




Sliding window-based LightGBM model for electric load forecasting using anomaly repair

Sungwoo Park¹ · Seungmin Jung¹ · Seungwon Jung¹ · Seungmin Rho² · Eeunjun Hwang¹ 

Accepted: 30 March 2021 / Published online: 14 April 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Smart grids have attracted much attention recently for their potential to reduce power system operating and management costs. Smart grid core components include energy storage, renewable energy source(s), and smart meters. Smart meters collect diverse data regarding smart grid operation, which can lead to inefficient operation if the meter data are damaged or tampered with during collection or transmission. Therefore, it is important to identify abnormalities in smart grid data and process them accordingly. Various anomaly detection models have been proposed using statistical methods, but they cannot detect some anomaly patterns accurately, and the models generally did not consider repair strategies for the detected anomalies. Anomaly repair should be included with model training to improve forecasting performance. This paper proposes a robust sliding window-based LightGBM model for short-term load forecasting using anomaly detection and repair. We first show how to detect anomalies using a variational autoencoder and then how they can be repaired using a random forest method. Finally, we verify that the proposed sliding window-based LightGBM achieves superior forecasting performance in combination with anomaly detection and repair.

Keywords Anomaly detection · Data repair · Electric load forecasting · Variational autoencoder · Random forest · LightGBM · Sliding window.

This paper is an extended version of our paper published in Proceedings of the 2020 International Conference on Artificial Intelligence (ICAI), Las Vegas, USA, 27–30 Jul 2020.

✉ Eeunjun Hwang
ehwang04@korea.ac.kr

Extended author information available on the last page of the article

1 Introduction

Recent advances in the microelectromechanical system (MEMS) and flexible manufacturing system (FMS) have enabled significant sensor size reductions while retaining or extending advanced functionality and reducing their price [1]. Wireless communication technology has also enabled sensors to be embedded into many devices [2]. As a result, the continuous collection of large amounts of data through various sensors has become a very common phenomenon in modern systems [3].

Sensors measure physical quantities or objects that exist in nature, such as temperature, humidity, pressure, and convert them into electrical signals [4]. Most of the data collected through the sensor are time-series data, i.e., recorded at regular time intervals. The data change due to various variations such as trend variation, cyclical variation, and seasonal variation [5]. The importance of time-series data is growing because data analysis enables understanding of data changes and predicts future changes [6].

Sensor-based time-series data have been used in many fields [7]. The smart city is one of the fields that actively use these data. Many smart city systems utilize time-series data to develop applications [8]. For example, smart grid systems optimize energy operations by analyzing data collected from all power utilization processes within the smart city [9]. A smart grid, which is an intelligent power grid that combines information and communication technologies with existing power grids, can solve environmental problems and energy shortage problems by optimizing energy efficiency [10]. Through the smart grid, consumers can reduce electricity bills, and suppliers can optimize energy efficiency by solving demand and supply imbalances through real-time information sharing and control [11].

One of the main components that enable smart grid technology is the smart meter [12]. Smart meters are digital electronic meters that record energy consumption in real-time and control energy use by communicating information to both power suppliers and consumers through a communication network. The data recorded by the smart meter are used to analyze current or predict future energy usage.

Anomaly detection is essential to ensure smart meter data security and integrity [13]. Anomalies (outliers) are data points that differ significantly from other observations. They are commonly caused by malfunctioning smart meters, consumer behavior changes, energy leakage, or tampering. If the smart meter data are damaged or tampered with during acquisition or transmission, many problems such as incorrect electricity bills or inefficient smart grid operation can arise [14]. Therefore, it is essential to determine whether smart meter data include anomalies to avoid such problems [15].

Various statistical techniques have been proposed for anomaly detection [16], with interquartile range (IQR) being one of the most popular approaches [17]. However, IQR cannot detect some anomaly types, such as high leverage anomalies. To solve this problem, machine learning-based anomaly detection models were proposed [18]. While many studies make many efforts to improve anomaly

detection performance, they do not consider how to repair the detected anomalies [19]. If the use of anomalies-removed data for training the predictive model can be expected to improve the prediction performance, but if the data including the anomalies are repaired to an appropriate value and then used for training the predictive model, further improvement in prediction performance can be expected [20].

Therefore, this paper proposes an accurate electric load forecasting based on anomaly detection and repair. We used a variational autoencoder (VAE) for anomaly detection, a random forest (RF) for data repair, and a sliding window-based LightGBM for electric load forecasting. Figure 1 shows the overall structure for the proposed model.

The main contributions from this paper are as follows.

1. We propose a VAE anomaly detection scheme and verify it can achieve better accuracy than other statistical anomaly detection schemes.
2. We propose an RF anomaly repair scheme and verify its effectiveness by reflecting input variables.
3. We propose a sliding window-based LightGBM model for electric load forecasting and verify it can forecast power consumption more accurately than other popular machine learning electric load forecasting models.

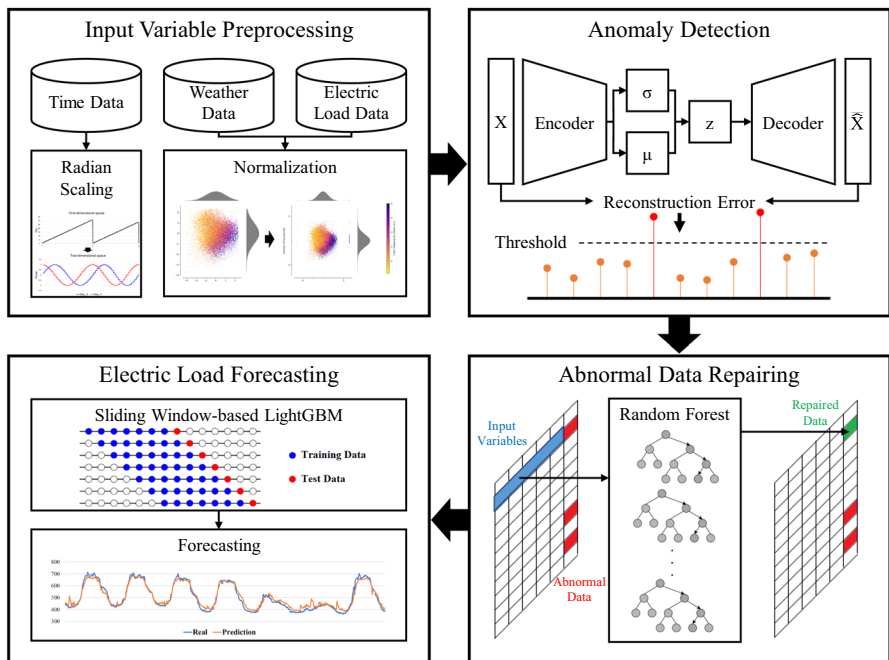


Fig. 1 Overall structure of proposed electric load forecasting scheme

The remainder of this paper is organized as follows. Section 2 discusses related anomaly detection, data repair, and electric load forecasting studies. Section 3 describes input variable configurations to constructing the proposed forecasting model, and Sect. 4 discusses the proposed VAE anomaly detection and repair schemes. Section 5 presents the proposed sliding window-based LightGBM model for load forecasting, and Sect. 6 discusses several experiments we performed to evaluate the proposed model performance. Finally, Sect. 7 summarizes and concludes the paper.

2 Related works

Many previous studies have considered anomaly detection, data repair, and electric load forecasting [21]. This section introduces various anomaly detection models and data repair methods and discusses relevant studies regarding electric load forecasting.

2.1 Anomaly detection

Breunig et al. [22] proposed an anomaly detection model based on the local outlier factor (LOF), i.e., how isolated an object is with respect to its surrounding neighborhood. Experimental results verified that LOF-based anomaly detection was a promising approach, identifying meaningful local outliers that previous approaches could not. Liu et al. [23] proposed an isolation forest (IForest) approach. IForest can create algorithms with linear time complexity, low constant, and low memory requirement. The proposed approach performed well in terms of area under the curve and processing time, particularly for large datasets. Chen et al. [24] proposed a randomized neural network (NN), i.e., a randomly connected autoencoder-based ensemble model combining adaptive sample size with random edge sampling, to achieve high-quality results while avoiding overfitting and improving robustness compared with conventional NN outlier detection techniques. Akouemo et al. [25] employed autoregression with exogenous inputs (ARXs) and an artificial neural network (ANN) to detect and impute anomalies in time-series data. They performed hypothesis testing on residual extrema to verify their proposed approach could identify and impute anomalous data points. Araya et al. [26] proposed two frameworks for anomaly detection in building energy consumption: collective contextual anomaly detection using a sliding window (CCAD-SW) framework and ensemble anomaly detection (EAD). The CCAD-SW framework identified anomalous consumption patterns using overlapping sliding windows, and the EAD framework combined several anomaly detection classifiers using majority voting. They verified that the EAD framework improved CCAD-SW sensitivity by 3.6% and reduced the false alarm rate by 2.7%.

2.2 Data repair

Xu et al. [27] proposed a point estimation model for biased sentinel hospital area disease estimation to interpolate missing data in temperature datasets. This technique employed a weighted summation of observed stations to estimate the missing data's

unbiased minimum error variance, using ratio and covariance between stations to calculate the weights. They achieved improved interpolation accuracy for the missing data from the temperature data and obtained the best linear unbiased estimation. Habermann et al. [28] proposed cubic spline interpolation as an alternative to linear or standard B-spline interpolation. The proposed interpolation could be implemented faster and easier than B-spline interpolation but had limited preconditions. However, the approach could provide a quick solution to cubic spline interpolation once the preconditions were satisfied. Therefore, it could be used for various approximation problems as in computational economics. Gan et al. [29] proposed a seislet transform for sparsity-based interpolation from highly undersampled seismic data based on the classic projection onto convex sets framework. Seismic data undersampled at very low boundary frequency can be low-pass filtered to obtain accurate estimates and subsequently interpolated through this estimate. They verified that the proposed approach achieved better performance than traditional frequency wavenumber-based approaches.

2.3 Electric load forecasting

Jurado et al. [30] constructed several prediction models using RF, ANN, and fuzzy inductive reasoning (FIR) approaches. They then compared the prediction models with an auto-regressive integrated moving average model by predicting electric energy consumptions in three different buildings at Catalonia Technical University in Catalonia, Spain, verifying that FIR approaches achieved the best prediction performance. Grolinger et al. [31] proposed electric load forecasting models based on support vector machine (SVM) and ANN for a large entertainment building in Canada and compared their performance under various model configurations to discuss the strengths and weaknesses of each model. They also presented a model selection algorithm to determine SVM and ANN hyperparameters. ANN achieved better accuracy than SVM models with daily data. Abbasi et al. [32] proposed an extreme gradient boosting (XGBoost) electrical load forecasting model, using feature importance to extract input variables from historical load over a week. They verified that historical loads close to or a week before the prediction time point had high importance for model construction. They used Australian Energy Market Operator electrical load data to confirm prediction performance. The proposed XGBoost model exhibited mean absolute percentage error, MAPE=10% with accuracy=97%. Kuo et al. [33] proposed an electric load forecasting model based on a convolutional NN, using historical electric load data as input variables to build the forecasting model. They verified that their proposed model was more accurate than models based on SVM, RF, decision tree (DT), etc.

3 Input variable configuration

This study used electric load data collected at a private university in Seoul, South Korea. The university grouped its buildings into four clusters according to the purpose or location and collected their power consumption data in real-time using an i-Smart system operated by the Korea Electric Power Corporation (KEPCO). The

data were collected every 15 min from January 1, 2016, to December 31, 2019. Cluster A comprised 32 academic buildings, including the central library and college of humanities buildings. Cluster B contained 16 residential buildings, and Clusters C and D contained 19 and 5 science and engineering buildings, respectively. Table 1 summarizes the collected data.

We also used time and weather data as input variables for anomaly detection and electric load forecasting. The following sections describe various relevant data details.

3.1 Time Data

Since electric load patterns differ depending on various timescales (minutes, hours, days of the week, months, etc.), we considered all variables that express date and time as an input variable [34], including month, day, hour, minute, day of the week, and holiday. Month, day, hour, and minute have a sequence form which is difficult to reflect periodic information in machine learning algorithms. For example, 11:59 pm and midnight are continuous in time, but the difference of the minute data in sequence form is 59. To solve this problem, we enhanced the time data into two dimensions,

$$\text{time}_x = \sin((360/\text{cycle}) \times \text{time}) \quad (1)$$

and

$$\text{time}_y = \cos((360/\text{cycle}) \times \text{time}), \quad (2)$$

where cycle represents time data periodicity, e.g., month and minute cycles = 12 and 60, respectively. We retained one- and two-dimensional data to better represent temporal characteristics [35]. In addition, we used a vector of 0 or 1 for each day of the week and holiday data. Days of the week can be expressed in continuous and

Table 1 Statistical analysis of electric load data in the four building clusters

Statistic	Cluster A	Cluster B	Cluster C	Cluster D
Mean	643.77	315.97	640.76	510.03
Standard error	0.94	0.21	0.6	0.32
Median	535.68	305.43	562.08	472.2
Mode	271.2	278.64	454.56	417.6
Standard deviation	353.64	77.32	225.06	121.18
Sample variance	125,059.3	5,978.61	50,651.71	14,683.4
Kurtosis	-0.71	0.42	0.34	-0.53
Skewness	0.69	0.75	1.06	0.71
Range	1,565.76	522.36	1,116	572.2
Minimum	159.36	133.56	294.24	315.2
Maximum	1,725.12	655.92	1,410.24	887.4
Count	140,256	140,256	140,256	140,256

binary format. If the days of the week are represented as continuous data, the difference between two consecutive days is 1, while the difference between Sunday and Monday is 6. This could have a negative impact on the forecasting model. Thus, we represented the day of the week as binary data using the one-hot encoding method. Likewise, if the input day of the week is a holiday, the input variable for a holiday is 1 and 0 otherwise. Table 2 provides the resultant 20-time data input variable data types.

3.2 Weather data

As power usage is closely related to weather conditions, we considered weather data as an input variable [36]. The Korea Meteorological Administration (KMA) provides diverse short- and long-range weather forecasts. We considered short-range weather forecasts for daily minimum temperature, daily maximum temperature, temperature, humidity, wind speed, cloudiness, and precipitation. Short-range weather forecasts provide weather data for up to 67 h with 3 h resolution, and we calculated smaller resolution weather data using linear interpolation. Figure 2 shows an example short-range weather forecast provided by KMA.

We also calculated wind chill (WC) and discomfort index (DI) to establish a more direct association with power consumption [37],

Table 2 Input variables configuration for time data

No	Input variable	Type
1	Month	Continuous on [1, 12]
2	Day	Continuous on [1, 31]
3	Hour	Continuous on [0, 23]
4	Minute	Continuous on [0,59]
5	Month _x	Continuous on [- 1, 1]
6	Month _y	Continuous on [- 1, 1]
7	Day _x	Continuous on [- 1, 1]
8	Day _y	Continuous on [- 1, 1]
9	Hour _x	Continuous on [- 1, 1]
10	Hour _y	Continuous on [- 1, 1]
11	Minute _x	Continuous on [- 1, 1]
12	Minute _y	Continuous on [- 1, 1]
13	Monday	Binary
14	Tuesday	Binary
15	Wednesday	Binary
16	Thursday	Binary
17	Friday	Binary
18	Saturday	Binary
19	Sunday	Binary
20	Holiday	Binary

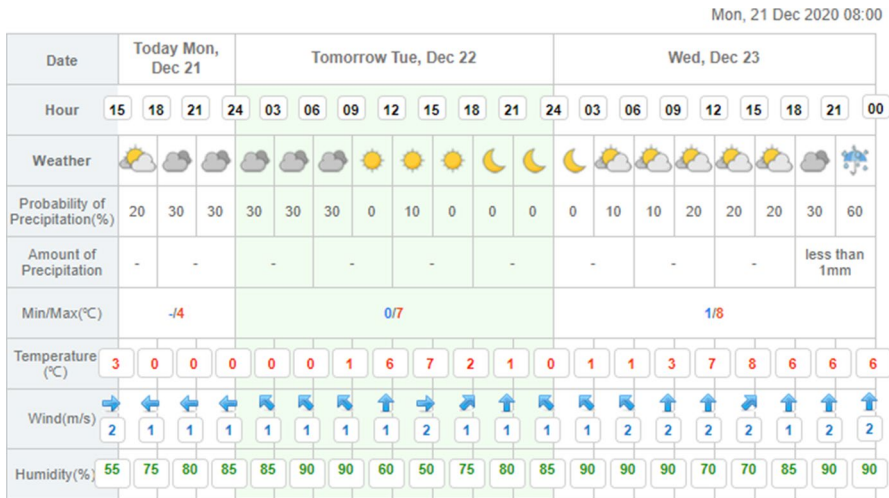


Fig. 2 Example of short-range weather forecast provided by KMA

$$WC = 13.12 + 0.0615 \times T - 11.37 \times WS^{0.16} + 0.3965 \times T \times WS^{0.16} \quad (3)$$

and

$$DI = 1.8 \times T - 0.55(1.8 \times T - 26) \times (1 - 0.01 \times H) + 32, \quad (4)$$

where T , H , and WS represent temperature, humidity, and wind speed, respectively. Thus, we used nine weather data types. Table 3 provides the Pearson correlation coefficient (PCC) value of the weather data for each cluster.

Table 3 PCC value of weather data for each cluster

Weather data	Cluster A	Cluster B	Cluster C	Cluster D
Daily minimum Temperature	0.017	0.196	0.171	0.137
Daily maximum Temperature	0.041	0.145	0.142	0.110
Temperature	0.119	0.277	0.295	0.278
Humidity	0.323	0.138	0.253	0.286
Wind Speed	0.270	0.136	0.230	0.286
Cloudiness	0.012	0.064	0.053	0.042
Precipitation	0.005	0.027	0.007	0.006
Wind chill	0.099	0.265	0.279	0.255
Discomfort index	0.127	0.298	0.304	0.290

4 Anomaly detection model configuration

An autoencoder (AE) is a deep learning network based on unsupervised learning comprising encoder and decoder networks [38]. The encoder maps input data from high-dimensional space to low-dimensional space and expresses it as a latent variable. Latent variables compressed by the encoder preserve input data characteristics, so the decoder can restore the original input data by analyzing the latent variable. The difference between output and the input values becomes a loss function in AE and is used for learning by backpropagation, enabling unsupervised learning.

The basic VAE principle is the same as for AE, but the latent variable is generated from a Gaussian distribution [39]. Latent variables generated by AE are random discrete values and hence difficult to understand what each latent variable means; whereas VAE constructs a Gaussian probability distribution that can derive mean and standard deviation of the latent variable and then uses variables randomly obtained from the probability distribution as input values for the decoder. Figure 3 shows a typical AE model structure.

There are considerable differences between AE and VAE loss functions for learning. Only the reconstruction error is used for the AE loss function, i.e., an index that determines the decoder resilience. Since it is impossible to learn how latent variables are generated from specific input data, completely different latent variables can be generated from similar input data. On the other hand, the VAE loss function combines reconstruction error and Kullback–Leibler (KL) divergence, an index that determines whether the VAE latent variable has a specific distribution.

The VAE loss function can be expressed as

$$L_i(\phi, \theta, x_i) = -\mathbb{E}_{q_\phi(z|x_i)} [\log(p_\theta(x_i|z))] + KL(q_\phi(z|x_i)||p(z)), \tag{5}$$

where the first term represents reconstruction error, i.e., cross-entropy between x_i and the result of recovering x_i based on z generated by the encoder; and the second

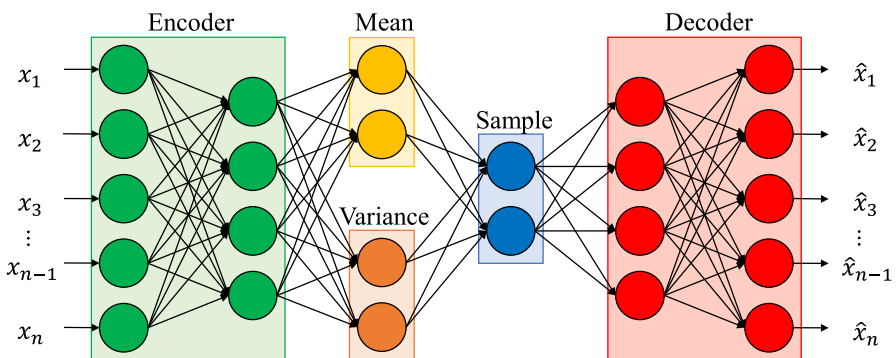


Fig. 3 Typical variational autoencoder model structure

term represents KL divergence, i.e., the probability distribution difference between sampled z and z generated by the encoder. The latent variable distributions converge with increased second term value.

Using KL divergence in the learning process can help the encoder generate a common cluster depending on the data class. Thus, latent variable characteristics, i.e., anomalies, can be more clearly defined using VAE compared with AE. Therefore, this study employed a VAE-based anomaly detection model.

AE-based anomaly detection uses reconstruction errors from all input variables, and any reconstruction is determined to be an outlier if the reconstruction error exceeds a given threshold. However, if load data anomalies are detected using the reconstruction error for all input variables, normal load data may be classified as abnormal data. Therefore, we only used the load data reconstruction error after calculating the reconstruction error for each input variable in VAE.

5 Load forecasting model configuration

LightGBM is a popular ensemble model released in 2016 that uses a boosting algorithm [40] to combine several weak learners into a more accurate model. The boosting concept trains weak models sequentially, compensating for previous model problems in the subsequent model. LightGBM shortens data processing time, a disadvantage of previous boosting algorithms, using gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB) techniques. GOSS excludes data instances with small gradients and uses the remainder to estimate information. Since data with large gradients are more critical, information can be estimated quickly and accurately, even from small-scale datasets. EFB bundles mutually exclusive variables and processes them to reduce the number of variables. The number of variables can be effectively reduced by bundling and processing variables that rarely have simultaneous nonzero values without significantly impairing accuracy. Thus, LightGBM achieves good performance with short training times.

We also employ a sliding window algorithm to reflect the latest trends [41], which use previous steps to predict the next step. Figure 4 shows the sliding window approach for time-series data.

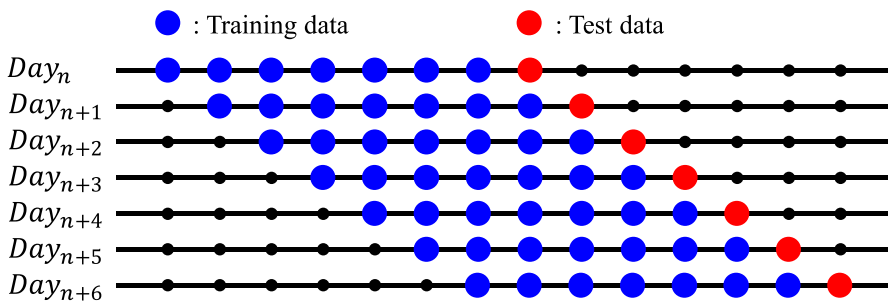


Fig. 4 Sliding window approach for time-series data

The sliding window algorithm requires considerable learning time because the model needs to be newly trained to predict the next point. Thus, we need a model that provides excellent prediction performance even with short learning times to support this effectively, and hence we selected LightGBM.

6 Experimental results

6.1 Anomaly detection

This section verifies the proposed VAE anomaly detection scheme effectiveness by comparison with several popular anomaly detection models, including IQR, LOF, and IForest. For comparison, we constructed several datasets by increasing the ratio of anomalies to the total data amount from 1 to 10% in 1% increments. Anomalies were randomly generated with values less than 0.8 times the normal data value or more than 1.2 times the normal data value. Also, randomly generated anomalies include both one-point anomalies and continuous anomalies. Figure 5 illustrates an example of the collected electric load data with generated anomalies. In the figure, points 1 to 4 represent one-point anomalies, and point 5 represents a continuous anomaly.

Anomaly detection scheme accuracy was measured as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{6}$$

where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative metrics, respectively.

Figures 6, 7, 8, and 9 compare the selected model performances for the four building clusters, respectively. The proposed VAE anomaly detection scheme exhibits the best performance in most cases, with less accuracy reduction as the anomaly rate increases, compared with all other methods. For example, the LOF and proposed

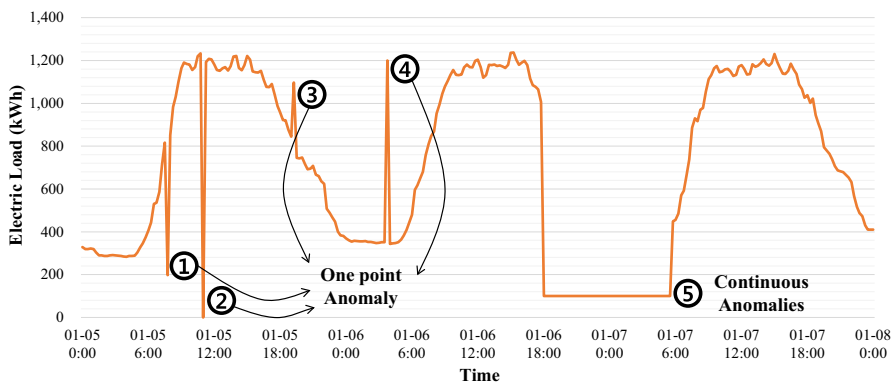


Fig. 5 Example of the electric load data including generated anomalies

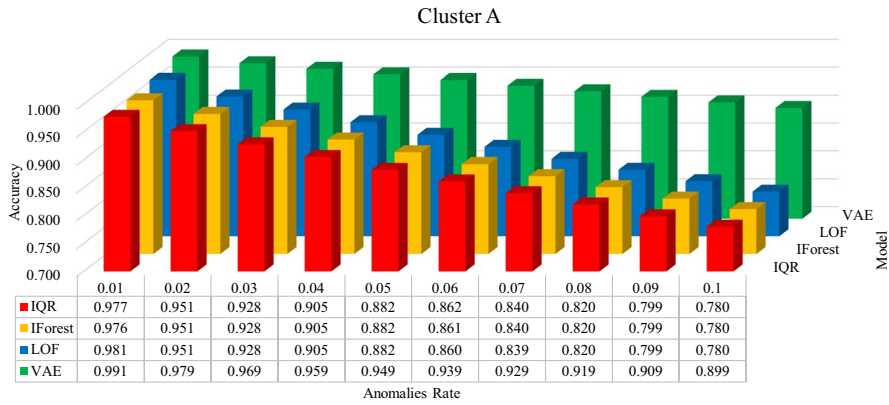


Fig. 6 Accuracy comparison of selected models according to anomalies rate in Cluster A

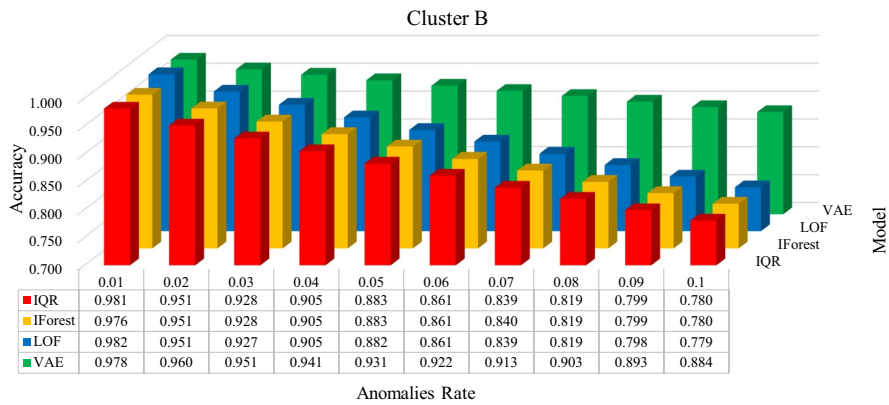


Fig. 7 Accuracy comparison of selected models according to anomalies rate in Cluster B

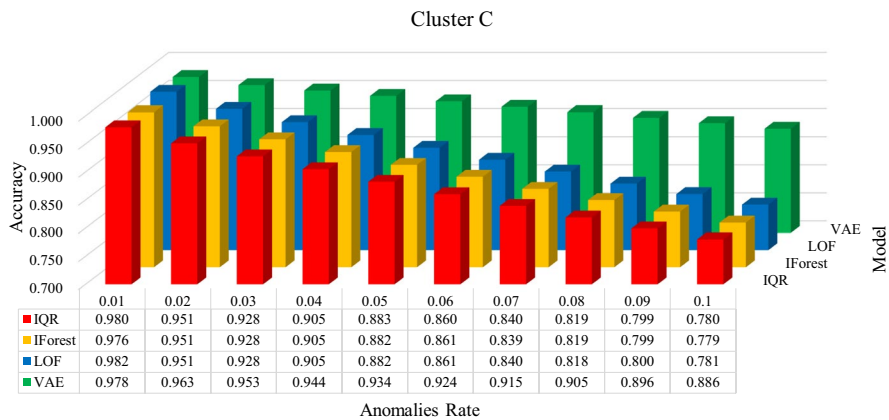


Fig. 8 Accuracy comparison of selected models according to anomalies rate in Cluster C

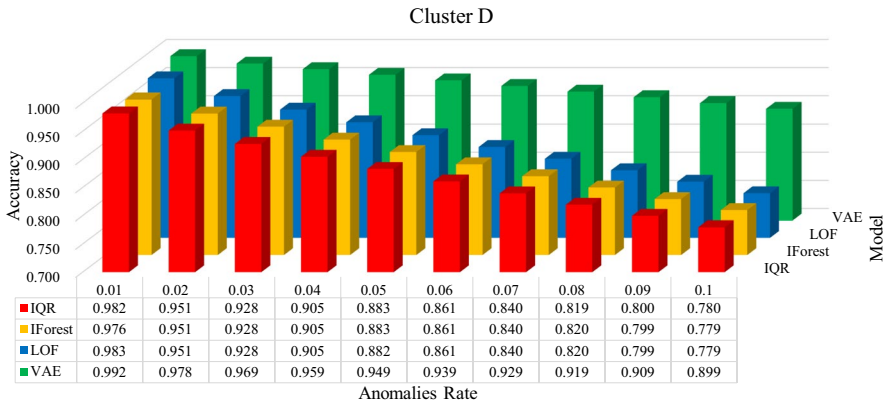


Fig. 9 Accuracy comparison of selected models according to anomalies rate in Cluster D

scheme overall accuracy reduction = 20.6% and 9.4%, respectively, for 1–10% anomaly rate.

6.2 Data repair

This section investigates the proposed RF data repair scheme effectiveness compared with linear interpolation. Linear interpolation is effective for repairing single point anomaly, but not for continuous anomalies. To repair continuous anomalies effectively, various external variables should be used to represent the situation at the time when the continuous anomalies occurred. We excluded popular models such as SVM and DNN in the comparison because they need a significant time for hyperparameter tuning and model training. On the other hand, RF is a flexible machine learning algorithm that performs well even without hyperparameter tuning. As RF works well with large amounts of data and large numbers of input variables, it is suitable for repairing anomalies using various external variables. Hence, we proposed a data repair scheme using RF and compared it with zero and linear interpolations. We repaired the randomly generated anomalies for the different anomaly ratio cases and compared the repaired and original data using mean absolute percentage error (MAPE), defined as

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|, \tag{7}$$

where n , y_i , and \hat{y}_i represent data amount, actual electric load data, and forecasted electric load data, respectively. MAPE is effective at comparing the results of all clusters at once. Table 4 compares the repair methods for the defined building clusters. The proposed RF repair method achieves better repair performance than linear interpolation for all cases. Values in bold font indicate the best repair performance for each anomaly rate.

Table 4 MAPE comparison of repair methods for each cluster

Anomaly rate	Cluster A			Cluster B		
	Zero	Linear	RF	Zero	Linear	RF
0.01	99.46	3.43	2.87	99.36	2.92	1.64
0.02	98.48	3.43	2.92	98.47	2.82	1.79
0.03	97.98	3.34	2.95	97.99	2.79	1.88
0.04	97.49	3.31	2.89	97.49	2.85	1.98
0.05	97.01	3.30	2.88	97.01	2.93	1.85
0.06	96.51	3.40	2.93	96.50	2.87	1.93
0.07	96.00	3.36	2.93	96.01	2.95	1.93
0.08	95.55	3.39	3.00	95.52	2.94	1.90
0.09	95.10	3.34	2.92	95.10	2.86	1.86
0.1	94.61	3.34	2.92	94.61	2.87	1.88
Anomaly rate	Cluster C			Cluster D		
	Zero	Linear	RF	Zero	Linear	RF
0.01	99.40	1.85	1.19	99.38	1.60	1.38
0.02	98.47	1.75	1.30	98.47	1.67	1.55
0.03	97.98	1.84	1.21	97.98	1.65	1.52
0.04	97.49	1.81	1.28	97.49	1.63	1.50
0.05	97.02	1.78	1.27	97.00	1.66	1.49
0.06	96.51	1.81	1.28	96.51	1.66	1.50
0.07	96.01	1.81	1.24	96.02	1.67	1.50
0.08	95.51	1.79	1.31	95.52	1.68	1.50
0.09	95.12	1.81	1.27	95.12	1.69	1.50
0.1	94.61	1.83	1.29	94.63	1.68	1.51

6.3 Determining optimal window size

We constructed a sliding window-based LightGBM model for electric load forecasting, determining the optimal window size empirically by comparing performance for various window sizes (1–10 days). Each day was represented by 96 points since time resolution = 15 min. For the same reason using MAPE as an indicator of data repair, we used MAPE to compare the forecasting performance of sliding window-based LightGBM models with different window sizes.

As a result of conducting experiments on all clusters, prediction performance improved as window size increased up to 7 days. From then on, there is no further significant improvement compared to the increase in training time. Therefore, we set window size = 7 days. Figure 10 shows the training times and MAPE of the proposed model with different window sizes (Table 5).

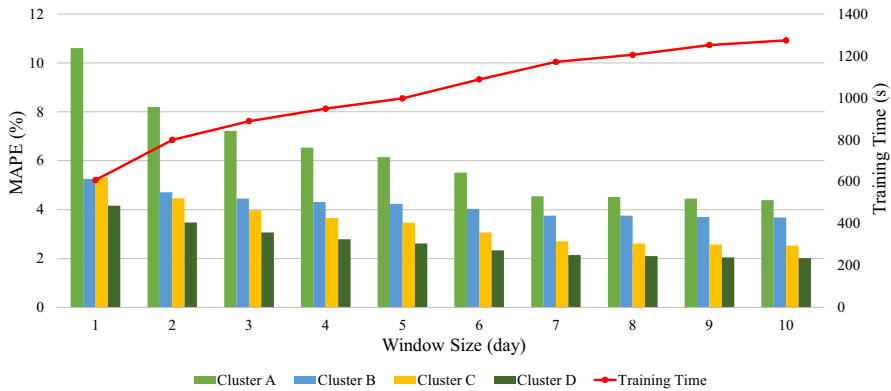


Fig. 10 MAPE and training time of proposed model

Table 5 Hyperparameter settings for each model

Model	Hyperparameter
LR	–
DT	1. Criterion: “friedman_mse”
GBM	1. Max_depth: 4 2. N_estimators: 256
RF	1. N_estimators: 256
XGBoost	1. Colsample_bylevel: 0.5 2. Max_depth: 6 3. N_estimators: 256
LightGBM	1. Max_depth: 4 2. N_estimator: 256 3. Num_leaves: 32 4. Subsample: 0.5
DNN	1. Batch_size: 1000 2. Hidden_layer: 6 3. Hidden_node: 24 4. Learning_rate: 0.1
LSTM	1. Batch_size: 1000 2. Hidden_node: 24 3. Learning_rate: 0.1 4. Seq_len: 96

6.4 Electric load forecasting

Tables 6, 7, 8, and 9 compare the forecasting performance of various machine learning models: linear regression (LR), DT, gradient boosting machine (GBM), RF, XGBoost, LightGBM, deep neural network (DNN), and long short-term memory (LSTM). DNN and LSTM models were implemented using Tensorflow, and the rest of the models were implemented using scikit-learn library. Table 5 shows the hyperparameter settings for each forecasting model. We divided the

Table 6 MAE comparison of forecasting models

Forecasting model	Cluster A			Cluster B		
	Zero	Linear	RF	Zero	Linear	RF
LR	137.574	141.111	141.109	51.950	42.667	42.662
DT	131.844	67.548	66.184	77.842	41.229	40.549
GBM	51.190	50.279	50.242	47.526	31.426	31.155
RF	66.494	51.017	50.976	60.251	31.466	31.432
XGBoost	50.390	47.716	47.693	47.936	31.447	31.243
LightGBM	50.719	50.266	50.251	46.817	31.390	30.995
DNN	29.176	29.170	28.418	12.435	12.426	12.456
LSTM	27.182	26.452	24.805	12.085	11.958	11.747
Proposed	25.233	24.809	24.492	12.515	12.465	11.469
	Cluster C			Cluster D		
	Zero	Linear	RF	Zero	Linear	RF
LR	117.739	101.843	101.842	67.856	67.858	64.653
DT	148.247	75.016	73.789	106.423	56.346	55.732
GBM	104.571	65.012	64.825	53.486	48.762	48.692
RF	129.975	66.687	66.615	66.523	49.993	49.976
XGBoost	105.020	64.742	64.666	53.901	48.849	48.707
LightGBM	103.554	65.093	65.051	52.870	48.825	48.674
DNN	19.228	19.710	19.707	12.378	12.377	12.073
LSTM	21.657	18.086	18.378	13.137	11.301	10.916
Proposed	18.749	18.382	17.662	10.908	10.710	10.709

dataset into training and test sets, comprising data collected from January 1, 2016, to December 31, 2018, and January 1, 2019, to December 31, 2019, respectively.

We compared forecasting performance using mean absolute error (MAE), root-mean-square error (RMSE), root-mean-square logarithmic error (RMSLE), and MAPE, respectively, defined as

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n}, \quad (8)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}, \quad (9)$$

$$\text{RMSLE} = \sqrt{\frac{\sum_{i=1}^n (\log(y_i + 1) - \log(\hat{y}_i + 1))^2}{n}}. \quad (10)$$

Table 7 RMSE comparison of forecasting models

Forecasting model	Cluster A			Cluster B		
	Zero	Linear	RF	Zero	Linear	RF
LR	170.086	170.086	167.66	66.632	55.125	55.116
DT	258.058	101.247	98.282	130.862	54.357	53.112
GBM	72.369	69.091	69.249	57.305	41.036	40.731
RF	96.460	72.840	72.690	71.589	41.229	41.180
XGBoost	70.724	66.696	66.680	57.486	41.001	40.811
LightGBM	70.041	69.773	68.550	56.228	41.133	40.642
DNN	49.604	49.599	49.113	18.415	18.420	18.089
LSTM	48.074	47.926	47.025	18.110	18.040	17.887
Proposed	40.576	39.657	39.656	17.187	17.056	17.045
	Cluster C			Cluster D		
	Zero	Linear	RF	Zero	Linear	RF
LR	155.969	133.267	133.263	87.537	87.534	85.042
DT	264.216	95.386	94.811	189.74	88.124	87.129
GBM	117.878	78.325	78.086	81.246	80.437	80.414
RF	147.186	80.089	80.039	91.380	81.368	81.324
XGBoost	117.625	77.358	77.148	80.871	80.588	80.217
LightGBM	116.187	78.411	78.401	80.508	80.406	79.340
DNN	34.437	34.435	34.141	19.414	19.413	19.163
LSTM	34.941	33.553	33.270	19.602	18.636	18.397
Proposed	29.998	29.292	29.290	17.634	17.241	17.240

and (7). In Tables 6, 7, 8, and 9, values in bold font indicate the best performance for each model.

The proposed RF-based data repair method provides superior performance overall metrics. Also, the sliding window-based LightGBM model achieves the best forecasting performance compared with the other machine learning techniques for most cases.

Finally, we conducted a Wilcoxon test to verify that the results of the proposed model are statistically the same as those of the other models. In the test, if the p value is less than the significance level, the null hypothesis is rejected, which indicates that there is no significant difference between the two dependent samples. The results of the Wilcoxon test with a significance level of 0.05 are shown in Table 10. As the p value in all cases is lower than the significance level, it was proved that the results of the proposed model were not statistically different from those of other models. This means that the data used for the sliding window-based LightGBM model, albeit in a small amount, are sufficient for training.

Table 8 RMSLE comparison of forecasting models

Forecasting model	Cluster A			Cluster B		
	Zero	Linear	RF	Zero	Linear	RF
LR	0.470	0.470	0.468	0.198	0.163	0.163
DT	2.133	0.158	0.155	2.068	0.175	0.170
GBM	0.117	0.116	0.116	0.175	0.122	0.122
RF	0.149	0.118	0.118	0.231	0.124	0.124
XGBoost	0.112	0.112	0.111	0.123	0.123	0.177
LightGBM	0.117	0.116	0.111	0.172	0.123	0.121
DNN	0.085	0.085	0.085	0.059	0.059	0.058
LSTM	0.083	0.081	0.083	0.058	0.058	0.057
Proposed	0.070	0.069	0.069	0.054	0.054	0.054
	Cluster C			Cluster D		
	Zero	Linear	RF	Zero	Linear	RF
LR	0.253	0.218	0.218	0.173	0.173	0.168
DT	2.283	0.155	0.153	2.102	0.166	0.165
GBM	0.188	0.122	0.122	0.153	0.149	0.149
RF	0.245	0.127	0.127	0.182	0.151	0.151
XGBoost	0.188	0.122	0.122	0.153	0.150	0.149
LightGBM	0.184	0.122	0.122	0.149	0.149	0.149
DNN	0.048	0.048	0.048	0.037	0.037	0.037
LSTM	0.050	0.046	0.045	0.038	0.036	0.035
Proposed	0.043	0.041	0.041	0.034	0.033	0.033

7 Conclusion

This paper proposed an accurate electric load forecasting scheme that detects anomalies using VAE, repairs data using RF, and forecasts electric load using sliding window-based LightGBM. We collected 15-min resolution electric load data collected at a private university in Seoul, South Korea, and performed data preprocessing for the proposed scheme.

We proposed a VAE-based anomaly detection method and compared its performance to popular anomaly detection methods such as IQR, LOF, and IForest. The proposed VAE-based anomaly detection method shows the best performance in most cases, with less accuracy reduction as the anomaly rate increases, compared with all other methods. In addition, we used the RF model to repair the anomalies to appropriate values. To repair continuous anomalies effectively, various external variables should be used to represent the situation at the time when the continuous anomalies occurred. As RF works well with large amounts of data and large numbers of input variables, it is suitable for repairing anomalies using various external variables. As a result of comparing the proposed RF-based data repair method with the widely used missing data interpolation methods, such as zero

Table 9 MAPE comparison of forecasting models

Forecasting model	Cluster A			Cluster B		
	Zero	Linear	RF	Zero	Linear	RF
LR	30.169	30.167	27.394	14.652	12.493	12.492
DT	22.043	11.831	11.625	23.411	12.226	12.061
GBM	9.246	9.216	8.633	13.674	9.099	9.018
RF	10.663	9.189	9.189	17.533	9.149	9.133
XGBoost	8.776	8.764	8.452	13.856	9.129	9.077
LightGBM	9.257	9.226	8.525	13.488	9.083	8.965
DNN	5.314	5.313	5.310	4.060	4.057	3.895
LSTM	5.086	5.003	4.597	3.897	3.875	3.892
Proposed	4.600	4.546	4.545	3.913	3.898	3.755

	Cluster C			Cluster D		
	Zero	Linear	RF	Zero	Linear	RF
LR	16.587	14.824	14.823	14.488	14.488	13.049
DT	22.103	11.247	11.061	21.922	11.616	11.517
GBM	15.548	9.770	9.755	10.074	10.065	10.461
RF	19.373	10.059	10.046	13.066	10.334	10.328
XGBoost	15.670	9.791	9.789	10.099	10.070	10.563
LightGBM	15.400	9.788	9.776	10.372	10.087	10.053
DNN	3.053	3.052	3.051	2.535	2.524	2.524
LSTM	3.536	2.821	2.702	2.719	2.351	2.191
Proposed	2.756	2.702	2.700	2.183	2.144	2.144

Table 10 Result of Wilcoxon test

Compared models		Wilcoxon test (p value)			
		Cluster A	Cluster B	Cluster C	Cluster D
Proposed model	LR	1.10×10^{-288}	0	0	0
	DT	0	0	0	0
	GBM	0	0	0	0
	RF	0	0	0	0
	XGBoost	0	0	0	0
	LightGBM	0	0	0	0
	DNN	3.28×10^{-90}	1.67×10^{-25}	3.53×10^{-6}	1.30×10^{-92}
	LSTM	5.06×10^{-45}	1.55×10^{-49}	2.70×10^{-50}	1.01×10^{-182}

interpolation and linear interpolation, it was confirmed that the use of RF could better repair anomalies. Finally, we proposed a sliding window-based LightGBM model for electric load forecasting. As a result of experimenting with various window sizes, prediction performance improved as window size increased up to

7 days, with no further significant improvement. Therefore, the proposed sliding window-based LightGBM model has a seven-day window size.

The performance of the proposed models was verified in terms of MAE, RMSE, RMSLE, and MAPE compared with other popular machine learning and deep learning methods. As a result of the experiment using the data repaired through the zero, linear, and RF interpolation techniques, it was confirmed that the best performance was obtained when using the data repaired by RF. In addition, as a result of comparing the performance of the proposed model with various models, it was confirmed that the performance of the proposed model shows the best performance in all indicators.

Despite meaningful experimental outcomes, our study has some limitations which present our future research directions. First, despite previous studies showing better performance, it could not be used due to time constraints. In order to apply to the actual smart grid system, we plan to explore new models that show good performance and take less time for model training. Second, it is difficult to explain how the proposed model derives the predicted values. So, we plan to develop a more accurate electrical load forecasting model by analyzing the influence of various input variables using an explainable artificial intelligence (XAI) technique. In addition, we will verify the possibility of application in the smart grid by research in link electrical load forecasting model with various systems such as Energy Management System (EMS) and Energy Storage System (ESS).

Acknowledgement This research was supported by Energy Cloud R&D Program (grant number: 2019M3F2A1073184) through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT.

References


1. Hodge VJ, O'Keefe S, Weeks M, Moulds A (2014) Wireless sensor networks for condition monitoring in the railway industry: a survey. *IEEE Trans Intell Transp Syst* 16(3):1088–1106
2. Hempstead M, Lyons MJ, Brooks D, Wei GY (2008) Survey of hardware systems for wireless sensor networks. *J Low Power Electron* 4(1):11–20
3. Jagannathan S (2016) Real-time big data analytics architecture for remote sensing application. In: *International Conference on Signal Processing, Communication, Power and Embedded System*, pp 1912–1916 (2016).
4. Kanoun O, Trankler HR (2004) Sensor technology advances and future trends. *IEEE Trans Instrum Meas* 53(6):1497–1501
5. Bell WR, Hillmer SC (1983) Modeling time series with calendar variation. *J Am Stat Assoc* 78(383):526–534
6. Fu TC (2011) A review on time series data mining. *Eng Appl Artif Intell* 24(1):164–181
7. Ding D, Cooper RA, Pasquina PF, Fici-Pasquina L (2011) Sensor technology for smart homes. *Maturitas* 69(2):131–136
8. Yu Z, Zheng X, Huang F, Guo W, Sun L, Yu Z (2020) A framework based on sparse representation model for time series prediction in smart city. *Front Comp Sci* 15(1):1–13
9. Chou JS, Ngo NT (2016) Smart grid data analytics framework for increasing energy savings in residential buildings. *Autom Constr* 72:247–257
10. Tabrizchi H, Javidi MM, Amirzadeh V (2019) Estimates of residential building energy consumption using a multi-verse optimizer-based support vector machine with k-fold cross-validation. *Evolv Syst* 1–13 (2019)

11. Park S, Moon J, Hwang E (2019) 2-Stage electric load forecasting scheme for day-ahead CCHP scheduling. In: IEEE 13th International Conference on Power Electronics and Drive Systems (PEDS), pp 1–4
12. Montazerolghaem A, Moghaddam MHY, Leon-Garcia A (2017) OpenAMI: Software-defined AMI load balancing. *IEEE Internet Things J* 5(1):206–218
13. Raciti M, Nadjm-Tehrani S (2013) Embedded cyber-physical anomaly detection in smart meters. In: Critical information infrastructures security, pp 34–45
14. Jiang R, Lu R, Wang Y, Luo J, Shen C, Shen X (2014) Energy-theft detection issues for advanced metering infrastructure in smart grid. *Tsinghua Sci Technol* 19(2):105–120
15. Moghaddass R, Wang J (2017) A hierarchical framework for smart grid anomaly detection using large-scale smart meter data. *IEEE Trans Smart Grid* 9(6):5820–5830
16. Zhang W, Yang Q, Geng Y (2009) A survey of anomaly detection methods in networks. In: International symposium on computer network and multimedia technology, pp 1–3
17. Wang C, Viswanathan K, Choudur L, Talwar V, Satterfield W, Schwan K (2011) Statistical techniques for online anomaly detection in data centers. In: 12th IFIP/IEEE international symposium on integrated network management and workshops, pp 385–392
18. Chandola V, Banerjee A, Kumar V (2009) Anomaly detection: a survey. *ACM Comput Surv (CSUR)* 41(3):1–58
19. Zhang A, Song S, Wang J, Yu PS (2017) Time series data cleaning: from anomaly detection to anomaly repairing. *Proc VLDB Endowm* 10(10):1046–1057
20. Jung S, Moon J, Park S, Rho S, Baik SW, Hwang E (2020) Bagging ensemble of multilayer perceptrons for missing electricity consumption data imputation. *Sensors* 20(6):1772
21. Armstrong JS (1989) Combining forecasts: the end of the beginning or the beginning of the end? *Int J Forecast* 5:585–588
22. Breunig MM, Kriegel HP, Ng RT, Sander J (2000) LOF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp 93–104
23. Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. In: 8th IEEE International Conference on Data Mining, pp 413–422
24. Chen J, Sathe S, Aggarwal C, Turaga D (2017) Outlier detection with autoencoder ensembles. In: Proceedings of the SIAM International Conference on Data Mining, pp 90–98
25. Akouemo HN, Povinelli RJ (2017) Data improving in time series using ARX and ANN models. *IEEE Trans Power Syst* 32(5):3352–3359
26. Araya DB, Grolinger K, ElYamany HF, Capretz MA, Bitsuamlak G (2017) An ensemble learning framework for anomaly detection in building energy consumption. *Energy Build* 144:191–206
27. Xu CD, Wang JF, Hu MG, Li QX (2013) Interpolation of missing temperature data at meteorological stations using P-BSHADE. *J Clim* 26(19):7452–7463
28. Habermann C, Kindermann F (2007) Multidimensional spline interpolation: theory and applications. *Comput Econ* 30(2):153–169
29. Gan S, Wang S, Chen Y, Zhang Y, Jin Z (2015) Dealised seismic data interpolation using seislet transform with low-frequency constraint. *IEEE Geosci Remote Sens Lett* 12(10):2150–2154
30. Jurado S, Peralta J, Nebot A, Mugica F, Cortez P (2013) Short-term electric load forecasting using computational intelligence methods. In: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp 1–8
31. Grolinger K, L'Heureux A, Capretz MA, Seewald L (2016) Energy forecasting for event venues: Big data and prediction accuracy. *Energy Build* 112:222–233
32. Abbasi RA, Javaid N, Ghuman MNJ, Khan ZA, Rehman SU (2019) Short term load forecasting using XGBoost. In: Workshops of the International Conference on Advanced Information Networking and Applications, pp 1120–1131
33. Kuo PH, Huang CJ (2018) A high precision artificial neural networks model for short-term energy load forecasting. *Energies* 11(1):213
34. Massana J, Pous C, Burgas L, Melendez J, Colomer J (2015) Short-term load forecasting in a non-residential building contrasting models and attributes. *Energy Build* 92:322–330
35. Park S, Moon J, Jung S, Rho S, Baik SW, Hwang E (2020) A two-stage industrial load forecasting scheme for day-ahead combined cooling, heating and power scheduling. *Energies* 13(2):443
36. Wang P, Liu B, Hong T (2016) Electric load forecasting with recency effect: a big data approach. *Int J Forecast* 32:585–597

37. Xie J, Chen Y, Hong T, Laing TD (2016) Relative humidity for load forecasting models. *IEEE Trans Smart Grid* 9:191–198
38. Kramer MA (1991) Nonlinear principal component analysis using autoassociative neural networks. *AIChE J* 37(2):233–243
39. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint <http://arxiv.org/abs/1312.6114>
40. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY (2017) LightGBM: a highly efficient gradient boosting decision tree. In: *Advances in neural information processing systems*, pp 3146–3154
41. Moon J, Jung S, Rew J, Rho S, Hwang E (2020) Combination of short term load forecasting models based on a stacking ensemble approach. *Energy Build* 109921

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Sungwoo Park¹ · Seungmin Jung¹ · Seungwon Jung¹ · Seungmin Rho² · Eeunjun Hwang¹ 

Sungwoo Park
psw5574@korea.ac.kr

Seungmin Jung
jmkstcom@korea.ac.kr

Seungwon Jung
jsw161@korea.ac.kr

Seungmin Rho
smrho@cau.ac.kr

¹ School of Electrical Engineering, Korea University, Seoul, Republic of Korea

² Department of Industrial Security, Chung-Ang University, Seoul, Republic of Korea