting models: DNN [10] and the state-of-the-art LSTM [12]. Third, we compare two domains used to input the forecasting model: 1) feature domain LSTM where the output of the *encoder* enters the input of the forecasting model, 2) time domain LSTM where the output of the *decoder* enters the input of the forecasting model.

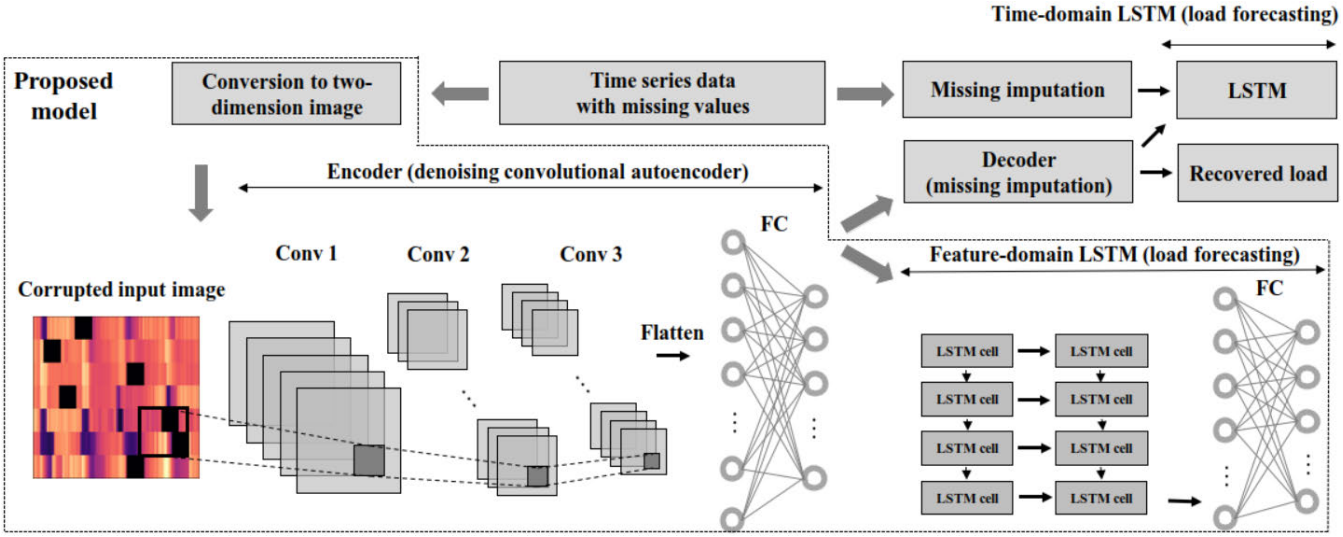


FIGURE 4. The proposed DCAE/LSTM combined model.

TABLE 5. Average MAPE Comparison of Missing Imputation Methods for the Input of LSTM.

Model	Random missing rate					Block missing rate				
	5%	10%	15%	20%	25%	5%	10%	15%	20%	25%
Zero-filled	26.63	27.23	28.41	30.15	33.19	26.08	26.38	28.44	30.37	32.16
Historical average	25.91	27.10	28.55	30.26	32.98	25.87	27.84	29.63	31.53	32.37
Forward-filled	22.73	23.79	24.56	25.85	26.38	25.17	26.36	27.95	29.58	31.59
Linear interpolation	22.96	23.56	24.63	25.00	25.97	23.12	24.77	26.42	27.49	29.01
DCAE (proposed)	**21.67**	**22.02**	**22.98**	**23.47**	**23.91**	**22.24**	**22.47**	**23.02**	**23.59**	**24.00**

A. COMPARISON WITH OTHER MISSING IMPUTATION METHODS

As shown in Table 5, we compare the proposed DCAE with various missing imputation methods: zero-filled (ZF), historical average, forward-filled, linear interpolation (LI). The zero-filled method simply fills the missing values with zeros [17]. The average of $t-1$ to $t-96$ (day-ahead) load data is used for the historical average method [18]. The forward-filled method fills the missing values with intact values before one unit time [30]. If the value before one unit time is also missing, it goes back until the intact value appears. The linear interpolation method replaces missing values with mean measurements that occur before and after missing [15]. In the case of random missing, linear interpolation performs almost as good as the proposed DCAE/LSTM because of the nature of random missing. However, in the case of block missing, the DCAE/LSTM is superior in all missing rates. As the block missing rate increases, the method of extracting important features for forecasting model's input is more efficient than separate methods of processing the missing values.

B. FORECASTING WITH INTACT DATA

First, we analyze each model using the intact data that do not have any missing values. As can be seen in Table 7 and Fig. 5, all four combined models show better forecasting accuracy than single forecasting models such as DNN and LSTM. Measured in MAPE, forecasting error becomes the lowest with (D)CAE/LSTM models. In the case of CAE/LSTM and DCAE/LSTM, we change the one-dimensional time series load data into two-dimensional load image data, which improves the overall forecasting accuracy because, by extracting features over several days, it prevents the attributes from being overfitted just for one day.

We analyze the forecasting results in the first quartile (0-25%, denoted by Q_1) and fourth quartile (75-100%, denoted by Q_4), based on MAPE to the degree of difficulty in forecasting. In the case of Q_1, when forecasting is easy, all models show good performance. In the case of Q_4, when forecasting becomes difficult, all four feature-based models show far better performance. This confirms that extracting important features and applying them to the input of the forecasting model works well; this is because the harder the forecasting is, the more focus should be on preventing overfitting. Also, it confirms that CAE is a good fit when forecasting is challenging; by using the convolution method, it is possible to encode information of several days of similar time because it brings in the features by referring to the receptive field, which is directly related to the forecasting accuracy. We also

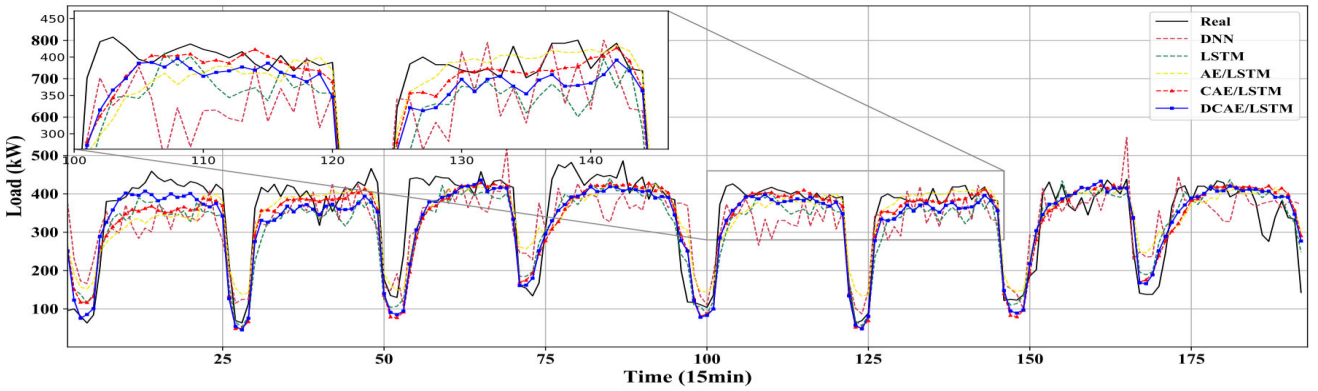


FIGURE 5. Forecasting result with intact data.

TABLE 6. Average MAPE with Different Missing Rates (Random Missing, Block Missing).

Model	Intact	Random missing rate					Block missing rate				
		5%	10%	15%	20%	25%	5%	10%	15%	20%	25%
DNN-ZF	25.31	28.89	32.54	34.03	36.61	39.21	30.85	33.54	35.92	39.58	44.30
DNN-LI		26.23	27.77	28.73	31.81	34.45	29.35	31.60	33.22	36.16	40.12
LSTM-ZF	22.41	26.63	27.23	28.41	30.15	33.19	26.08	26.38	28.44	30.37	32.16
LSTM-LI		22.96	23.56	24.83	25.20	26.57	23.12	24.77	26.42	27.49	29.01
AE/LSTM	20.20	22.64	24.09	25.01	25.95	27.02	22.70	23.75	24.84	26.46	27.88
DAE/LSTM	20.49	22.83	23.46	24.23	24.78	25.80	22.52	23.44	24.30	25.35	25.56
CAE/LSTM	19.17	21.95	22.41	23.07	24.17	24.95	22.52	22.90	23.24	23.93	24.52
DCAE/LSTM (proposed)	**18.90**	**21.67**	**22.02**	**22.98**	**23.47**	**23.91**	**22.24**	**22.47**	**23.02**	**23.59**	**24.00**

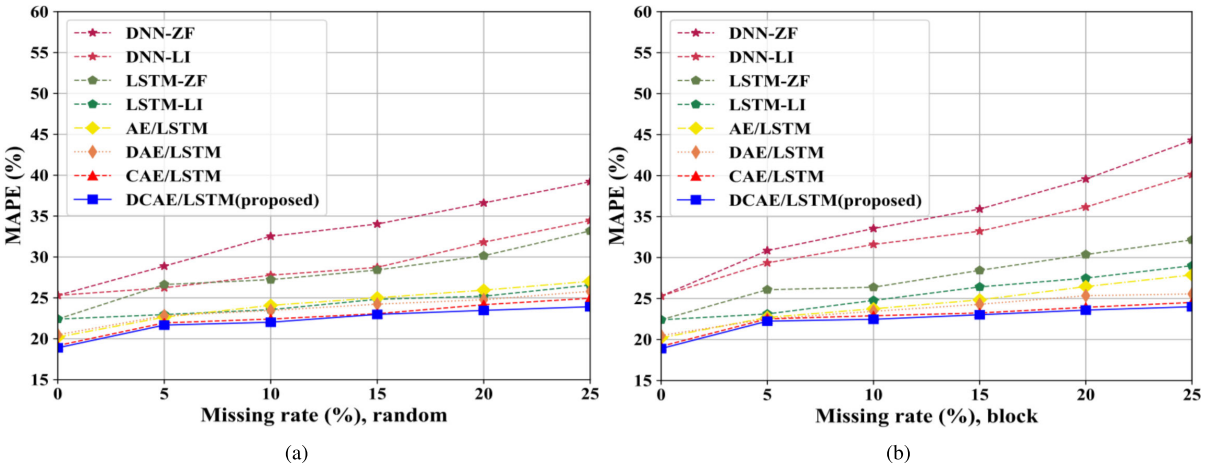


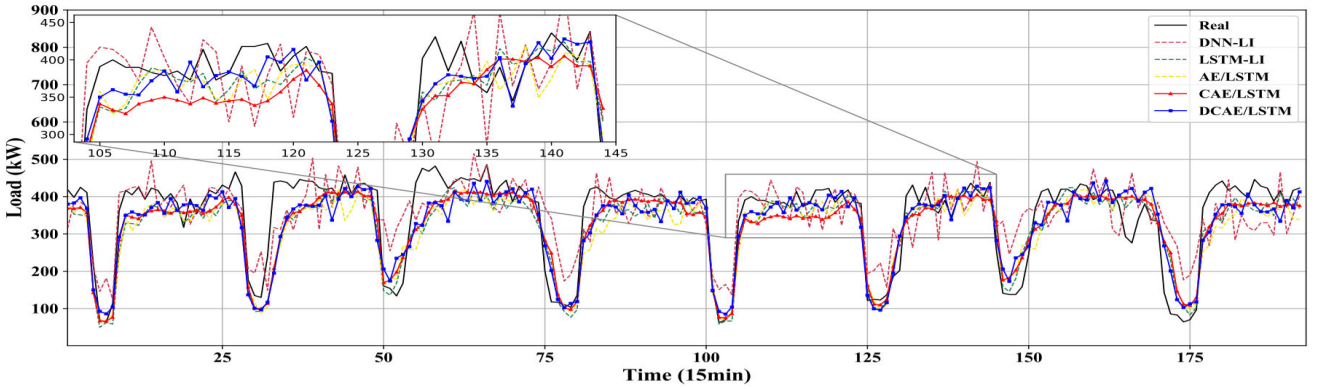
FIGURE 6. Average forecasting errors in terms of missing rate: (a) random missing (b) block missing.

find that denoising contributes to improving performance. Training is based on Google Colab's GPU [31], and all models have a fast training speed of fewer than 10 seconds.

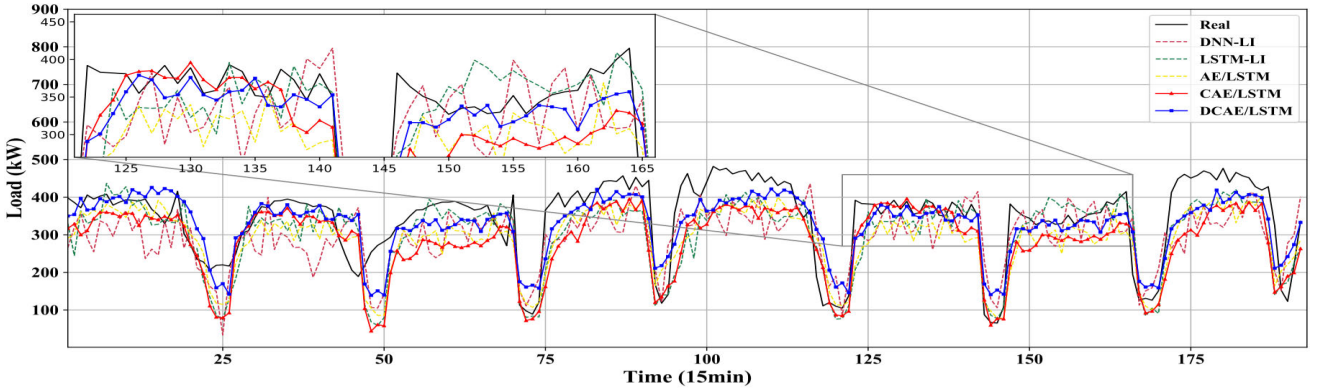
C. FORECASTING WITH MISSING DATA

Next, we analyze each model when missing occurs. As shown in Table 8, all the forecasting models combined with feature extraction perform better than DNN and LSTM. For example, DCAE/LSTM outperforms the traditional DNN-ZF and LSTM-ZF by 32.33% and 19.13%, respectively. The DCAE/LSTM model shows the best forecasting performance regardless of customers in different industries. In the case of DNN and LSTM models, the forecasting error increases sharply for customers who are relatively difficult to forecast.

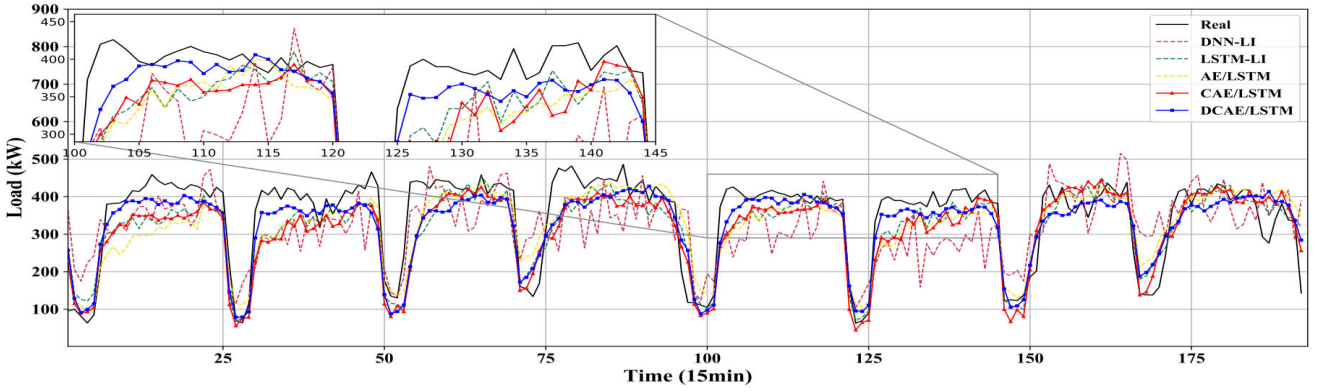
When we compare Table 8 with Table 7, the MAPEs of 10% random missing are increased compared to the case with intact data. Meanwhile, denoising models (DAE/LSTM, DCAE/LSTM) show better accuracy than non-denoising models (AE/LSTM, CAE/LSTM), respectively. It is because denoising models learn robustness to missing and partial destruction of input features. The benefit of denoising becomes more clear in the case of Q_4.



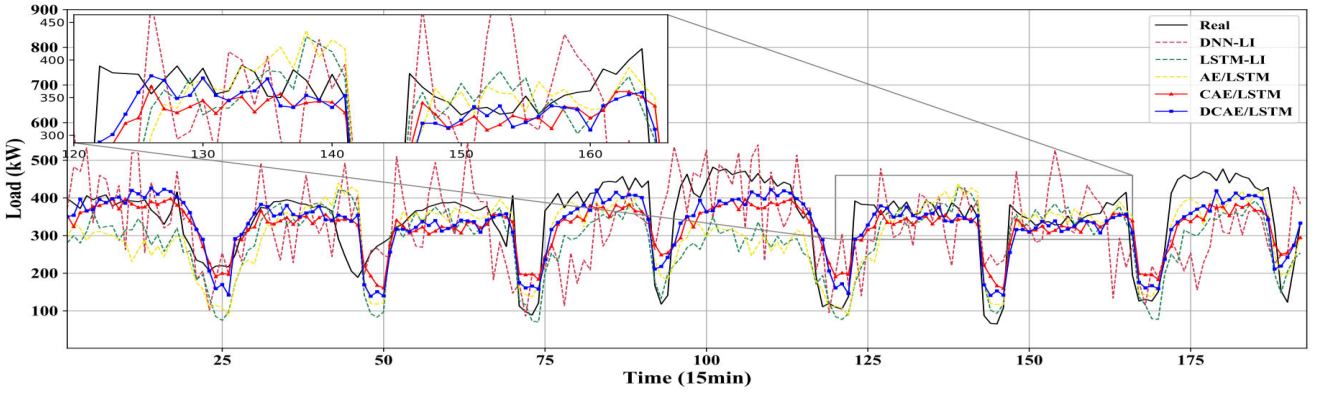
(a)



(b)



(c)



(d)

FIGURE 7. Forecasting results with various missing rates: (a) random 10% (b) random 25% (c) block 10% (d) block 25%.

TABLE 7. Forecasting Results with Intact Data.

Model	MAPE (%)			Training speed (s)
	Average	Q_1	Q_4	
DNN	25.31	6.40	62.26	1.47
LSTM	22.41	6.03	56.22	2.19
AE/LSTM	20.20	6.25	46.27	2.79
DAE/LSTM	20.49	6.43	45.04	2.84
CAE/LSTM	19.17	**5.90**	42.92	7.36
DCAE/LSTM (proposed)	**18.90**	6.02	**42.41**	7.55

TABLE 8. Forecasting Results With 10% Random Missing Data.

Model	MAPE (%)		
	Average	Q_1	Q_4
DNN-ZF	32.54	8.73	78.36
DNN-LI	27.77	8.50	65.26
LSTM-ZF	27.23	8.29	61.31
LSTM-LI	23.56	8.06	49.32
AE/LSTM	24.09	8.01	50.09
DAE/LSTM	23.46	8.10	47.41
CAE/LSTM	22.41	8.05	44.44
DCAE/LSTM (proposed)	**22.02**	**7.62**	**44.08**

D. VARIOUS MISSING RATE

1) RANDOM MISSING

We apply to various missing rates (5%, 10%, 15%, 20%, 25%, see Section II-C for range selection) to show robust forecasting performance of the proposed models. Fig. 6 and Table 6 show the forecasting results as the missing rate increases. The MAPEs of DNN and LSTM undesirably increase as the missing rate increases. By contrast, the (D)CAE/LSTM show the best performance for all the range of missing rate, followed by DAE/LSTM and AE/LSTM. Compared to DNN and LSTM, the combined models of extracting feature and forecasting achieve much smaller error for all missing rates. Furthermore, the result shows that the proposed DCAE/LSTM is the most robust to missing rates among the considered AE-based models.

2) BLOCK MISSING

We also verify the performance of the proposed models when the block-wise missing occurs. MAPE results with various missing rates (5%, 10%, 15%, 20%, 25%) are given in Table 6: 5%, 10%, 15%, 20%, 25% correspond to 75, 150, 225, 300, 375 minutes of successive missing occurrence in a day, respectively. As shown in Table 6, the MAPEs of DNN, AE/LSTM, DAE/LSTM surge. When block missing occurs, the performance of extracting features using (D)CAE is superior to that of extracting features using (D)AE. This is because multiple days can be considered when selecting features corresponding to each timestep by using two-dimensional filters. The result shows that obtaining information on different days using the receptive field is directly related to forecasting performance.

Graph results of random missing and block missing are given in Fig. 7. Overall, CAE/LSTM and DCAE/LSTM forecast close to the real values. When the missing rate is 25%, especially in the block missing, the models excluding CAE/LSTM and DCAE/LSTM show significantly poor forecasting performance.

TABLE 9. Comparing the use of Autoencoder (10% random missing).

Model	Feature domain LSTM		Time domain LSTM	
	Intact	Missing	Intact	Missing
AE/LSTM	20.20	24.09	21.48	24.84
DAE/LSTM	20.48	23.46	**21.17**	24.10
CAE/LSTM	19.17	22.41	21.96	24.07
DCAE/LSTM (proposed)	**18.90**	**22.02**	21.21	**23.95**

E. COMPARING THE USE OF AUTOENCODER

Finally we are interested in the effectiveness of using the feature domain LSTM against the time domain LSTM. Unlike our approach, LSTM has been originally used for time domain regression, and thus one may wonder what if, as shown in Fig. 4, the decoder output of autoencoder is used for the input of time domain LSTM. Recall that so far we focus on the feature domain LSTM that takes the output of the encoder as an input. Hence, the output of encoder as well as the output of decoder can be used for the input of feature domain LSTM and time domain LSTM, respectively. Table 9 shows that the performance of the feature domain LSTM outperforms the time domain LSTM for both intact and missing cases. It performs the role of smoothing, and the feature is well extracted to prevent overfitting, which implies that the proposed model does not need to handle missing imputation separately.

V. CONCLUSION

In this article, we proposed a new forecasting technique that is insensitive to missing data in nature. In doing this we considered a family of autoencoders such as AE, DAE, CAE, and DCAE, each of which is combined with LSTM. In overall, the proposed DCAE/LSTM model shows the best forecasting accuracy under various missing conditions such as intact data, random missing, and block missing, up to 25% missing rate. The forecasting improvements are by 18.6%–28.0% (random missing) and by 14.7%–25.4% (block missing), compared to the baseline LSTM. The proposed DCAE is also better than the traditional missing imputation methods, specifically in the case of block missing. Finally, we verified that our *unified* feature-based model (forecasting along with missing imputation) is better than the separate processing (forecasting after missing imputation).

REFERENCES

[1] G. Ciulla and A. D'Amico, "Building energy performance forecasting: A multiple linear regression approach," *Appl. Energy*, vol. 253, Nov. 2019, Art. no. 113500.

[2] J. F. Rendon-Sanchez and L. M. de Menezes, "Structural combination of seasonal exponential smoothing forecasts applied to load forecasting," *Eur. J. Oper. Res.*, vol. 275, no. 3, pp. 916–924, Jun. 2019.

[3] J. C. Lóez, M. J. Rider, and Q. Wu, "Parsimonious short-term load forecasting for optimal operation planning of electrical distribution systems," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1427–1437, Mar. 2019.

[4] C. Carquex, C. Rosenberg, and K. Bhattacharya, "State estimation in power distribution systems based on ensemble Kalman filtering," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6600–6610, Nov. 2018.
[5] Y. Chen, P. B. Luh, C. Guan, Y. Zhao, L. D. Michel, M. A. Coolbeth, P. B. Friedland, and S. J. Rourke, "Short-term load forecasting: Similar day-based wavelet neural networks," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 322–330, Feb. 2010.
[6] F. Y. Xu, X. Cun, M. Yan, H. Yuan, Y. Wang, and L. L. Lai, "Power market load forecasting on neural network with beneficial correlated regularization," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 5050–5059, Nov. 2018.
[7] H. Jiang, Y. Zhang, E. Muljadi, J. J. Zhang, and D. W. Gao, "A short-term and high-resolution distribution system load forecasting approach using support vector regression with hybrid parameters optimization," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3341–3350, Jul. 2018.
[8] M. Rafiei, T. Niknam, J. Aghaei, M. Shafie-Khah, and J. P. S. Catalão, "Probabilistic load forecasting using an improved wavelet neural network trained by generalized extreme learning machine," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6961–6971, Nov. 2018.
[9] Y. Yang, W. Li, T. A. Gulliver, and S. Li, "Bayesian deep learning based probabilistic load forecasting in smart grids," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4703–4713, Jul. 2019.
[10] S. Ryu, J. Noh, and H. Kim, "Deep neural network based demand side short term load forecasting," *Energies*, vol. 10, no. 1, p. 3, Dec. 2016.
[11] J. Vermaak and E. C. Botha, "Recurrent neural networks for short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 13, no. 1, pp. 126–132, Feb. 1998.
[12] K. Park, Y. Choi, W. J. Choi, H.-Y. Ryu, and H. Kim, "LSTM-based battery remaining useful life prediction with multi-channel charging profiles," *IEEE Access*, vol. 8, pp. 20786–20798, 2020.
[13] H. Choi, S. Ryu, and H. Kim, "Short-term load forecasting based on ResNet and LSTM," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Oct. 2018, pp. 1–6.
[14] L. Li, J. Zhang, Y. Wang, and B. Ran, "Missing value imputation for traffic-related time series data based on a multi-view learning method," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 8, pp. 2933–2943, Aug. 2019.
[15] J. Luengo, S. García, and F. Herrera, "A study on the use of imputation methods for experimentation with radial basis function network classifiers handling missing attribute values: The good synergy between RBFNS and event covering method," *Neural Netw.*, vol. 23, no. 3, pp. 406–418, Apr. 2010.
[16] S. Ryu, M. Kim, and H. Kim, "Denoising autoencoder-based missing value imputation for smart meters," *IEEE Access*, vol. 8, pp. 40656–40666, 2020.
[17] S. Poddar and M. Jacob, "Clustering of data with missing entries using non-convex fusion penalties," *IEEE Trans. Signal Process.*, vol. 67, no. 22, pp. 5865–5880, Nov. 2019.
[18] S. Ryu, H. Choi, H. Lee, and H. Kim, "Convolutional autoencoder based feature extraction and clustering for customer load analysis," *IEEE Trans. Power Syst.*, vol. 35, no. 2, pp. 1048–1060, Mar. 2020.
[19] P. K. Sharpe and R. J. Solly, "Dealing with missing values in neural network-based diagnostic systems," *Neural Comput. Appl.*, vol. 3, no. 2, pp. 73–77, Jun. 1995.
[20] D. Talwar, A. Mongia, D. Sengupta, and A. Majumdar, "AutoImpute: Autoencoder based imputation of single-cell RNA-seq data," *Sci. Rep.*, vol. 8, no. 1, p. 16329, Nov. 2018.
[21] R. Asadi and A. Regan, "A convolution recurrent autoencoder for spatio-temporal missing data imputation," 2019, *arXiv:1904.12413*. [Online]. Available: http://arxiv.org/abs/1904.12413
[22] S. Qu, K. Li, S. Zhang, and Y. Wang, "Predicting achievement of students in smart campus," *IEEE Access*, vol. 6, pp. 60264–60273, 2018.
[23] X. Xu, W. Chong, S. Li, A. Arabo, and J. Xiao, "MIAEC: Missing data imputation based on the evidence chain," *IEEE Access*, vol. 6, pp. 12983–12992, 2018.
[24] F. Saitoh, "An ensemble model of self-organizing maps for imputation of missing values," in *Proc. IEEE 9th Int. Workshop Comput. Intell. Appl. (IWCIA)*, Hiroshima, Japan, Nov. 2016, pp. 9–14.
[25] U. Hwang, S. Choi, H.-B. Lee, and S. Yoon, "Adversarial training for disease prediction from electronic health records with missing data," 2017, *arXiv:1711.04126*. [Online]. Available: http://arxiv.org/abs/1711.04126
[26] J. Dai, H. Song, G. Sheng, and X. Jiang, "Cleaning method for status monitoring data of power equipment based on stacked denoising autoencoders," *IEEE Access*, vol. 5, pp. 22863–22870, Aug. 2017.
[27] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, "Short-term residential load forecasting based on resident behaviour learning," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 1087–1088, Jan. 2018, doi: 10.1109/TPWRS.2017.2688178.
[28] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. OSDI*, vol. 16, 2016, pp. 265–283.
[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[30] Z. C. Lipton, D. C. Kale, and R. Wetzel, "Directly modeling missing data in sequences with RNNs: Improved classification of clinical time series," 2016, *arXiv:1606.04130*. [Online]. Available: http://arxiv.org/abs/1606.04130
[31] E. Bisong, "Google colaboratory," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Berkeley, CA, USA: Apress, 2019, pp. 59–64, doi: 10.1007/978-1-4842-4470-8_7.

KYUNGNAM PARK received the B.S. and M.S. degrees in electronic engineering from Sogang University, in 2017 and 2019, respectively. He is currently a Researcher with the Applied Research Team, Encored Technologies, Seoul, South Korea. His research interests include energy analytics, load forecasting, and deep learning. He was a recipient of CSAT Math and Science Scores, in 2012, in the Department of Electronic Engineering, Sogang University, and Full Scholarship for Excellent Undergraduate GPA, in 2017.

JAEIK JEONG received the B.S. and M.S. degrees in electronic engineering from Sogang University, South Korea, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the Department of Electronic Engineering. His research interests include deep reinforcement learning, smart grid, and energy forecasting.

DONGJOO KIM received the B.S. and M.S. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2013 and 2015, respectively. He is currently a Researcher with the Smart Power Distribution Laboratory, Korea Power Electric Corporation (KEPCO) Research Institute, Daejeon, South Korea. His research interests include energy economics, energy data analysis, and smart energy city.

HONGSEOK KIM (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Seoul National University, in 1998 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin, in 2009. From 2000 to 2005, he was a Member of Technical Staff with Korea Telecom Labs. From 2009 to 2010, he was a Postdoctoral Research Associate with the Department of Electrical Engineering, Princeton University. From 2010 to 2011, he was a Member of Technical Staff with Bell Labs, USA. He is currently a Professor with the Department of Electronic Engineering, Sogang University, South Korea. His research interests include smart grid and energy ICT, specifically focused on machine learning for energy forecasting, energy trading and electricity market, energy storage systems, microgrid, optimal power flow, and wireless networks. He was a recipient of the Korea Government Overseas Scholarship, from 2005 to 2008. He received the Haedong Young Professional Award, in 2016. He served as an Editor for the *Journal of Communications and Networks* and the Guest Editor for *Energies* with the Special Issue of Machine Learning and Optimization with Applications of Power System.