# Decomposition-based Data Augmentation for Time-series Building Load Data


Yang Deng
Hong Kong Polytechnic Univ.
Hong Kong
yang2.deng@connect.polyu.hk

Rui Liang
Hong Kong Polytechnic Univ.
Hong Kong
19079853d@connect.polyu.hk

Dan Wang
Hong Kong Polytechnic Univ.
Hong Kong
dan.wang@polyu.edu.hk

Ao Li
Hong Kong Polytechnic Univ.
Hong Kong
18070016r@connect.polyu.hk

Fu Xiao
Hong Kong Polytechnic Univ.
Hong Kong
linda.xiao@polyu.edu.hk



## ABSTRACT

Building load data, i.e., building electricity demands, are important for many downstream applications such as load forecasting, demand response, and others. Recent applications, in particularly, those based on machine learning models, require a large amounts of data. Unfortunately, many buildings do not have sufficient data. To augment data, recent schemes are relying on generative adversarial networks (GANs). GAN-based schemes can generate new samples for the same distribution, i.e., to enrich data diversity. However, they are not suitable for augmenting the data with insufficient data distributions, e.g., a data shortage caused by insufficient time coverage, a common problem for new buildings. This paper aims to address this problem.

We propose a decomposition-based data augmentation scheme. Intrinsically, decomposition-based schemes assume that time-series data consist of several components. We analyze data from 407 buildings to understand whether and what components exist. This analysis gives us prior knowledge of the decomposable components. We then develop DAST with appropriately designed decomposition, augmentation, and combination schemes. We evaluate DAST using six real-world buildings, and show that the distribution of the data augmented by DAST matches the distribution of raw data and reduces the error by 47.8% as compared to state-of-the-art GAN-based schemes. We apply the data augmented by DAST to two building load forecasting tasks and find a 46.2% reduction in errors relating to forecasting.



## CCS CONCEPTS

• **Applied computing** → **Engineering**.

## KEYWORDS

Smart building, Machine learning, Data augmentation




*BuildSys '23, November 15–16, 2023, Istanbul, Turkey*



ACM ISBN 979-8-4007-0230-3/23/11…$15.00
https://doi.org/10.1145/3600100.3623727

**ACM Reference Format:**
Yang Deng, Rui Liang, Dan Wang, Ao Li, and Fu Xiao. 2023. Decomposition-based Data Augmentation for Time-series Building Load Data. In *The 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys '23), November 15–16, 2023, Istanbul, Turkey*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3600100.3623727


## 1 INTRODUCTION

The building electrical load (or "the building load" for short) refers to the electricity demand of a building. The building load is important for many applications, e.g., electricity consumption forecasting [15, 16, 29], waste identification [36], demand response [24], and others. However, for certain applications, particularly those based on machine learning, there is commonly a shortage of data.

To address the data shortage problem, data augmentation (DA) schemes have been proposed. For example, generative adversarial networks (GANs)-based schemes have been proposed for buildings, e.g., to augment data for load forecasting [2, 35], HVAC control [12, 28], and other purposes. GAN-based schemes are suitable for generating new samples for the same distribution; i.e., they can enrich data diversity so that ML models can be trained to avoid the problem of overfitting. GAN-based schemes, however, are not suitable for augmenting the data with insufficient data distribution.

This paper aims to address the problem of data augmentation for building load data with insufficient distribution of data, particularly, the situation of data shortages caused by insufficient time coverage in the collecting of data. This is common in practice. For example, new buildings have short data collection periods, e.g., two weeks. The building operators want to apply machine learning applications but they have yet to see full data distribution. In this paper, we leverage decomposition-based (DA) schemes [43]. In decomposition-based DA schemes, the assumption is that the data are synthesized by and can be decomposed into several *components*. A set of typical components, e.g., trend, seasonal, cyclical, and remainder have been studied [20]; and various decomposition techniques [6, 7] and data augmentation techniques for specific components [3, 13, 25] have been developed. To apply decomposition-based DA schemes in our scenario, the first challenge is to validate whether the building load data are indeed composed of well-studied components; and the second challenge is to develop and apply appropriate decomposition and augmentation schemes.

In this paper, we first analyze 407 building load data from the open source Genome project [34]. Our analysis shows that three components widely exist in certain buildings: (1) a daily-load component, reflecting daily variations. This is a repeated and periodic component; (2) a seasonal context component, reflecting yearly variations (e.g., summer, winter, etc.). This is also a repeated and periodic component; and (3) an irregular component, reflecting short irregular influences such as noises. These three components widely exist in certain types of buildings, which make up 76% of all buildings. For other buildings, our analysis shows that we need additional knowledge, e.g., about functional events. In this paper, we emphasize a problem in which we do not assume that we have extra knowledge beyond building load data, and we leave further investigations for future work. Our analysis demonstrates that we can apply decomposition-based DA schemes and quantify prior knowledge for decomposition.

We then propose DAST, a new decomposition-based DA scheme for building load data. The objective is to minimize the distance between the distribution of the generated synthetic load and the distribution of the real data in the target building. DAST has three phases: (1) load decomposition: to decompose the three abovementioned components based on the classical season-trend decomposition (STL) algorithm; (2) data augmentation: to augment the daily-load component using a clustering-assisted pattern mixing method; to augment the seasonal context component using a domain translation algorithm; and to augment the irregular component using a widely used kernel density estimation (KDE) scheme; and (3) component combination: to recombine the augmented components through a contrastive learning algorithm.

We evaluate DAST in two dimensions: (1) we study whether the generated synthetic data distribution and the real-world distribution are close. We use both quantitative and qualitative measures on the distance between distributions, i.e., maximum mean discrepancy (MMD) and t-SNE visualization. Our results show that DAST outperforms state-of-the-art GAN-based schemes by 47.8%; and (2) we study whether the data augmented by DAST can better support downstream applications. We develop two case studies by applying DAST to a load forecasting model training task and a load forecasting model testing task. The result is that DAST outperforms state-of-the-art GAN-based schemes by 46.2%.

The contributions of the paper can be summarized as follows:

- We for the first time study a building data augmentation problem where there is insufficient data distribution. This problem is important in practice, e.g., new buildings.
- We demonstrate that we can apply decomposition-based DA schemes by analyzing data from 407 buildings, and quantify the prior knowledge about three decomposable components in certain types of buildings.
- We develop DAST, a decomposition-based DA scheme with carefully designed sub-schemes on decomposition, augmentation, and combination.
- We evaluate DAST using real-world data. We further present case studies to show that the data augmented by DAST can improve the accuracy of downstream applications.

## 2 BACKGROUND AND RELATED WORK

**Building load and building data augmentation:** Given a building and its equipment, building loads can be measured using smart meters. It depends on the behavioral and environmental factors [44]. Behavioral factors relate to occupant behaviors, e.g., the number of occupants, occupant electricity requirements [40], office hours and arrival/departure periods [36], functional events [33], and others. Environmental factors include diurnal rhythms [36], outdoor weather conditions [36], seasonal changes [11], and others.

However, it is common to have a shortage of building load data. Data augmentation (DA) [1, 42] refers to the methods for augmenting the diversity of the data without collecting data. Building load data are time-series data. DA for time-series data includes basic technologies such as random transformation [42] and pattern mixing [23]; as well as advanced approaches such as generative models, e.g., GANs, which use the feature distributions in datasets to generate new samples [21], and decomposition-based methods, which first decompose the data into components, then augment the data for each component, and finally assemble a new time-series [43].

Studies have been conducted on building data augmentation. TimeGAN was developed to generate building load profiles for HVAC control [12]. Wasserstein GAN was developed to generate time-series energy data for large buildings [35]. Conditional GAN was developed to generate load data in the multiple buildings [2]. Current studies rely heavily on GAN-based schemes [21]. GAN-based schemes are suitable for generating new samples given the data distribution, i.e., to enrich data diversity. With enriched data, machine learning models, for example, can be trained to avoid overfitting problems.

GAN-based schemes, however, are not suitable for generating data with insufficient data distribution, i.e., a data shortage caused by insufficient time coverage, a situation that is common in new buildings. Addressing this problem is the aim of this paper.

**Decomposition-based Data Augmentation:** A decomposition-based DA scheme assumes that the time-series can be regarded as a collection of observations with several components, each representing one of the underlying categories of patterns. The typical components are [43]: (1) trend components: reflecting long-term progression; (2) seasonal components: reflecting repeated and periodic fluctuations; (3) cyclical components: reflecting repeated and non-periodic fluctuations; and (4) irregular components: reflecting irregular influences, e.g., noise.

Intrinsically, the decomposition-based DA schemes [13, 25] assume that components exist. With prior knowledge of the components, augmenting the data of each component can be a more targeted process. In many practical applications, components exist and prior knowledge can be analyzed.

In addition to the division of components mentioned above, time-series can also be decomposed into multiple signals with different frequency ranges and amplitudes, which is widely applied in frequency-domain augmentation [47] for non-linear signals.

## 3 ANALYZING THE COMPONENTS OF A BUILDING LOAD TIME-SERIES

As mentioned, building load is influenced by environmental factors (e.g., diurnal rhythms, climate) and occupant behaviors (e.g., the

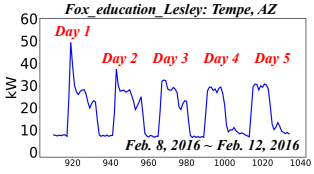



**Figure 1: The daily-load examples.**

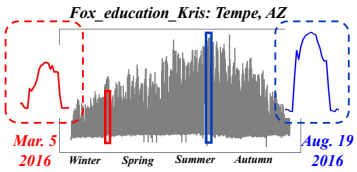



**Figure 2: Load variation in seasonal contexts.**

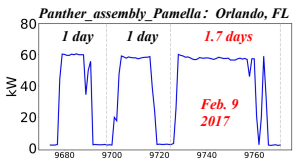



**Figure 3: Non-periodic event occurs.**

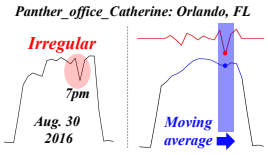



**Figure 4: The irregular(s) in the load**

number of occupants, office hours). To apply decomposition-based DA, we need to study how these factors can be classified into the aforementioned typical components in decomposition-based DA. Our analysis in this section serves this purpose.

**Dataset:** We analyze the building load data from Genome [34], an open-source building data dataset. This dataset contains hourly electrical meter data from 1636 buildings for two years (from 2016 January to 2017 December). The Genome dataset covers the USA, Canada, European countries, and others. The primary use of the buildings can be classified into six types: Education, Office, Assembly, Lodging, Public services, and Others. In this paper, we study the first five types of buildings. We note that data is missing for some of the buildings, which can have a negative effect on our analysis. We remove those buildings for which more than 1% of the data are missing. For the sake of brevity, we also focus on the buildings in the USA, although we plan to analyze buildings in other countries in the future. We ended up with 407 buildings for our analysis.

**Analysis:** We now study whether building data are composed of the aforementioned components.

*The Trend Component:* reflects long-term progression in the building in terms of its functions and occupant supports. To estimate the trend component, in addition to a time-series of building load data, additional knowledge on the building is needed, e.g., the rate at which equipment is aging. Since we do not have additional knowledge, we do not differentiate this component. As we show later in our analysis, other components dominate the building load data and the impact of the trend component is minimal, even if it exists.

*The Seasonal Component:* reflects repeated and periodic fluctuations. We observe two types of repeated and periodic fluctuations:

*(1) Daily-load*: is the load variation in a period of a day (see an example in Figure 1). This reflects both behavioral (e.g., office hours) and environmental (e.g., diurnal rhythms) factors. The Lawrence Berkeley National Laboratory (LBNL) proposed a *five-parameter* method [36] that, given building load time-series data, estimates the proportion of daily loads. The five parameters are (1) the base load, (2) the rise time, (3) the peak load, (4) the high-load duration, and (5) the fall time. We develop a simple rule-based traversal algorithm to roughly detect these parameters and thus, the daily loads.

In Table 1, we show the proportion of the daily-load component for each building type, denoted by $Ratio_d$. For example, in Education buildings $Ratio_d$ = 91%, i.e., 91% of the building load data in Education buildings have daily periods. This conforms to intuition since Education buildings have clear daily activities. Thus, Education buildings contain a daily-load component. On the contrary, in Lodging buildings $Ratio_d$ = 68%. This means that many Lodging buildings do not contain a clear daily-load component.

**Table 1: The statistics of the components in the 407 buildings.**

| Building type | $Ratio_d$ | $Ratio_s$ | $Ratio_{ir}$ |
|---|---|---|---|
| *Education (47%) | 91% | 86% | 94% |
| *Public (17%) | 87% | 88% | 91% |
| Assembly (14%) | 78% | 65% | 97% |
| *Office (12%) | 88% | 89% | 95% |
| Lodging (9%) | 68% | 84% | 97% |
| **The three types (76%)** | 89% | 86% | 93% |

*(2) Seasonal context:* is the load variation in the period of a year that contains multiple (e.g., four) seasons (see an example in Figure 2). This reflects both behavioral (e.g., changes in dress) and environmental (e.g., climate switch) factors. We design a simple statistical algorithm to detect the load seasonal variations in the load. This algorithm estimates the similarity in the building load data across multiple years. If the load data in multiple years are similar, then we say that there is seasonal context in this building load data. This is a rough estimation, and we conduct this statistic in the first year and the second year (data on the buildings in the Genome are two years in length). In detail, the yearly-level similarity is calculated using a normalized Euclidean distance with the threshold set to 0.2 at the monthly granularity. Note that our goal here is to estimate the proportion of buildings with seasonal contexts, not to accurately determine the seasonal contexts. From our observation, it is apparent that buildings with seasonal characteristics exhibit consistent patterns in their daily-load during different seasons, such as winter and summer. For example, Figure 2 shows that the pattern on August 19, 2016 appears to be a scaled version of the pattern observed on March 5, 2016. This observation indicates the presence of seasonal context in this specific building.

In Table 1, we show the proportion comprised by the seasonal context component for each building type, denoted by $Ratio_s$. For example, in Education buildings, $Ratio_s$ = 83%, i.e., 83% of Education buildings show load variations with changes in seasonal context. Thus, Education buildings contain a seasonal context component. By contrast, in Assembly buildings $Ratio_s$ = 65%.

*The Cyclical Component:* reflects repeated and non-periodic fluctuations. This may be due to human events (e.g., multi-day conference events) or environmental factors of weather conditions that are longer than one day (e.g., wildfire, typhoon). For example, in Lodging buildings, the proportion of the daily-load is 68%. We conjecture that there are cyclical components. We investigated some data in detail and we see building loads as in Figure 3, even though we cannot tell the exact reason/event giving rise to the load. This

is a multi-day building load and such events do not appear periodically. Capturing cyclical components requires additional knowledge (e.g., building activity schedules). In this paper, we assume that we do not have such additional knowledge. Thus, we cannot decompose cyclical components. We leave the decomposition of cyclical components to a future work.

*The Irregular Component:* reflects noises for a short duration, i.e., spike/dip loads within a stable load period (see the example in the left side of Figure 4). We develop a simple smooth window algorithm to detect irregular components. This algorithm performs auto-regression in a window to self-construct regular loads and identify the irregular components (see the right side of Figure 4).

In Table 1, we show the proportion of the irregular components for each building type denoted by $Ratio_{ir}$, i.e., the proportion of days with obvious noise. We observe that all building types have irregular components.

**Summary:** To apply decomposition-based DA, we need to show that components do exist in building load data and we need prior knowledge of the components. Our analysis in Table 1 shows that the daily-load component, seasonal context component, and irregular component are widely present in Education, Office, and Public buildings. No additional knowledge is needed for these three components. Education, Office, and Public buildings make up the majority of buildings (76%). This is the fitted scenario for this paper. There are buildings with other components, e.g., cyclical components. Our scheme is not suitable for those buildings. With additional prior knowledge, more fine-grained decomposition-based DA schemes can be developed, which we will explore in a future work.

## 4 DESIGN OVERVIEW

**Problem Statement:** We aim to solve the building load data augmentation problem for buildings with an insufficient distribution of data, i.e., a shortage of data caused by insufficient time coverage, which is common for new buildings with a short data collection period. Our objective is to minimize the distance between the distribution of the synthetic load and the distribution of the real data.

We first present the notations for daily-load, seasonal context and irregulars. Let $p$ be a daily-load. Note that a seasonal context is intrinsically a stretching or a shrinking of a daily-load. For example, the spring season and summer season have similar daily-load patterns, yet the summer season stretches the electricity load (see Figure 2). As a result, we can let $C(p)$ be a daily-load with a seasonal context. We also use $C_{spring}(p)$, to $C_{winter}(p)$ to denote the daily-load at a specific season. Let $R$ be the irregular component. Finally, we can let the building load be $D(C(p), R)$.

We now present DAST, a new augmentation scheme based on time-series decomposition for building loads. Intrinsically, given building load data of $D(C(p), R)$ for a time period of $[t_0, t_k]$, DAST can generate $D'(C(p), R)$ in period $[t_0, t_n]$ where $n >> k$. DAST (Figure 5) has a Load Decomposition module §5.1 to decompose $D$ into $C(p)$ and $R$. Then, DAST has three augmentation modules: Daily-load Augmentation §5.2 , Seasonal Context Augmentation §5.3, and Irregular Augmentation §5.4, for $p$, $C(\cdot)$, and $R$, respectively. Finally, DAST has a Component Combination module §5.5 to combine the augmented components.

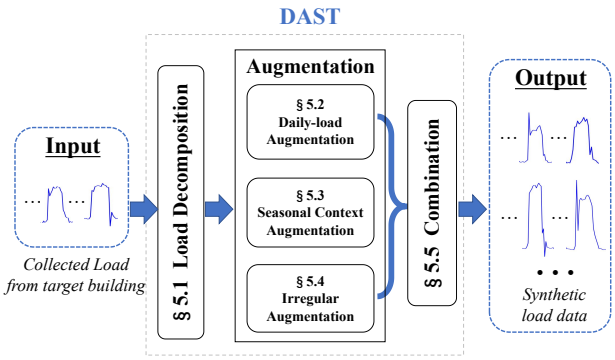



**Figure 5: The pipeline of decomposition-based load data augmentation (DAST).**

Note that after our analysis, we have two repeated and periodic components (daily-load and seasonal context), as well as an irregular component. Individual decomposition, augmentation, and combination algorithms of these types of components have been widely studied [3, 13, 23, 25]. We leverage existing algorithms and briefly introduce these five modules.

**Load Decomposition:** This module decomposes $D$ into $C(p)$ and $R$. This requires capturing $C(p)$, $R$ and avoiding misidentification between $C(p)$ and $R$, for example, the identification of a morning catch-up load into a noise. We develop a decomposition algorithm based on the season-trend decomposition (STL) algorithm [6], which is a classical time-series decomposition method.

**Daily-load Augmentation:** Based on $C(p)$ in one season-context, this module generates a set of synthetic daily-load $C(p')$. This requires $C(p')$ to span diversity (to enrich the daily-load data), yet maintain fidelity (to avoid deviating from the true daily-load distribution). We adopt a pattern-mixing method, which is widely used to generate diverse time-series. To achieve more fidelity, we develop an additional clustering-based mechanism for clustering similar $p$ before conducting pattern mixing.

**Seasonal Context Augmentation:** This module generates $C_{\neg a}(p)$ given $C_a(p)$, where $C_a$ is one of the four seasonal contexts and $C_{\neg a}$ is any one of the other seasonal contexts. This is a data transformation problem and the transformation process contains multiple operations, e.g., scaling, warping, and others. We develop a learning-based domain translation [50] algorithm, which is widely used for transforming the context of data (in DAST, the seasonal contexts). In addition, to improve fidelity, a multi-tasks algorithm is adopted to select training data to train the domain translation models.

**Irregular Augmentation:** This module generates irregulars of different sizes, i.e., the length of spikes or dips. This requires estimating the underlying probability distribution of the size of the irregulars. We adopt a kernel density estimation (KDE) scheme.

**Component Combination:** This module loads the augmented $C(p)$ and $R$ into a building load $D'$. This requires fidelity, i.e., we should avoid a situation where the meaning of the $C(p)$ is altered after combining $R$. We develop a contrastive learning [37] algorithm, which is widely used in decomposition-based augmentation for judging whether or not the combination is acceptable.

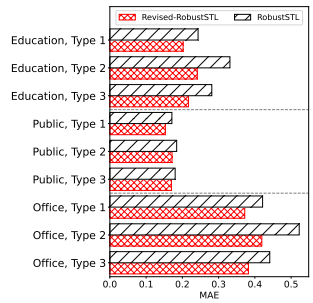



**Figure 6: Evaluation of the decomposition scheme**

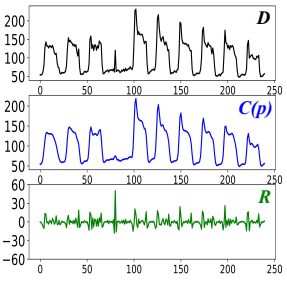



**Figure 7: An example of decomposition results**

# 5 DECOMPOSITION-BASED DATA AUGMENTATION SCHEME

## 5.1 Load Decomposition

Given the load time-series data of a target building: $\{D\}$, we need to decompose $\{D\}$ into $\{C(p)\}$ and $\{R\}$, i.e., the sequence of daily profiles and irregulars. Our goal is to minimize both distance of the decomposed $R'$ and the real $R$, as well as the distance of decomposed $C(p)'$ and real $C(p)$, i.e.,

$$\frac{1}{|\{D\}|} \sum_{C(p),R \in \{D\}} [\alpha \times l(C(p)', C(p)) + (1-\alpha) \times l(R', R)] \quad (1)$$

, $l(\cdot,\cdot)$ denotes the loss function and we use MAE in this work, $\alpha$ is the weight of the two components (we set $\alpha$ as 0.5). Note that $\alpha$ can be carefully set, and we will put it to future work.

Time-series decomposition has model-free and model-based solutions. Model-free methods do not make assumptions on patterns in the data, whereas model-based methods assume that the data has intrinsic patterns. Model-based methods model the patterns and conduct decomposition based on patterns. Clearly, building load data have patterns. Therefore, we propose a model-based method. We leverage the idea of the classical season-trend decomposition STL method, which has been successfully been used to decompose seasonality and noises in other time-series, such as $CO_2$ and electricity prices. Our rationale follows STL, i.e., to model the seasonality component $C(p)$, and the remainder is $R$. Our algorithm works as follows.

Step 1: DAST focuses on buildings with clear daily seasonality; hence, the season window is 24h. Then, based on the given $D$, a LOESS regression is applied for each timestamp (e.g., all the loads at 8 am in $\{D\}$); hence, a smooth general daily seasonal pattern is extracted. Note that, all of the processes in this step follow the original STL algorithm.

Step 2: Considering that there are different $p$ in $\{D\}$, i.e., different daily behaviors. This corresponds to *seasonality fluctuation* [9, 41] in the time-series decomposition field, and one widely used algorithm is RobustSTL [41]. In RobustSTL, an additional daily seasonality calibration mechanism is designed, where one $C(p)$ is calibrated by the weighted linear combination of the load within $k$ neighbor $D$, and the closer time points and similar $D$ are assigned greater weights. We drew inspiration from RobustSTL to adapt to the characteristics of the building domain, where the calibration will skip the neighborhood $D$ if the type of day is different (weekend/holiday and weekday), since this factor greater affects the accuracy of decomposition in the buildings (e.g., the accuracy on Monday and Friday).

**Decomposition evaluation:** To validate the effectiveness of our algorithm, we simulate $\{\hat{D}\}$ by the pre-prepared $\{C(\hat{p})\}$ and $\{\hat{R}\}$, i.e., the de-noised load by the moving-average (the same as the statistical method in the irregulars analysis) and simulated irregulars. Note that this is a common evaluation methodology because it is difficult for a real-world time-series to have the ground truth on the decomposed components [41]. For $\{C(\hat{p})\}$, we can get various load profiles from Genome buildings. For $\{\hat{R}\}$, we simulate three types of $\{\hat{R}\}$ 1) white noise, 2) obvious spikes/dips but (with the statistic on frequency from the analysis section), and, 3) a mix of the first two types. These synthetic irregulars are widely used to evaluate decomposition algorithms. We conducted the evaluation in the 300+ studied buildings, which were education, public, and office buildings.

As shown in Figure 6, our decomposition algorithm achieves better accuracy than the baseline RobustSTL[1] in all three types of buildings, as well as in the three types of simulated testing data. An example of a period of decomposition is shown in Figure 7.

The subsequent augmentations are based on the decomposed $\{C(p)'\}$ and $\{R'\}$.

## 5.2 Daily-load Augmentation

For a target building, given a period of a decomposed seasonality component, i.e., $\{p\}$ [2], our goal is to generate synthetic $p$, and the density $\hat{p}(p)$ can best approximate $p(p)$ for the target building (in the same season). This process can be formulated as:

$$\min_{\hat{p}} Dis(\hat{p}(p) || p(p)) \quad (2)$$

, where $Dis(\cdot,\cdot)$ is a measure of distance between distributions and we apply the maximum mean discrepancy (MMD) [38].

Considering that the size of $\{p\}$ may be insufficient for the learning-based augmentation scheme, then the $p$ augmentation scheme would resort to a conventional transformation, such as random transformation or pattern mixing. Random transformation involves conducting the operations like rotation and scaling without making assumptions about the inherent pattern, while pattern mixing involves combineing two or more *similar* samples to produce new ones [22]; hence, synthetic data can be generated with high fidelity.

One challenge in applying pattern mixing is that similar $p$ needs to be identified in advance. To address this, we develop a $p$ clustering sub-scheme to group the similar $p$ that correspond to similar behaviors. This sub-scheme helps in identifying patterns that can be mixed effectively during augmentation. Therefore, our $p$ augmentation process has two steps:

**$p$ clustering:** A clustering mechanism is designed to cluster the similar $p$. We use the k-means algorithm and choose DTW (Dynamic time warping) as the similarity metric, which can handle temporal distortions and find the best alignment of two $p$. Note that the number of clusters depends on the size of $\{p\}$, i.e., the given time-series from the target building. We leverage the Calinski-Harabasz (CH) Score to set the cluster number, which offers a

[1]We omit the results of the original STL algorithm which had a poor performance.
[2]We omit the seasonal context $C(\cdot)$ because they are the same in this part.

trade-off between separation and cohesion. Note that, we conduct $p$ clustering on the source dataset[3], and that all the $p$ are normalized by its peak load. As a result, we obtained 16 clusters. Therefore, each $p$ in the set $\{p\}$ from the target building can be associated with one of these clusters with the shortest distance to $p$.

**Pattern mixing method:** We apply a classical *pattern mixing* method called SMOTE [5] to mix the data in the magnitude domain. This is how it works: First, one $p$ is selected from the given set $\{p\}$, and another $p_c$ is sampled from the cluster $cluster_p$ corresponding to $p$. Then, a new $p'$ is generated by

$$p' = p + \lambda(p - p_c) \tag{3}$$

, where $\lambda$ is a random sampling value to give the synthetic data diversity (with a range of $(0, 1]$ [5]). Finally, the synthetic $\{p'\}$ are generated for the target building.

## 5.3 Seasonal Context Augmentation

For a target building, given a short period of decomposed $\{C_a(p)\}$ corresponding to one specified season context $C_a(\cdot)$, our goal is to generate $C_{\neg a}(p)$, i.e., the synthetic daily profiles in other season contexts, and the density $\hat{p}(C_{\neg a}(p))$ can best approximate $p(C_{\neg a}(p))$ for the target building. It can be formulated as:

$$\min_{\hat{p}} Dis(\hat{p}(C_{\neg a}(p)) \,||\, p(C_{\neg a}(p))), \quad \forall C_{\neg a} \in U - C_a. \tag{4}$$

, where $Dis(\cdot, \cdot)$ represents the measurement of MMD. The Universe $U$ includes four seasonal contexts, i.e., spring, summer, autumn, and winter (refer to ASHRAE [19])[4].

The core of this task is to modify the $C(\cdot)$ of $p$. We suggest that *Domain Translation (DT)* techniques are well suited for this purpose. DT is a special GAN-based technique for finding mappings of $domain_1 \leftrightarrow domain_2$ such that the mapping yields meaningful pairings [30], and thus the input and output can share certain *content* features, i.e., $p$, even though they may differ in *style*, i.e., $C(\cdot)$. DTs are widely applied in seasonal/climate context change and sensor-related time-series transformation scenarios [4, 17, 32].

Based on the previously introduced $p$ clustering, which groups the normalized daily profiles, i.e., roughly seasonal-agnostic $p$, we can leverage DT to transform $C(\cdot)$ for each $p$ cluster. Note that the given $\{C_a(p)\}$ usually covers a small subset of the $p$ clusters; thus, only the corresponding DT models need to be prepared for a target building (details of the training are shown later).

**DT-based $C(\cdot)$ change.** We leverage the idea from one classical DT model, UNIT [31], which aims to embed the data from two domains into a common shared latent space. We assume that this idea is suitable for our scenario since our goal is to "translate" between $C_a(\cdot)$ and $C_{\neg a}(\cdot)$ pair while sharing the same $p$. Figure 8 (left) shows the structure of domain translation. $E_1$ and $E_2$ denote two autoencoders that project the $C(p)$ from the associated domains to a shared-latent space, which is the $C$-agnostic of the $p$ representation. Suppose $C_1(p)$ and $C_2(p)$ are two daily profiles with the same type of $p$. Ideally, $E_1$ and $E_2$ would encode them to the same embedding that represents $p$. This embedding can be translated back to Domain 1 and Domain 2 by two specific generators $G_1$ and $G_2$, respectively. Moreover, there are discriminators $D_1$ and $D_2$ for $G_1$ and $G_2$ in GAN [21], and we omit them in Figure 8 for the sake of brevity.

[3]We select multiple buildings from education, office, and public service as the source buildings to support the learning-based augmentation schemes.

[4]Each season is represented by a three-month period. For example, the winter season corresponds to (Dec, Jan, Feb), and so on.

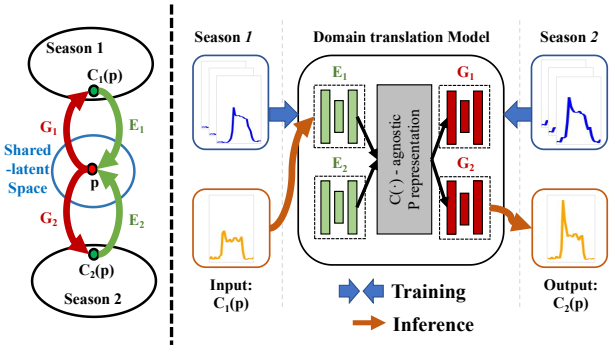



**Figure 8: Seasonal augmentation process.**

Note that all $E_i$, $G_i$, and $D_i$ are neural networks, and that the learning objective of the DT model is to optimize the following three associated loss functions.

*1) Reconstruction loss:* $C(p)$ reconstruction can be denoted as $\{E_i, G_i\}_{i=1,2}$. That is, the generated $\hat{C(p)} = G_i(E_i(C(p)))$ can be regarded as to reconstruct the original $C(p)$. Considering the time-series property, we leverage both DTW and MAE to evaluate the shape similarity of reconstruction. The loss function can be expressed as follows: here we use $x$ to replace $C(p)$ for clarity, and $\alpha$ is a hyperparameter (we set it as 0.5):

$$L_{Rec_i}(E_i, G_i) = \alpha \cdot DTW(\hat{x}, x) + (1 - \alpha) \cdot MAE(\hat{x}, x) \tag{5}$$

*2) GAN loss:* to achieve the equilibrium point in the minimax game for $\{D_i, G_i\}_{i=1,2}$. And the loss function $L_{GAN_i}(D_i, G_i)$ follows GAN [21].

*3) Cycle-consistency loss:* The core of DT is the invariance of the domain theory in topology, i.e., $x_1 \rightarrow G_2(E_1(x)) = \hat{x}_2$ and then $\hat{x}_2 \rightarrow G_1(E_2(x)) \approx x_1$, in which the loss is denoted as $L_{CC_1}(E_1, G_2, E_2, G_1)$ (details are given in [50]) and vice versa.

The total loss can be summarized as Formula 6. This ultimately corresponds to solving the domain translation model $f_\theta$ according to the optimization problem as Formula 7.

$$\mathcal{L}_{trans} = L_{Rec_1} + L_{Rec_2} + L_{GAN_1} + L_{GAN_2} + L_{CC_1} + L_{CC_2} \tag{6}$$

$$f_\theta^* = \arg\min_{f_\theta} \min_{\{E_1,E_2,G_1,G_2\}} \max_{\{D_1,D_2\}} \mathcal{L}_{trans} \tag{7}$$

**DT models training.** For the target building, specified DT models are trained using data from the source building and only buildings with properties similar to those of the target building are suitable. Thus, we seek another building-level clustering algorithm to select the source buildings. We leverage a metadata clustering-based method [49], which designs a two-phase clustering scheme and performs well in building-related tasks. The metadata used in our scenario is the building location, building type, and the id of the $p$ clusters that are involved.

## 5.4 Irregular Augmentation

The decomposed irregular component $\{R\}$ (shown in Figure 7) is a noise sequence that contains white noise and an irregular spike/dip at each timestamp, with each irregular point having a certain size

(or height). In this section, our goal is to estimate the underlying distribution of the size of the irregulars for the target building, while given a limited $\{R\}$ for the period of $[t_0, t_k]$.

In existing works on decomposition-based time-series augmentation, the augmentation schemes for irregular component is usually based on various statistics-based methods [3, 25]. We adopt kernel density estimation (KDE), a nonparametric statistics method, to estimate the probability density function (PDF) by setting a kernel function for each given piece of data. KDE is used in scenarios where the samples are limited and the overall distribution is not the normal distribution. We use the Gaussian distribution as the kernel function, which is widely used in KDE. After fitting the KDE model using the given $\{R\}$, we can sample the irregulars for the generated $\{R'\}$ for a long period.

## 5.5 Augmented Components Combination

Given the synthetic daily profiles $\{C(p)\}$ and irregular sequences $\{R\}$ ($R$ is one day in length), the combinations are denoted as $D = C(p) + R$. Our goal is to recognize and remove the low-fidelity $D'$, and the remaining $D$ can then reproduce the real distribution, i.e.,

$$\min_{\hat{p}} Dis(\hat{p}(D)||p(D)) \tag{8}$$

, where $Dis(\cdot,\cdot)$ is the MMD measurement.

We argue that a $D$ is certainly low-fidelity if the $R$ alters the meaning of $C(p)$. For example, if an evident dip irregular is mistakenly combined with the morning catchup, it can change the overall daily profile pattern $p$. Consequently, the new pattern may become unreasonable or fall outside the scope of the target building. Since it is difficult to model the relation of various $C(p)$ and $R$, we seek a learning-based method for recognizing such $D'$.

The existing solutions can be broadly categorized under two directions: supervised learning and unsupervised learning. Supervised learning methods, like time-series classification, have the ability to determine whether or not a given $D'$ is low-fidelity. However, these methods require labeled data, which necessitates domain expertise for accurate labeling.

Considering that alterations should be avoided for $C(p)$ after merging $R$. This means that the representation of $C(p) + R$, i.e., $D$, should have a similar representation as that of $C(p)$. In unsupervised learning methods, *contrastive learning (CL)* is a suitable class for learning data representation. CL leverages positive and negative sample pairs to learn the representation of the sample, with the goal of making the representation of similar samples similar. The core of the CL task is: 1) to design the positive and negative data pair, i.e., $(x, x^+)$ and $(x, x^-)$, corresponding to *short distance* and *large distance*, respectively; and 2) to train the discriminator model, i.e., a binary classification model, to judge whether or not the input pair is similar (i.e., positive). Note that CL is widely used for data augmentation scenarios in which data need to be combined [18, 26].

We develop the following contrastive learning scheme.

*1) Positive/negative data pair definition.* We define a real $D$ as having less distance with its decomposed $C(p)$, corresponding to ($x^+$, $x$), respectively. And there is a large distance between the decomposed $C(p)$ and that added with random residuals: $C(p) + R_{Random}$, i.e., $(x, x^-)$. Note that, when a $D$ is decomposed to $C(p)$ and $R$, one positive pair is collected, and we can simulate the multiple negative pairs (In this work, we only simulate one negative pair for a positive pair to avoid sample imbalance).

**Table 2: Target building Dataset specification**

| Building | The ID in Genome | Location | Floor area (sqft) |
|---|---|---|---|
| A | Rat_public_Emilee | Washington | 22500 |
| B | Rat_public_Isabel | Washington | 16576 |
| C | Fox_office_Joy | Tempe | 70837 |
| D | Fox_education_Virginia | Tempe | 12773 |
| E | Bear_education_Iris | Berkeley | 58733 |
| F | Peacock_education_Ophelia | Princeton | 120836 |

*2) Discriminator model.* We collect the pairs from the source buildings to train a binary-classification model (denoted as $f_\theta(\cdot,\cdot)$). The loss function is a triplet loss [37] (a classical loss function in CL), as $\mathcal{L}(x, x^+, x^-) = \sum_{x\in\mathcal{X}} max(0, \left\|f(x) - f(x^+)\right\|_2^2 - \|f(x) - f(x^-)\|_2^2 + \epsilon)$, where function $f(\cdot)$ encodes $x$ into an embedding vector. In the inference phase, the pair of $data_{pair} = (C(p)' + R', C(p)')$ is input into $f_\theta$. Then $f_\theta$ classifies $data_{pair}$ as positive or negative pair, i.e., whether the synthetic data is considered high-quality or not.

Moreover, to enhance the effectiveness of usage, we perform the aforementioned CL tasks separately on weekdays, weekends, and holidays. After this process, we filter out the high-fidelity daily profile set $\{D\}$ for the target building. For applications that require multiple days of load profile, we carefully concatenate $\{D\}$ with the appropriate order, e.g., ensuring that every five-day weekday period was followed by a two-day weekend period.

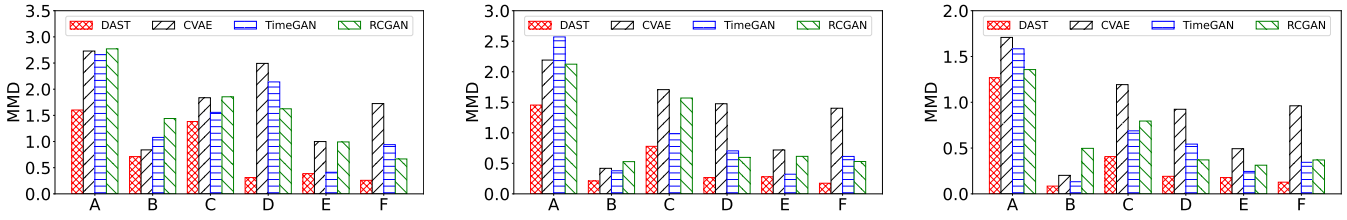
# 6 EVALUATION

In this section, we first present the methodology for conducting the evaluation, and then present the results of the quantitative and qualitative evaluations of the generated synthetic data. We also present an ablation study to show the key components contributing to the performance of DAST.

## 6.1 Methodology

**Datasets:** We evaluate DAST with the Genome dataset. To evaluate the effectiveness of DAST, our rationale was to select the target buildings according to different dimensions: building type, location (i.e., climate condition), and building size. Moreover, the target buildings should have a very low percentage of missing data, which is important for evaluating the quality of the generated data. Finally, in this work, we present the results of DAST on six representative target buildings, with details given in Table 2. The analysis and evaluation of more target buildings will be given in future work.

**Baseline methods:** We compare DAST against three widely used time-series generation methods as baselines. All of these baselines have been applied to augment building or energy data. (1) RCGAN [10], i.e., Recurrent Conditional GAN, a benchmark model for time-series data augmentation. RCGAN adopts RNN in Vanilla GAN. It is widely used to generate synthetic electrical load time-series [45]. (2) cVAE [27], i.e., conditional variational autoencoder. In the building energy field, it is applied to generate data for specific types of months [11]. (3) TimeGAN [46], a state-of-the-art GAN that incorporates temporal dynamics. TimeGAN has been used in the building field, e.g., heating load prediction [48].




(a) Given 15 days data (b) Given 30 days data (c) Given 90 days data

**Figure 9: The MMD measurement under the different sizes of given data for six target buildings (the lower the better).**

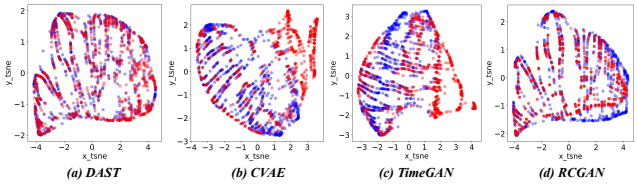



**Figure 10: t-SNE visualization.**

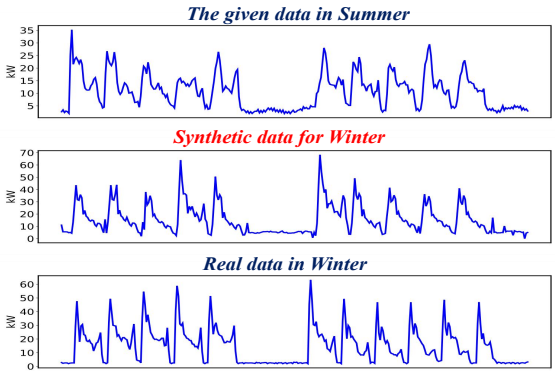



**Figure 11: An example of two weeks of synthetic data.**

**Evaluation Metrics:** To assess the quality of the generated data, we observe two aspects that need to be evaluated in time-series data augmentation tasks: (1) Fidelity: samples should be indistinguishable from the real data. For a quantitative measure of similarity, we apply Maximum Mean Discrepancy (MMD) [38]. (2) Diversity: samples should be distributed to cover the real data. We apply t-SNE [39] analysis, a statistical visualization method, on the original and synthetic data. This visualizes the extent to which the distribution of the generated samples resembles that of the original in two-dimensional space, giving a qualitative assessment of diversity.

**Experiment Setup:** We conducted data augmentation for six target buildings listed in Table 2. Each specified building was considered as the target building, with the remaining 300+ buildings serving as source buildings for supporting the learning-based augmentation sub-tasks . To assess the effectiveness of data augmentation, different sizes of the given data were used from the target building, specifically 15 days, 30 days, and 90 days.

## 6.2 Overall Performance

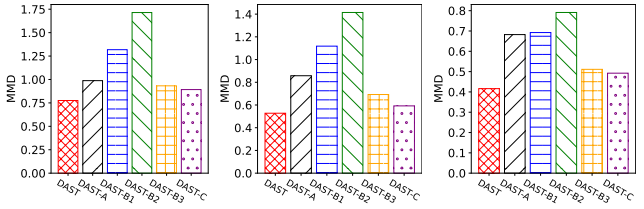
*6.2.1 Improvement in Fidelity.* Figure 9 compares the fidelity of the augmented data of DAST and the baselines based on MMD (the lower the better), under different sizes of given data from the target building. Figure 9 (a) shows the results when the given data size is 15 days. We can see that DAST outperforms the other baselines on all six buildings. On average, the MMD of DAST is 0.77, while for the other three methods, it is 1.77, 1.46, and 1.55, respectively. Thus, DAST achieves an improvement in accuracy (i.e., the distance of the distributions between real and synthetic data) of 56%, 46%, and 50% compared to CVAE, TimeGAN, and RCGAN, respectively. More specifically, in building $D$, the MMD value of DAST is 0.31, while the MMD of CVAE, TimeGAN, and RCGAN is much larger, i.e., 2.49, 2.13, and 1.62, respectively. This is because $D$ has diverse patterns in the given data, and the seasonal variation is also obvious. It is quite difficult for the baselines to generate high-fidelity data if the generative model is only trained on the given 15-day data.

We see that DAST still outperforms the baselines when the size of the given data is 30 and 90 days (as shown in Figure 9 (b) and (c)). We see an improvement in the performance of all the methods when the volume of the given data increases. For a given data size of 30 days, the MMD of DAST is 0.52, with the figures being 1.31, 0.92, and 0.99 for the other three methods. DAST achieves an average improvement of 49% compared to the baselines. For a given data size of 90 days, the MMD of DAST is 0.37, while the MMD of the baselines is 0.91, 0.58, and 0.61, respectively. DAST achieves an average improvement of 44% compared to the baselines.

*6.2.2 Qualitative assessment on diversity.* The t-SNE visualization of the distribution of the generated time-series and the distribution of the real data are shown in Figure 10. Specifically, t-SNE converts the daily profiles into a 2-D scatter plot, where each scatter represents one sample. In our settings, red denotes synthetic data and blue denotes original data. We then check whether the points with two different colors overlap well. This evaluation methodology is widely used in data augmentation research in the AI community. As a result, we can observe that the proposed DAST can achieve better performance (i.e., the red points and blue points can have a better overlap) than other baseline methods. We present a real case of two weeks of synthetic data (shown in Figure 11) for Building D. The results show that DAST can augment high-quality data, even when only a two-week dataset was provided.

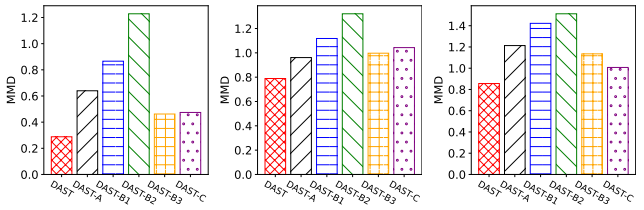
## 6.3 Model ablation study

We now study our designs for each individual component of DAST. We implemented five breakdown versions of DAST to take a closer look at the contribution of each component: **DAST-A** has all the daily pattern $p$ and seasonal context $C$ augmentation schemes, but without decomposition; **DAST-B1** is DAST without daily pattern $p$ augmentation; **DAST-B2** is DAST without seasonal context $C$ augmentation; **DAST-B3** is DAST without irregular $R$ augmentation;




(a) Given 15 days data (b) Given 30 days data (c) Given 90 days data

**Figure 12: The ablation study on different sizes of given data.**




(a) Education (b) Public (c) Office

**Figure 13: The ablation study on the three building types.**

**DAST-C** is DAST without filtering mechanism in augmented component combination. Hence, all of the combinations are acceptable.

Figure 12(a-c) shows the results of the comparison of DAST and the other variants under different sizes of given data. We see that DAST achieves the best performance and that the performance of the other versions of DAST- decreases by 13% to 63%. Moreover, the performance of DAST-B2 is the worst of all of the settings of the given days. Thus, the contribution of seasonal context augmentation is the most important if only limited data from the target building is given. Figure 13(a-c) shows the results of the comparison of DAST and the variants in different building types. DAST outperforms the other DAST- schemes in the three different building types, especially in education buildings.

These results show that our components are necessary and that our designs successfully improve the quality of the synthetic data.

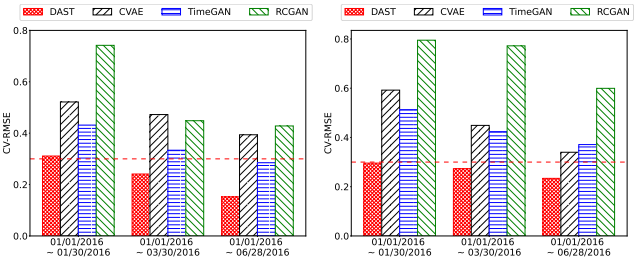
## 7 CASE STUDY: LOAD FORECASTING TASKS

We present a case study where we use our DAST prototype to support building load forecasting (BLF) tasks, and we target two main tasks. Our results demonstrate that our data augmentation approach can benefit both tasks when the length of the given data for a target building is relatively short.

- ***Task 1 - Data-driven BLF model training:*** The synthetic time-series should be as useful as the real data when used for the same predictive purposes (train-on-synthetic and test-on-real, i.e., TSTR [14]).
- ***Task 2 - The differentiation between multiple trained BLF models:*** With the rapid growth of ML, it is easy to collect massive trained ML model files (e.g., .h5 file). This is common in the AI community. Thus, the performance of the trained models[5] on the target building needs to be differentiated (i.e., ranked).

**BLF models and the tested building:** We implement two 24-hour ahead data-driven BLF models, i.e., an RNN-based model and an LSTM-based model, which are widely used for building electrical load forecasting. For task 2, we simulate the trained BLF models by training the two models on 15 different buildings in Genome and then collect $2 \times 15 = 30$ trained models. The features needed for these ML models are regular: time-related (one-hot encoding of the day of the week and hour index) and last 24-hour load history.

[5]In this case study, the parameters of the trained BLF models are fixed, i.e., they will not be continually trained on the synthetic data.




(a) RNN model (b) LSTM model

**Figure 14: The error of the models trained on synthetic data.**

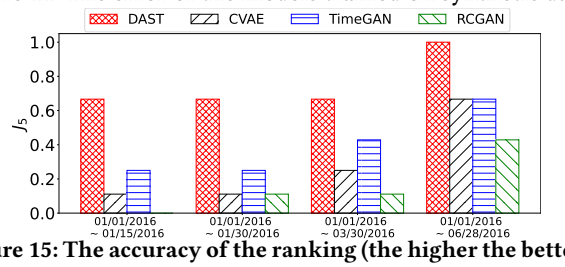



**Figure 15: The accuracy of the ranking (the higher the better).**

We select Building C (in Table 2) to conduct the case study because the quality of the synthetic data for that building is average (the MMD result) among the studied six buildings. We argue that this means that Building C represents a general case of the usefulness of the synthetic data generated by DAST. As always, we set the start of the given data to 01.01.2016 (the start of the holistic data trace) and set the given data from building C as 15 days, 30 days, 90 days, and 180 days, respectively. For example, if the given data is from 01.01.2016 to 02.01.2016, then the remaining data would cover the period from 02.01.2016 to 12.31.2017.

**Metrics:** In task 1, the accuracy of the BLF model is CV-RMSE. In task 2, we introduce the Jaccard similarity coefficient (denoted as $J_k$) to evaluate the similarity between the the estimated rank and actual rank of the top-$k$ model. For example, the estimated rank is $M_1, M_3, M_5, ...$, and the actual rank is $M_1, M_3, M_2, ....$ Thus, the Jaccard similarity coefficient between the two top-3 model sets $\{M_1, M_3, M_5\}$ and $\{M_1, M_3, M_2\}$ is calculated as: $J_3 = \frac{|r_1 \cap r_2|}{|r_1 \cup r_2|} = \frac{2}{4} = 0.5$. In our case study, we use $J_5$.

**The results of the two tasks:** For task 1, Figure 14 shows the accuracy of the two BLF models (i.e., RNN and LSTM), in which the training data is generated by four different data augmentation methods, i.e., DAST, and the other three introduced baselines. We also draw the line of acceptable accuracy, i.e., CV-RMSE $\leq$ 30% for engineering purposes (defined in ASHRAE [8]). We observe that for both BLF models, the accuracy based on DAST is better than the accuracy based on the other three data augmentation baselines, which shows that DAST outperforms other baselines in task 1. In the RNN model, the CV-RMSE of DAST is 31.1%, 24.1%, and 15.1% for the given sizes of 30 days, 90 days, and 180 days, respectively. In general, the CV-RMSE decrease of DAST is 49.2%, 32.9%, and 56.4% compared to that for RCGAN, TimeGAN, and CVAE, respectively.

For task 2, Figure 15 shows the accuracy of the rankings of the trained models (top-5) based on different data augmentation methods. The $J_5$ of the DAST-assisted ranking is 0.67, 0.67, 0.67, and 1.0 for different given data sizes. DAST outperforms baselines. For example, when the given data of the building is from 01.01.2016 to 01.15.2016, the $J_5$ of RCGAN, TimeGAN, and CVAE, are 0.11, 0.25, and 0. This reveals that the capacity of the RCGAN-based and CVAE-based testing is close to random.

## 8 CONCLUSION

In this paper, we presented DAST, a decomposition-based data augmentation scheme to augment time-series building data. As compared to GAN-based methods, DAST fits the scenario with insufficient data distributions, e.g., new buildings with insufficient data collection periods. In practice, decomposition-based methods have been widely applied to various scenarios. Such a method is scenario-dependent and the challenges are to carefully analyze prior knowledge on decomposable components and to develop appropriate decomposition and augmentation schemes. We overcame these challenges by analyzing load data from a large number of buildings and the careful design of DAST. Our evaluation of DAST using real-world data demonstrated that DAST outperforms state-of-the-art baselines and can successfully augment data to support various machine learning tasks in building load forecasting.

## ACKNOWLEDGMENTS

Dan Wang's work is supported by RGC-GRF 15210119, 15209220, 15200321, 15201322, from ITC via project No. K-BBY1, RGC-CRF C5018-20G, ITC ITF-ITS/056/22MX. Fu Xiao's work is supported by RGC (C5018-20GF) and ITF (ITP/002/22LP).

## REFERENCES

[1] A. Adadi. 2021. A survey on data-efficient algorithms in big data era. *Journal of Big Data* (2021).
[2] G. Baasch, G. Rousseau, et al. 2021. A Conditional Generative adversarial Network for energy use in multiple buildings using scarce data. *Energy and AI* (2021).
[3] C. Bergmeir, R. J. Hyndman, and J. Benítez. 2016. Bagging exponential smoothing methods using STL decomposition and Box–Cox transformation. *International journal of forecasting* (2016).
[4] K. Chang, Y. Liao, et al. 2022. Enhancing Recognition Accuracy of Urban Flooding by Generating Synthetic Samples Using CycleGAN. In *IEEE ICCE-TW 2022*.
[5] N. V. Chawla, K. W. Bowyer, et al. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* (2002).
[6] R. B. Cleveland, W. S. Cleveland, et al. 1990. STL: A seasonal-trend decomposition. *J. Off. Stat* (1990).
[7] E. B. Dagum and S. Bianconcini. 2016. *Seasonal adjustment methods and real time trend-cycle estimation*.
[8] Y. Deng, J. Fan, et al. 2022. Behavior testing of load forecasting models using BuildChecks. In *Proc. of ACM e-Energy '22*.
[9] Alexander Dokumentov, Rob J Hyndman, et al. 2015. STR: A seasonal-trend decomposition procedure based on regression. *Monash econometrics and business statistics working papers* 13, 15 (2015), 2015–13.
[10] C. Esteban, S. L. Hyland, and G. Rätsch. 2017. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv:1706.02633* (2017).
[11] C. Fan, M. Chen, R. Tang, and J. Wang. 2022. A novel deep generative modeling-based data augmentation strategy for improving short-term building energy predictions. In *Building Simulation*.
[12] M. Fochesato, Fa. Khayatian, D. F. Lima, and Z. Nagy. 2022. On the use of conditional TimeGAN to enhance the robustness of a reinforcement learning agent in the building domain. In *Proc. of ACM Buildsys '22*.
[13] J. Gao, X. Song, et al. 2020. Robusttad: Robust time series anomaly detection via decomposition and convolutional neural networks. *arXiv:2002.09545* (2020).
[14] N. Gao, H. Xue, et al. 2022. Generative adversarial networks for spatio-temporal data: A survey. *ACM TIST 2022* (2022).
[15] Y. Gao and Y. Ruan. 2021. Interpretable deep learning model for building energy consumption prediction based on attention mechanism. *Energy and Buildings* (2021).
[16] Y. Gao, Y. Ruan, et al. 2020. Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data. *Energy and Buildings* (2020).
[17] A. Geiger, D. Liu, et al. 2020. TadGAN: Time series anomaly detection using generative adversarial networks. In *IEEE Big Data 2020*.
[18] H. Gowda and J. Channegowda. 2022. Contrastive learning for practical battery synthetic data generation using seasonal and trend representations. *International Journal of Energy Research* (2022).
[19] ASHRAE Guideline. 2002. 14: Measurement of energy and demand savings. *American Society of Heating, Refrigerating and Air-Conditioning Engineers* (2002).
[20] R. J. Hyndman and G. Athanasopoulos. 2018. *Forecasting: principles and practice*.
[21] G. Ian, J. Pouget-Abadie, M. Mirza, B. Xu, and D. Warde-Farley. 2014. Generative adversarial nets." In Advances in neural information processing systems. (2014).
[22] B. K. Iwana and S. Uchida. 2021. An empirical survey of data augmentation for time series classification with neural networks. *Plos one* (2021).
[23] B. K. Iwana and S. Uchida. 2021. Time series data augmentation for neural networks by time warping with a discriminative teacher. In *IEEE ICPR 2020*.
[24] S. Joseph and J. Erakkath A. 2018. Real-time retail price determination in smart grid from real-time load profiles. *International Transactions on Electrical Energy Systems* (2018).
[25] L. Kegel, M. Hahmann, and W. Lehner. 2018. Feature-based comparison and generation of time series. In *Proc. of ACM SSDBM '18*.
[26] E. Kharitonov, M. Rivière, et al. 2021. Data augmenting contrastive learning of speech representations in the time domain. In *IEEE SLT 2021*.
[27] D. P. Kingma, M. Welling, et al. 2019. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning* (2019).
[28] K. Kurte, K. Amasyali, J. Munk, and H. Zandi. 2022. Deep reinforcement learning with online data augmentation to improve sample efficiency for intelligent HVAC control. In *Proc. of ACM Buildsys '22*. 479–483.
[29] A. Li, F. Xiao, C. Zhang, and C. Fan. 2021. Attention-based interpretable neural network for building cooling load prediction. *Applied Energy* (2021).
[30] J. Lin, Y. Wang, et al. 2020. Learning to transfer: unsupervised domain translation via meta-learning. In *Proc. of AAAI '20*.
[31] M. Liu, T. Breuel, and J. Kautz. 2017. Unsupervised image-to-image translation networks. *NeurIPS* (2017).
[32] X. Liu, D. Hong, et al. 2021. Modality translation in remote sensing time series. *IEEE Transactions on Geoscience and Remote Sensing* (2021).
[33] X. Luo and T. andothers Hong. 2017. Electric load shape benchmarking for small-and medium-sized commercial buildings. *Applied energy* (2017).
[34] C. Miller, A. Kathirgamanathan, et al. 2020. The building data genome project 2, energy meter data from the ASHRAE great energy predictor III competition. *Scientific data* (2020).
[35] G. Năstăsescu and D. Cercel. 2022. Conditional Wasserstein GAN for Energy Load Forecasting in Large Buildings. In *IJCNN'22*. IEEE.
[36] P. Price. 2010. *Methods for analyzing electric load shape and its variability*. Technical Report. Lawrence Berkeley National Lab.(LBNL), Berkeley, CA.
[37] F. Schroff, D. Kalenichenko, and J. Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proc. of IEEE CVPR '15*.
[38] I. O. Tolstikhin, B. K. Sriperumbudur, and B. Schölkopf. 2016. Minimax estimation of maximum mean discrepancy with radial kernels. *Advances in Neural Information Processing Systems* (2016).
[39] L. Van der M. and G. Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* (2008).
[40] Z. Wang and T. Hong. 2020. Generating realistic building electrical load profiles through the Generative Adversarial Network (GAN). *Energy and Buildings* (2020).
[41] Q. Wen, J. Gao, and OTHERS. 2019. RobustSTL: A robust seasonal-trend decomposition algorithm for long time series. In *Proc. of AAAI '19*.
[42] Q. Wen, L. Sun, et al. 2020. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478* (2020).
[43] M. West. 1997. Time series decomposition. *Biometrika* (1997).
[44] R. Yao and K. Steemers. 2005. A method of formulating energy load profile for domestic buildings in the UK. *Energy and buildings* (2005).
[45] B. Yilmaz and R. Korn. 2022. Synthetic demand data generation for individual electricity consumers: Generative Adversarial Networks (GANs). *Energy and AI* (2022).
[46] J. Yoon, D. Jarrett, et al. 2019. Time-series generative adversarial networks. *Advances in neural information processing systems* (2019).
[47] X. Zhang, R. Chowdhury, et al. 2023. Towards Diverse and Coherent Augmentation for Time-Series Forecasting. In *ICASSP 2023-2023*. IEEE, 1–5.
[48] Y. Zhang, Z. Zhou, et al. 2022. Data augmentation for improving heating load prediction of heating substation based on TimeGAN. *Energy* (2022).
[49] Z. Zheng, Y. Wang, et al. 2019. Metadata-driven Task Relation Discovery for Multi-task Learning.. In *IJCAI*.
[50] J. Zhu, T. Park, et al. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proc. of IEEE ICCV'17*.