Deep Reinforcement Learning with Online Data Augmentation to Improve Sample Efficiency for Intelligent HVAC Control

Kuldeep Kurte
Oak Ridge National Laboratory, USA
kurtekr@ornl.gov

Kadir Amasyali
Oak Ridge National Laboratory, USA
amasyalik@ornl.gov

Jeffrey Munk
National Renewable Energy Laboratory, USA
jeff.munk@nrel.gov

Helia Zandi
Oak Ridge National Laboratory, USA
zandih@ornl.gov

ABSTRACT

Deep Reinforcement Learning (DRL) has started showing success in real-world applications such as building energy optimization. Much of the research in this space utilized simulated environments to train RL-agent in an offline mode. Very few research have used DRL-based control in real-world systems due to two main reasons: 1) sample efficiency challenge—DRL approaches need to perform a lot of interactions with the environment to collect sufficient experiences to learn from, which is difficult in real systems, and 2) comfort or safety related constraints—user's comfort must never or at least rarely be violated. In this work, we propose a novel deep **R**einforcement **L**earning framework with online **D**ata **A**ugmentation (RLDA) to address the sample efficiency challenge of real-world RL. We used a time series Generative Adversarial Network (TimeGAN) architecture as a data generator. We further evaluated the proposed RLDA framework using a case study of an intelligent HVAC control. With a ≈28% improvement in the sample efficiency, RLDA framework lays the way towards increased adoption of DRL-based intelligent control in real-world building energy management systems.

CCS CONCEPTS

• **Computing methodologies** → **Reinforcement learning**; • **Hardware** → **Power and energy**.

KEYWORDS

deep reinforcement learning, data augmentation, intelligent HVAC control, demand response, building energy

ACM Reference Format:
Kuldeep Kurte, Kadir Amasyali, Jeffrey Munk, and Helia Zandi. 2022. Deep Reinforcement Learning with Online Data Augmentation to Improve Sample Efficiency for Intelligent HVAC Control. In *Third ACM SIGEnergy Workshop on Reinforcement Learning for Energy Management in Buildings & Cities (RLEM) (RLEM '22), November 9–10, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3563357.3566168

1 INTRODUCTION

Reinforcement Learning (RL) is a branch of Machine Learning (ML) that is concerned with learning optimal policies through reward maximization [14]. It is a trial and error method, in which RL-agent interacts with the environment through actions, collects data through observations, and improves the policy. The contemporary Deep Learning (DL) based RL algorithms such as Deep Q-Network (DQN), Deep Deterministic Policy Gradient (DDPG), Proximal Policy Optimization (PPO), etc. have demonstrated to be effective in simulated environments where RL-agent can perform a large number of interactions with the environment before it learns the optimal policy. In real-world applications such as building energy management, RL-agent may not have an opportunity to perform a large number of interactions with the environment. This is called the sample efficiency challenges of real-world RL. Additionally, the comfort and safety-related constraints must never or rarely be violated. This means RL-agent needs to be very cautious during the exploration while performing random actions in real-world settings. Due to these reasons: 1) sample efficiency challenges, and 2) comfort and safety-related constraints, the uptake of DRL-based controls in real-world problems is still limited.

A similar trend has been observed in the research of DRL-based building energy optimization. For instance, the research in [16], [4], [10], and [6] used simulated environments of buildings and HVAC systems to train RL-agent to obtain optimal HVAC control policy. In similar research, [2], [1] used simulation environments of buildings and water heaters for training RL-agent to control water heater in a cost-efficient way. The more recent interesting approaches in this direction are RL-based energy-efficient data center [12], energy-efficient personalized thermal comfort in office buildings [18], and multi-agent multi-objective RL for controlling residential appliance scheduling [11]. These approaches used either energy plus or a custom build simulated environments for training RL-agent.

In the United States, more than 50% of buildings' energy use is attributed to HVAC [7]. Therefore, a significant energy saving can be achieved by intelligently controlling the HVAC system. To achieve this, wide-scale adoption of such intelligent systems in real-world is necessary. However, very few of the above works demonstrated RL's capability in real-world energy management applications. Most of these algorithms fall under the model-free RL category that does not assume any prior knowledge of the

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).
RLEM '22, November 9–10, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9890-9/22/11…$15.00
https://doi.org/10.1145/3563357.3566168

environment and learn optimal policy only by interacting with the environment. Hence such approaches need a large number of interactions with the environment to obtain a good policy. This is difficult in real-world settings due to the aforementioned challenges of the real-world setting.

In this work, we proposed a DRL framework with online Data Augmentation (RLDA) capability to address the sample efficiency challenges of real-world building energy management, specifically HVAC control. DRL setup uses a replay memory that stores the agent's experiences in terms of a set of tuples comprising of current state (s_t), action (a_t), reward (r_{t+1}), and next state (s_{t+1}), i.e. $<s_t, a_t, r_{t+1}, s_{t+1}>$. A set of tuples stored consecutively in the replay memory represents a trajectory that the agent has traversed through the environment. We use a time series Generative Adversarial Network (TimeGAN, [17]) architecture to generate synthetic trajectories and use them to augment the replay memory with synthetic tuples. In the current RLDA setup, we invoke TimeGAN on day 10 which then uses all the tuples from the replay memory until day 10 and generates the synthetic tuples. Further, DRL training will use synthetic tuples along with real experiences. We compared the average cumulative energy cost of 50 executions of operating HVAC using RLDA and conventional DRL with the cumulative energy cost of a fixed setpoint baseline. The average cumulative energy cost of conventional DRL crossed baseline's cumulative energy cost on the 39th day whereas the proposed RLDA crossed the baseline's cumulative energy cost on the 28th day. This shows ≈28% improvement in the sample efficiency. This result provides experimental evidence that performing data augmentation in an online fashion during DRL training is a potential way to address the sample efficiency challenges of the real-world RL.

Contributions: The contributions of this paper are: 1) We proposed a DRL framework with a TimeGAN-based online data augmentation module to address the sample-efficiency challenges of the real-world RL. 2) We provide experimental evidence that the proposed RLDA framework improves the sample-efficiency of DRL training.

The rest of this paper is structured as follows: Section 2 describes the proposed RLDA framework. It provides the details of TimeGAN architecture and how we utilized it to generate synthetic experiences. Next, Section 3 discusses the experimental setup and results showing the sample-efficiency improvement achieved by the RLDA framework. Finally, Section 4 concludes the paper and summarizes the future work that needs to be done in this direction.

2 DEEP REINFORCEMENT LEARNING WITH ONLINE DATA AUGMENTATION FRAMEWORK

2.1 RL preliminaries

RL is a powerful paradigm for solving control optimization problems such as HVAC control of building energy management. RL-agent interacts with the environment through actions and state. Environment evolves to the next state (s_{t+1}) in the response to the action. The objective of RL algorithms is to obtain an optimal policy (π^*) that dictates what action (a_t) to take in a current state s_t. For instance, the setpoint to set for a given current indoor condition. The optimal policy maximizes the cumulative discounted reward, G_t (refer Eq. 1).

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots = \sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k} \tag{1}$$

RL-agent keeps a record of state-action value called Q-value, $Q(s_t, a_t)$. It dictates how good it is to take an action a in a state s. Q-learning is a model-free RL algorithm that learns optimal Q-value by iteratively executing the Q-learning equation (refer Eq. 2).

$$\begin{aligned} Q_{t+1}(s_t, a_t) &\leftarrow Q_t(s_t, a_t) \\ &+ \eta(r_{t+1} + \gamma \max_{a_{t+1}} Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)) \end{aligned} \tag{2}$$

2.2 Deep-Q-Network

The Q-learning algorithm works well for discrete states and actions, where maintaining a Q-table is easy. In the case where state or actions are continuous, Q-table can not be maintained. DRL algorithms such as Deep-Q-network (DQN) address this problem by using neural networks to approximate Q-table [13]. Figure 1 shows the architecture of DQN. It consists of two networks: 1) evaluation network—approximates Q-value of the current state and action, and 2) target network—approximates the Q-value of the next state and action. DQN-agent interacts with the environment and stores the experiences of tuples in a replay memory. During training, a batch of tuples is randomly drawn from the replay memory and a loss is calculated using the Q-value of the current state and action, and the target Q-value. This loss is used to update the weights of the evaluation network. The weights of the evaluation network are copied to the target network periodically (say at every ΔT training steps).

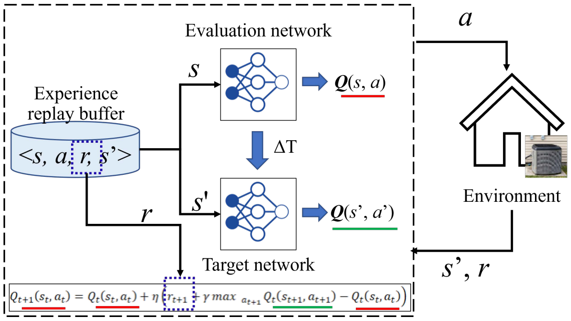
a
Evaluation network
s
Q(s, a)
Experience replay buffer
<s, a, r, s'>
ΔT
Environment
r
s'
Q(s', a')
Target network
s', r
$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \eta(r_{t+1} + \gamma \max_{a_{t+1}} Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t))$

Figure 1: DQN-architecture.

2.3 TimeGAN

TimeGAN [17] is a variant of Generative Adversarial Network (GAN [8] for time-series data generation task. It was shown to demonstrate excellent performance on a time-series data augmentation task by Jinsung Yoon et al., in 2019 [17]. In building energy domain, GAN was used to generate energy use and load characteristics for multiple buildings [3], to create uncertainty-infused synthetic profiles of building performance [9], and to generating realistic building electrical load profiles [15]. Figure 2 shows TimeGAN architecture which includes autoencoder, GAN, and supervisor architectures. Autoencoder learns the temporal dynamics in the hidden dimensions. The generated features in the hidden dimension by both encoder ($h_{1:T}$) and generator ($\tilde{h}_{1:T}$) are used in a supervised learning fashion to train the generator to capture the temporal dynamics of the real data, i.e. $p(X_t|X_{1:t-1})$. In this way, the generative model of TimeGAN learns the temporal correlations as well as relationships among features. The generator and discriminator architectures of GAN compete with each other. On one hand, the generator aims to generate synthetic time-series samples that can achieve real feature distributions. On the other hand, the discriminator aims to distinguish whether a given time-series sample is synthetic or not. Zhang et al., 2022 [19] demonstrated the use of TimeGAN for data augmentation for improving heating load prediction of the heating substation.

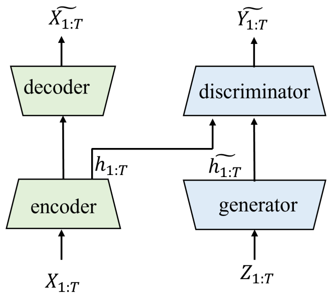

Figure 2: TimeGAN architecture.

2.4 RLDA framework

The experience replay memory stores tuples $<S_t, A_t, R_{t+1}, S_{t+1}>$ where the consecutive tuples are correlated. It represents the trajectory traversed by an RL-agent through the environment. Since these experiences are stored sequentially, they can be treated as a time-series of tuples. In a real-world setting, we want RL-agent to learn an optimal policy with a limited number of samples. In this work, we propose an online data-augmentation approach to generate synthetic experiences during training. We use TimeGAN to generate synthetic trajectories of experiences. During training, the data augmentation process is invoked after Δt_a time period, e.g. 10 days. The data augmentation will use the tuples of experiences collected until Δt_a time from replay memory. Further, a set of partial trajectories of tuples of a particular sequence length (say 16) are produced to train TimeGAN. TimeGAN uses these partial trajectories and learns to generate similar partial trajectories of tuples of the same sequence length. After this process, the synthetic tuples from these generated trajectories are used along with the real tuples during DQN training.

3 EXPERIMENTAL RESULTS

3.1 Simulation setup

We evaluated the performance of the proposed RLDA framework in a simulated environment. We trained a DQN model using weather from TMY3 Knoxville, TN, USA data for July and August months in a cooling mode. We used a grey-box model that simulates the thermal response of a 2-story unoccupied research house located in Knoxville, TN, USA. A particle swarm optimization algorithm was used to train the simulation model parameters. More details about simulation model development are described in [5]. The simulation model training and validation results are presented in [10].

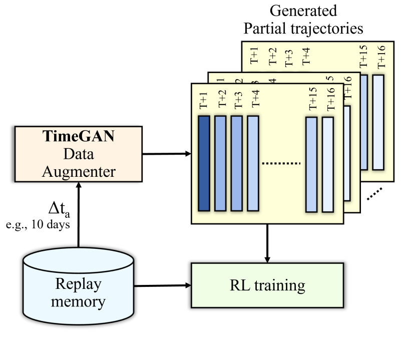

Figure 3: Proposed RLDA framework.

3.2 DQN training

We formulated the HVAC control problem as a Markov Decision Process (MDP) which involves defining state and action spaces, and reward function.

State space: We used features such as time of the day, indoor temperatures of both the zones, outdoor temperature, and Time Of Use (TOU) electricity price as a state. In the TOU price, the peak price is \$0.25 during 2:00–8:00 p.m. and the off-peak price is \$0.05.

Action space: Actions consist of a set of setpoints, one for each zone, i.e. 70°F, 74°F. It indirectly generates AC's ON/OFF behavior based on the current indoor temperature.

Reward function: The reward function used in this work is $RF = -Energy\_cost$.

3.3 Real-world RL setting and constraints

RL algorithms use multiple episodes of training when trained in a simulated environment. However, in real-world RL problems, it is not possible to perform multiple episodes. In fact, RL-agent is deployed in the environment, continues to interact with the environment, and learns the optimal policy. Moreover, in a real-world setting, RL-agent may get very little chance to explore the environment. To mimic this real-world setting in simulation, we put the following additional constraints:

- DQN training is carried out only for one episode and we performed 50 independent repetitions of the training to captures the effects of random actions and random initialization of agent.
- RL-agent is allowed to explore for a short period at the beginning of the training period. This is controlled by a parameter called Exploration Rate Factor (ERF). For example, ERF=0.01 allows RL-agent to take random actions during the first 1% of the total training iterations.
- We invoke the data augmentation on day 10.

Various DQN and TimeGAN parameters in the RLDA framework and their values used during training are shown in Table 1.

Table 1: DQN and TimeGAN parameters used

Parameter	Value
Episodes	1
Simulation step (Δt_s)	1 min
Control step (Δt_c)	15 min
Learning rate	0.01
Optimizer	Adam
Reward decay (γ)	0.9
ϵ-greedy value	0.99
Target replacement iterations	200
Initial steps	1440
Batch size	64
Experience replay memory size	100,000
Exploration Rate Factor (ERF)	0.01
[Lower Threshold (LT), Upper Threshold (UT)]	[72°F, 74°F]
Data augmentation day (Δt_a)	day 10
TimeGAN sequence length	16
TimeGAN training iterations	10000

3.4 Evaluation criteria

We used a fixed setpoint baseline of 74°F. Further, we ran RL and RLDA for 62 days of July and August months. Here we refer DQN algorithm without data augmentation as RL. We repeated this training 50 times. During training, we recorded the cumulative energy cost to capture the learning progress. These cumulative energy costs from 50 repetitions, were used to calculate average cumulative energy cost. We then compared the day when average cumulative costs of both RL and RLDA have crossed the baseline. This was used to calculate the improvement in sample-efficiency, i.e. reduction in the data used observed by RLDA over RL.

3.5 Results

Figure 4 shows the comparison of the average cumulative energy cost of 50 repetitions of RLDA and RL training with the baseline's cumulative energy cost. The ribbon plot shows the range of 25% quantile and 75% quantile of cumulative energy cost of 50 repetitions. Figure 4(b) shows the zoomed-in plot of the blue rectangle in Figure 4(a). We observed that RLDA's average cumulative energy cost crossed the baseline on day 28, and RL's (i.e. without data augmentation) average cumulative energy cost crossed the baseline on day 39. This provides us with experimental evidence that the proposed online data augmentation module clearly provided benefits and allowed RL-agent to learn faster, i.e. ≈11 days earlier than the RL without any data augmentation. This shows ≈28% improvement in the sample-efficiency.

3.6 Limitation of the current work

We identified the following limitations of the present work:

- The synthetic trajectories of tuples may contain bad tuples which need to be filtered, which is missing in the current RLDA framework. In the future, we plan to integrate a module to filter the bad tuples based on some quantitative criteria.
- The TimeGAN's training is computationally expensive. Currently, with ≈960 tuples (10 days of tuples), the sequence length of 16, and 10,000 training iterations, it takes an average of ≈50 minutes for training. This can be accelerated in the future using GPUs and distributed training.

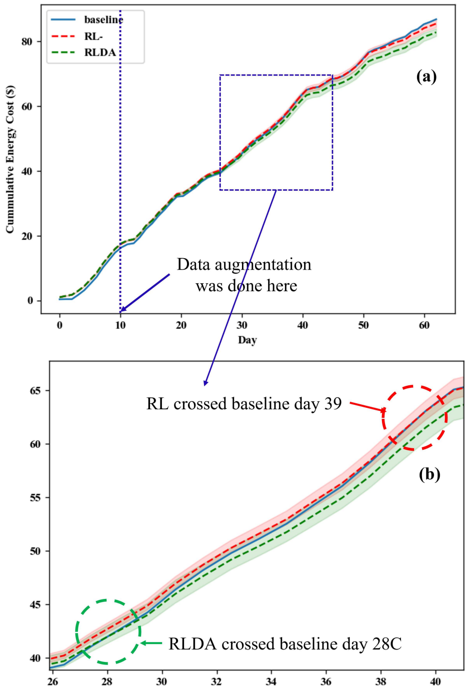

Figure 4: Cumulative cost comparison of RLDA and RL.

- Here, day 10 was chosen arbitrarily to invoke data augmentation. This needs to be automatically computed on-the-fly based on RL-agent's interactions with the environment.
- The current work uses limited seasonal variation, i.e. only one summer season was consider. To obtain more robust results, we will use a year's worth of simulation.

4 CONCLUSION AND FUTURE WORK

In this paper, we presented a Deep Reinforcement Learning framework with online data augmentation (RLDA) to address the sample-efficiency challenge of the real-world RL. We used HVAC control as our use case. For data augmentation, we used the TimeGAN-based time-series generator. The preliminary results are very promising and showed ≈28% of improvement in sample-efficiency, which shows the applicability of the proposed RLDA framework. More research is needed in this direction. In the future, we plan to implement a module that filters the bad tuples generated by TimeGAN based on some quantitative criteria. Also, we will accelerate TimeGAN's computation through the use of GPUs. Moreover, in this work, day 10 was chosen arbitrarily to invoke data augmentation. More research is required in this direction to identify the perfect time to invoke data augmentation. Also, instead of invoking data augmentation only once throughout the training, it can be invoked multiple times periodically during training. Also, we will perform an year worth of simulation to evaluate TimeGAN's response to the seasonal variation.

ACKNOWLEDGMENTS

This research is sponsored by the Artificial Intelligence Initiative as part of the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the US Department of Energy under contract DE-AC05-00OR22725.

REFERENCES

[1] Kadir Amasyali, Kuldeep Kurte, Helia Zandi, Jeffrey Munk, Olivera Kotevska, and Robert Smith. 2021. Double Deep Q-Networks for Optimizing Electricity Cost of a Water Heater. In *2021 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. 1–5. https://doi.org/10.1109/ISGT49243.2021.9372205

[2] Kadir Amasyali, Jeffrey Munk, Kuldeep Kurte, Teja Kuruganti, and Helia Zandi. 2021. Deep Reinforcement Learning for Autonomous Water Heater Control. *Buildings* 11, 11 (2021). https://doi.org/10.3390/buildings11110548

[3] Gaby Baasch, Guillaume Rousseau, and Ralph Evins. 2021. A Conditional Generative adversarial Network for energy use in multiple buildings using scarce data. *Energy and AI* 5 (2021), 100087.

[4] Enda Barrett and Stephen Linder. 2015. Autonomous HVAC Control, A Reinforcement Learning Approach. In *Machine Learning and Knowledge Discovery in Databases*, Albert Bifet, Michael May, Bianca Zadrozny, Ricard Gavalda, Dino Pedreschi, Francesco Bonchi, Jaime Cardoso, and Myra Spiliopoulou (Eds.). Springer International Publishing, Cham, 3–19.

[5] Borui Cui, Jeffrey Munk, Roderick Jackson, David Fugate, and Michael Starke. 2017. Building thermal model development of typical house in US for virtual storage control of aggregated building loads based on limited available information. In *Proceedings of ECOS*.

[6] Yan Du, Helia Zandi, Olivera Kotevska, Kuldeep Kurte, Jeffery Munk, Kadir Amasyali, Evan Mckee, and Fangxing Li. 2021. Intelligent multi-zone residential HVAC control strategy based on deep reinforcement learning. *Applied Energy* 281 (2021), 116117. https://doi.org/10.1016/j.apenergy.2020.116117

[7] EIA. 2015. Use of energy explained Energy use in homes (U.S. Energy Information Administration (EIA), 2015). https://www.eia.gov/energyexplained/use-of-energy/homes.php

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[9] Fazel Khayatian, Zoltán Nagy, and Andrew Bollinger. 2021. Using generative adversarial networks to evaluate robustness of reinforcement learning agents against uncertainties. *Energy and Buildings* 251 (2021), 111334.

[10] Kuldeep Kurte, Jeffrey Munk, Olivera Kotevska, Kadir Amasyali, Robert Smith, Evan McKee, Yan Du, Borui Cui, Teja Kuruganti, and Helia Zandi. 2020. Evaluating the Adaptability of Reinforcement Learning Based HVAC Control for Residential Houses. *Sustainability* 12, 18 (2020). https://doi.org/10.3390/su12187727

[11] Junlin Lu, Patrick Mannion, and Karl Mason. 2022. A multi-objective multi-agent deep reinforcement learning approach to residential appliance scheduling. *IET Smart Grid* 5, 4 (2022), 260–280. https://doi.org/10.1049/stg2.12068 arXiv:https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/stg2.12068

[12] Muhammad Haiqal Bin Mahbod, Chin Boon Chng, Poh Seng Lee, and Chee Kong Chui. 2022. Energy saving evaluation of an energy efficient data center using a model-free reinforcement learning approach. *Applied Energy* 322 (2022), 119392. https://doi.org/10.1016/j.apenergy.2022.119392

[13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[14] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[15] Zhe Wang and Tianzhen Hong. 2020. Generating realistic building electrical load profiles through the Generative Adversarial Network (GAN). *Energy and Buildings* 224 (2020), 110299.

[16] Tianshu Wei, Yanzhi Wang, and Qi Zhu. 2017. Deep reinforcement learning for building HVAC control. In *Proceedings of the 54th annual design automation conference 2017*. 1–6.

[17] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. 2019. Time-series generative adversarial networks. *Advances in neural information processing systems* 32 (2019).

[18] Liang Yu, Zhanbo Xu, Tengfei Zhang, Xiaohong Guan, and Dong Yue. 2022. Energy-efficient personalized thermal comfort control in office buildings based on multi-agent deep reinforcement learning. *Building and Environment* 223 (2022), 109458. https://doi.org/10.1016/j.buildenv.2022.109458

[19] Yunfei Zhang, Zhihua Zhou, Junwei Liu, and Jianjuan Yuan. 2022. Data augmentation for improving heating load prediction of heating substation based on TimeGAN. *Energy* (2022), 124919.