Energy & Buildings 224 (2020) 110299

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enb

Generating realistic building electrical load profiles through the Generative Adversarial Network (GAN)

Zhe Wang, Tianzhen Hong *

Building Technology and Urban Systems Division, Lawrence Berkeley National Laboratory, One Cyclotron Road, Berkeley, CA 94720, USA

ARTICLE INFO

Article history:
Received 5 March 2020
Revised 27 May 2020
Accepted 7 July 2020
Available online 11 July 2020

Keywords:
Electrical load profile
Generative Adversarial Network (GAN)
Machine learning
Smart meter data
Load shape

ABSTRACT

Building electrical load profiles can improve understanding of building energy efficiency, demand flexibility, and building-grid interactions. Current approaches to generating load profiles are time-consuming and not capable of reflecting the dynamic and stochastic behaviors of real buildings; some approaches also trigger data privacy concerns. In this study, we proposed a novel approach for generating realistic electrical load profiles of buildings through the Generative Adversarial Network (GAN), a machine learning technique that is capable of revealing an unknown probability distribution purely from data. The proposed approach has three main steps: (1) normalizing the daily 24-hour load profiles, (2) clustering the daily load profiles with the k-means algorithm, and (3) using GAN to generate daily load profiles for each cluster. The approach was tested with an open-source database – the Building Data Genome Project. We validated the proposed method by comparing the mean, standard deviation, and distribution of key parameters of the generated load profiles with those of the real ones. The KL divergence of the generated and real load profiles are within 0.3 for majority of parameters and clusters. Additionally, results showed the load profiles generated by GAN can capture not only the general trend but also the random variations of the actual electrical loads in buildings. The proposed GAN approach can be used to generate building electrical load profiles, verify other load profile generation models, detect changes to load profiles, and more importantly, anonymize smart meter data for sharing, to support research and applications of grid-interactive efficient buildings.

1. Introduction

1.1. Background

The term "building electrical load profile" refers to the variation of the building's electrical load versus time.[1] Building electrical load profile is influenced by both physical and behavioral factors [1]. Building load profile analytics have wide applications, such as identification of unnecessary waste [2], load forecasting [3], customer segmentation and demand-side management [4], demand response planning and pricing [5], long-term resource planning, and renewable energy integration [6]. Therefore, collecting and analyzing load profiles plays a significant role in enhancing our understanding of how best to integrate buildings with a changing electric grid and to finally reduce building carbon footprints.

Building load profiles can be collected by smart meter data, which are the data (mostly electrical power consumption) collected by the smart meter at fine temporal granularities [7]. A smart meter is an electronic device that records consumption of electric energy and communicates the information to a remote server. Compared with conventional electricity meters, the smart meter has two key characteristics: fine temporal granularity and real-time communication. The temporal granularity of smart meter data is usually 5 to 15 mins.

With the pervasive deployment of smart meters, fine-grained electricity consumption data are becoming increasingly available [8]. While the use of smart meter data is unlocking large benefits, it also raises data privacy and security concerns [9]. Partly due to those privacy concerns, utility companies are unwilling or unable to share customer meter data for general research and analysis, which is a major barrier to advancing smart meter data analytics. To address the privacy concerns, existing studies have proposed privacy-preserving approaches, such as data aggregation [10], clustering [11], principal component analysis [12], and wavelet-

* Corresponding author.
E-mail address: thong@lbl.gov (T. Hong).

[1] A similar term to load profile is "load shape," which is defined as the curve that represents load as a function of time [2]. It can be observed from the definitions that the terms of load profile and load shape are interchangeable. To avoid confusion, we will only use the term "load profile" in this study.

https://doi.org/10.1016/j.enbuild.2020.110299
0378-7788/

Nomenclature

Abbreviations

Adam	Adaptive moment estimation
CBECS	Commercial Building Energy Consumption Survey
D	Discriminator
DOE	Department of Energy
DBI	Davies-Bouldin Index
GAN	Generative Adversarial Network
G	Generator
KL Divergence	Kullback–Leibler divergence
LeakyReLU	Leaky version of a Rectified Linear Unit
NREL	National Renewable Energy Laboratory
SC	Silhouette Coefficient

Subscripts/superscripts

d_{loss}	Loss function of discriminator
$D(x)$	Discriminator output given input x, usually binary (0 or 1)
g_{loss}	Loss function of generator
$G(x)$	Generator output given input x, usually 24-dimension vector

based representation [13]. However, those privacy-preserving methods suffer from reduced data resolution and unavoidable information loss.

1.2. Building electrical load profile analysis

As building electrical load profiles can yield useful information and have wide applications, many studies have been done to analyze building load profiles, and these dated back to as early as the 1980s [14]. Studies on building load profiles can be classified into descriptive and generative studies. Descriptive studies aim to describe the characteristics of building load profiles. Price [2] defined five statistics to characterize daily load profile: (1) near-peak load, (2) near-base load, (3) high-load duration, (4) rise time, and (5) fall time, and demonstrated how load profile analysis can be used to estimate demand response effectiveness.

Generative studies aim to generate load profiles that are as realistic as possible, and they can avoid the privacy concerns of directly using smart meter data. Additionally, the relatively high cost of smart meters prevents them from being deployed in some regions; this problem can be solved by using generative studies instead. The generated load profiles can be used to test or verify demand control techniques [15] and to evaluate demand response policies.

Two common approaches are used to generate realistic building load profiles. The first is through energy simulation using white-box models. The white-box approach is physics based, which simulates the energy consumption with detailed building energy models. To facilitate physics-based building energy simulation, detailed assumptions about the building's physics (envelope, system efficiency), predicted occupant behavior [16], and electrical appliance schedules [17] need to be provided, which demands expertise and is labor intensive. To avoid the tedious efforts needed to collect data on building physical and behavioral factors for white-box simulation, some studies use schedules and assumptions from reference building models to generate load profiles [18]. For instance, using U.S. Department of Energy (DOE) reference models, the National Renewable Energy Laboratory (NREL) generated and open sourced hourly load profile data for 16 commercial building types on OpenEI.[2] However, the assumptions proposed by the modelers or in the reference building model may not necessarily reflect the reality of actual buildings [19], resulting in a gap in the load profiles between the white-box simulation and the real building [20].

The second approach is to use black-box models to regress building energy consumption, with features such as demographics [21], energy price [22], local climate [23], and presence of end-use appliances (aka "conditional demand analysis") [24]. The black-box model can be developed through a probabilistic method [15], regression [25], or a neural network [26]. However, the black-box approach usually does not address individual end uses [27], and more importantly, it is unable to provide high temporal-granularity load variations. Fig. 1 summarizes major approaches taken to collect or generate building load profiles and their constraints.

1.3. Objectives

The major contribution of this study is to propose a data-driven approach to generate realistic building electrical load profiles using the Generative Adversarial Network (GAN). This new approach can generate load profiles at the individual building or household level.

As a data-driven black-box approach, this approach can save tedious work of making assumptions on building physics and occupant schedules, which are required by physics-based models. In terms of privacy concern, GAN retains important statistical information such as the dynamic and stochastic behaviors of building loads, while anonymizing user-sensitive information to protect users' privacy.

The proposed approach can be used to generate or forecast building load profiles, as well as to anonymize collected smart meter data. By protecting customer privacy, it is expected that more building owners will be encouraged to share their data for research and analysis.

The remaining of this paper is organized as follows. Section 2 proposes a two-step approach to applying GAN to generate realistic building load profiles. Section 3 presents the result of testing our proposed approach on smart meter data from real buildings. Hyper-parameter tuning, potential applications, and limitations of this approach are discussed in Section 4. Section 5 offers conclusions.

2. Method

Fig. 2 presents the technical roadmap to generate building load profiles through GAN, which has three major steps: data preprocessing, load profile clustering, and GAN, which is discussed in detail in this section.

2.1. Data source

In this study, we selected the open source Building Data Genome Project database [28] as our testbed. Testing our approach on an open source database can demonstrate the reproducibility of this new method. The Building Data Genome Project has collected and open sourced one-year hourly whole building electrical meter data from 507 non-residential buildings. Those buildings are mostly from the education industry in the United States and

[2] https://openei.org/doe-opendata/dataset

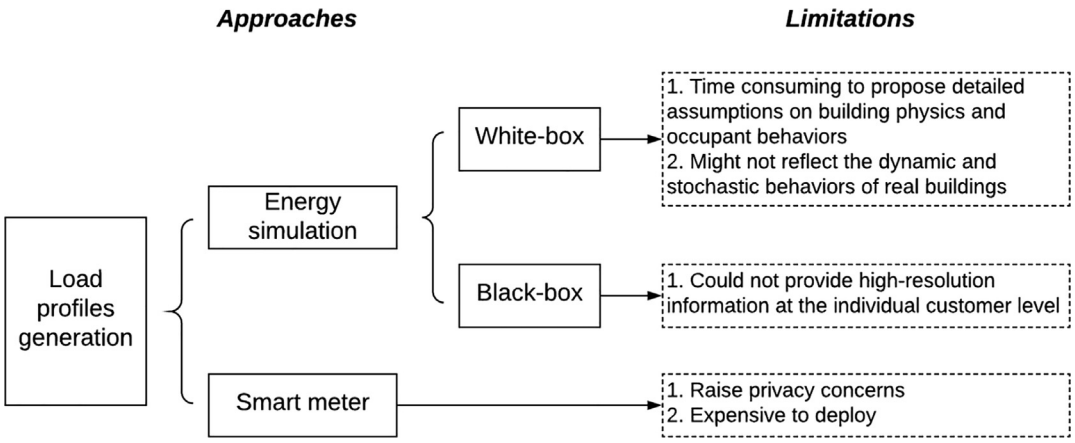


Fig. 1. Approaches to generating building load profiles and their constraints.

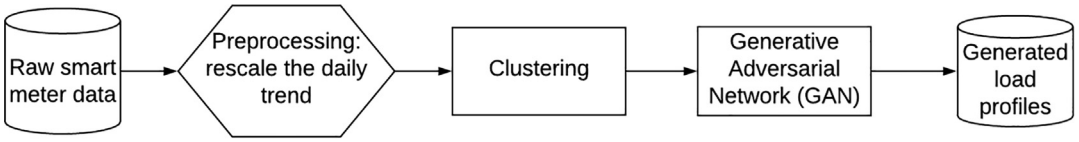


Fig. 2. Technical roadmap to generate building electrical load profiles using GAN.

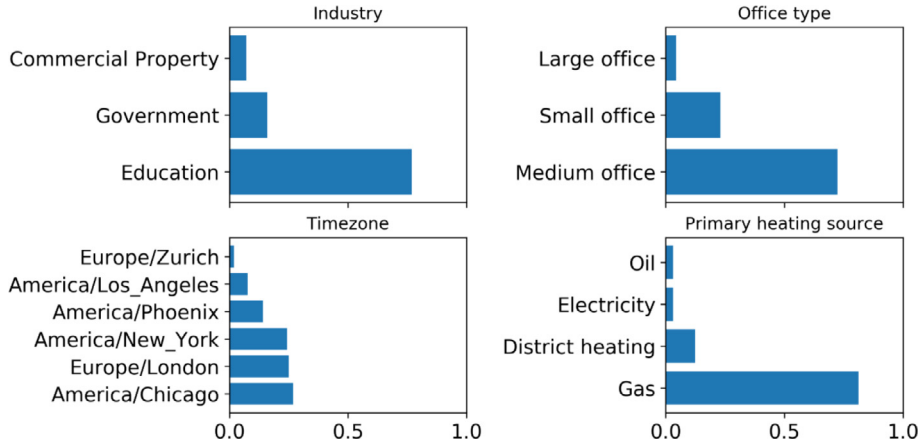


Fig. 3. Overview of the buildings used in this study. ("Small office" is defined as floor area less than 30,000 square feet, "medium office" is defined as floor area between 30,000 and 300,000 square feet, and "large office" is defined as floor area above 300,000 square feet.) The majority of building samples are medium office buildings in educational institutions, such as researchers' offices in universities.

Europe. Metadata of floor area, occupant counts, and primary heating source are released for some but not all buildings.

In this study, we used the smart meter data of 156 office buildings from the Building Data Genome Project database. Office buildings represent the largest sample in the Genome database and represent the largest floor area and highest total energy use in the U.S., according to the Commercial Building Energy Consumption Survey (CBECS 2012) [29]. From the 156 office buildings, we extracted 56,957 daily electrical load profiles.[3] The temporal granularity of this dataset is one hour.

2.2. Data preprocessing

Fig. 3 presents key building information for the office buildings investigated in this study. Building electrical load profiles vary markedly among different office buildings due to different floor area, climate conditions, occupancy schedules, primary heating/-cooling source and other factors. As we focused on time-varying behavior, rather than the overall energy usage, we normalized each building's daily load profiles by their annual peak load, to make sure the load profiles of different buildings were similar in magnitude, in the range of 0 to 1.

Due to the existence of extreme weather conditions and uncommon events, the definition of "peak load" varies among the different literature. We prefer the peak load to be more stable and free from the influence of extreme events. Luo et al. proposed the 95 percentile as the peak load [30], while Price defined the near-peak load as the 97.5 percentile [2]. The choice of the peak load definition is beyond the discussion of this study, and we used the 95 percentile as the peak load for normalization. As we did not use the absolute peak as the denominator, the normalized load can be above 1, which has not been clipped in this study.

[3] The electrical meter recordings of 17 buildings are from the leap year.

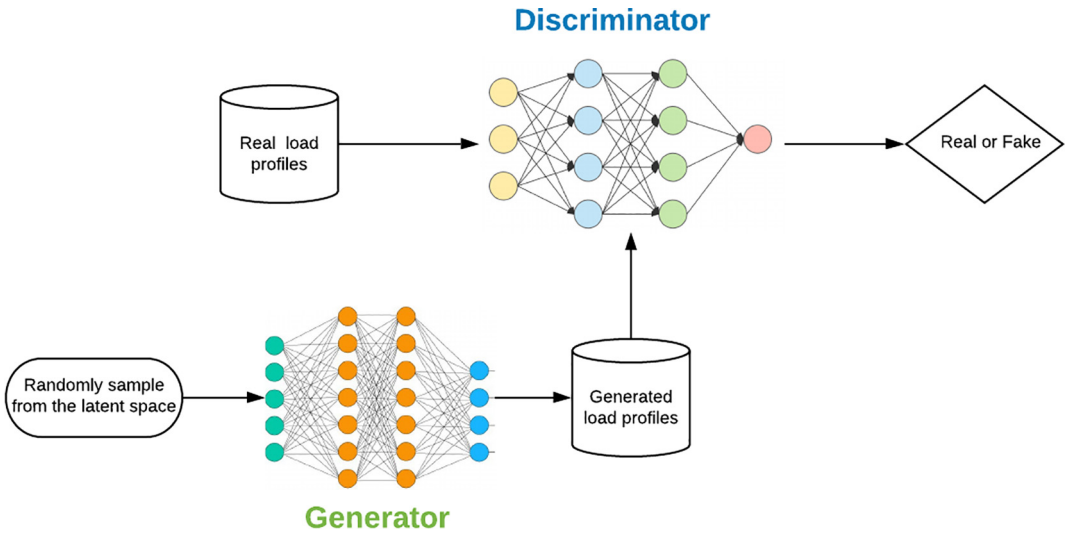


Fig. 4. Generative Adversarial Network (GAN).

2.3. Load profile clustering

Clustering is a necessary step of generating realistic load profiles. Daily load profiles vary a lot, even for the same building, due to different working schedules and weather conditions. It makes sense to generate the building load profiles for each cluster, i.e., capturing the key explanatory variables and characteristics of each cluster, rather than for the whole dataset.

As a non-supervised machine learning technique, clustering is widely utilized in many fields to discover meaningful patterns and structures in data. In the field of electrical usage, clustering, sometimes also referred to as "load profiling," had been widely used in customer segmentation and grouping. Accordingly, several previous studies developed different clustering methods. In this study, we selected k-means as the clustering algorithm, as k-means [31–33] and its variants [34,35] are frequently used for load profile clustering.

To evaluate the performance of clustering, we selected the Davies-Bouldin Index (DBI), defined in equations 1 and 2 [36].

Where s_i and s_j are the average distance between each point of cluster i, cluster j and the centroid of that cluster; and d_{ij} is the distance between cluster centroids i and j. The clustering configuration with a lower DBI is preferred, as a low DBI indicates that the distance of points within each cluster is small compared with the distance between different clusters.

In this study, we selected DBI rather than Silhouette Coefficient (SC) [37] because SC needs to calculate the distance between any two points of the same cluster, which has the computational complexity of $o(N^2)$[4], while DBI only needs to calculate the distance between each point and the cluster centroid, with the computational complexity of $o(N)$. With the increase of sample points, DBI is much more computationally efficient than SC.

2.4. Generative Adversarial Network (GAN)

GAN was first proposed by Goodfellow et al. in 2014 [38], and has since been applied widely in creative activities such as generating paintings or music. As shown in Fig. 4, GAN consists of two neural networks: the generator and the discriminator. The generator generates load profiles with the seeds randomly sampled from the latent space. The discriminator judges whether an input load profile is real or fake (generated). The latent space may have a structure that can be explored. For instance, in the image generation case, some dimensions in the latent space may correspond to 'happy vs. unhappy' or 'male vs. female'. In our building load generation case, some dimensions in the latent space may correspond to 'large variation vs. small variation'. Therefore, performing vector arithmetic between points in latent space can have meaningful and targeted effects on the generated profiles.

The objective of the generator is to generate load profiles which can resemble the real load profiles as much as possible, while the goal of the discriminator is to detect the real load profiles from the generated ones as accurately as possible. Given a properly defined loss function, the generator becomes more capable of generating realistic load profiles, while the discriminator becomes more sensitive to detecting the subtle differences between the real and the generated load profiles after epochs of training. The competition between the generator and the discriminator helps them both to evolve and improve.

The loss function of the discriminator and the generator are defined in equations 3 and 4, respectively, where G stands for *generator*, and D stands for *discriminator*. In Equation 3 and 4, $D(x)$ and $G(x)$ denotes the output of the discriminator and generator, respectively. $G(x)$ is the generated load profile, which has the same dimension as the real load. In this study, $G(x)$ is a 24-dimension vector. $D(x)$ is a binary variable, indicating whether the input load profile is real or fake, 1 for real and 0 for fake.

To train the discriminator, we usually sampled half the data points from the real load profile database, and another half from the fake load profiles generated by the generator (equal weight for identifying the real load profiles and detecting the fake ones, as in Equation 3). The goal of training the discriminator was to minimize d_{loss}, defined in Equation 3, which enhances the capability of the discriminator to distinguish the real load profiles from the fake ones. Ideally, we hope the generator is accurate enough. When the load profile is from the real database ($x\ X_{true}$), the discriminator outputs $D(x) = 1$; and when the load profile is generated, the discriminator outputs $D(G(seed)) = 0$.[5] In this case, d_{loss} is minimized to 0.

[4] N is the number of points in each cluster, which is on the same scale of the number of total samples.

[5] seed is a random sampling from the latent space. Seed is the input to the generator to generate load profiles

Pseudocode

---

define the *discriminator neural network* and compile the *discriminator model*

define the *generator neural network*

define and compile *GAN model* by integrating the *generator* and *discriminator neural network*, and setting the parameter of *discriminator neural network* untrainable

for *epoch* in range(*epochs*):

 # train the discriminator

sample points randomly from the real load profile dataset

generate fake load profiles from randomly sample seeds with the *generator neural network*

combine and shuffle the real and fake load profiles together

train the *discriminator model* with the combined data points to minimize d_loss defined in *Equation 3* (training the parameters in the *discriminator neural network*)

 # train the generator

sample seeds randomly from pre-defined normal distribution

train the *GAN model* with the sampled seeds to minimize g_loss defined in *Equation 4* (as the parameter of *discriminator* was set untrainable in the *GAN model*, we are essentially training the parameters in *generator neural network* only in this phase)

Fig. 5. Pseudocode for training the GAN model..

To train the generator, we randomly sampled seeds from the pre-defined latent space[6]; then generated load profiles from the seeds with the generator. The dimension of the generated load profiles is the same as the real load profiles. In this case, we used hourly smart meter data. Therefore, the dimension of the generated and real load profiles are 24. As the input of the generator is randomly sampled, only the inputs of the discrimator, i.e., the real load profiles, are the information fed into the GAN to train both the generator and discrimitor. The goal of training the generator is to minimize $g_{loss,}$ defined in Equation 4. In the ideal scenario, the discriminator believes the generated load profiles ($G(seed)$) are real and accordingly outputs $D(G(seed)) = 1$. In this case, g_{loss} is minimized to -1.

The training process is summarized in the pseudocode (Fig. 5), which has three main steps: (1) define the discriminator and the generator, (2) train the discriminator, and (3) train the generator. In this study, we used Keras [39] backend by Tensorflow [40] to implement and train the GAN model. Keras is a model-level library, providing high-level building blocks for developing deep learning models. TensorFlow is an open-source symbolic tensor manipulation framework developed by Google, Inc.

In Keras or Tensorflow, the model needs to be compiled before training. To train the discriminator network, we need to compile the discriminator model first, then use the loss function defined in Eq (3) to train it. However, to train the generator network, we need to compile the generator and discriminator together and set the parameters of the discriminator untrainable. Because we need to use *D(G(seed))*, as shown in Eq (4), to train the generator. To get *D(G(seed))*, we need to compile the generator and discriminator together. Meanwhile, the parameters of the discriminator need to be set as untrainable when we train the generator, because, we want to train one network at a time. To reduce the g_{loss} defined in Eq (4), there are two ways: a better generator that is able to generate more realistic load profile, or a worse discriminator, which is less capable to distinguish the fake loads from the real ones. We do not want the second to occur, therefore, we need to freeze the parameters of the discriminator when we train the generator.

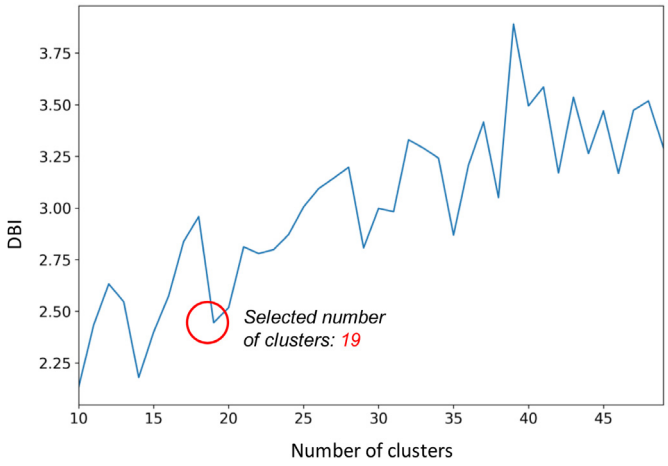
3. Results

3.1. Load profile clusters

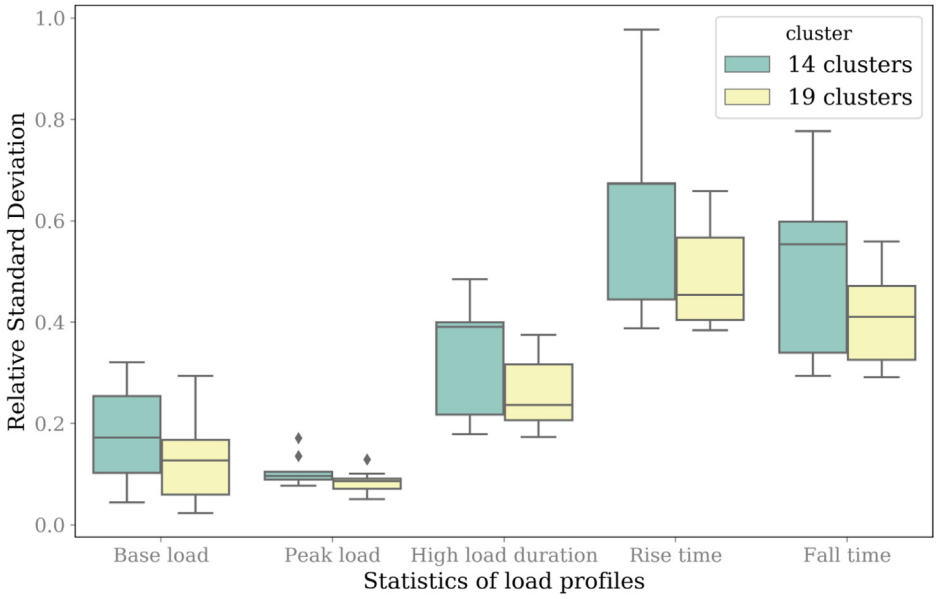
The first step of clustering is to select the optimal number of clusters. We used DBI to select the optimal number of clusters. Clustering configuration with a lower DBI is always preferred, as a low DBI indicates the distance between data points within the same cluster is small compared with the distance between different clusters. We ran the clustering algorithm with different cluster numbers and selected the optimal number of clusters that had a low DBI value. As the *k-means* algorithm is sensitive to the initial values of the cluster centers, we ran *k-means* 50 times with a different initial value and recorded only the best result of each cluster number. As shown in Fig. 6 (a), local minimum DBI values can be achieved when the optimal number is 14 or 19. The clustering analysis have been conducted for 14 and 19 number of clusters.

We used five key parameters to quantify the load shape of building daily electricity consumption: base load, peak load, peak load duration, rise time and fall time. The above five parameters

[6] The dimension of the latent space is an important hyper-parameter. In this study, we selected the dimension as 20, as shown in Table 3



(a) Davies–Bouldin index



(b) In-cluster Standard Deviation of key statistics

Fig. 6. Selection of optimal number of clusters.

are proposed by Price, P. to analyze electric load shape and its variability [2], with the following definition:

Near-Peak load: 97.5 percentile of daily load, denoted as P_{peak}

Near-Base load: 2.5 percentile of daily load, denoted as P_{base}

High-load duration: Duration for which load is closer to near-peak load (P_{peak}) than to the near-base (P_{base}), i.e., the duration of $P > \frac{1}{2}(P_{peak} + P_{base})$

Rise time: Duration for load to go from base load to the start of the high-load period

Fall time: Duration for load to go from the end of the high-load period to the base load

In this study, we made two slight changes to Price, P.'s definition [2]. First, we distinguish working days from non-working days, as they demonstrate different load shapes. We only calculated the above five parameters for the working day, as in non-working days, the daily load is less variant and stable on the base load. Therefore, the concepts of peak load, peak load duration, rise time, and fall time are not applicable to non-working days. In this study, the working day is defined as if the near-peak load is more than 1.2 times of the near-base demand. Contrarily, if the near-peak load is less than 1.2 times of the near-base load, it is a non-working day. The peak load, peak load duration, rise time, and fall time are only defined for working days and shown in Tables 1 and 2. The second revision is we used 95% and 5%, rather than 97.5% and 2.5%, as the threshold of near-peak and near-base load. The key statistics of each cluster are presented in Tables 1 and 2 respectively.

As observed in Tables 1 and 2, the standard deviation within each cluster is smaller if the building load is clustered into 19 clusters rather than 14. For instance, the standard deviation of base load is between 0.03 and 0.09 for 14 clusters, compared with 0.02 and 0.07 for 19 clusters. Similar results can be found in all the other four parameters. Since the standard deviation within each cluster quantifies the dissimilarities of each instances within the same cluster, a lower standard deviation means the instances of the cluster are similar to its peers in the same cluster. As shown in Fig. 6(b), if we select 19 as the optimal number of clusters, the relative in-cluster standard deviation of each cluster will be lower for all of the five key statistics of the load profile. Therefore, we finally selected the number of cluster as 19 because 19 clusters can deliver a cluster result with lower in-cluster standard deviation.

Table 1
Clustering result if the number of cluster is 14.

Cluster	Day Type	Percentage	Base Load		Peak Load		High Load Duration		Rise Time		Fall Time	
			mean	std	mean	std	mean	std	mean	std	mean	std
0	Non-working	0.04	0.17	0.05								
1	Working	0.04	0.23	0.05	0.35	0.06	13.38	6.49	2.67	2.61	3.32	2.58
2	Non-working	0.09	0.37	0.03								
3	Working	0.08	0.30	0.08	0.51	0.05	11.48	4.49	3.49	2.35	4.21	2.52
4	Working	0.04	0.32	0.09	0.59	0.08	11.11	4.42	3.59	2.42	4.80	2.66
5	Working	0.05	0.42	0.09	0.62	0.06	13.35	5.58	2.93	2.42	3.53	2.46
6	Working	0.06	0.28	0.09	0.78	0.07	9.65	1.73	4.02	1.79	5.92	1.79
7	Working	0.04	0.35	0.06	0.84	0.08	10.14	1.92	4.10	1.59	5.88	1.73
8	Working	0.06	0.50	0.08	0.76	0.08	11.84	4.21	3.48	2.01	4.59	2.46
9	Working	0.10	0.46	0.08	0.90	0.07	10.76	2.34	3.97	1.62	5.27	1.79
10	Working	0.05	0.64	0.06	0.85	0.07	12.48	4.99	3.34	2.25	4.07	2.38
11	Non-working	0.10	0.62	0.08								
12	Non-working	0.09	0.70	0.06								
13	Non-working	0.14	0.90	0.04								

Table 2
Clustering result if the number of cluster is 19.

Cluster	Day Type	Percentage	Base Load		High Load		High Load Duration		Rise Time		Fall Time	
			mean	std	mean	std	mean	std	mean	std	mean	std
0	Non-working	0.03	0.17	0.05								
1	Non-working	0.03	0.28	0.02								
2	Non-working	0.07	0.34	0.02								
3	Non-working	0.07	0.40	0.03								
4	Working	0.08	0.24	0.07	0.62	0.08	9.38	1.96	3.37	2.19	5.25	2.27
5	Non-working	0.05	0.49	0.03								
6	Working	0.09	0.34	0.06	0.67	0.06	10.58	3.49	3.64	1.90	5.24	2.54
7	Working	0.06	0.28	0.08	0.79	0.07	9.44	1.68	4.09	1.80	5.87	1.73
8	Working	0.04	0.44	0.06	0.69	0.07	12.21	4.58	3.34	2.20	4.52	2.53
9	Working	0.03	0.37	0.07	0.83	0.07	10.12	2.09	4.07	1.58	5.10	1.92
10	Working	0.02	0.48	0.07	0.76	0.07	11.94	4.13	3.47	2.02	4.32	2.21
11	Working	0.05	0.44	0.07	0.85	0.07	11.12	3.10	3.72	1.66	5.02	2.09
12	Working	0.06	0.45	0.07	0.93	0.06	10.58	1.84	4.06	1.56	5.59	1.63
13	Working	0.03	0.55	0.07	0.88	0.06	11.34	2.99	3.92	1.81	4.64	1.88
14	Non-working	0.09	0.77	0.03								
15	Working	0.03	0.62	0.07	0.98	0.05	10.35	2.18	4.07	1.60	5.33	1.65
16	Non-working	0.06	0.85	0.02								
17	Non-working	0.06	0.89	0.03								
18	Non-working	0.05	0.93	0.04								

Fig. 7 presents the clustering results. Each cluster accounts for 2%-9% of the total 56,957 daily load profiles. Some clusters, such as clusters 4, 7, 9, 11 and 12, demonstrated a clear working day pattern, i.e., electricity usage increased in the morning and decreased in the late afternoon. Some clusters, such as cluster 1 and 2, demonstrated a non-working day pattern, i.e., the electricity usage was low throughout the whole day. Though some clusters were similar in general shape, some subtle but important differences did exist, such as the base load value, the ramp height, the time of morning warm up, and the duration of the peak load.

3.2. Generating load profiles with GAN

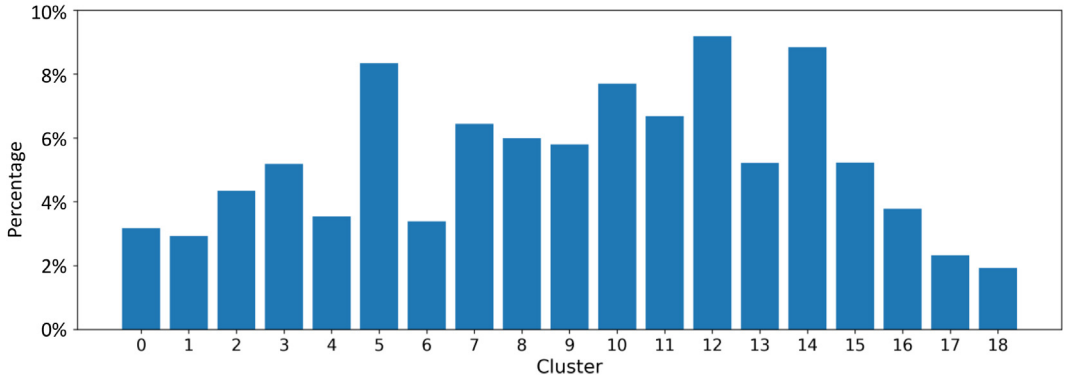
As observed in Fig. 7, each cluster load profile demonstrates similar patterns. The goal of GAN is to learn and reproduce those patterns to generate artificial load profiles which can maintain the key statistical information of real load profiles.

Because there are two neural networks (the generator and the discriminator) to be trained at the same time, GAN is a dynamic system, where the optimization minimum is not fixed. Therefore, it is difficult to train GAN [41]. This section focuses on the results; however, we present a discussion of hyper-parameter tuning in Section 4.1. Fig. 8 uses cluster 12 as an example to illustrate how the generated load profiles evolved with the training process. Fig. 8a shows 100 randomly selected load profiles from the real dataset. Fig. 8b1 through 8b6 show the load profiles generated from the nine randomly selected seeds during the training process.

Because the weights of the generator were totally randomized at the beginning of the process, the generated load profiles at the early epochs were dissimilar to the actual load profiles. As the training continued, the overall shape (load increasing at around 05:00, starting to peak at about 09:00, and starting to decrease at about 15:00) was gradually learned by the generator. Again, as the training continued, some details and random variations, such as the possibility of load decreasing during the lunch break (as shown in Fig. 8a) had been learned as well (Fig. 8b6). This process enables the generated load profiles to capture not only the general trend but also the random variation of the actual ones.

3.3. Accuracy of the generator and discriminator

To present a clear picture of the evolution of the generator and discriminator, we plotted how the accuracy of the discriminator and generator change as the training epochs increase, as shown



(a) Distribution of each cluster

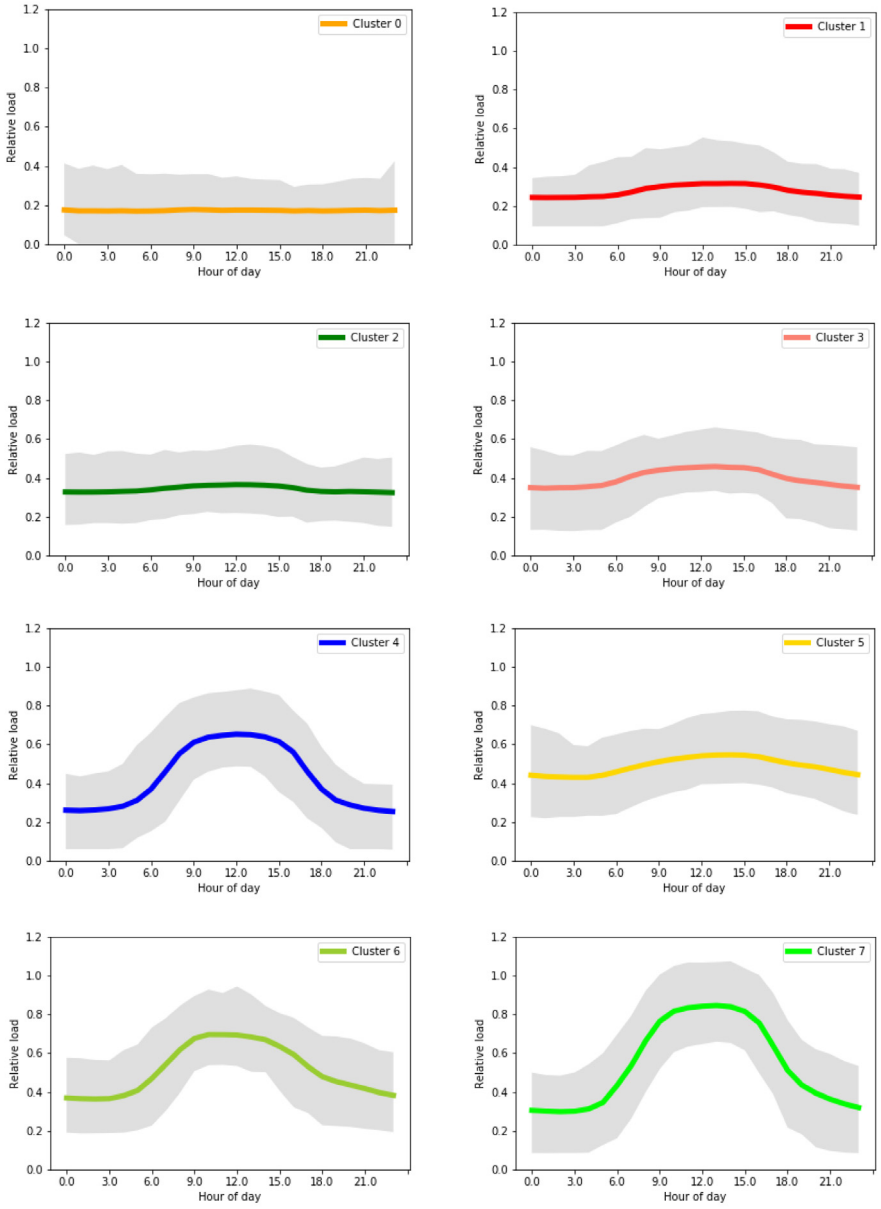
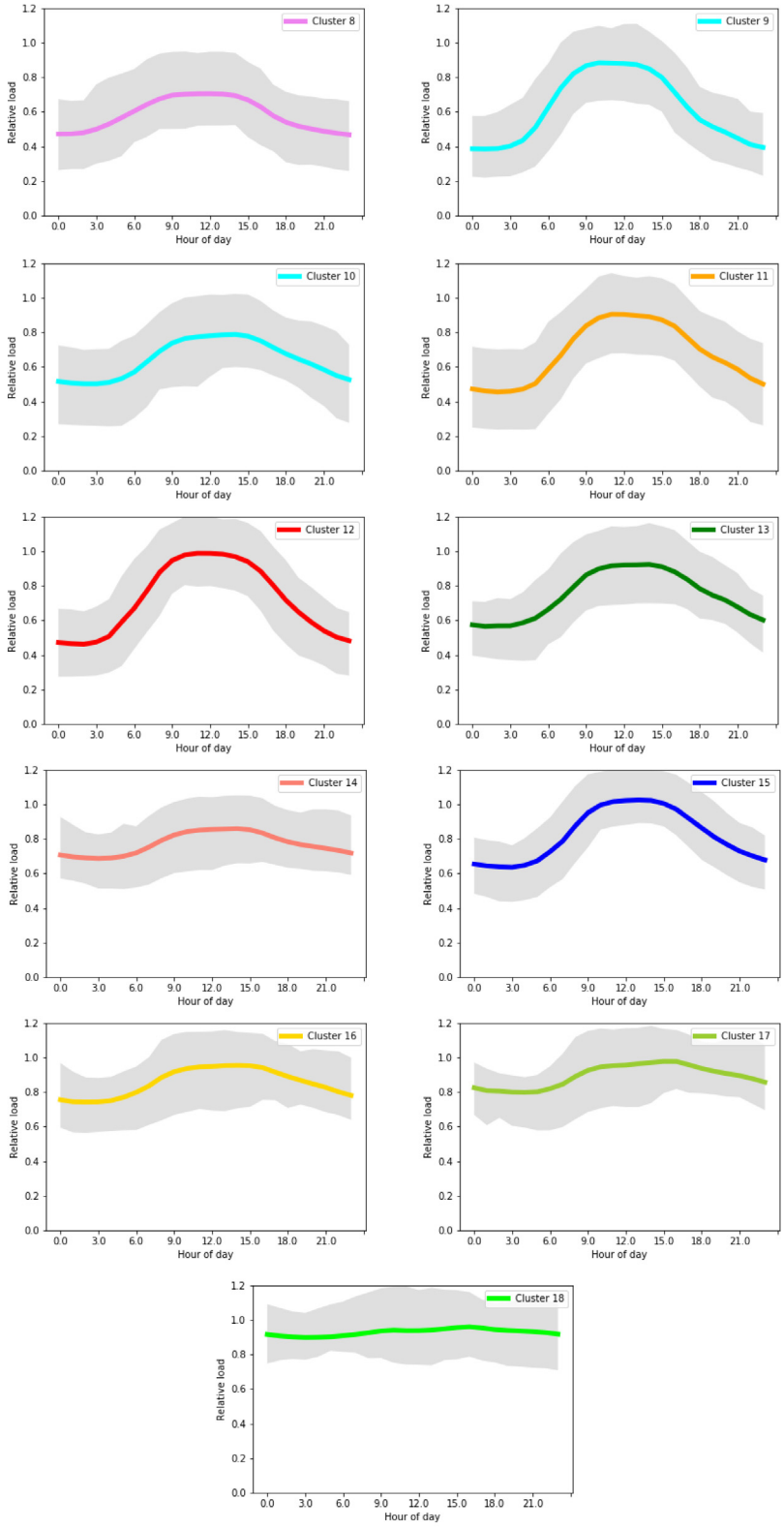


Fig. 7. Load profile clustering results.

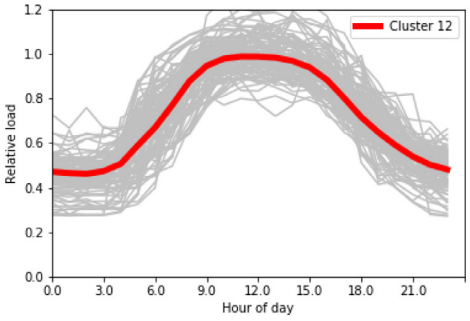
in Fig. 9. The accuracy of the discriminator is defined as the proportion of load profiles that can be detected correctly. The accuracy of the generator is defined as the percentage of generated load profiles that are detected as "real" by the discriminator. The accuracy plotted in Fig. 9 is the average accuracy of the 19 clusters on the test dataset, which accounts for 20% of the total data points.



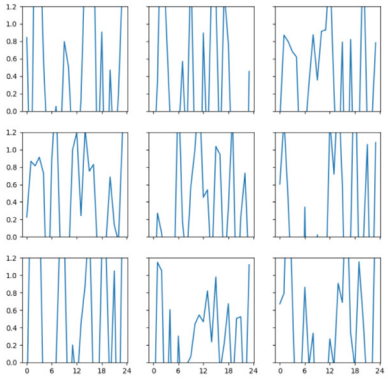
(b) The 19 clusters identified. The silver band denotes the load ranges, the bold line denotes the cluster centroid

Fig. 7 (*continued*)

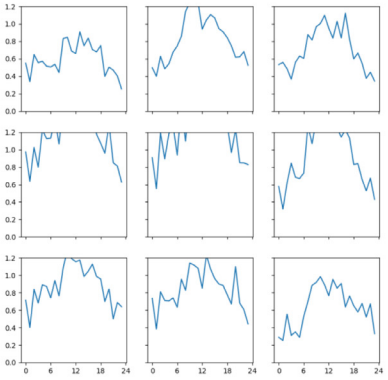
The oscillating saw shape of the discriminator's and generator's accuracy evolving with epochs (Fig. 9) is due to GAN's competing nature, i.e., the enhancing of the discriminator or generator imposes additional challenges to the other. There are two phases in each epoch: the discriminator training phase and the generator training phase. In typical GAN settings, the discriminator will be



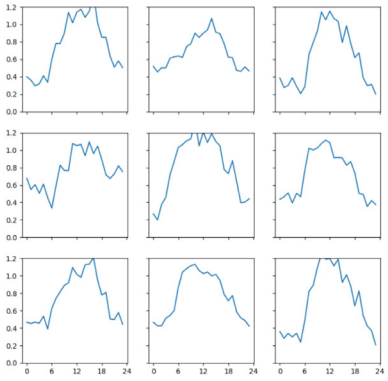
(a) Raw data



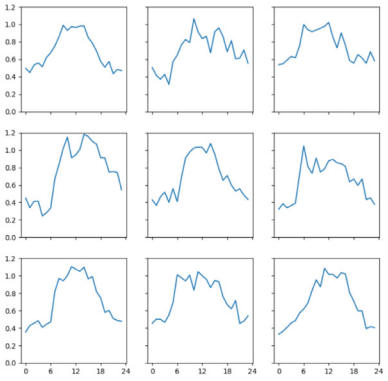
(b1) Epoch 1



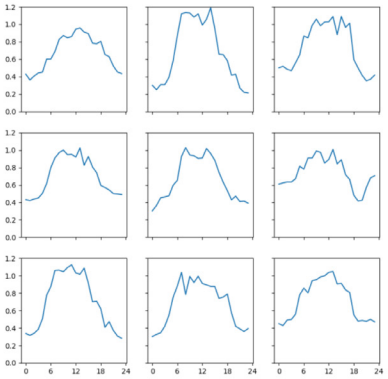
(b2) Epoch 30



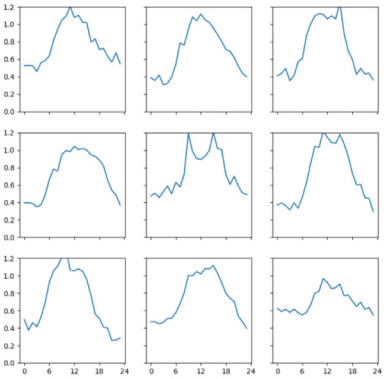
(b3) Epoch 60



(b4) Epoch 90



(b5) Epoch 120



(b6) Epoch 150

Fig. 8. Evolution of the generated load profiles (using cluster 12 as an example).

trained first, and then the generator. As shown in Fig. 9, after the first half of each epoch (the discriminator training phase), the accuracy of the discriminator improves; meanwhile, the generator's accuracy decreases, as the enhanced discriminator becomes more capable and sensitive in detecting the generated fake load profiles. In the second half of each epoch (the generator training phase), the

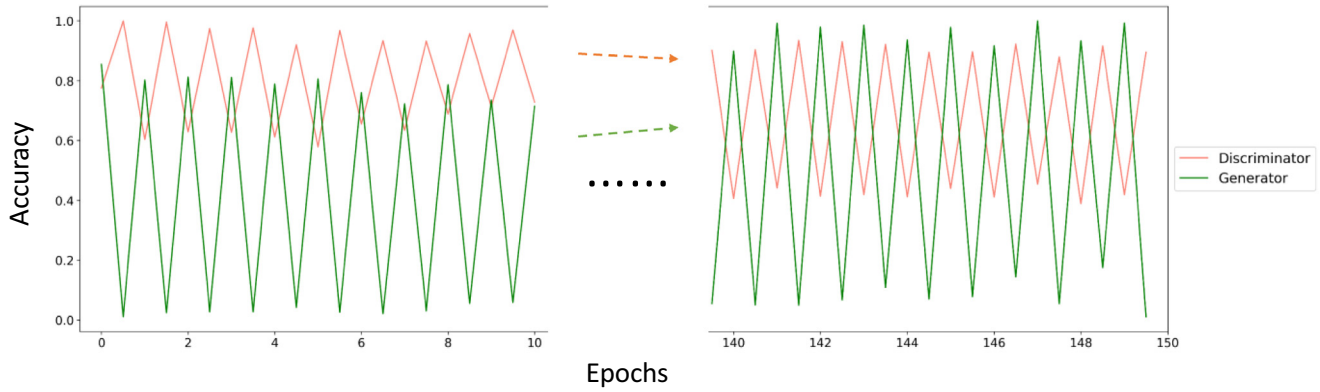


Fig. 9. Training accuracy of GAN.

generated load profiles become more similar to the real ones, with the generator weights being optimized to reduce the generator loss defined in Equation 4. The enhanced generator poses additional challenges for the discriminator to differentiate the real and the generated load profiles. As a result, the accuracy of the discriminator decreases after the second half of each epoch. These cycles repeat throughout the whole training process, and finally improve both the generator and the discriminator.

Another observed trend was that as the training progressed, the accuracy of the discriminator decreased, while the accuracy of the generator increased. During early epochs of the training, the generator weights are random and less trained. As a result, the generated load profiles are distinct from the real ones. It is not difficult for the discriminator to distinguish the real load profiles from the generated ones. Therefore, at the beginning of the training process, the accuracy of discriminator is high, and it can almost reach 100% after each epoch's training. With the training process going on, the generator can generate load profiles that are more similar to the real ones. However, it is more challenging for the discriminator to detect the real load profiles from the fake ones. As a result, the accuracy of the discriminator between epoch 140 and 150 is not as high as it is in earlier epochs.

3.4. Validation

Two approaches were adopted to validate whether GAN can generate load profiles which are similar to the reality. The first approach is to compare the mean and standard deviation of the key parameters of the generated and real smart meter data for each cluster. Again, the five key parameters proposed by Price [2] were used. The difference between the generated and real smart meter data is presented in the form of heat map in Fig. 10 (a). It can be seen the generated load profiles have very similar mean value and standard deviation of the five key parameters. The difference of base load is a little bit higher in some of non-working clusters (such as cluster 5 and 14).

The second approach we used is to compare the distribution of the five key parameters of the generated and original smart meter data using the metrics of Kullback–Leibler (KL) divergence. The KL divergence is a measure of how one probability distribution is different from another, reference probability distribution. A KL divergence of 0 indicates that the two distributions in comparison are identical. KL divergence is mathematically defined as Equation 5. Where, P and Q are probability distributions that defined on the same space X. KL divergence is an asymmetric metrics, P is usually the distribution of the real data, while Q is the distribution of the generated data.

We plotted the KL divergence of the distributions of the five key statistics for each cluster in Fig. 10 (b). It can be observed that the KL divergence of majority metrics for each cluster is less than 0.3, indicating the parameter distributions of the generated load profiles are very similar to those of the original data. The only parameter which has a relatively large KL divergence is the base load, which can be due to the base load has a smaller absolute value, and accordingly has a larger generation error. To further illustrate the implication of KL divergence, we plotted the distribution of base load for the cluster 2, which has a lower KL divergence; and the cluster 8, which has the highest KL divergence. Even for cluster 8, the distribution of the generated load is similar to that of the real load, confirming the validity of our proposed load generation approach.

4. Discussion

4.1. Hyper-parameter tuning

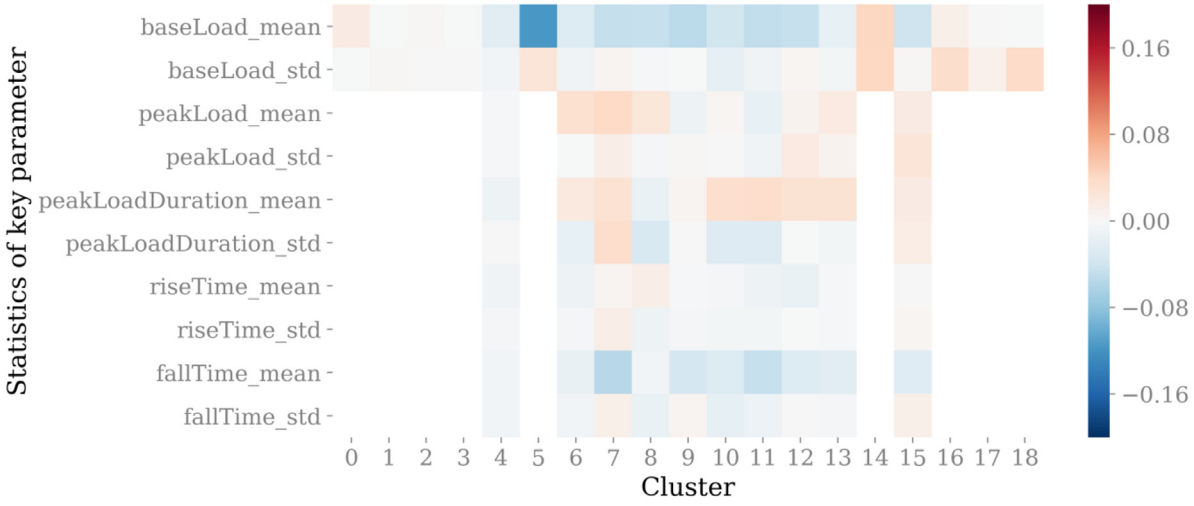
Hyper-parameter tuning is especially challenging in training a GAN model because GAN is a dynamic system consisting of two neural networks, with adversarial loss functions. The typical approach to train a deep neural network gradient descent method normally consists of going downhill in a static loss landscape. But with GAN, the loss landscape is no longer static. Every step taken down the hill changes the entire landscape a little, which makes it more difficult to find the minimal point through the gradient descent method [41].

To efficiently train GAN, tricks provided by previous studies were tested. Some of them work in this case (for instance, adding a dropout layer in the discriminator), but some do not (for instance, adding random noise to the labels for the discriminator). Table 3 presents the detailed hyper-parameters selected in this study, which can be helpful for other researchers.

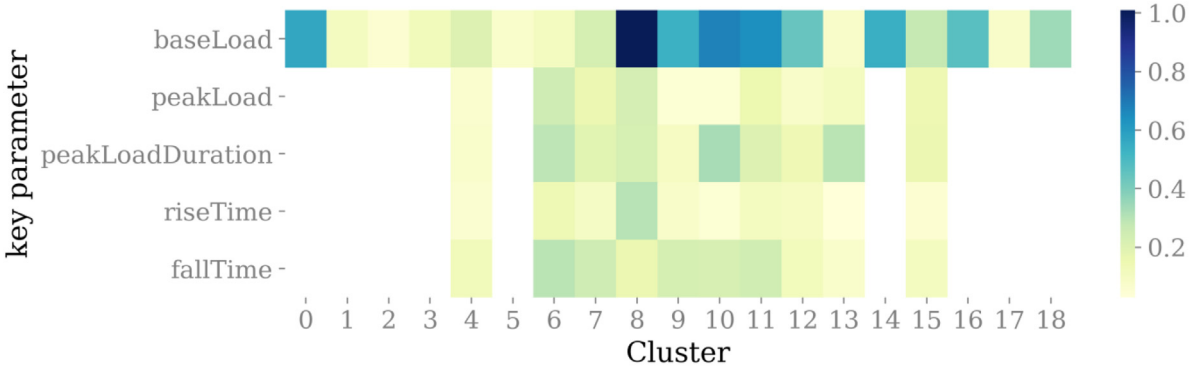
For generative models such as GAN, there are no ground truths to evaluate the performance of the model. Therefore, it is challenging to decide when we should stop training because the concepts of underfitting and overfitting are defined on ground truths. In this study, we used two approaches: the typology of the generated load profile (as shown in Fig. 8) and some quantitative metrics (as shown in Fig. 10) to decide when we stop training.

4.2. Potential applications

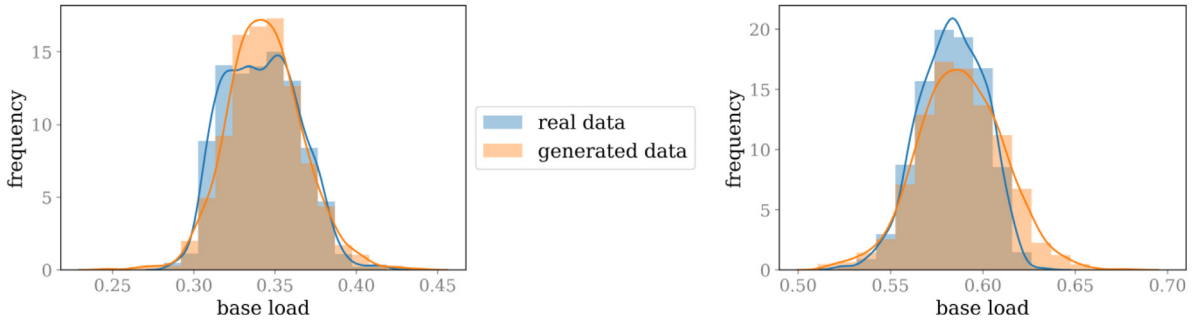
As a new approach to generating building electrical load profiles, it is important to know what GAN can do and what it cannot.



(a) difference of statistics (mean and standard deviation) of key parameters



(b) KL divergence of key parameters



(c) cluster 2 (low KL divergence)

(d) cluster 8 (high KL divergence)

Fig. 10. Model validation through comparing mean, standard deviation, and KL divergence of key parameters (Clusters 0, 1, 2, 3, 5, 14, 16, 17, 18 only have the value of key parameter of base load because those clusters are from non-working day, which do not have peak load periods, rise time or fall time. The peak load duration, rise time and fall time were normalized by dividing 24 h).

We will discuss what it can do in this section and what it cannot do in the next.

The first potential application of GAN is to anonymize real building electrical load profiles. The privacy concern is considered to be a major barrier for smart meter data analytics, because it prevents end users and utility companies from donating and sharing their data [9]. GAN can be used to anonymize smart meter data, because it is capable of generating load profiles that maintain key statistical information (both the general trend and random variations) while removing sensitive information, because the load profile is randomly generated. This process contains two steps: first, using the real load profiles to train the generator; and second, using the trained generator to generate load profiles and open source the generated load profiles rather than the original real one. To train the generator, we still need to get access to the real smart meter data, but it can be done in a secure and private way. Compared with directly sharing the original real data with the public, sharing the generated load profiles (which are different from the real data but have same statistics) preserve the value of data while addressing data privacy concerns. Using GAN to anonymize smart meter data may encourage more data owners and utility operators to share their data.

Table 3
Hyper-parameters of GAN settings in this study.

Latent space	Dimension	20
	Sampling method	Normal distribution, NOT uniform distribution
Generator	Layer 1	Densely connected; 64 units; activation = LeakyReLU
	Layer 2	Densely connected; 128 units; activation = LeakyReLU
	Layer 3	Batch normalization; momentum = 0.8
	Layer 4	Densely connected; 64 units; activation = LeakyReLU
	Output Layer	Densely connected; 24 units
	Loss function	Binary cross entropy
	Optimizer	Adam (**Ada**ptive **m**oment estimation) [42]: learning rate = 0.0002; clipvalue = 1; learning rate decay = 4e-6
	Training epochs	100
	Training callback	early stopping, patience = 3
Discriminator	Layer 1	Densely connected; 64 units; activation = LeakyReLU
	Layer 2	Densely connected; 128 units; activation = LeakyReLU
	Layer 3	Densely connected; 128 units; activation = LeakyReLU
	Layer 4	Densely connected; 64 units; activation = LeakyReLU
	Layer 5	Dropout; dropout rate = 0.4
	Output Layer	Densely connected; 1 unit; activation = sigmoid
	Loss function	Binary cross entropy
	Optimizer	Adam: learning rate = 0.0004; clipvalue = 1; learning rate decay = 8e-6
	Training epochs	50
	Training callback	early stopping, patience = 3
Training	Total epochs	150
	Train-test split	Training fraction = 0.8
	Batch size	2048

The second application of GAN is to generate electrical load profiles based on exogenous variables, such as building type, vintage, day type (working or non-working) and weather. The two-step workflow to predict load profiles is shown in Fig. 11. The first step is to predict which cluster the load profile will belong to, given critical information of the building and the weather. Then we can use GAN to generate the load profiles, given the predicted cluster.

The GAN discriminator can also be used to verify other load profile generation models by leveraging the trained discriminator to test if it can distinguish the generated load profiles from the real ones. If not, then the load profile generation model, which can be either physics-based white-box or data-driven black-box, is validated.

The last but not least, a potential application of GAN is to detect or diagnose changes to building electrical load profiles. The load profile changes can be diagnosed if the discriminator detected a marked difference between the newly measured load profile and the previous load profile patterns, which can be due to building

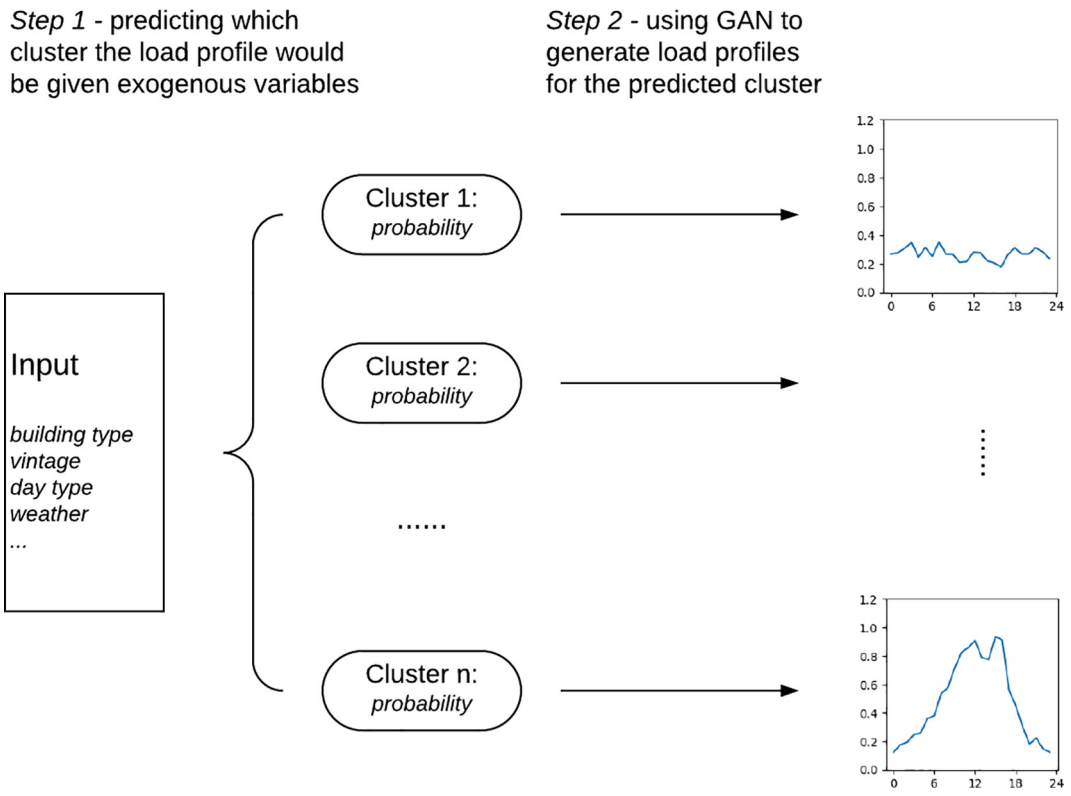


Fig. 11. Load profile prediction with GAN.

operational changes (e.g., occupancy, use hours), equipment or thermostat malfunction [1], or extreme events. This information can be helpful for fault detection and building operation improvement.

4.3. Limitations

As a data-driven approach, GAN can not explain what it generates. We have little information about how the load is constituted or whether novel control strategies can be used to shift or shed the peak load, which may be possible to know in a physics-based white-box model.

Additionally, GAN can only work within the domain of training data, and can not generate load profiles it has never seen before. For instance, if the building's energy system was retrofitted, its building load profiles may change dramatically. In this case, we can not input load profiles before renovation into GAN to generate load profiles after renovation.

5. Conclusions

Building electrical load profiles have wide applications in research and analysis of building operations, energy efficiency, and building-grid interactions. Current approaches to obtain these load profiles may be either time-consuming, unable to reflect the dynamic and stochastic behaviors of real buildings, or have privacy concerns. In this study, we proposed a novel approach to generate realistic electrical load profiles through the Generative Adversarial Network (GAN), an unsupervised machine learning technique that can be used to reveal an unknown probability distribution from the smart meter data.

Our proposed approach has three steps. We first normalized the daily load profiles to 95 percentile of the annual peak load to make sure the load profiles of different buildings are similar in magnitude. Then we use k-means to cluster the daily load profiles into 19 clusters. The optimal number of clusters is selected with the Davies-Bouldin Index. Lastly, we use GAN to generate the load profiles for each cluster.

We tested our approach with the open source Building Data Genome Project database. We validated the method we proposed by comparing the mean, standard deviation, and distribution of the key parameters of the generated load profiles with those of the real ones. It can be observed that the generated load profiles have very similar mean and standard deviation of the five key parameters (base load, peak load, peak load duration, rise time, and fall time) with the real ones. The KL divergence of the above five parameters between the generated and real load profiles are within 0.3 for majority clusters. Additionally, the results showed the load profiles generated by GAN can capture not only the general trend but also the random variations of actual building loads.

The proposed GAN approach can be used to anonymize smart meter data, generate load profiles, verify other load profile generation models, and detect changes to load profiles. Future research includes verifying the GAN approach for other building types and on a larger dataset of smart meter data, as well as interpretation of the clustering results with real building characteristics, weather, operational parameters, and occupant behavior.

CRediT authorship contribution statement

Zhe Wang: Conceptualization, Methodology, Investigation, Data curation, Writing - original draft, Writing - review & editing. **Tianzhen Hong:** Conceptualization, Methodology, Writing - review & editing, Resources, Visualization, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. Yao, K. Steemers, A method of formulating energy load profile for domestic buildings in the UK - ScienceDirect, Energy Build. 37 (6) (2005) 663–671.
[2] P. Price, "Methods for Analyzing Electric Load Shape and its Variability," Lawrence Berkeley National Laboratory, LBNL-3713E, May 2010.
[3] X. Liu, A. Heller, P.S. Nielsen, CITIESData: a smart city data management framework, Knowl. Inf. Syst. 53 (3) (2017) 699–722, https://doi.org/10.1007/s10115-017-1051-3.
[4] Y. Wang, Q. Chen, C. Kang, Q. Xia, M. Luo, Sparse and redundant representation-based smart meter data compression and pattern extraction, IEEE Trans. Power Syst. 32 (3) (2017) 2142–2151, https://doi.org/10.1109/TPWRS.2016.2604389.
[5] S. Joseph, J.E. Abdu, Real-time retail price determination in smart grid from real-time load profiles, Int. Trans. Electr. Energy Syst. 28 (3) (2018), https://doi.org/10.1002/etep.2509 e2509.
[6] N. A. Mims, T. Eckman, and C. Goldman, "Time-varying value of electric energy efficiency," Jun. 2017.
[7] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building, 2010, pp. 61–66.
[8] Y. Wang, Q. Chen, T. Hong, C. Kang, Review of smart meter data analytics: applications, methodologies, and challenges, IEEE Trans. Smart Grid 10 (3) (2019) 3125–3148, https://doi.org/10.1109/TSG.2018.2818167.
[9] J. Hu, A.V. Vasilakos, Energy big data analytics and security: challenges and opportunities, IEEE Trans. Smart Grid 7 (5) (2016) 2423–2436, https://doi.org/10.1109/TSG.2016.2563461.
[10] C. Fan, S. Huang, Y. Lai, Privacy-enhanced data aggregation scheme against internal attackers in smart grid, IEEE Trans. Ind. Inform. 10 (1) (2014) 666–675, https://doi.org/10.1109/TII.2013.2277938.
[11] K. Xing, C. Hu, J. Yu, X. Cheng, F. Zhang, Mutual Privacy Preserving k -Means Clustering in Social Participatory Sensing, IEEE Trans. Ind. Inform. 13 (4) (2017) 2066–2076, https://doi.org/10.1109/TII.2017.2695487.
[12] L. Wei, A.D. Sarwate, J. Corander, A. Hero, V. Tarokh, "Analysis of a privacy-preserving PCA algorithm using random matrix theory", in, IEEE Global Conference on Signal and Information Processing (GlobalSIP) 2016 (2016) 1335–1339, https://doi.org/10.1109/GlobalSIP.2016.7906058.
[13] D. Engel, "Wavelet-based load profile representation for smart meter privacy", in, IEEE PES Innovative Smart Grid Technologies Conference (ISGT) 2013 (2013) 1–6, https://doi.org/10.1109/ISGT.2013.6497835.
[14] C.F. Walker, J.L. Pokoski, Residential Load Shape Modelling Based on Customer Behavior, IEEE Trans. on Power Apparatus and Syst. PAS-104 (7) (1985) 1703–1711, https://doi.org/10.1109/TPAS.1985.319202.
[15] J.K. Gruber, M. Prodanovic, "Residential Energy Load Profile Generation Using a Probabilistic Approach," in: 2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation, 2012, pp. 317–322, doi: 10.1109/EMS.2012.30.
[16] W. Wang, T. Hong, N. Li, R.Q. Wang, J. Chen, Linking energy-cyber-physical systems with occupancy prediction and interpretation through WiFi probe-based ensemble classification, Appl. Energy 236 (2019) 55–69, https://doi.org/10.1016/j.apenergy.2018.11.079.
[17] Z. Wang, T. Hong, M.A. Piette, Data fusion in predicting internal heat gains for office buildings through a deep learning approach, Appl. Energy 240 (2019) 386–398, https://doi.org/10.1016/j.apenergy.2019.02.066.
[18] M. Deru et al., "U.S. Department of Energy commercial reference building models of the national building stock," NREL Report No. TP-5500-46861, Feb. 2011.
[19] Z. Wang, T. Hong, R. Jia, Buildings. Occupants: a Modelica package for modelling occupant behaviour in buildings, J. Build. Perform. Simul. 12 (4) (2019) 433–444, https://doi.org/10.1080/19401493.2018.1543352.
[20] A.C. Menezes, A. Cripps, D. Bouchlaghem, R. Buswell, Predicted vs. actual energy performance of non-domestic buildings: Using post-occupancy evaluation data to reduce the performance gap, Appl. Energy 97 (2012) 355–364, https://doi.org/10.1016/j.apenergy.2011.11.075.
[21] Z. Wang, Z. Zhao, B. Lin, Y. Zhu, Q. Ouyang, Residential heating energy consumption modeling through a bottom-up approach for China's Hot Summer-Cold Winter climatic region, Energy Build. 109 (2015) 65–74, https://doi.org/10.1016/j.enbuild.2015.09.057.
[22] R. Nesbakken, Price sensitivity of residential energy consumption in Norway, Energy Econ. 21 (6) (1999) 493–515, https://doi.org/10.1016/S0140-9883(99)00022-5.
[23] E. Hirst, R. Goeltz, D. White, Determination of household energy using 'fingerprints' from energy billing data, Int. J. Energy Res. 10 (4) (1986) 393–405, https://doi.org/10.1002/er.4440100410.
[24] M. Aydinalp-Koksal, V.I. Ugursal, Comparison of neural network, conditional demand analysis, and engineering approaches for modeling end-use energy consumption in the residential sector, Appl. Energy 85 (4) (2008) 271–296, https://doi.org/10.1016/j.apenergy.2006.09.012.

[25] B.E. Tonn, D.L. (Oak R. N. L. White, "Residential electricity use, wood use, and indoor temperature; An econometric model," Energy Syst. Policy USA, vol. 12:3, Jan. 1988.
[26] M. Aydinalp, V. Ismet Ugursal, A.S. Fung, Modeling of the space and domestic hot-water heating energy-consumption in the residential sector using neural networks, Appl. Energy 79 (2) (2004) 159–178, https://doi.org/10.1016/j.apenergy.2003.12.006.
[27] L.G. Swan, V.I. Ugursal, Modeling of end-use energy consumption in the residential sector: A review of modeling techniques, Renew. Sustain. Energy Rev. 13 (8) (2009) 1819–1835, https://doi.org/10.1016/j.rser.2008.09.033.
[28] C. Miller, F. Meggers, The Building Data Genome Project: An open, public data set from non-residential building electrical meters, Energy Procedia 122 (2017) 439–444, https://doi.org/10.1016/j.egypro.2017.07.400.
[29] Energy Information Administration (EIA), "Commercial Buildings Energy Consumption Survey (CBECS)." [Online]. Available: https://www.eia.gov/consumption/commercial/. [Accessed: 30-May-2019].
[30] X. Luo, T. Hong, Y. Chen, M.A. Piette, Electric load shape benchmarking for small- and medium-sized commercial buildings, Appl. Energy 204 (2017) 715–725, https://doi.org/10.1016/j.apenergy.2017.07.108.
[31] R. Granell, C.J. Axon, D.C.H. Wallom, Impacts of raw data temporal resolution using selected clustering methods on residential electricity load profiles, IEEE Trans. Power Syst. 30 (6) (2015) 3217–3224, https://doi.org/10.1109/TPWRS.2014.2377213.
[32] G. Chicco, Overview and performance assessment of the clustering methods for electrical load pattern grouping, Energy 42 (1) (2012) 68–80, https://doi.org/10.1016/j.energy.2011.12.031.
[33] T. Zhang, G. Zhang, J. Lu, X. Feng, W. Yang, A new index and classification approach for load pattern analysis of large electricity customers, IEEE Trans. Power Syst. 27 (1) (2012) 153–160, https://doi.org/10.1109/TPWRS.2011.2167524.
[34] O.Y. Al-Jarrah, Y. Al-Hammadi, P.D. Yoo, S. Muhaidat, Multi-layered clustering for power consumption profiling in smart grids, IEEE Access 5 (2017) 18459–18468, https://doi.org/10.1109/ACCESS.2017.2712258.
[35] I. Benítez, A. Quijano, J.-L. Díez, I. Delgado, Dynamic clustering segmentation applied to load profiles of energy consumption from Spanish customers, Int. J. Electr. Power Energy Syst. 55 (2014) 437–448, https://doi.org/10.1016/j.ijepes.2013.09.022.
[36] M. Halkidi, Y. Batistakis, M. Vazirgiannis, On clustering validation techniques, J. Intell. Inf. Syst. 17 (2) (2001) 107–145, https://doi.org/10.1023/A:1012801612483.
[37] P.J. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65, https://doi.org/10.1016/0377-0427(87)90125-7.
[38] I. Goodfellow et al., "Generative Adversarial Nets," in Advances in Neural Information Processing Systems 27, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 2672–2680.
[39] "Home - Keras Documentation." [Online]. Available: https://keras.io/. [Accessed: 14-Jun-2019].
[40] "TensorFlow," TensorFlow. [Online]. Available: https://www.tensorflow.org/. [Accessed: 14-Jun-2019].
[41] F. Chollet, "Deep Learning with Python," in Deep Learning with Python, Second edition., Manning Publications Company, 2017, p. 337.
[42] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014.