Smart Home Energy Management: VAE-GAN Synthetic Dataset Generator and Q-Learning

Mina Razghandi, *Graduate Student Member, IEEE*, Hao Zhou, *Graduate Student Member, IEEE*, Melike Erol-Kantarci, *Senior Member, IEEE*, and Damla Turgut, *Senior Member, IEEE*

Abstract—In recent years, there has been a growing interest in academia and industry in the analysis of electrical consumption in residential buildings and the implementation of smart home energy management systems (HEMS) to reduce household energy usage and costs. HEMS have been designed to emulate the statistical and functional characteristics of real smart grids. However, a major challenge in this research area is the limited availability of publicly accessible datasets. To address this challenge and further leverage the potential of artificial HEMS applications, it is crucial to develop time series that accurately represent diverse operating conditions of synthetic systems. This paper introduces a novel approach based on the combination of variational auto-encoder-generative adversarial network (VAE-GAN) techniques to generate time-series data of energy consumption in smart homes. Additionally, we investigate the performance of the generative model when integrated with a Q-learning based HEMS. The effectiveness of the Q-learning based HEMS is assessed through online experiments using real-world smart home data. To evaluate the quality of the generated dataset, we employ various metrics including Kullback–Leibler (KL) divergence, maximum mean discrepancy (MMD), and the Wasserstein distance, which quantify the disparities between probability distributions of the real and synthetic data. Our experimental results demonstrate that the synthetic data generated by VAE-GAN closely aligns with the distribution of real data. Furthermore, we demonstrate that the utilization of the generated data facilitates the training of a more efficient Q-learning based HEMS, surpassing the performance achieved with datasets generated using baseline approaches.

Index Terms—Synthetic data, load consumption, smart grid, deep learning, generative adversarial network, q-learning.

I. INTRODUCTION

THE DESIGN of modern smart homes must take into consideration energy-efficient technologies and renewable energy sources. Recent advances in smart grid research include new technologies and strategies for managing energy generation, storage, supply, and demand [1]. Smart homes, as a critical component of the smart grid, are projected to increase household energy efficiency, save energy costs, and improve user comfort [2]. Utilities and smart home controllers can acquire information from smart meters for use in forecasting consumption and generation, demand-side management, and economic power dispatch [3]. Thus, acquiring fine-grained data about the living environment is becoming an essential precondition for a smart home.

It has become increasingly common to use artificial intelligence and statistical analysis for demand response and energy management tasks requiring accurate short-term (second-to-minute scale) representations of load behavior. These techniques usually require large datasets of representative data for training. However, the collection of such data presents significant security and privacy challenges, with relatively few high-quality publicly available datasets [4].

Therefore, synthetic data generation become an attractive alternative to training machine learning algorithms that can decide, for example, the optimal time for implementing demand response or charging an EV [5]. A variety of techniques had been used to generate such datasets, including Markov chains [6], statistical models [7], and physical simulator-based methods [8]. A major drawback of these methods can be attributed to targeting a limited number of problems, applications tailored to specific scenarios, and the lack of scalability. Specifically, these methods require case-by-case analyses for each device, user behavior, and environment. But it is impractical to model all the detailed energy consumption behaviors and state transitions when a large number of user devices are involved. Recent studies, on the other hand, have utilized deep learning-based approaches that can be applied to large-scale datasets and can be conducted directly on raw data, without the need for further analysis [9].

In our recent paper [10], our approach involved generating electric load profiles and PV generation synthetic data for a smart home using variational autoencoder-generative adversarial networks (VAE-GAN), and we compared the distributions of synthetic data for the VAE-GAN model and vanilla GAN network to the real data distributions. According to the statistical metrics such as KL divergence, Wasserstein distance, and maximum mean discrepancy, the distribution of data points was extremely close between synthetic data

and real data for both PV power generation and energy consumption load profiles. Nevertheless, it is a key defining characteristic of a realistic time series to maintain an awareness of temporal differences between synthetic and real-time series. For instance, If PV power production shifted earlier or later in the day rather than following sunrise and sunset patterns, the distribution would still be similar to the real data, but it would not be realistic. The addition of electric vehicles (EV) charging load consumption synthetic data could provide additional insights and value in understanding smart home electricity consumption patterns. Also, no experiments were conducted to investigate if the synthetic dataset is applicable in real-life practice. To address these shortcomings, we propose a revised architecture for the variational autoencoder-generative adversarial network (VAE-GAN) technique for generating time-series data of smart homes from a variety of synthetic data sources and investigate the performance of synthetic data generative models in the presence of control operations. In contrast to the schemes mentioned above, this strategy enables the learning of various types of data distributions in a smart home, including electric load profiles, PV power generation, and EV charging load consumption, and can subsequently generate plausible samples without carrying out any prior analysis before the training phase. The VAE-GAN architecture is favored in this model since it allows us to fine-tune the regulated latent space that influences the generated output, as opposed to vanilla GAN, in which the generator maps the input noise to the generated output and makes it susceptible to mode collapse, in which the generator repeatedly produces the same optimal input sequence to mislead the discriminator [11]. Furthermore, we compare one model-based and two data-driven generative models, including Gaussian Mixture Model (GMM), vanilla GAN, and VAE-GAN.

Taking advantage of the synthetic data, we propose a Q-learning-based smart home energy management system (HEMS). The generated data is used for the offline training of Q-learning HEMS agents, which aims to maximize long-term management profit. Then, the trained agents are tested online in an environment based on real-world data. Finally, we compare online profits to further investigate how the data generation method will affect the HEMS performance. The idea behind this scheme is that we assume good-quality synthetic data in the offline training can better prepare the agent for real-world operations [12]. Thus, the performance of the Q-learning-based HEMS is used to demonstrate the quality of the synthetic data.

Compared with conventional model-based optimization algorithms, such as convex optimization, the reinforcement learning (RL) based method can avoid the complexity of defining a sophisticated optimization model since the optimization problem can be transformed into a unified Markov decision process (MDP) scheme. On the other hand, most existing HEMS models assume a perfect environment for data collection and algorithm training [13]. By contrast, the proposed Q-learning HEMS uses synthetic data for training, overcoming the data availability bottleneck.

The main contributions of this paper are as follows:

- We propose a VAE-GAN-based scheme to generate high temporal resolution synthetic time-series data for energy consumption in a smart home.
- We compare the performance of the proposed approach with techniques based on Gaussian Mixtures and vanilla GANs.
- We further investigate the quality of the synthetic data by using it to train a Q-learning-based HEMS model. Evaluating the HEMS agent online in environments based on real-world data, we find that the VAE-GAN method allows higher HEMS profit than other baselines.

The rest of this paper is organized as follows. Section II introduces related work. Section III shows the proposed smart home synthetic data generation model and Q-learning-based HEMS model. Section IV presents the simulation settings and results, and Section V concludes the paper.

II. RELATED WORK

The majority of previous work in generating synthetic energy consumption data for smart homes can be classified into model-based or data-driven approaches. Model-based approaches describe the features of household devices with hand-crafted features and mathematical equations. For instance, [14] proposes a bottom-up residential building energy simulation, which simulates occupant behavior patterns with a Markov chain clustering algorithm. Reference [6] combined non-intrusive load decomposition and Markov chain methods for user energy consumption simulation. Reference [8] defines a smart residential load simulator based on MATLAB-Simulink, and it includes dedicated physical models of various household devices. References [7] and [15] utilize Gaussian Mixture Modeling to estimate the distribution of the arrival and departure time of electric vehicles (EVs) and to perform temporal modeling of the charging sessions.

On the other hand, recent developments in using GANs have achieved great success in producing synthetic time-series data. Reference [16] coupled non-intrusive load decomposition with a conditional GAN to generate synthetic labeled (e.g., appliances) load for a smart home. For smart grid applications, [17] proposes a GAN-based approach that captures spatial and temporal correlations between renewable energy sources. Reference [18] applies a GAN to the features derived by ARIMA and Fourier transform for generating realistic energy consumption data. Reference [4] uses a GAN to learn the level and pattern features of smart home time-series data, both of which are defined by household consumption and activity, and generate synthesized data with a distribution similar to the real data. Reference [19] clusters daily load profiles with known clustering techniques and uses a GAN to synthesize daily loads for each cluster. Reference [20] proposed a GAN with a Wasserstein loss function to learn temporal and power patterns of EV charging sessions and create a synthetic load dataset.

Reinforcement learning-based energy management models have been extensively investigated in the smart grid. For example, a correlated Q-learning-based approach is proposed
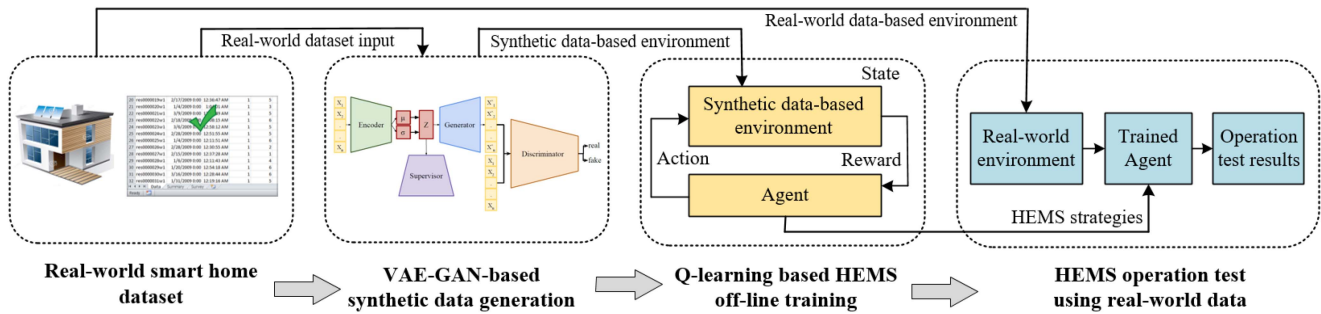
Fig. 1.   Overall architecture of the proposed scheme.

in [5] for microgrid energy management, and a multi-agent reinforcement learning method is introduced in [21] to minimize the energy cost of different smart home devices. A self-learning HEMS model is presented in [22], which includes price forecasting, price clustering, and power alert systems. Reference [23] introduced a Bayesian deep reinforcement learning method for microgrid energy management under communication failure.

However, most aforementioned make the strong assumption of a perfect data collection and agent training environment. In a real-world environment, fine-grained and good-quality data may be inaccessible due to privacy concerns or measurement errors. As such, synthetic data-based HEMS is much more realistic since the agent can use pre-generated and fine-grained data for training.

III. METHODOLOGY

The overall architecture of the proposed approach is described in Fig. 1, and the organization of this work is introduced as followings:

- *Synthetic data generation:* Given a real-world smart home dataset, we first apply the VAE-GAN-based method to produce synthetic data, including the smart home load profiles, PV power generation, and EV charging load consumption, which is introduced in the following Sections III-A–III-D.
- *Q-learning based HEMS training:* Next, the generated synthetic data is used for the offline training of Q-learning based HEMS. The intelligent HEMS agent will interact with a synthetic data-based environment and produces HEMS strategies to maximize long-term profit, which is included in Section III-E.
- *Real-world HEMS operation test:* Finally, the trained HEMS agent will be tested in a real-world data-based environment. The test is designed to evaluate the real-world performance of the synthetic data-based HEMS, providing an in-depth evaluation of the synthetic data quality. Specifically, it is expected to demonstrate that our synthetic data can be applied to HEMS training, and the agent can obtain a satisfying real-world performance without touching real-world data.

Among the well-known deep learning-based generative models, GAN and Variational Autoencoders are two of the best known. In this network, the encoder module first encodes the input sequence as a Gaussian distribution over the latent space. Then, the supervisor module trains the encoder module to approximate the next time step. In the generator module, the input sequence is reconstructed from the latent space in an attempt to fool the discriminator. On the other hand, the discriminator module trains the generator to create realistic sequences by identifying fake samples.

In the following, we first introduce the autoencoder (AE) and variational autoencoder (VAE), then we present generative adversarial network (GAN) and VAE-GAN generative models. Finally, we include the Q-learning-based HEMS control.

A. Autoencoder (AE)

An autoencoder is a neural network architecture that uses an unsupervised learning technique to compress the input dimensions into a compressed knowledge representation, called latent space, and then reverses the compressed knowledge representation into the original input dimensions. This architecture has two essential modules: the encoder and the decoder. The encoder module maps the input sequence (x) into the meaningful latent space (z), and based on z the decoder module outputs a reconstruction of the original input sequence ($\hat{x}$). The original input is unlabeled, but the decoder is trained to make reconstructions as close as possible to the original input by minimizing the reconstruction error, $\mathcal{L}_{reconstr}$, which is the distance between the original input and the subsequent reconstruction.

$$\mathcal{L}_{reconstr} = \|\hat{x} - x\|^2 \tag{1}$$

where x and $\hat{x}$ are original input and the reconstruction, respectively.

B. Variational Autoencoder (VAE)

After training the autoencoder model, the decoder module can produce new content given a random latent space. Due to the unregulated latent space, which can lead to severe overfitting, the decoder may not be able to interpolate indefinitely in the absence of data points included in the input sequence. Latent space regularity is reliant on the distribution of the input sequence and the dimension of the latent space, hence it is not always guaranteed. As a solution to this limitation, variational autoencoders include a regularization parameter, such as Kullback–Leibler (KL) divergence, in the learning process to achieve the Gaussian distribution of the latent space and avoid overfitting. In this architecture, the VAE encoder (E) encodes the input sequence (x) as a distribution over
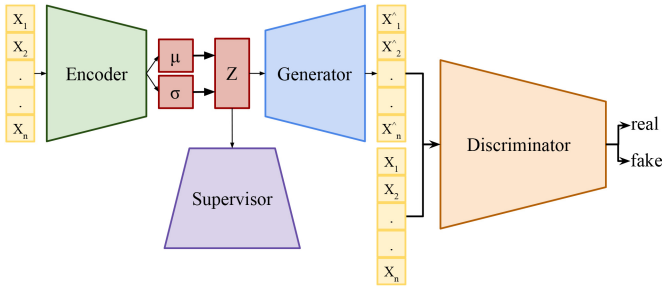
Fig. 2. VAE-GAN model architecture. In this network, the encoder module encodes the input sequence as a Gaussian distribution over the latent space, defined by mean and variance vectors. The supervisor module trains the encoder module to approximate the next time step closely in the latent space. In the generator module, the input sequence is reconstructed from the latent space in an attempt to fool the discriminator so the generated sequence is considered real. The discriminator module trains the generator module to create realistic sequences by identifying fake samples from real ones.

latent space defined by Gaussian distribution first moment, mean, and second moment, standard deviation, and parameters. Ultimately, the encoder is trained to minimize the $\mathcal{L}_{prior}$ loss and output a latent space with a Gaussian distribution. In the training process, the VAE minimizes a loss function ($\mathcal{L}_{VAE}$) that consists of two terms: the reconstruction term and the regularization term, which is the KL divergence between the latent space distribution and standard Gaussian distribution.

$$\mathcal{L}_{prior} = D_{KL}(E(x)||\mathcal{N}(0, 1)) \qquad (2)$$

$$\mathcal{L}_{VAE} = \mathcal{L}_{prior} + \mathcal{L}_{reconstr} \qquad (3)$$

where D_{KL} is the KL divergence, E is VAE encoder, $\mathcal{N}(0, 1)$ is the Gaussian distribution, and $\mathcal{L}_{prior}$ is the loss.

C. Generative Adversarial Network (GAN)

Generative models, such as GAN, are used to generate new data samples. This architecture consists of two neural networks: one attempts to detect and learn patterns in the input data and produce a new sample, while the other aims to distinguish between real input data and the synthesized sample. These two networks play against each other and actively seek equilibrium.

- *Generator* (G)*:* Takes an input noise z, which usually has a normal distribution, and maps that into data samples $G(z; \theta_g)$, with θ_g as the network parameter.
- *Discriminator* (D)*:* $D(x; \theta_d)$ returns the likelihood of x being classified as a member of the original data, where 0 represents fake data and 1 represents original data.

G and D engage in an adversarial game, equation (4), where G attempts to fool the D and maximize the final classification error, while D is trained to classify fake data more accurately and minimize the error.

$$\min_{G} \max_{D} \mathcal{L}_{GAN}(D, G) = E_x\big[log(D(x))\big]$$
$$+ E_z\big[1 - log(D(G(z)))\big] \qquad (4)$$

where G and D has been defined as generator and discriminator, x and z are input sequence and noise, respectively.

D. Data-Driven Generative Model

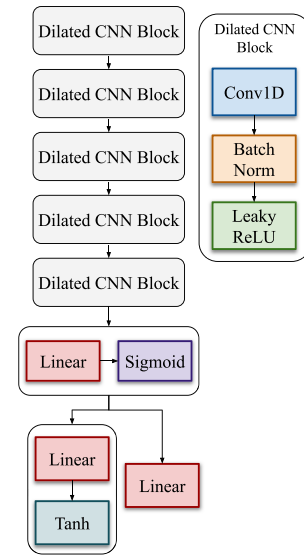To generate synthetic smart home data we adopted a VAE-GAN architecture, as presented in Fig. 2 that avoids the

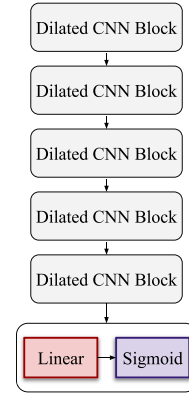

Fig. 3. VAE-GAN **encoder** module structure.

Fig. 4. VAE-GAN **generator**, **supervisor**, and **discriminator** modules structure.

shortcomings of the vanilla GAN or VAE. For example, vanilla GAN is prone to mode collapse, a situation where the generator finds an input sequence that fools the discriminator and repeatedly produces it over and over again. The VAE-GAN architecture, introduced by Larsen et al. [24], generates new data samples based on a regulated latent space rather than producing new data samples from a noise input. The discriminator will further classify the generated sample. Considering that the data for smart homes is time-series data, it is essential that the encoder be able to not only detect the underlying pattern and distribution in the input data but also retain the realistic order in which events occur. As a result, we took TimeGAN's [25] concept of incorporating a supervisor module into our VAE-GAN architecture. We will discuss each module in more detail.

As opposed to mapping the input sequence into a meaningful latent space, The **encoder** module (E) here translates the original input sequence x into two vectors that represent the mean (μ) and variance (σ) of a standard Gaussian distribution. Minimizing $\mathcal{L}_{prior}$ (equation (2)) forces the encoder to compress the data over a standard Gaussian distribution. μ and σ are mapped into the latent space z using the *reparameterization* technique. The encoder uses five layers of **Dilated One**

Dimensional Convolutions (dilated Conv1D) stacked one on top of the other (see Fig. 3). This architecture promises similar performance but lower complexity and faster convergence compared to the Long Short-Term Memory (LSTM) networks typically used for time-series analysis. The Conv1D layer architecture resembles the WaveNet architecture [26], as each layer has a stride length of one and kernel size of two, but the dilation rate varies in each layer. As the second layer dilates at any second of the input sequence (skipping every other timestep), the third layer dilates at any fourth, etc., the lower layers are more focused on short-term dependencies while the higher layers capture long-term dependencies.

The **supervisor** module computes the distance between the latent representation at the current time step (z) and the next time step ($\hat{z}$), minimizing $\mathcal{L}_{supervisor}$. Added to the encoder loss function, equation (6), allows the encoder to approximate the next time step in the latent space, retaining the original sequence of events.

$$\mathcal{L}_{supervisor} = \|z - \hat{z}\|^2 \tag{5}$$

$$\mathcal{L}_E = \mathcal{L}_{prior} + \mathcal{L}_{supervisor} \tag{6}$$

The **generator** module (G) is trained to reconstruct the original input sequence, given latent space z as the input by minimizing the Mean Squared Error (MSE) between the reconstructed sequence and original input sequence ($\mathcal{L}_{reconstr} + \mathcal{L}_{prior}$). The generator loss also contains the term $\mathcal{L}_{dG}$, computed as in equation (8), quantifying the likelihood of the discriminator classifying the reconstructed sequence as a fake sequence. The goal is to make the reconstructed sequence so realistic that it can mislead the discriminator.

$$\mathcal{L}_{dG} = E_x\big[log(D(G(z)))\big] \tag{7}$$

$$\mathcal{L}_{generator} = \mathcal{L}_{prior} + \mathcal{L}_{reconstr} + \mathcal{L}_{dG} + \mathcal{L}_{supervisor} \tag{8}$$

$$\mathcal{L}_{real} = E_x\big[log(D(x))\big] \tag{9}$$

$$\mathcal{L}_{fake} = E_z\big[1 - log(D(G(z)))\big] \tag{10}$$

$$\mathcal{L}_{noise} = E_z\big[1 - log(D(\mathcal{N}(0, 1)))\big] \tag{11}$$

$$\mathcal{L}_D = \mathcal{L}_{real} + \mathcal{L}_{fake} + \mathcal{L}_{noise} \tag{12}$$

The **discriminator** module (D) is responsible for classifying the original input sequence and fake data samples. D minimizes $\mathcal{L}_D$, equation (12), in the training process, which has three terms. $\mathcal{L}_{real}$, equation (9), is the likelihood of original input data being classified as fake data, $\mathcal{L}_{fake}$, equation (10), is the likelihood of reconstructed data sample being classified as real, and $\mathcal{L}_{noise}$, equation (11), is the likelihood of classifying a random noise input as a real data which is added to $\mathcal{L}_D$ to improve the convergence of the discriminator.

E. Q-Learning Based HEMS Control

In this section, we introduce the Q-learning-based HEMS model. As shown in Fig. 5, the components of the smart home involved in the energy production and consumption include the PV panels, loads, EV, and the ESS:
- *PV:* The PV is considered a pure energy supplier. The PV power will first serve the energy demand of EVs and other smart home devices, then the surplus energy can be
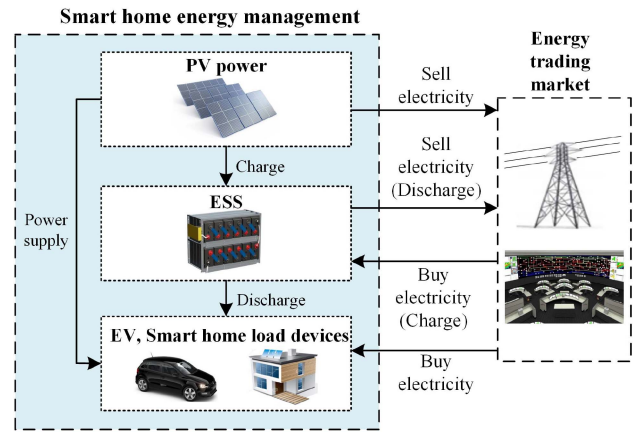

Fig. 5. The proposed smart home energy management system.

used for ESS charging or selling to the energy trading market for profit.
- *EV and other smart home loads:* EV and other smart home loads are pure energy consumers. They will first receive the power supply from PV or ESS for operation or buy electricity from the market if the internal power supply is insufficient.
- *ESS:* Finally, the ESS can be a power supplier when discharging, or a consumer when charging. For charging, it uses the surplus PV power or buys electricity from the market. For discharging, it supplies energy to EVs and smart home devices or sells electricity to the energy trading market for profit.

The HEMS model takes advantage of the flexibility of ESS to minimize the total energy cost and maximize profit. The optimization task of the centralized HEMS model is described as follows:

$$\max \quad \sum_{t=1}^{T} P^{total}\Big(\gamma p_t^{sell} + (1 - \gamma)p_t^{buy}\Big) \tag{13}$$

$$\text{s.t.} \quad P^{total} = P_t^{ESS} + P_t^{PV} - P_t^{L} - P_t^{EV} \tag{13a}$$

$$\gamma = \mathbb{1}\Big\{P^{total} > 0\Big\} \tag{13b}$$

$$P_t^{ESS} = P_{ch}q_t \tag{13c}$$

$$q_t = \begin{cases} -1 & ESS \quad charges \\ 0 & ESS \quad unchanged \\ 1 & ESS \quad discharges \end{cases} \tag{13d}$$

$$Soc_{t+1} = Soc_t - \frac{P_t^{ESS}}{C^{ESS}} \tag{13e}$$

$$Soc_{min} \le Soc_t \le Soc_{max} \tag{13f}$$

where T is the total optimization period, p_t^{sell} and p_t^{buy} represent the price of selling energy to the energy trading market and buying energy from the market, respectively. P_t^{ESS}, P_t^{PV}, P_t^{L} and P_t^{EV} represent the power of ESS, PV, smart home load, and EV charging load at time slot t, respectively. P^{total} denotes the total power consumption/demand of the smart home, and (13a) is the energy balance constraint. $\mathbb{1}$ is an indicator function. $\mathbb{1}\{P^{total} > 0\} = 1$ when $P^{total} > 0$, which means the agent will sell surplus energy for profit; otherwise, $\mathbb{1}\{P^{total} > 0\} = 0$ if $P^{total} < 0$, which means buying

energy from the market. P_{ch} is the fixed ESS charging power. $q_t = -1, 0, 1$ when ESS charges, remain unchanged and discharges, respectively. C^{ESS} is the fixed capacity of ESS, and Soc is the ESS state of charge (SOC). Equations (13c) to (13e) are ESS operation constraints, and (13f) is the SOC upper and lower bound constraint.

To transform this optimization task into the context of Q-learning, we define a Markov decision process (MDP) as follows.

- *State:* The agent state is defined as $s_t = \{Soc_t, P_t^{PV}, P_t^{Load}\}$, where $P_t^{Load} = P_t^L + P_t^{EV}$ represents the total energy consumption of smart home.
- *Action:* Based on the total energy demand and PV power generation, the agent decides on the action $a_t = q_t$, which indicates the charging, discharging, or remaining unchanged status of ESS.
- *Reward:* By selecting actions intelligently, the agent intends to maximize the total profit in the optimization period, and the reward function is defined by:

$$r_t = P^{total}\left(\gamma p_t^{sell} + (1 - \gamma)p_t^{buy}\right), \qquad (14)$$

which is the objective function of our problem formulation equation (13).

In Q-learning, the agent aims to maximize the long-term expected reward

$$V^\pi(s) = E_\pi\left[\sum_{n=0}^{\infty} \gamma^n r(s_n, a_n)|s = s_0\right], \qquad (15)$$

where s_0 is the initial state, n is the number of iterations, E_π is the expected value under action selection policy π, $r(s_n, a_n)$ is the reward of selecting action a_n under state s_n, and γ is the reward discount factor. A lower γ means focusing on immediate rewards, and a high γ indicates that future reward is more important.

Then, we define the state-action value

$$Q^{new}(s_n, a_n) = Q^{old}(s_n, a_n) + \alpha\Big(r_n + \gamma \max_a Q(s_{n+1}, a) - Q^{old}(s_n, a_n)\Big), \qquad (16)$$

where $Q^{old}(s_n, a_n)$ and $Q^{new}(s_n, a_n)$ are old and new Q-values, respectively, indicating the accumulated reward of selecting action a_n under state s_n. s_{n+1} is the state of $n+1$ iteration, and α is the learning rate. A high learning rate will lead to a fast learning process, but the results can be unstable; otherwise, a lower learning rate may result in very slow convergence and a long training time.

In addition, we use the ϵ-greedy policy for the action selection.

$$\pi(s) = \begin{cases} \arg\max_a Q(s_{n+1}, a), & rand > \epsilon, \\ \text{random action selection}, & rand \le \epsilon. \end{cases} \qquad (17)$$

where *rand* represents a random number ($0 \le rand \le 1$), and $\epsilon < 1$. When $rand > \epsilon$, the agent takes greedy policy to select the action with maximum Q-value; otherwise, the action is selected randomly for exploration. ϵ-greedy policy can balance the exploration and exploitation of the Q-learning agent to maximize the long-term expected reward.

Algorithm 1 Q-Learning for HEMS

1: **Initialize:** Q-learning and smart home parameters
2: **Phase 1: Offline training using synthetic data:**
3:    **Input:** generated synthetic data of smart home load, EV load, and PV power generation.
4:    **for** $episode = 1$ to E **do**
5:       With probability ϵ choose action a randomly. Otherwise, $a = arg\ max(Q(s, a))$
6:       Agent calculates reward based on equation (13).
7:       Update agent state $\{t, S_t, w_{g,t}\}$ and Q-value:
8:       $Q(s, a) = (1 - \alpha)Q(s, a) + \alpha(r + \gamma\ maxQ(s', a'))$
9:    **end for**
10: **Output:** Trained Q-learning based HEMS strategy.
11: **Phase 2: Online test using real-world data:**
12:    **Input:** Real-world data.
13:    Deploy the pre-trained Q-learning agent for an operation test.
14: **Output:** HEMS profit under real-world data.

Finally, the Q-learning-based HEMS algorithm is summarized in Algorithm 1, which consists of two phases. In the offline training phase, given the generated synthetic data, the intelligent agent is trained to maximize the total profit. Then, in the online test phase, we apply real-world data for online operation, which aims to test the performance of agents that are trained on synthetic data from various datasets.

IV. EVALUATION STUDY

A. Dataset

1) Smart Home Dataset: The iHomeLab RAPT dataset [27] is a real-world dataset for residential power traces. Five households in Switzerland have been surveyed for 1.5 to 3.5 years with a sampling frequency of 5 minutes. The dataset contains residential electricity consumption and PV energy production, including appliance-level and aggregated household consumption data. Studies of PV power generation show recurring patterns with corresponding noises originating from rain and clouds. Consumption patterns show weekly trends, but with occasional irregular spikes.

The experiment was conducted on residential house D from the dataset. Data cleansing was done by choosing days with full 24-hour records, resulting in 594 days in total. As part of the training process, electrical load and PV power production data were downsampled to a resolution of 15 minutes. For the train and test datasets, a split ratio of 80:20 was adopted.

2) Residential Electric Vehicle Charging Dataset: Sørensen et al. [28] presented a residential electric vehicle charging dataset recorded from apartment buildings. From December 2018 to January 2020, 97 users in Norway reported real-world EV charging experiences. Each charging session includes plug-in time, plug-out time, and charged energy. This dataset provides a synthetic charging load for level-1 charging and level-2 charging assuming 3.6 kW or 7.2 kW of charging power. In our experiments, we interpolate the data for a 15-minute resolution. We note that a higher time scale resolution such as 5-minute will increase the simulation

running time, while a lower resolution such as 20 minutes can not capture the environment dynamics. In addition, the proposed synthetic data generator scheme can adapt to any time resolution without loss of generality.

Our analysis revealed that the user with the largest number of available records has only 62 days of 24-hour data. This data contains only 11 charging sessions, which is insufficient for the training of a deep learning model. To address this issue we generated a larger synthetic dataset by combining the charging data from several users and assuming that the data belonged to an indoor charging station shared by the building's residents, thus accumulating 263 days of historical data, We are thereby able to have more significant charging events for the model, resulting in a more efficient training process.

B. Baseline Models and Performance Metrics

In our experiments, we compared the performance of our VAE-GAN-based approach with two other state-of-the-art approaches:

- *Gaussian Mixture Model (GMM):* is a probabilistic approach that partitions data into groups using soft clustering based on multiple multidimensional Gaussian probability distributions. The mean and variance of the distributions are calculated using Expectation-Maximization. Upon fitting the GMM to some data, a generative probabilistic model can sample synthetic data, following the same distribution.
- *Vanilla GAN:* As described in Section III-C.
- *VAE-GAN:* As described in Section III-D

In the following, we introduce the metrics that we will employ to evaluate the quality of the generated synthetic data.

1) Kullback–Leibler (KL) Divergence: The Kullback-Leibler divergence (KLD) is one of the most often used measures for evaluating the similarity of two probability distributions. Equation (18) is the formal definition of KL divergence, where x and y are sampled data points, and $p(x)$ and $p(y)$ are their respective probability distributions. The KL divergence ranges from 0 when two probability distributions almost match everywhere, to ∞ for completely different distributions.

$$D_{KL}(p\|q) = \sum_{i=1}^{N} p(x_i) log\left(\frac{p(x_i)}{q(y_i)}\right) \qquad (18)$$

2) Maximum Mean Discrepancy (MMD): The MMD approach represents the distance between distributions by the distance between the mean embeddings of features into a reproducing kernel Hilbert space. Given the distributions $p(x)$ for $\{x_i\}_{i=0}^{N}$ and $q(y)$ for $\{y_j\}_{j=0}^{M}$, MMD is calculated as follows:

$$MMD(p, q)^2 = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} K(x_i, x_j)$$

$$- \frac{2}{MN} \sum_{i=1}^{N} \sum_{j=1}^{M} K(x_i, y_j)$$

$$+ \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} K(y_i, y_j) \qquad (19)$$

$$K(x, y) = \exp\left(\frac{-\|x - y\|^2}{2\sigma^2}\right) \qquad (20)$$

3) Wasserstein Distance: Intuitively, the Wasserstein distance, also called the earth mover's distance, models a probability distribution as a pile of soil and computes how much soil needs to be moved to transform one probability distribution to the other.

$$l_1(p, q) = \inf_{\pi \in \Gamma(p,q)} \int_{\mathbb{R} \times \mathbb{R}} |x - y| d\pi(x, y) \qquad (21)$$

Equation (21) represents the formal definition of Wasserstein distance, where $p(x)$, $q(y)$ are probability distributions, $\Gamma(p, q)$ is the set of probabilistic distributions on $\mathbb{R} \times \mathbb{R}$ whose marginals represent p and q on the first and second moments, respectively.

4) HEMS Models: The previous techniques measured the quality of the generated data based on its distance from the probability distribution of the real data. In this technique, we measure the quality of the synthetic data by its ability to be used in the training of a Q-learning-based smart home energy management system (HEMS). We implemented the Q-learning model in the MATLAB platform. The ESS fixed charging/discharging power is 4 kW, and the ESS capacity is 16 $kW \cdot h$. The learning rate is 0.8, the discount factor is 0.7, and the initial ϵ value for the ϵ-greedy exploration algorithm is 0.05. The energy trading price follows a fixed pattern as in [5]. We recorded the average results over 10 randomized runs.

C. Distance Metrics Evaluation Results

We used the KL-divergence, MMD, and Wasserstein distance metrics to measure how close is the distribution of synthetic data to the real data. Table I shows these statistical metrics for smart home electrical load consumption, PV production, and EV charging load consumption synthetic data generated by the GMM, GAN, and VAE-GAN generative models. For all metrics, the lower the errors are the better, corresponding to a closer match between the synthetic and real data.

The results allow us to draw several conclusions. First, we find that all three models have relatively close errors for synthetic EV charging datasets, which can be explained by the comparative regularity of EV charging datasets versus PV generation, which is influenced by the weather or load, depending on many personal choices. Another possible explanation is that PV generation and load consumption training data were almost twice the size of EV charging data, suggesting that rich historical training data could have a positive impact on accuracy. Second, we find that for the KL-divergence and Wasserstein distance, our VAE-GAN-based approach almost always provides the best performance. For the MMD, in contrast, the best performance is provided by the GAN approach. This reversal in the relative order of the generator approaches for MMD is likely a result of the choice of mean embedding features. Understanding the exact mechanism of this difference requires future work.

TABLE I
DISTANCE BETWEEN REAL AND SYNTHETIC SMART GRID DATA DISTRIBUTION USING KL-DIVERGENCE, WASSERSTEIN DISTANCE,
AND MMD. BEST RESULTS ARE HIGHLIGHTED IN **BOLD**

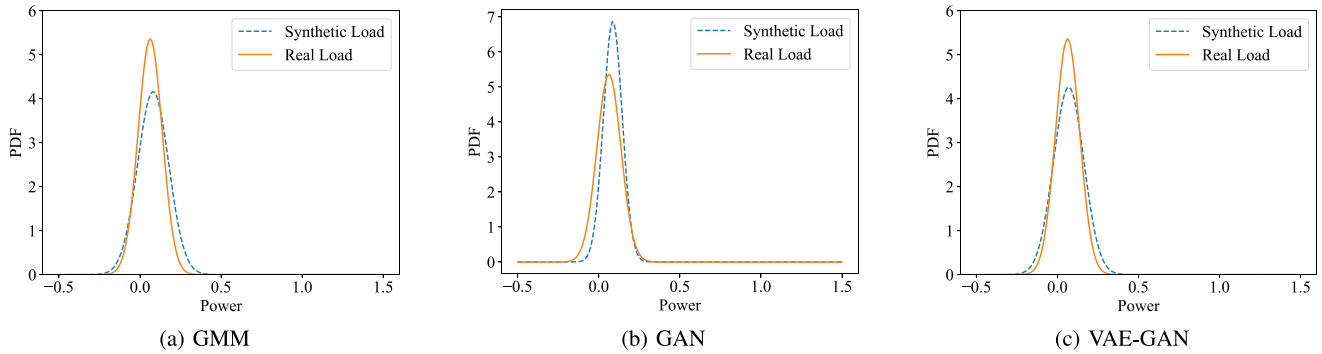| Model	KL divergence			Wasserstein distance			MMD		
	Load	PV	EV	Load	PV	EV	Load	PV	EV
GMM	0.277	1.067	0.127	619.3	1319.3	0.006	0.168	0.074	0.000047
GAN	0.104	0.454	0.104	437.42	1080.5	**0.002**	**0.010**	**0.049**	**0.000028**
VAE-GAN	**0.065**	**0.071**	**0.085**	**295.458**	**648.87**	0.005	0.011	0.186	0.000071

Fig. 6. Electrical load consumption real and synthetic data probability density for (a) GMM, (b) GAN, and (c) VAE-GAN generative models. The orange line shows the real data PDF, the blue line shows the synthetic data PDF.
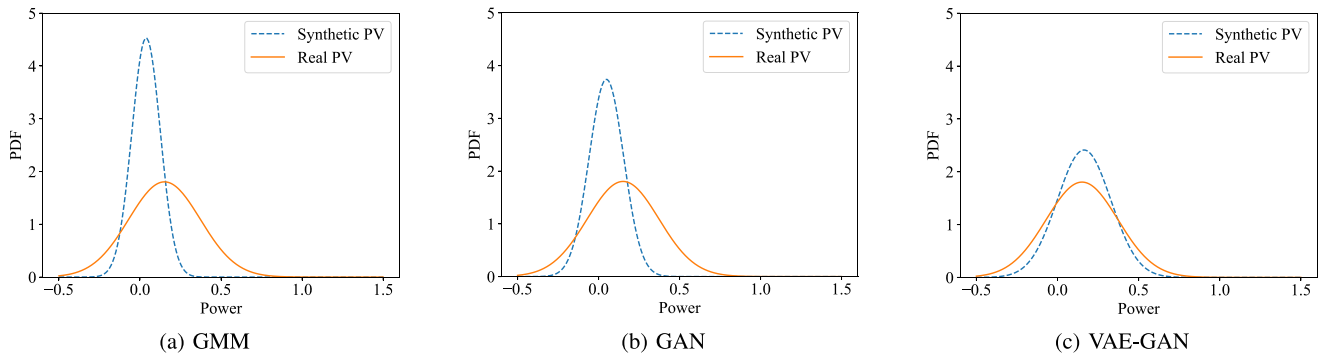

Fig. 7. PV power production real and synthetic data probability density for (a) GMM, (b) GAN, and (c) VAE-GAN generative models. The orange line shows the real data PDF, the blue line shows the synthetic data PDF.
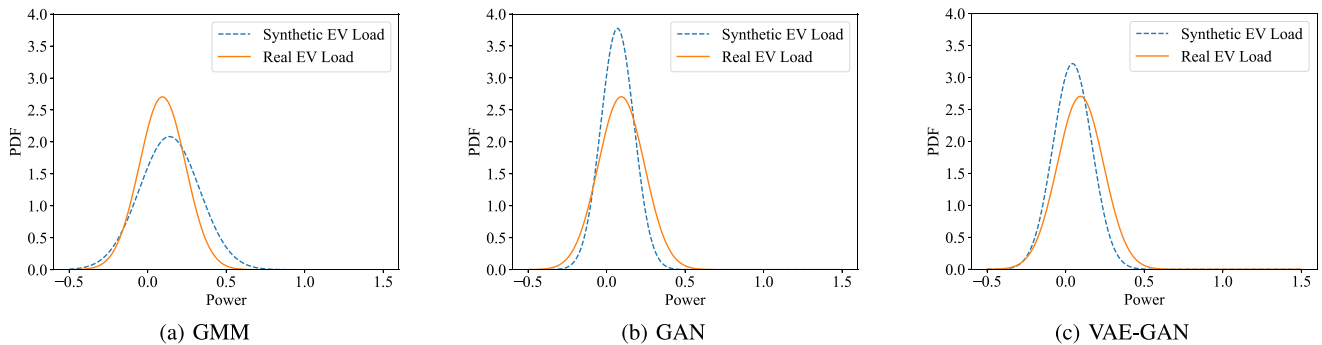

Fig. 8. EV charging load consumption real and synthetic data probability density function for (a) GMM, (b) GAN, and (c) VAE-GAN generative models. The orange line shows the real data PDF, the blue line shows the synthetic data PDF.

While single numerical metrics quantifying the distance between the probability density functions are useful, examining the distributions as a whole provides us with a better understanding of the quality of the generated data. Figs. 6–8 show the Probability Density Functions (PDF) for real and synthetic smart home data.

Fig. 6(d) shows the PDF for normalized real and synthetic electrical load consumption for each generative model. While both the GMM and VAE-GAN distributions are relatively close to the real data, the synthetic data generated by VAE-GAN provides the best matches with the real data distribution in terms of the standard deviation from the mean value, as

(a) Aggregated Load

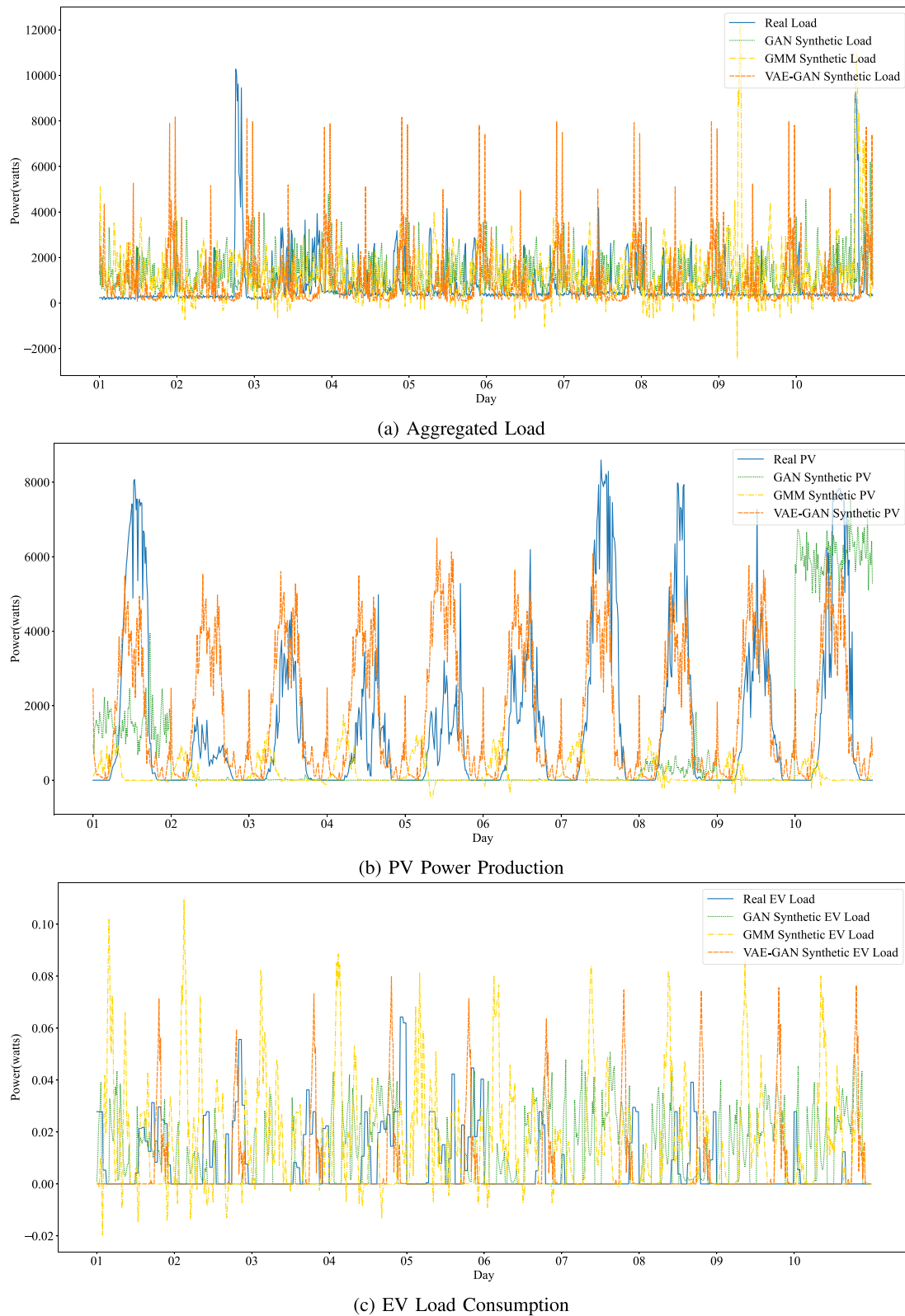

(b) PV Power Production

(c) EV Load Consumption

Fig. 9. A sample of 10 days of synthetic data generated from GAN, GMM, and VAE-GAN models compared with the real test data.

supported by the distance metrics. Fig. 9(a) represents the pattern of real and synthetic electrical load consumption during the test data. It is evident from the close similarity of these patterns that the VAE-GAN network is capable of learning the distribution and patterns of smart home aggregated load

consumption data and producing samples that reflect the same characteristics.

Fig. 7(c) illustrates that the VAE-GAN model performs better compared to two other generative models in generating synthetic PV production data, a conclusion that is supported
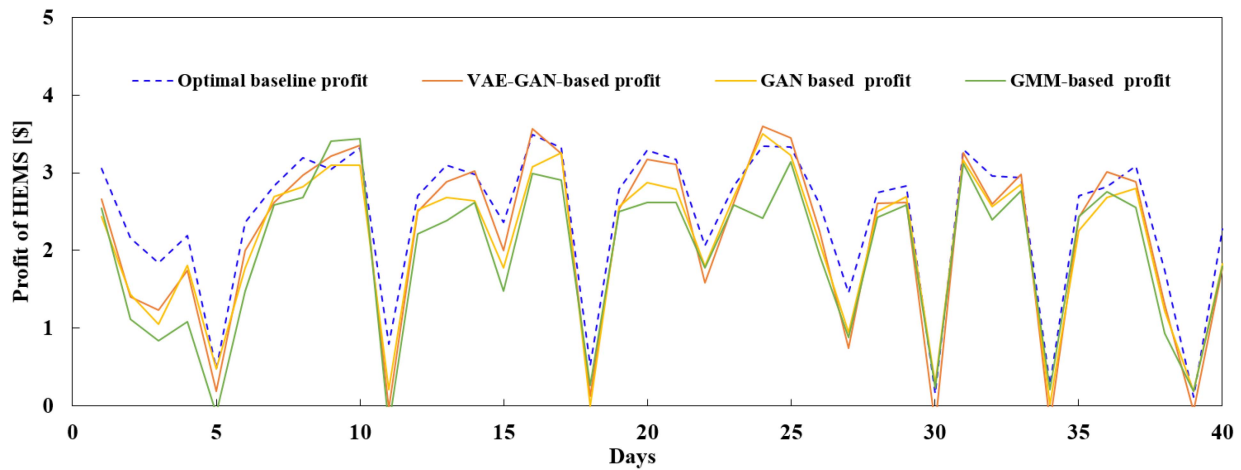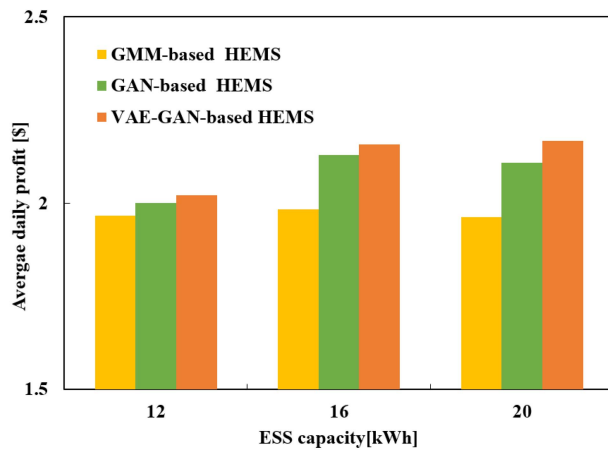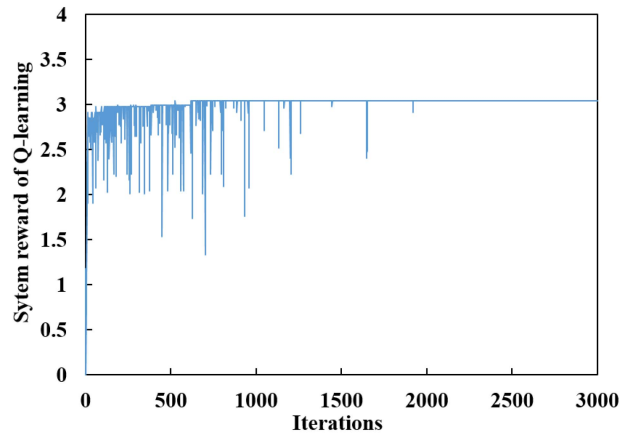
Fig. 10. HEMS test performance comparison in 40 days by using real-world data (16 kWh ESS Capacity) The the total profit of optimal based, VAE-GAN, GAN, and GMM based methods are 97.54$, 86.31$, 85.18$, and 79.36$, respectively.

(a) Average daily HEMS profit comparison under various ESS capacities

(b) Convergence performance of Q-learning based HEMS in one episode

Fig. 11. Smart home HEMS performance analyses.

by Fig. 9(b), where the pattern of synthetic PV production data generated by VAE-GAN is reasonably close to the pattern of real PV production data. We note that although PV production follows sunrise and sunset patterns, it is also highly affected by unpredictable environmental factors, therefore it is not expected for them to be the same equivalent.

Finally, Fig. 9(c) presents the EV charging load consumption real and synthetic data PDF for GMM, GAN, and VAE-GAN generative models. In Fig. 8 it is shown that synthetic EV charging load consumption generated by the GMM model is larger in power range compared to the real EV charging load consumption. On the other hand, the GAN model in Fig. 8(b), generates data centered around the average. Although the VAE-GAN model, Fig. 8(c), generates a slightly smaller power consumption than actual EV charging data, the distribution of load consumption is comparable to that of real data. Accordingly, each generative model shows the same characteristics in the sample EV data presented in Fig. 9(c).

D. HEMS Performance Evaluation Results

In this section, we investigate the performance of the Q-learning-based HEMS. For each type of synthetic data, we trained a corresponding policy, which was then evaluated using real-world data (see Algorithm 1). We compare these policies against an optimal baseline that uses real-world data for both offline training and online operation testing, indicating that the Q-learning agent can best learn the hidden data patterns and prepare for the test. The optimal baseline is expected to achieve the best overall performance by using the same real-world dataset for both training and testing, but it cannot guarantee that the optimal baseline can obtain higher profit every single day.

Fig. 10 shows the 40 days HEMS profit of different data generation methods. While the profit varies with the environmental conditions each day, as expected, the strategy trained on real-world data achieves the highest profit on most days. However, we found that the HEMS trained on the data generated by the proposed VAE-GAN model achieves

a comparable performance. The GAN-based model achieves a somewhat lower performance, while the HEMS trained on the GMM-generated data shows the least profit.

The performance of the HEMS also depends on the capacity of the energy storage system (ESS) of the smart home. The average profit of the differently trained HEMS systems for various ESS capacities is shown in Fig. 11(a). As expected, a larger ESS improves the profit of the HEMS, as the agent has higher flexibility in making decisions. We found that for all ESS values, the VAE-GAN-based HEMS performs best, followed by the GAN and GMM-based methods. The higher the ESS capacity increases the advantage of the VAE-GAN compared to the other methods.

Finally, for practical applications, it is essential to establish that the learning process is stable. Fig. 11(b) shows the reward of the Q-learning process function of the learning iterations. The figure shows that the agent has a stable convergence after the exploration phase.

E. Complexity and Feasibility Analysis

Based on the simulation results, this section will discuss the complexity and feasibility of the proposed scheme, including a) feasibility and scalability, b) implementation complexity, c) management costs, and d) economic benefits.

- Feasibility and scalability: Compared with conventional model-based methods such as Markov chain and simulators, the applied data-driven method has higher scalability and can be easily generalized to other scenarios. In the experiment, note that the only requirement for applying the proposed methods is the dataset, indicating that it can be easily generalized from smart homes to smart builds and communities with proper datasets. By contrast, it is impractical to build very huge simulators or Markov chains to describe all the detailed behaviors of a large area.
- Organizational complexity: The main complexity of the proposed method is the neural network architecture and design, which is a well-known issue for ML-based methods. However, in this work, note that no dedicated models are designed for any household devices or energy consumption behaviors, significantly reducing the organizational complexity.
- Management costs: As above-analyzed, the dataset is the only requirement of the proposed method. When the physical system changes, e.g., adding or removing new devices, our model can be easily updated by feeding new datasets to the neural networks, which means a very low system management cost.
- Economic benefits: The simulation results in Fig. 10 and 11(a) have demonstrated that the proposed method can obtain satisfying profit for the smart home operation. The benefit can be greater when considering larger application scenarios such as smart buildings and communities.

V. Conclusion

In recent years, smart home management increasingly takes advantage of advanced data-demanding machine learning approaches, but the availability of find-grained data sets may prevent the applications. Therefore, synthetic data generation has become a critical enabler for smart home management. In this paper, we present a variational autoencoder GAN (VAE-GAN) approach for the synthetic data generation of smart home datasets. We have shown that the proposed VAE-GAN method can generate high-quality data that better matches the statistical properties of real-world data compared to benchmarks. In addition, we use the synthetic data to train a Q-learning-based smart home energy management system (HEMS), achieving a higher profit than other data-generation methods. The simulations prove that the generated synthetic data can be used for HEMS offline training, and the agent obtains a satisfying real-world online performance without touching the real-world data.

References

[1] M. S. Aliero, K. N. Qureshi, M. F. Pasha, and G. Jeon, "Smart home energy management systems in Internet of Things networks for green cities demands and services," *Environ. Technol. Innov.*, vol. 22, May 2021, Art. no. 101443.

[2] B. Sovacool and D. Rio, "Smart home technologies in Europe: A critical review of concepts, benefits, risks and policies," *Renew. Sustain. Energy Rev.*, vol. 148, pp. 1–22, Mar. 2020.

[3] A. Al-Ali, I. Zualkernan, M. Rashid, R. Gupta, and M. Alikarar, "A smart home energy management system using IoT and big data analytics approach," *IEEE Trans. Consum. Electron.*, vol. 63, no. 4, pp. 426–434, Nov. 2017.

[4] C. Zhang, S. R. Kuppannagari, R. Kannan, and V. K. Prasanna, "Generative adversarial network for synthetic time series data generation in smart grids," in *Proc. IEEE Int. Conf. Commun. Control Comput. Technol. Smart Grids (SmartGridComm)*, 2018, pp. 1–6.

[5] H. Zhou and M. Erol-Kantarci, "Decentralized microgrid energy management: A multi-agent correlated q-learning approach," in *Proc. IEEE Int. Conf. Commun. Control Comput. Technol. Smart Grids (SmartGridComm)*, 2020, pp. 1–7.

[6] S. Ge, J. Li, H. Liu, X. Liu, Y. Wang, and H. Zhou, "Domestic energy consumption modeling per physical characteristics and behavioral factors," *Energy Procedia*, vol. 158, pp. 2512–2517, Feb. 2019.

[7] M. Lahariya, D. Benoit, and C. Develder, "Synthetic data generator for electric vehicle charging sessions: Modeling and evaluation using real-world data," *Energies*, vol. 13, p. 4211, Aug. 2020.

[8] J. López, E. Pouresmaeil, C. Cañizares, K. Bhattacharya, A. Mosaddegh, and B. Solanki, "Smart residential load simulator for energy management in smart grids," *IEEE Trans. Ind. Electron.*, vol. 66, no. 2, pp. 1443–1452, Feb. 2019.

[9] I. Goodfellow et al., "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, Dec. 2014, pp. 1–18.

[10] M. Razghandi, H. Zhou, M. ErolKantarci, and D. Turgut, "Variational autoencoder generative adversarial network for synthetic data generation in smart home," Jan. 2022, *arXiv:2201.07387*.

[11] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANS," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 2226–2234.

[12] M. Razghandi, H. Zhou, M. Erol-Kantarci, and Turgut, "Smart home energy management: Sequence-to-sequence load forecasting and Q-learning," in *Proc. IEEE Globecom*, Dec. 2021, pp. 1–6.

[13] C. Tu, X. He, Z. Shuai, and F. Jiang, "Big data issues in smart grid–A review," *Renew. Sustain. Energy Rev.*, vol. 79, pp. 1099–1107, Nov. 2017.

[14] L. Diao, Y. Sun, Z. Chen, and C. Jiayu, "Modeling energy consumption in residential buildings: A bottom-up analysis based on occupant behavior pattern clustering and stochastic simulation," *Energy Build.*, vol. 147, pp. 47–66, Jul. 2017.

[15] J. Quirós-Tortós, A. Navarro-Espinosa, L. F. Ochoa, and T. Butler, "Statistical representation of EV charging: Real data analysis and applications," in *Proc. Power Syst. Comput. Conf. (PSCC)*, 2018, pp. 1–7.

[16] S. Kababji and P. Srikantha, "A data-driven approach for generating synthetic load patterns and usage habits," *IEEE Trans. Smart Grid*, vol. 11, no. 6, pp. 4984–4995, Nov. 2020.

[17] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, "Model-free renewable scenario generation using generative adversarial networks," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3265–3275, May 2018.

[18] M. Fekri, A. Ghosh, and K. Grolinger, "Generating energy data for machine learning with recurrent generative adversarial networks," *Energies*, vol. 13, p. 130, Dec. 2019.

[19] Z. Wang and T. Hong, "Generating realistic building electrical load profiles through the generative adversarial network (GAN)," *Energy Build.*, vol. 224, Oct. 2020, Art. no. 110299.

[20] R. Buechler, E. Balogun, A. Majumdar, and R. Rajagopal, "EVGen: Adversarial networks for learning electric vehicle charging loads and hidden representations," 2021, *arXiv:2108.03762*.

[21] X. Xu, Y. Jia, Y. Xu, Z. Xu, S. Chai, and C. S. Lai, "A multi-agent reinforcement learning-based data-driven method for home energy management," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3201–3211, Jul. 2020.

[22] W. Li, T. Logenthiran, V.-T. Phan, and W. L. Woo, "Implemented IoT-based self-learning home management system (SHMS) for Singapore," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2212–2219, Jun. 2018.

[23] H. Zhou, A. Aral, I. Brandic, and M. Erol-Kantarci, "Multi-agent Bayesian deep reinforcement learning for microgrid energy management under communication failures," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 11685–11698, Jul. 2022.

[24] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. 33rd Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 1558–1566. [Online]. Available: https://proceedings.mlr.press/v48/larsen16.html

[25] J. Yoon, D. Jarrett, and M. Van der Schaar, "Time-series generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 5509–5519.

[26] A. Oord et al., "WaveNet: A generative model for raw audio," Sep. 2016, *arXiv:1609.03499*.

[27] P. Huber, M. Ott, M. Friedli, A. Rumsch, and A. Paice, "Residential power traces for five houses: The iHomeLab RAPT dataset," *Data*, vol. 5, no. 1, p. 17, Feb. 2020.

[28] Å. L. Sørensen, K. B. Lindberg, I. Sartori, and I. Andresen, "Residential electric vehicle charging datasets from apartment buildings," *Data Brief*, vol. 36, Jun. 2021, Art. no. 107105.

Hao Zhou (Graduate Student Member, IEEE) received the B.Eng. degree from the Huazhong University of Science and Technology, China, in 2016, and the M.Eng. degree from Tianjin University, China, in 2019. He is currently pursuing the Ph.D. degree with the University of Ottawa. His research interests include microgrid energy trading, 5G network slicing and mmWave, and reconfigurable intelligent surfaces. He is devoted to applying machine-learning techniques for smart grid and wireless network applications. He won the Best Paper Reward at 2023 IEEE ICC conference.

Melike Erol-Kantarci (Senior Member, IEEE) is the Canada Research Chair of AI-Enabled Next-Generation Wireless Networks and an Associate Professor with the School of Electrical Engineering and Computer Science, University of Ottawa. She is the Founding Director of the Networked Systems and Communications Research (NETCORE) Laboratory. She is the co-editor of three books on smart grids, smart cities, and intelligent transportation. She has over 200 peer-reviewed publications. She has delivered over 70 keynotes, plenary talks, and tutorials around the globe. Her main research interests are AI-enabled wireless networks, 5G and 6G wireless communications, smart grid, and Internet of Things. She has received numerous awards and recognitions. She is on the editorial board of the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE INTERNET OF THINGS JOURNAL, IEEE COMMUNICATIONS LETTERS, IEEE NETWORKING LETTERS, *IEEE Vehicular Technology Magazine*, and IEEE ACCESS. She has acted as the general chair and the technical program chair for many international conferences and workshops. She is an IEEE ComSoc Distinguished Lecturer and an ACM Senior Member.

Mina Razghandi (Graduate Student Member, IEEE) received the Ph.D. degree in computer science from the University of Central Florida in 2022. She has dedicated her academic pursuit to exploring the intersection of advanced AI technologies and sustainable, and smart infrastructure. Her Ph.D. research was primarily focused on the application of deep-learning techniques to enhance the capabilities and efficiency of smart homes and smart grids. This intellectual journey has instilled in her a keen interest in AI, Internet of Things, and energy microgrids. She is continually driven by the desire to leverage these cutting-edge tools and methodologies to create a more connected and energy-efficient world.

Damla Turgut (Senior Member, IEEE) is a Professor and the Chair of the Computer Science Department, University of Central Florida. Her research interests include wireless ad hoc, sensor, underwater and vehicular networks, cloud computing, smart cities, smart grid, IoT-enabled healthcare and augmented reality, as well as considerations of privacy in the Internet of Things. She is also interested in applying big data techniques for improving STEM education for women and minorities. Her recent honors and awards include the NCWIT 2021 Mentoring Award for Undergraduate Research Award, the UCF Research and Teaching Incentive Awards, the UCF Women of Distinction Award, and the University Excellence Award in Professional Service. She serves on several editorial boards and program committees of prestigious ACM and IEEE journals and conferences. She is an IEEE ComSoc Distinguished Lecturer, the Chair of IEEE Technical Committee of Social Networks, an ACM Senior Member, and the Chair of the IEEE Technical Community on Computer Communications.