IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON ARTIFICIAL INTELLIGENCE TECHNOLOGIES FOR ELECTRIC POWER SYSTEMS

Received January 28, 2020, accepted February 14, 2020, date of publication February 27, 2020, date of current version March 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2976500

Denoising Autoencoder-Based Missing Value Imputation for Smart Meters

SEUNGHYOUNG RYU, MINSOO KIM, AND HONGSEOK KIM, (Senior Member, IEEE)
Department of Electronic Engineering, Sogang University, Seoul 04107, South Korea

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

This work was supported in part by Smart City R&D 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 Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT under Grant NRF-2017R1A1A1A05001377.

ABSTRACT Electric load data are essential for data-driven approaches (including deep learning) in smart grid, and advanced smart meter technologies provide fine-grained data with reliable communications. Despite the recent development of smart metering devices, however, missing data still arise due to unexpected device power off, communication failure, measuring error, or other unknown reasons. In this paper, we investigate a deep learning framework for missing imputation of smart meter data by leveraging a denoising autoencoder (DAE). Then, we compare the performance of the proposed DAE with traditional methods as well as other recently developed generative models, e.g., variational autoencoder and Wasserstein autoencoder. The proposed DAE based imputation shows significantly better results compared to other methods in terms of root mean square error (RMSE) by up to 28.9% for point-wise error, and by up to 56% for daily-accumulated error.

INDEX TERMS Deep learning, smart grid, missing imputation, smart meters, denoising autoencoder, generative model, daily load profile (DLP).

I. INTRODUCTION

ASmart grid is an intelligent power grid system that is enabled by the convergence between the legacy power grid and the information & communication technology (ICT); the paradigm of power grid has expanded from infrastructure-based industry to hybrid of ICT and infrastructure. In this respect, *intelligence* is the core of the smart grid that differentiates from the legacy system.

Meanwhile, recent advent of the fourth industrial revolution was sparked by the development of artificial intelligence (AI). Specifically, deep learning leads the field of AI research. Based on higher abstraction ability and training from massive data, deep learning outperforms other algorithms in various fields, e.g., image classification, natural language processing, machine translation, generative model, etc. In this respect, the intelligence of the smart grid can be achieved by deep learning, which implies the emergence of energy AI as the future of the smart grid.

This shift is being realized through worldwide deployment of smart meters. Since smart meters are installed from large-scale sites to small-scale end-customers, i.e., building/factories to households, massive metering data are generated. By leveraging deep learning on energy big data, intelligent solutions for smart grid applications have been actively developed recently, e.g., load forecasting [1]–[4], system monitoring [5], clustering [6] [7], battery management [8], and load control [9].

Hence, metering data are the core of energy AI, and data-driven intelligent systems can be developed by utilizing energy big data. For all data-driven approaches, the well-known data mining framework is the knowledge discovery in databases (KDD) process [10]. The KDD process consists of five steps: data selection, preprocessing, transformation, data mining, and evaluation. Among the five stages, preprocessing is important as a groundwork for data mining process. Although it is possible to achieve fine-grained data with reliable communications through advanced smart metering technologies, there still exist missing values in databases. Such missing values occur due to unexpected device power off, communication failure, measuring error, or other unknown reasons [11], [12]. Therefore, missing values should be properly recovered during the preprocessing stage.

The associate editor coordinating the review of this manuscript and approving it for publication was Canbing Li.

There are various strategies of dealing missing value, which can be classified by three categories [13]: a) do not impute (DNI), b) deletion, and c) imputation. The first option leaves missing values as it is. However, in practice, most data analysis techniques are not robust to missing values and thus missing values need to be filled in advance. The second option is excluding missing values from the dataset. It is simple, but the main drawback is that the size of valid data reduces. In addition, typical forecasting models require complete past data with no missing values. From these reasons, we focus on the third option, i.e., missing imputation that replaces missing values with estimated ones. Specifically, we leverage a denoising autoencoder (DAE) to fill in *multiple* missing values in a daily load profile (DLP). So far, simple imputation techniques have been applied to missing smart meter data imputation. Two widely used methods are linear interpolation (LI) and historical average (HA) [14]. LI method replaces missing values from a straight line between the last/first measured value before/after missing value occurs. Hence, it is simple, fast, and effective for short-term missing values. However, if missing data occur for a long duration, LI based imputation may not work. On the other hand, HA method imputes missing values with the average of highly correlated time slots, e.g., the average of hour-ahead, day-ahead, or week-ahead metering data [1], [7]. HA method shows good results for load data at an aggregated level that has regular load patterns. However, electric loads of end customers (households or small buildings) are more random and volatile, which makes missing value imputation challenging.

Although missing imputation is common in data analytics, there are few studies focusing on the imputation of electric load in the literature. To overcome the disadvantages of LI and HA methods, optimally weighted average imputation method was proposed in [14], where missing values are imputed by the weighted sum of both LI and HA results. In [15], the authors developed a learning-based adaptive missing data imputation method (LAI) by applying kNN algorithm to the optimal length of historical data. Then, the extended LAI method, which combines LAI and LI, was proposed to alleviate the unexpected variations in the missing durations. Based on the low intrinsic dimensionality of load profile and the sparsity of bad data (missing values and outliers), distributed principal components pursuit is utilized for cleansing of bad data in [16]. K-means clustering algorithm based imputation framework was proposed in [17]. The authors compared different lengths of past data and distance measure for clustering, and the use of segmented cluster center and Canberra distance showed better imputation performance than others. In power grid perspective, topology based missing value estimation was proposed in [18]. In case, topology, voltage, and current information of neighboring nodes are available, missing meter readings can be well estimated.

Recently, deep learning has been utilized for missing data imputation in various fields, e.g., RNA sequences [19], traffic data [20], images [21]. For smart grids, a stacked DAE is utilized for missing power system monitoring data [22]. In [23], an enhanced DAE and long short-term memory (LSTM) based missing reconstruction framework was proposed and verified with phasor measurement unit data. In this paper, we investigate a deep learning based missing value imputation framework for household load data, which are highly irregular and thus missing value imputation becomes more challenging.

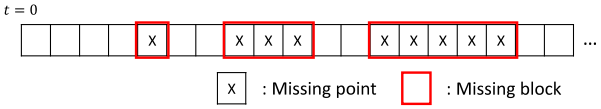


FIGURE 1. Missing points and missing blocks in load data.

Our contributions are summarized as follows. First, we propose a DLP based missing value imputation framework by leveraging a DAE. Based on the observed meter readings in an input DLP, the proposed DAE fills in *multiple* missing values of DLP at once. Compared to the previous methods, the proposed method does not require iteratively applying imputation for multiple missing values or determining the proper number of clusters and neighbors. Hence, once trained, it is easily applicable. Then, we analyze and compare the error results of the proposed framework with traditional LI, HA methods as well as other deep learning based generative methods such as denoising variational autoencoder and denoising Wasserstein autoencoder. Our extensive experiments with various types of missing (e.g., uniformly random and block-wise missing with 4 different configurations, and 4 predefined missing scenarios) for 15-minute interval residential load data verify that the proposed DAE model shows robust and accurate missing imputation performance; the point-wise imputation error of the proposed DAE is up to 28.9% lower and daily-accumulated error is up to 56% lower than the others in terms of RMSE and normalized accumulated absolute error.

The rest of this paper is organized as follows. First, we analyze the occurrence of missing data in a real dataset in Section II. In Section III, we briefly introduce deep learning methodologies (autoencoder and generative models). Section IV describes the proposed autoencoder based missing imputation framework in overall. Experimental results with residential load data are given in Section V. Finally, the paper is concluded in Section VI.

II. DESCRIPTION OF MISSING SMART METER DATA IN PRACTICE

Before describing the proposed missing value imputation framework, we analyze the statistical information of missing data obtained from a real dataset. As illustrated in Figure 1, missing data can occur block-wise. As an example in Figure 1, there are 9 missing points and 3 missing blocks. Let $l \geq 1$ denote the length of missing block.

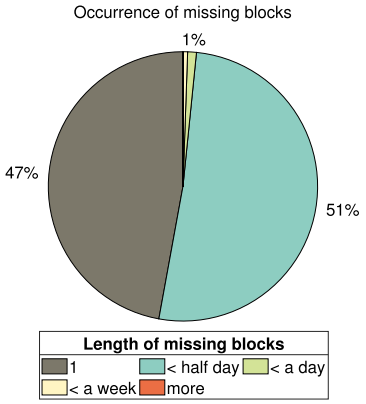


FIGURE 2. Pie chart about the length of missing block.

TABLE 1. Statistics of missing values.

Class	Period	l	Missing blocks	
			Number	Percentage (%)
I	single missing	1	34k	47.14
II	up to half day	$(1, 48]$	37k	51.26
III	up to day	$(48, 96]$	760	1.05
IV	up to one week	$(96, 672]$	386	0.53
V	more than a week	$(672, \infty)$	15	0.02
Total			72k	100

Next, we analyze the statistics of missing blocks of residential customers in South Korea. The dataset contains 360 days (March 1, 2016, to February 23, 2017) 15-minute interval load data of 1,445 residential customers. Hence, there are total 49,939,200 (360 × 96 × 1445) data points. Among 50 million load measurements, there exist 420k (420,253) missing points, which is roughly 1% of total measurements. In addition, there are 72k (72,412) missing blocks, and 47% of them are length 1. Details of missing statistics are given in Figure 2 and Table 1. We classify missing data into five classes depending on the length of missing block (l). As described in Table 1, Class I takes a large portion (47.14%). Class II ($1 < l \leq 48$, up to half day) takes the largest portion such as 51%. Since Class I missing block can be easily imputed with simple interpolation method and Class III, IV, V (more than half days) are only 1.6%, this paper mainly focuses on the random mixture of intra-day missing blocks (Class II).

For better understanding of missing value distribution, Figure 3 shows the number of missing points occurred at a specific time slot. As can be seen, missing points occur evenly in most areas, except two regions marked by (A) and (B). For example, more than 90% of households in the dataset have missing data in (A) area. For (B) area, missing data consecutively occur during 24 ∼ 25 and 27 ∼ 31 of December (20% of households). Such high occurrence of missing data is abnormal compared to other periods, and it might be due to external impact or event rather than the malfunction of smart meter. Figure 4 describes missing data occurrence in terms of the time of day, which is equal to the

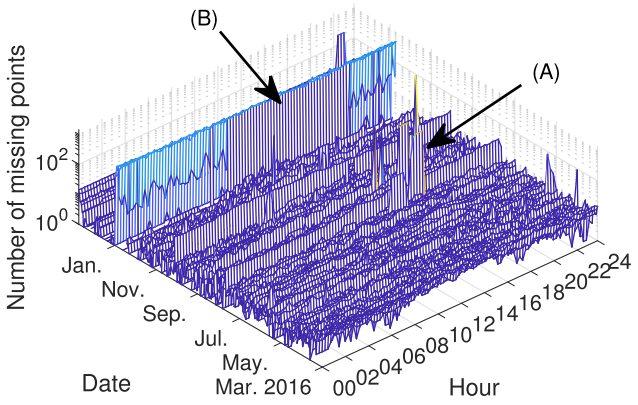


FIGURE 3. Illustration of missing occurrence.

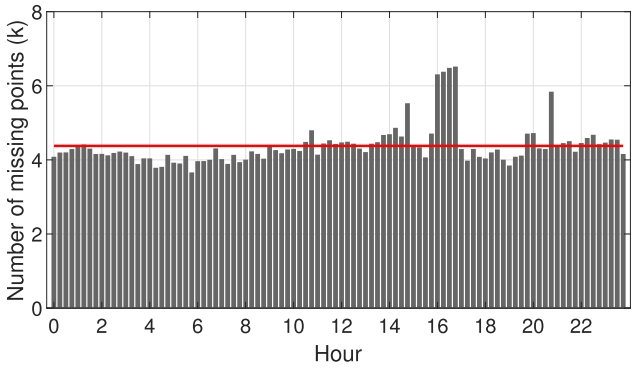


FIGURE 4. Bar graph of missing occurrence in each time slot. Red line is the average.

sum over Date axis in Figure 3. Missing points are uniformly distributed over a day except around 16:00, which is affected by abnormal missing occurrence of (A) in Figure 3. So far we emphasize the high occurrence of missing blocks in real environment. This motivates us to develop a deep learning technique, which is particularly effective for imputing random mixture of missing blocks.

III. METHODOLOGIES

In this section, we introduce deep learning methodologies (a family of autoencoders) for missing load data imputation.

A. DENOISING AUTOENCODER

An autoencoder is one kind of artificial neural networks which can be summarized by two nonlinear encoding and decoding functions f and g, respectively. For a given input vector $\mathbf{x}$, f encodes input $\mathbf{x}$ to latent feature $\mathbf{z}$, then g decodes transformed $\mathbf{z}$ to original $\mathbf{x}$. Therefore, the autoencoder learns to extract a feature vector that contains crucial information of input data. As the name suggests, DAE is an autoencoder having denoising property. Instead of input $\mathbf{x}$, DAE reconstructs $\mathbf{x}$ from partially corrupted input $\hat{\mathbf{x}}$. Hence, DAE achieves robustness to the partial destruction of input by learning common latent representation over the original and corrupted data [24]. For a given corruption function h, intentionally corrupted data $\hat{\mathbf{x}} = h(\mathbf{x})$ can be obtained. Then, DAE is trained to minimize the error between $\mathbf{x}$ and $\tilde{\mathbf{x}} = g(f(\hat{\mathbf{x}}))$.

Even though additive Gaussian noise is commonly used for h, DAE for missing value imputation needs to utilize different corruption functions; instead of adding noise to data, corruption function replaces selected data with predetermined value that represents missing status. Hence, the network is trained to restore the correct data from missing status value.

B. VARIATIONAL AUTOENCODER

A variational autoencoder (VAE) is a generative model having autoencoding structure [25]. The basic autoencoder learns deterministic function f and g by neural networks. However, generative models focus on the distribution of the data P_X; we denote capital letters (e.g., $P(X)$, $Q(Z|X)$) for probability distribution and lower-case letters (e.g., $p(x)$) for corresponding density function by following the notations in [26], respectively. In doing this, VAE starts from the latent variable model P_G to achieve P_X where $p_G(x) = \int p_G(x|z)p(z)dz$ holds. This indicates high dimensional and complicated x is driven from simple latent vector z which lies in a lower dimensional space. Prior distribution P_Z is typically assumed to be Gaussian. Hence, the objective becomes maximizing log likelihood of $p_G(x)$, but, $\int p_G(x|z)p(z)dz$ is intractable. Thus, VAE solves this problem by maximizing the lower bound of likelihood with encoder $Q(Z|X)$ and decoder $P_G(X|Z)$. Consequently, the objective function of VAE is given by minimizing

$$-\mathbb{E}_{Q(Z|X)}[\log p_G(x|z)] + D_{KL}(Q(Z|X), P_Z). \quad (1)$$

Intuitively, the first term indicates maximizing log likelihood $p_G(x|z)$ when z follows $Q(Z|X)$, and the latter term leads $Q(Z|X)$ to approximate prior latent distribution P_Z, respectively. Details of VAE is given in the original work of [25]. Note that denoising framework also can be applied to generative models, and denoising criterion for VAE is introduced in [27].

C. WASSERSTEIN AUTOENCODER

Another approach of generative model is based on the optimal transport, i.e., Wasserstein autoencoder (WAE) [26]. A distance between two probability distributions can be measured by optimal transport (OT) cost. The underlying objective of generative model is achieving generator distribution P_G equal to the data distribution P_X. Hence, in terms of optimal transportation, generative model is minimizing OT cost $W_c(P_X, P_G)$ where P_G is a latent variable model specified by the prior distribution P_Z. OT cost $W_c(P_X, P_G)$ is defined by the infimum of $\mathbb{E}_{(X,Y)\sim\Gamma}[c(X, Y)]$ where $c(x, y)$ is any measurable cost function and Γ belongs to the set of joint distribution $\mathcal{P}(X \sim P_X, Y \sim P_G)$. Specifically for the metric space $(\mathcal{X}, d)$ and $c(x, y) = d^p(x, y)$ for $p \geq 1$, the p-th root of W_c is called the p-Wasserstein distance.

According to Theorem 1 in [26], the above OT cost $W_c(P_X, P_G)$ is equal to

$$\inf_{Q:Q_Z=P_Z} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}[c(X, G(Z))], \quad (2)$$

with deterministic $P_G(X|Z)$ and any function G maps $\mathcal{Z}$ to $\mathcal{X}$. Finally, with relaxation of $Q_Z = P_Z$ condition, the objective of WAE is minimizing

$$\inf_{Q(Z|X)\in\mathcal{Q}} \mathbb{E}_{P_X}\mathbb{E}_{Q(Z|X)}[c(X, G(Z))] + \lambda \cdot D_Z(Q_Z, P_Z), \quad (3)$$

with nonparametric set of encoders $\mathcal{Q}$, arbitrary divergence D_Z, and its weight $\lambda > 0$. In short, the first term is the cost (distance) between original X and generation by $G(Z)$. The latter is a regularizer to make Q_Z approximates P_Z. Theoretical ground and implementation details of WAE are given in the original work of [26].

The inherent problem of VAE is that the KL divergence term forces each $Q(Z|X)$ to match P_Z [26], [28]. On the other hand, overall Q_Z matches to P_Z in WAE. In addition, deterministic encoder-decoder can be utilized for WAE.

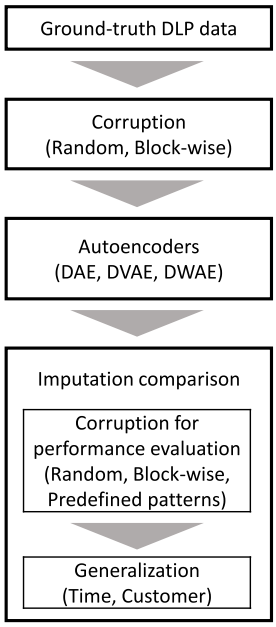


FIGURE 5. Overall procedure of DAE based missing value imputation.

IV. AUTOENCODER BASED MISSING IMPUTATION FRAMEWORK

In this section, we propose a DLP based missing value imputation framework by leveraging DAE. Let $\mathbf{x} = \{x_1, x_2, \ldots, x_{96}\}$ denote a daily load data of 15-minute interval, where x_t is metered value at time slot t. We assume $\mathbf{x}$ is ground-truth, i.e., there are no missing values in $\mathbf{x}$. Then, a DLP with missing values is denoted by $\hat{\mathbf{x}}$ where x_t is not available for some t. The objective of this framework is learning encoder network f and decoder network g by minimizing the error between the output $\tilde{\mathbf{x}} = g(f(\hat{\mathbf{x}}))$ and ground-truth $\mathbf{x}$, i.e., DAE. The overall framework of DAE based missing value imputation is illustrated in Figure 5. As can be seen, it consists of four major stages (data preprocessing, data corruption, autoencoder, and imputation comparison). Note that our scope is limited to the DLP based imputation. However, this can be further developed to longer missing data by utilizing adjacent load profiles.

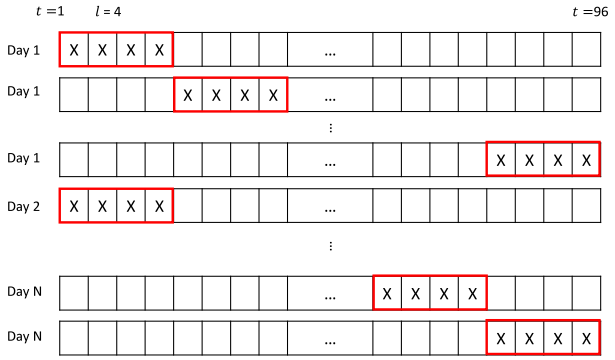


FIGURE 6. Example of block-wise missing with $l = 4$.

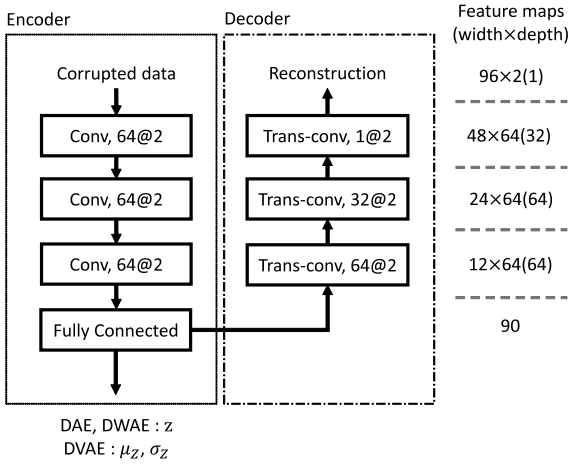


FIGURE 7. The CAE structure of denoising framework. The number and the size of filters are given in the form of *a@b*. The size of the feature map is given on the right and the parenthesis shows the depth in decoder.

A. GROUND-TRUTH DLP DATA CONSTRUCTION

We first build the ground-truth DLP data from the real dataset in Section II by replacing missing values with the average of previous time slots (e.g., hour and day ahead values). Note that this process is not a part of missing value imputation but just to make ground-truth dataset for experimental study. Then we normalize the load data by dividing the maximum value for each customer. Other normalization method (e.g., z-scores and max-min) can be applied with different missing status values.

B. DATA CORRUPTION

In data corruption stage, we generate the corrupted $\hat{\mathbf{x}}$ by enforcing artificial missing values to the ground-truth data $\mathbf{x}$. The number of all possible combinations of daily missing pattern is 2^{96}, which is impossible to train. Thus we utilize two corruption strategies, random corruption and block-wise missing as we observed the missing patterns in Section II. In random corruption, each value can be independently converted to missing value with given missing percentage (p_m). For example, if p_m is 50%, randomly selected half of measurements become missing values. In the case of block-wise missing, we generate artificial missing patterns according to the given missing duration l as illustrated in Figure 6. For example, there are 24 missing patterns per one DLP when $l = 4$; $\{x_t | t = 0, 1, 2, 3\}$ are missing in the first pattern, and $\{x_t | t = 4, 5, 6, 7\}$ are missing for the next.

Based on the above two corruption functions, selected artificial missing data are replaced by 0 to represent missing status. In addition, we attach binary missing status channel to DLP, which has 1 for missing and 0 for the others. Hence the dimension of input data is $1 \times 96 \times 2$. Finally, the DAE is trained with both random corruption and block-wise missing data.

C. MODEL SELECTION

Next, we train the DAE to reconstruct complete DLP from the above corrupted inputs. In doing this, two design options are *network structure* and *loss function*. First, we set modified 1-D convolutional autoencoder (CAE) based on the proposed structure in [7] and the validation result. Final CAE structure is illustrated in Figure 7. The CAE has three 1-D convolutional layers, one fully-connected layer, and three 1-D transposed convolutional layers, respectively. We set the dimension of latent vector $\mathbf{z}$ to 90. Leaky ReLU is used for the activation function, and the network is trained with ADAM optimizer [29].

After selecting the autoencoder structure (layers, neurons, and activation function), we build the network with three different loss function configurations based on DAE, VAE, and WAE. The overall loss function L consists of reconstruction loss L_{recon} and latent loss L_{latent} as follows.

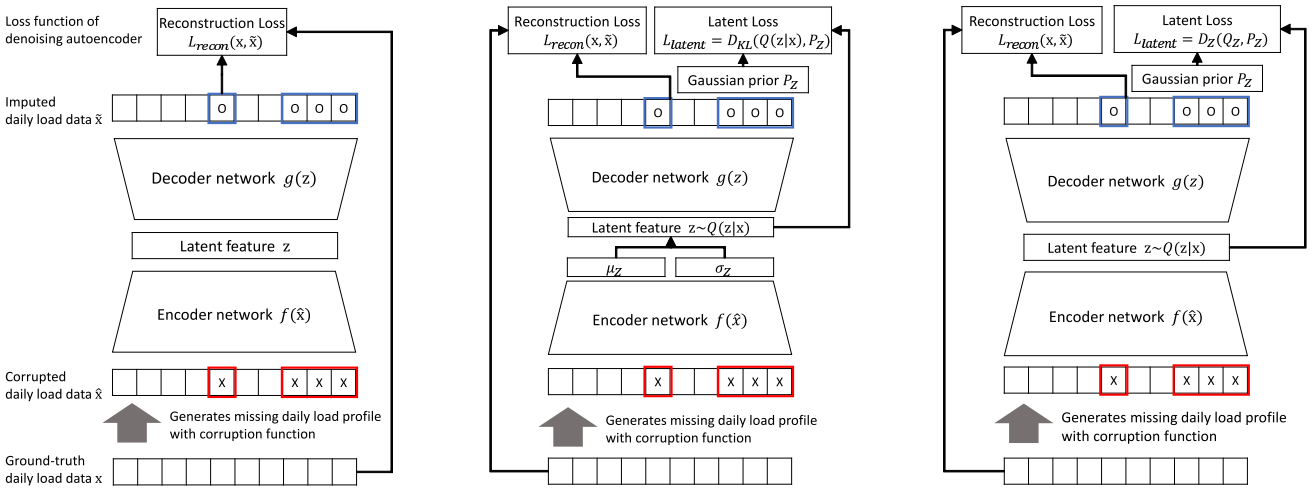
$$L = L_{recon} + \lambda L_{latent}, \tag{4}$$

where $\lambda > 0$ is a hyperparameter. The first case is with loss function for DAE which utilizes only L_{recon}, i.e., $L_{latent} = 0$. The second case is based on the VAE and L_{latent} is KL divergence between $Q(Z|X)$ and P_Z. The last one is based on the WAE where L_{latent} is arbitrary divergence $D_Z(Q_Z|P_Z)$. In this paper, maximum mean discrepancy (MMD) with inverse multi-quadratics kernel is utilized as in original work of [26]. In the rest of this paper, we call the above three autoencoders with different loss functions as DAE, denoising VAE (DVAE), and denoising WAE (DWAE), respectively.

In addition, we modify the reconstruction loss as follows.

$$L_{recon} = \alpha ||\mathbf{m} \circ (\mathbf{x} - \tilde{\mathbf{x}})||_2^2 + (1 - \alpha)||\bar{\mathbf{m}} \circ (\mathbf{x} - \tilde{\mathbf{x}})||_2^2, \tag{5}$$

where $\circ$ indicates element-wise product and $\mathbf{m}$, $\bar{\mathbf{m}}$ are binary mask vector (i.e., missing status channel in data corruption stage where 1 denotes missing and 0 for the others) and its complement. Hence, this gives more weight on the accuracy of imputed missing values rather than learning robust latent feature of corrupted DLPs. The overall structure of DAE and two generative models (DVAE, DWAE) for missing value imputation are summarized in Figure 8.



(a) DAE (the proposed method) (b) DVAE (c) DWAE

FIGURE 8. Summary of DAE, DVAE and DWAE methods.

D. IMPUTATION COMPARISON

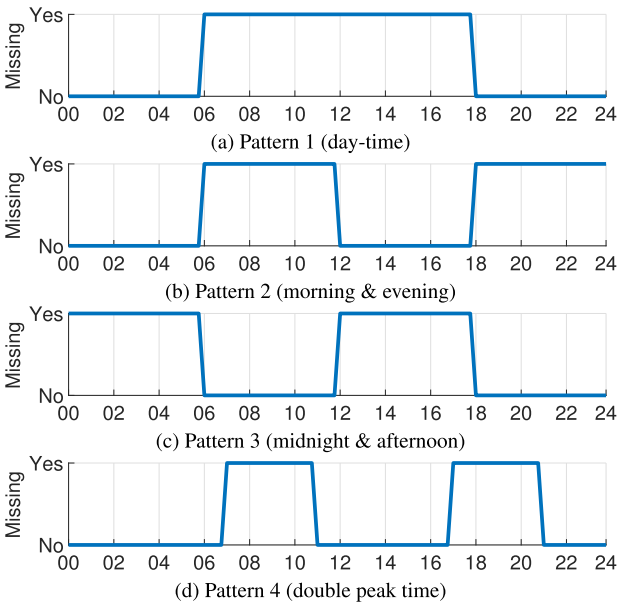
In the imputation comparison stage, we compare imputation performance with respect to 6 different configurations that are based on three corruption functions and two test set formulations. Corruption functions include random missing, block-wise missing, and predefined corruption patterns. The uniformly random and block-wise corruptions described in the data corruption stage are applied to test data. In addition, to compare the generalization according to the various missing patterns, we utilize 4 predefined missing patterns as illustrated in Figure 9; predefined patterns are not exposed during the training. From pattern 1 to 4, each missing pattern corresponds to missing during a) day-time, b) morning and evening, c) midnight and afternoon, d) double peak time, respectively.

V. RESULTS

A. EXPERIMENTAL SETUP

In the experiments, 15-minute interval electric load data of 60 residential customers are utilized. The dataset contains load profiles of total 360 days (from Mar. 1, 2016 to Feb. 23, 2017). Among the load data of 60 customers, 300 daily load data of customer 1 to 40 are used for training. The rest 60 daily load data of the *same* customers are used for test set, which is denoted as "Test 1". By contrast, 300 daily load data of remaining 20 customers are used for generalization over other customers and denoted by "Test 2". The details of the dataset configuration are described in Table 2, and parts of training data (i.e., Val.train and Validation) are used during model selection and hyperparameter setting.

During the training of autoencoders, we set block missing durations as $l \in \{4, 12, 24, 48\}$ and missing percentage for random corruption as $p_m \in \{5\%, 10\%, 20\%, 30\%, 50\%\}$, respectively. The networks are trained with 30 epochs according to the validation results. The hyperparameter λ and α are set to 1 and 0.8, respectively.



(a) Pattern 1 (day-time)
(b) Pattern 2 (morning & evening)
(c) Pattern 3 (midnight & afternoon)
(d) Pattern 4 (double peak time)

FIGURE 9. Predefined missing patterns (never experienced during training).

TABLE 2. Dataset configuration.

Dataset	Customers	Period (Days)	Total DLPs
Training	1∼40	Mar.1 ∼ Dec.25 (300)	12,000
Test 1	1∼40	Dec.26 ∼ Feb.23, (60)	2,400
Test 2	41∼60	Mar.1 ∼ Dec.25, (300)	6,000
*Val.train	1∼10	Mar.1 ∼ Dec.25, (300)	3,000
*Validation	1∼10	Dec.25 ∼ Feb.23, (60)	600

∗ training and validation sets for model and hyperparameter selection.

For the benchmark, we compare with the well-known methods of LI and HA. In doing this, we utilize two different HA configurations [30]; HA1 is based on the average of the values at time slots $t-1$ and $t-96$, and HA2 uses the average of the values at $t-96$, $t-1$, $t+96$. We evaluate imputation

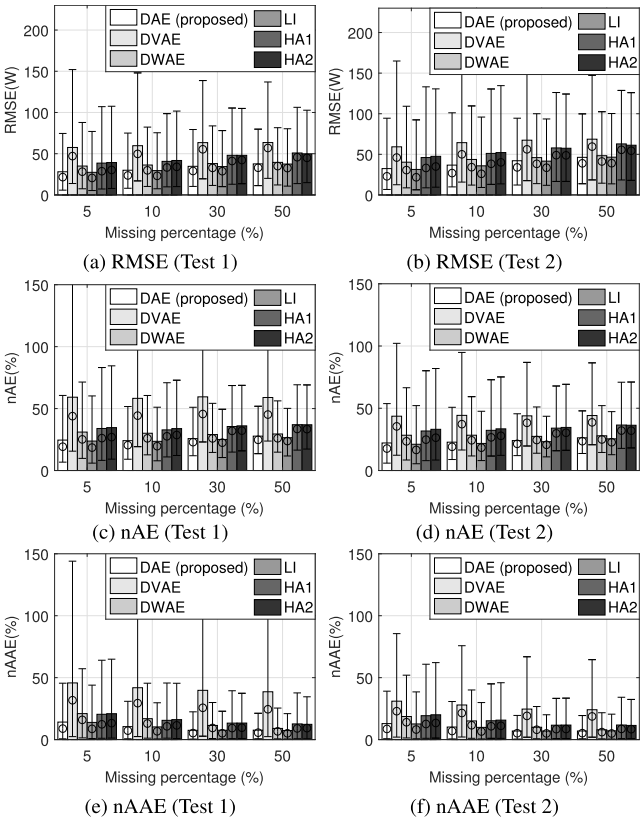


FIGURE 10. Bar plot of daily error $E_{i,d}$ of random corruption: mean and (5,50,95)-th percentile.

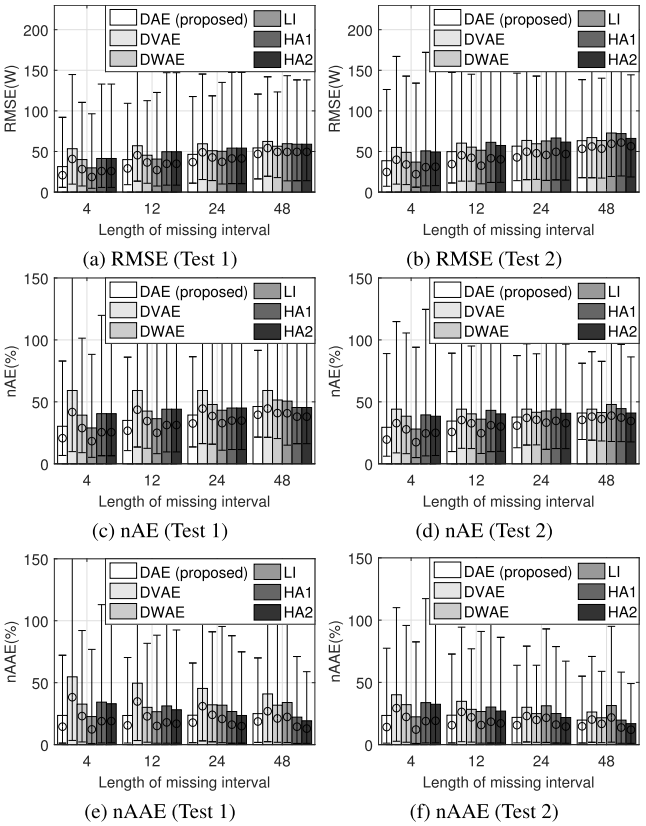


FIGURE 11. Bar plot of daily error $E_{i,d}$ of block-wise missing: mean and (5,50,95)-th percentile.

TABLE 3. Error metrics for imputation performance comparison.

Type	Metric	Definition
$E_{i,d}$, $i \in \mathcal{N}, d \in \mathcal{D}$	RMSE(W)	$\sqrt{\frac{1}{\lvert\mathcal{T}\rvert}\sum_{t\in\mathcal{T}}(x_t-\tilde{x}_t)^2}$
	nAE(W)	$\frac{100}{\lvert\mathcal{T}\rvert}\sum_{t\in\mathcal{T}}\frac{\lvert x_t-\tilde{x}_t\rvert}{\bar{X}_i}$
	nAAE(Wh)	$\frac{100}{\lvert\mathcal{T}\rvert\bar{X}_i}\lvert\sum_{t\in\mathcal{T}}x_t-\tilde{x}_t\rvert$
$E_i, i \in \mathcal{N}$ (Customer)		$\frac{1}{\lvert\mathcal{D}\rvert}\sum_{d\in\mathcal{D}}E_{i,d}$
E (Overall)		$\frac{1}{\lvert\mathcal{N}\rvert}\sum_{i\in\mathcal{N}}E_i$

performance in terms of RMSE (measured in W), and normalized absolute error (nAE). Since the proposed framework is based on DLP, we calculate the imputation error per DLP. Then, the customer and total errors are derived by averaging the values. Error metrics are given in Table 3, where x_t denotes the ground-truth value and , $\tilde{x}_t$ is the imputed value, $\mathcal{N}$ is a set of customers, $\mathcal{D}$ is a set of corrupted DLPs through the year and $\mathcal{T}$ is a set of missing time slot indices during a day. To check the imputation bias, we also compute the normalized accumulated absolute error (nAAE), and both nAE and nAAE are normalized by the average load of each customer (denoted by $\bar{X}_i$) to alleviate high MAPEs of near zero denominators.

Finally, based on the above implementation details, we compare the imputation performance of different methods for various missing patterns in the following sections.

B. CASE OF RANDOM MISSING

First, we compare the result of random corruption. Figure 10 shows overall errors when p_m is given by 5%, 10%, 30%, 50%, respectively. In the graph, bar indicates the average, ∘ indicates the median, and the lower/upper indicators are for 5th, 95th percentiles. Since missing occurs randomly, LI, the simplest method by linear interpolation mostly shows the best imputation performance in both point-wise errors (Figure 10 (a)-(d)) and accumulated errors (Figure 10 (e)-(f)), and the proposed DAE also shows similar performance in overall. Remaining four methods follows behind in the order of DWAE, HA2, HA1, and DVAE. Since each measurement is randomly changed to missing, the probability of long missing block in DLP (like $l = 48$) is low, even with high missing percentage (e.g., $p_m = 50\%$).

C. CASE OF BLOCK-WISE MISSING

Next, we compare overall imputation error of block-wise missing. The overall imputation errors for different missing durations ($l = 4, 12, 24, 48$) are given in Figure 11; $l = 4, 12, 24, 48$ correspond to 1h, 3h, 6h, 12h consecutive missing occurrence. Unlike the case of random missing, the proposed DAE shows the minimum errors in overall as

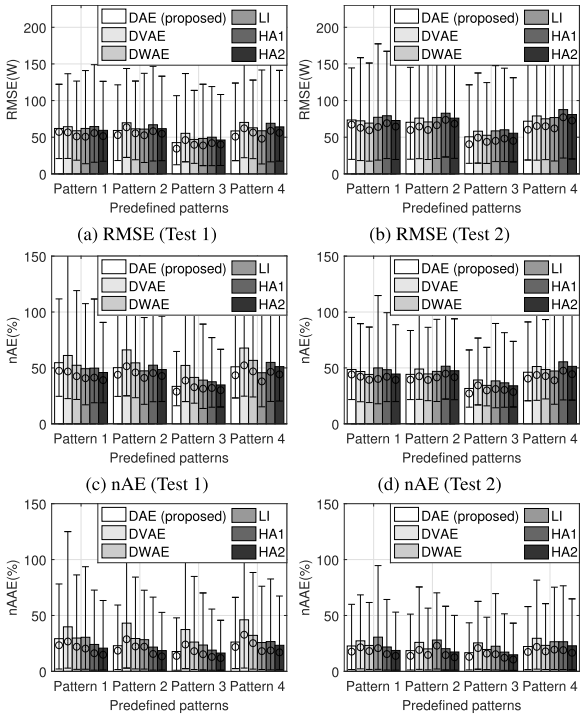


FIGURE 12. Bar plot of daily error $E_{i,d}$ of predefined missing patterns: mean and (5,50,95)-th percentile.

shown in Figure 11. DVAE, on the other hand, usually shows the worst results among all methods. Other methods show different aspects in terms of l and error measures. For example, LI shows low point-wise error for relatively short missing durations ($l = 4, 12$). HA2 shows the lowest *accumulated* error for long missing duration ($l = 48$). Interestingly, all methods have similar 5th percentiles, but difference in 95th percentile is prominent, and the proposed DAE has the lowest 95th percentiles in overall, which implies robustness of the proposed imputation framework.

D. CASE OF PREDEFINED MISSING PATTERNS

Finally, we analyze the imputation error with four predefined missing patterns as defined in Figure 9. Recall that these patterns are not exposed during training. As can be seen in Figure 12, the proposed DAE shows the lowest RMSE, whereas HA2 method shows the lowest accumulated errors, i.e., nAAE. Since the predefined missing patterns are composed of missing duration longer than $l = 24$ (6 hours), the result of LI method is relatively worse than the others, and HA methods show good performance. In all methods, Pattern 3 (midnight & afternoon) has lower error than Pattern 2 (morning & evening) in average, and this is because the imputation of morning and evening is more difficult compared to midnight and afternoon due to typical double peaks of residential energy consumption patterns.

E. DISCUSSION ON OVERALL RESULTS

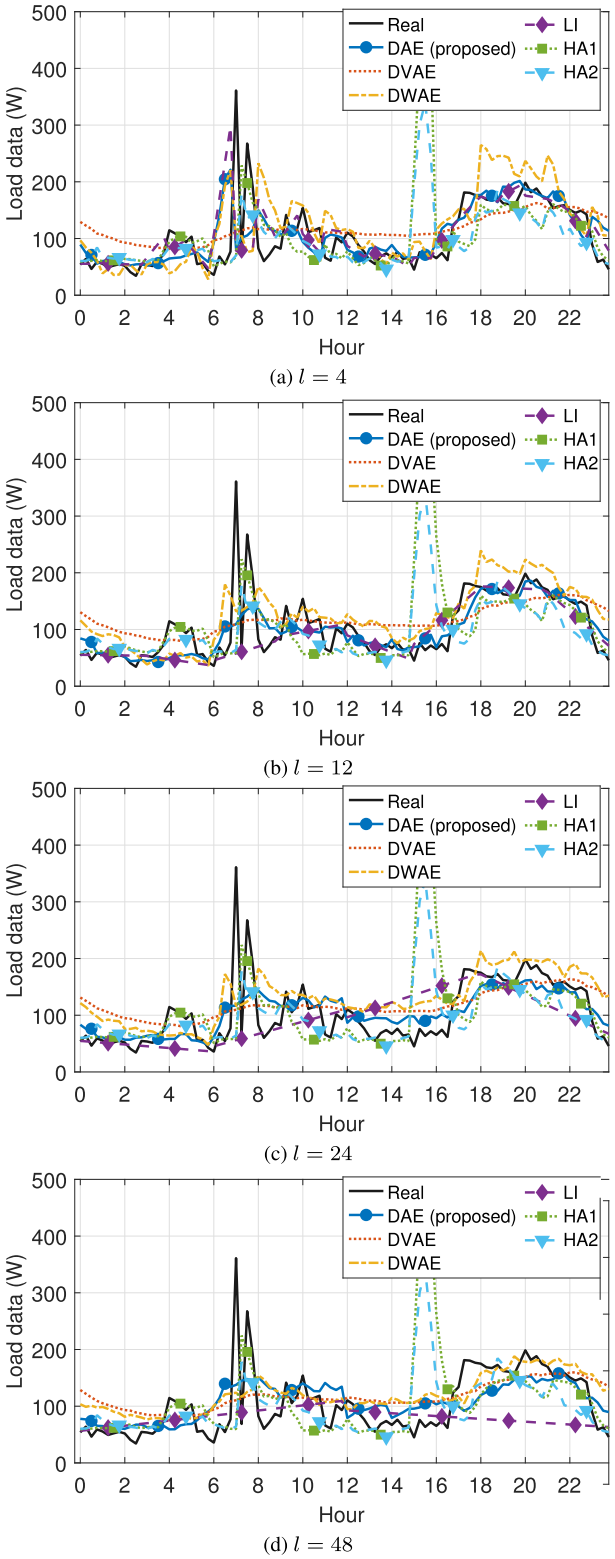
Table 4 summarizes the overall results. The proposed DAE shows lower imputation errors compared to the generative

TABLE 4. Summary of overall results.

Data	Missing	DAE (proposed)	DVAE	DWAE	LI	HA1	HA2
RMSE (W)							
Test 1	Random	32.8	61.2	37.7	32.2	44.6	44.8
	Block	43.2	58.2	46.1	45.1	51.1	47.8
	Pattern	55.1	65.0	57.3	57.6	62.6	57.9
	mean	43.7	61.5	47.0	44.9	52.8	50.1
Test 2	Random	39.6	64.9	44.8	38.8	54.5	54.6
	Block	52.3	61.5	54.9	56.0	62.7	58.6
	Pattern	66.2	71.4	68.2	72.4	77.5	71.3
	mean	52.7	66.0	56.0	55.8	64.9	61.5
nAE (%)							
Test 1	Random	25.7	59.3	30.2	24.8	34.8	35.3
	Block	37.9	59.5	41.6	39.8	43.8	41.0
	Pattern	46.6	62.2	47.7	45.4	48.7	45.1
	mean	36.7	60.3	39.8	36.6	42.4	40.5
Test 2	Random	24.0	44.1	28.2	22.9	33.6	34.4
	Block	35.9	44.2	37.9	38.6	42.8	40.1
	Pattern	42.1	46.7	42.3	45.7	48.0	44.4
	mean	34.0	45.0	36.2	35.7	41.5	39.6
nAAE (%)							
Test 1	Random	10.2	41.8	14.8	9.8	15.5	15.7
	Block	24.3	47.9	28.3	28.8	28.6	26.0
	Pattern	23.4	41.9	27.8	27.1	22.9	19.8
	mean	19.3	43.9	23.6	21.9	22.3	20.5
Test 2	Random	9.4	27.0	13.1	9.1	14.5	14.7
	Block	22.5	32.8	24.9	27.9	27.2	24.5
	Pattern	19.8	27.1	24.3	26.9	21.5	18.5
	mean	17.2	29.0	20.7	21.3	21.0	19.2

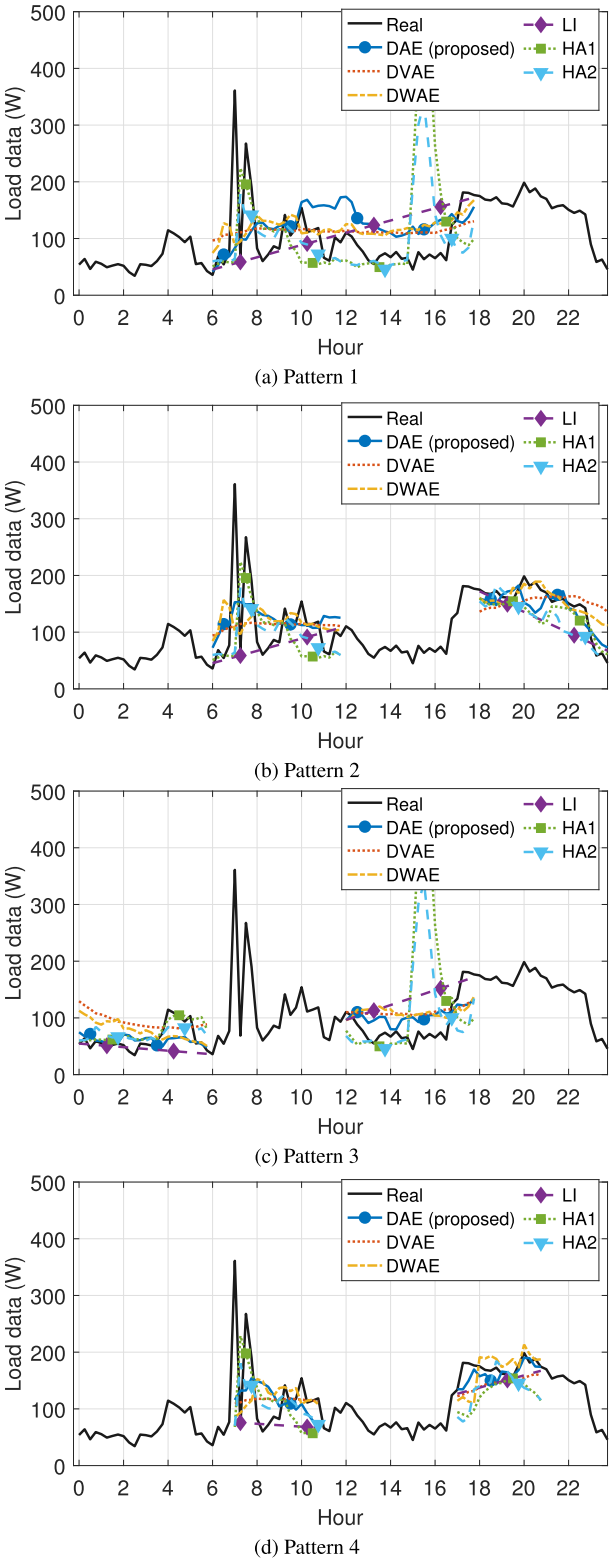
models. LI method shows good result with short missing duration (e.g., block-wise missing with $l = 4$, and random missing cases), and HA2 method has the lowest daily-accumulated errors for long missing sequences (e.g., block-wise missing with $l = 24, 48$ and missing with predefined patterns). However, regardless of the length of missing block, corruption functions and error metrics, the proposed DAE shows the lowest overall mean values than the others. For example, average RMSE of DAE is 43.7 (52.7) for Test 1 (Test 2), which is 2.7 $\sim$ 28.9% (5.5 $\sim$ 20.2%) lower than the others. In terms of daily-accumulated error of nAAE, the mean value of DAE is 19.3 (17.2) which is 5.8 $\sim$ 56% (10.4 $\sim$ 40.7%) lower than the others. This result implies the robustness of DAE based imputation, which is also observed in the bar plots of daily errors. According to the graphs, all six methods have similar 5th percentiles. However, 95th percentiles vary depending on the imputation methods and DAE mostly shows the lowest 95th percentile compared to the others.

Although DVAE and DWAE are in the form of DAE, the results are relatively worse than DAE, and the reasons are as follows. First, as can be observed in the reconstruction example of Figure 13, DVAE generates imputation that resembles averaged DLP regardless of different input. This reconstruction problem is a drawback of VAE, which is induced by the intersection of $Q(Z|X)$ [26]. Since Wasserstein autoencoder in [26] overcomes shortcomings of variational autoencoder, DWAE generates more plausible imputation



(a) $l = 4$ (b) $l = 12$ (c) $l = 24$ (d) $l = 48$

FIGURE 13. Example of load curve and missing imputation results of block-wise missing. The entire imputation graph is created by concatenating the imputation results of each missing section in the block-wise missing.



(a) Pattern 1 (b) Pattern 2 (c) Pattern 3 (d) Pattern 4

FIGURE 14. Example of load curve and missing imputation results of predefined patterns.

result and lower errors than DVAE. But the error is still higher than DAE. We observe that the error converges to that of DAE when decreasing the hyperparameter λ in (4). This result suggests that the probabilistic approach to the latent vector does not improve the performance within DLP (15 minute interval data) based imputation framework. Since the dimension of latent vector (90) is similar to the input

data (96), DAE may suffice within the proposed imputation framework, and the latent loss of DWAE (L_{latent}) works as a regularizer. However, DWAE can be utilized for multiple imputations.

Meanwhile, for the case of LI and HA, two methods are known to be good at short and long missing data respectively. However, LI method also shows low point-wise error for long missing data. For example, RMSE of $l = 48$ is similar to that of HA1. But this does not imply LI is suitable for long missing data, since LI shows worse results in terms of accumulated error. In addition, the imputation by LI is not plausible in terms of DLP shape; it is just a single straight line. For HA method, point-wise errors are worse than the others. This may be due to the variability of residential load, i.e., day-ahead electricity load at the same time slot can differ from current load. This result suggests that more intelligent imputation (or forecasting) method is required in dealing with highly variable load data (e.g., residential customer).

To visualize the imputation result, the load curve and missing imputation results of Customer 59 at Apr. 16, 2016 are given in Figures 13, 14 as examples. First, Figure 13 describes the imputation result of block-wise missing cases with $l = 4, 12, 24, 48$. Each line graph is concatenation of imputations in given missing blocks. For example, Figure 13(d) shows concatenation of $\{\tilde{x}_t | t = 0, \ldots 47\}$ and $\{\tilde{x}_t | t = 48, \ldots, 95\}$, hence two straight lines are observed for LI method. As l increases, it is hard to restore the original load curves by imputation. Finally, Figure 14 presents the imputed values of predefined missing patterns. Although it is hard to restore original peaks, imputation of DAE follows the trend of original load curve regardless of missing patterns. In all examples, DVAE provides the same, smoothed load curve that accounts for the highest error values. LI generates a single straight line which is not suitable for long missing sequence. HA methods generate unexpected peak around 15:00, and such difference in peak load leads to high point-wise error.

VI. CONCLUSION

In this paper, we proposed a novel missing imputation framework for electric load data leveraging the denoising convolutional autoencoder and showed that daily load profile (DLP) with multiple missing values can be accurately imputed. In doing this, we compared the proposed denoising autoencoder (DAE) with the traditional linear interpolation (LI), historical average (HA), and the recent two generative models (DVAE and DWAE). The proposed DAE outperformed the others in overall. Specifically, DAE has lower errors in both point-wise and daily-accumulated errors whereas the others have disadvantages in accuracy (DVAE, DWAE), shape and accumulated error (LI), unexpected peaks and point-wise error (HA), respectively. The imputation performance of the proposed DAE was verified with various missing patterns applied for residential DLPs, which are more random and volatile compared to industrial or commercial load patterns. Based on partially observed information, DAE exhibited accurate and robust missing imputation of corrupted DLPs. In addition, we compared three missing scenarios, i.e., uniformly random, block-wise missing, and predefined missing patterns. The proposed DAE framework showed $2.7 \sim 28.9\%$ lower point-wise RMSE and $5.8 \sim 56\%$ lower daily-accumulated nAAE than the others. The proposed method can be further developed in several directions, e.g., cluster-wise missing imputation with customer clustering or recurrent neural network based real-time missing imputation framework.

REFERENCES

[1] 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.

[2] A. Dedinec, S. Filiposka, A. Dedinec, and L. Kocarev, "Deep belief network based electricity load forecasting: An analysis of macedonian case," *Energy*, vol. 115, pp. 1688–1700, Nov. 2016.

[3] 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.

[4] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.

[5] L. Wang, Z. Zhang, J. Xu, and R. Liu, "Wind turbine blade breakage monitoring with deep autoencoders," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2824–2833, Jul. 2018.

[6] E. D. Varga, S. F. Beretka, C. Noce, and G. Sapienza, "Robust real-time load profile encoding and classification framework for efficient power systems operation," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 1897–1904, Jul. 2015.

[7] 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.

[8] Y. Choi, S. Ryu, K. Park, and H. Kim, "Machine learning-based lithium-ion battery capacity estimation exploiting multi-channel charging profiles," *IEEE Access*, vol. 7, pp. 75143–75152, 2019.

[9] B. J. Claessens, P. Vrancx, and F. Ruelens, "Convolutional neural networks for automatic state-time feature extraction in reinforcement learning applied to residential load control," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3259–3269, Jul. 2018.

[10] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Mag.*, vol. 17, no. 3, p. 37, Sep. 1996.

[11] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3125–3148, May 2019.

[12] Z. Jiang, D. Shi, X. Guo, G. Xu, L. Yu, and C. Jing, "Robust smart meter data analytics using smoothed ALS and dynamic time warping," *Energies*, vol. 11, no. 6, p. 1401, May 2018.

[13] 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 EventCovering method," *Neural Netw.*, vol. 23, no. 3, pp. 406–418, Apr. 2010.

[14] J. Peppanen, X. Zhang, S. Grijalva, and M. J. Reno, "Handling bad or missing smart meter data through advanced data imputation," in *Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT)*, Sep. 2016, pp. 1–5.

[15] M. Kim, S. Park, J. Lee, Y. Joo, and J. Choi, "Learning-based adaptive imputation method with kNN algorithm for missing power data," *Energies*, vol. 10, no. 10, p. 1668, Oct. 2017.

[16] G. Mateos and G. B. Giannakis, "Load curve data cleansing and imputation via sparsity and low rank," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 2347–2355, Dec. 2013.

[17] A. Al-Wakeel, J. Wu, and N. Jenkins, "K -means based load estimation of domestic smart meter measurements," *Appl. Energy*, vol. 194, pp. 333–342, May 2017.

[18] D. Kodaira and S. Han, "Topology-based estimation of missing smart meter readings," *Energies*, vol. 11, no. 1, p. 224, Jan. 2018.

[19] 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, Nov. 2018, Art. no. 16329.
[20] Y. Duan, Y. Lv, Y.-L. Liu, and F.-Y. Wang, "An efficient realization of deep learning for traffic data imputation," *Transp. Res. C, Emerg. Technol.*, vol. 72, pp. 168–181, Nov. 2016.
[21] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 85–100.
[22] 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, 2017.
[23] Y. Lin, J. Wang, and M. Cui, "Reconstruction of power system measurements based on enhanced denoising autoencoder," 2019, *arXiv:1907.11738*. [Online]. Available: http://arxiv.org/abs/1907.11738
[24] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1096–1103.
[25] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," Dec. 2013, *arXiv:1312.6114*. [Online]. Available: https://arxiv.org/abs/1312.6114
[26] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf, "Wasserstein auto-encoders," 2017, *arXiv:1711.01558*. [Online]. Available: http://arxiv.org/abs/1711.01558
[27] D. I. J. Im, S. Ahn, R. Memisevic, and Y. Bengio, "Denoising criterion for variational auto-encoding framework," in *Proc. AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 2059–2065.
[28] S. Zhao, J. Song, and S. Ermon, "InfoVAE: Balancing learning and inference in variational autoencoders," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 5885–5892, Jul. 2019.
[29] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, May 2015, pp. 1–15.
[30] S. Ryu, "Deep learning for electric load data analytics: Forecasting, feature extraction, and missing imputation," Ph.D. dissertation, Dept. Electron. Eng., Sogang Univ., Seoul, South Korea, 2020.

SEUNGHYOUNG RYU received the B.S., M.S., and Ph.D. degrees in electronic engineering from Sogang University, in 2014, 2016, and 2020, respectively. He is currently a Senior Research Engineer of the Intelligent Computing Lab, Korea Atomic Energy Research Institute, South Korea. His research interests are transactive energy, energy data analysis, energy forecasting, machine learning, and artificial intelligence in the smart grid.

MINSOO KIM received the B.S. degree in electronic engineering from Sogang University, South Korea, in 2019, where he is currently pursuing the M.S. degree with the Department of Electronic Engineering. His research interests include machine learning, artificial intelligence, and optimization in the smart grid.

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. He was a member of Technical Staff at Korea Telecom Labs, from 2000 to 2005. He was a Postdoctoral Research Associate with the Department of Electrical Engineering, Princeton University, from 2009 to 2010. He was also a member of Technical Staff at Bell Labs, USA, from 2010 to 2011. He is currently an Associate 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 *Journal of Communications and Networks* and a Guest Editor for *Energies* with the special issue of Machine Learning and Optimization with Applications of Power System.