Implementation of Deep Reinforcement Learning for Demand Response in Distribution Networks

Bahrami, S., Chen, Y. C., and Wong, V. W. S., "Deep reinforcement learning for demand response in distribution networks," IEEE Transactions on Smart Grid, vol. 12, no. 2, pp. 1496–1506, Mar. 2021.

INTRODUCTION

The Electric Power and Energy Systems Group research lab at the University of British Columbia (UBC) published this paper, Deep Reinforcement Learning for Demand Response (DR) in Distribution Networks. Dr. Shahab Bahrami (a Postdoctoral Research Fellow), Dr. Yu Christine Chen (an Assistant Professor) and Dr. Vincent W. S. Wong (a Professor & IEEE Fellow) are the paper authors. This paper was published in the IEEE Transactions on Smart Grids, which has an impact factor greater than 8, for the March 2021 edition.

This paper addresses one of the many problems distribution networks are facing with changing electricity grids due to variable renewable energy and decarbonisation through electrification. Electricity grids have been built over decades with significant capital expenditures to provide safe and reliable service to customers. These grids were designed to operate with centralized generation stations however with the shift to decarbonize all sectors, more variable renewable energy is being deployed in distribution networks. Sectors like transport are using electrification as a means to reduce carbon footprint, which is further changing distribution network operation.

Instead of investing significant capital to completely redesign the system, one of the well-recognized solutions in the electricity industry is to implement DR which reduces and/or shift loads (especially during peak times when grids are especially constrained). By controlling some loads strategically, the system could observe less loading during peak times as shown by Figure 1. While this concept provides the grid with relief, customers also expect have a threshold for changing their electricity consumption patterns. This introduces complexity to the problem as each unique customer will have different standards for which loads can be controlled and when.

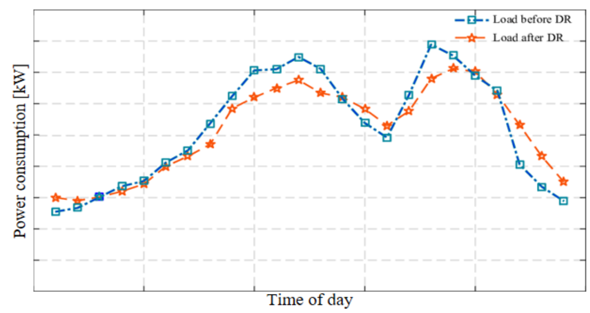

Figure 1: Load curve before and after DR [1].

Consider a residential home shown in Figure 2. By nature, some of these loads will not/cannot be controlled as customers would always want access to these loads without any issues (e.g. lighting as needed for safety, or clothes dryer to be used right after clothes have been washed). However, there are also controllable loads that the user may be comfortable adjusting or shifting, especially if incentivized (e.g. as long as an electric vehicle (EV) battery is charged for the next trip, hot water is available when needed, and dishes are cleaned when needed). The controllable loads can be adapted according to household preferences and yield reduced electricity bills. However, customer data including household preferences is sensitive information that needs to be protected. Furthermore, there is uncertainty in the electricity price and how much customers will be compensated because demand from other households could be reduced that would meet system requirements thereby making a customer's DR unnecessary.

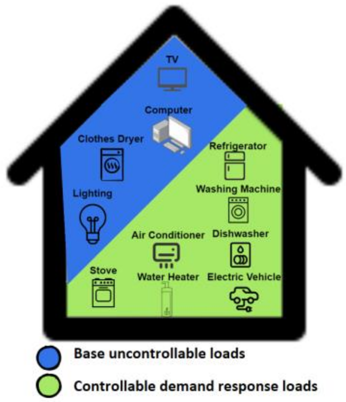

Figure 2: Sample residential home with controllable and uncontrollable loads.

This paper's solution addresses customer data privacy, in addition to uncertainty in price and demand on the distribution network while deploying DR in a residential network of 32 households.[1] This is important to demonstrate how distribution networks can deploy DR while still meeting customer needs to respond to system strain. These simulated systems strains could be due to variability patterns in renewable energy or increased demands due to electrification. This paper optimizes scheduling controllable loads while considering customer discomfort for DR and reducing customer electricity bills by implementing the actor-critic model, and the customer electricity data used to train this model will be protected through federated learning, a decentralized learning technique.

PROBLEM SOLUTION

The pseudo code used to solve this problem shown in Figure 3 and will be the structure to explain this section. The problem solution involves a load aggregator acting on behalf of the distribution network and coordinating how much DR each household should deploy for each time step. Each household runs its own deep neural network (DNN) actor critic model to act in its best interest but there is data sent between each household and the aggregator in the problem formulation using federated learning to protect customer privacy. This allows for reduced latency, reduced issues in connectivity and less power consumed to compute each model at the system level given that a central server is not responsible for all the training.

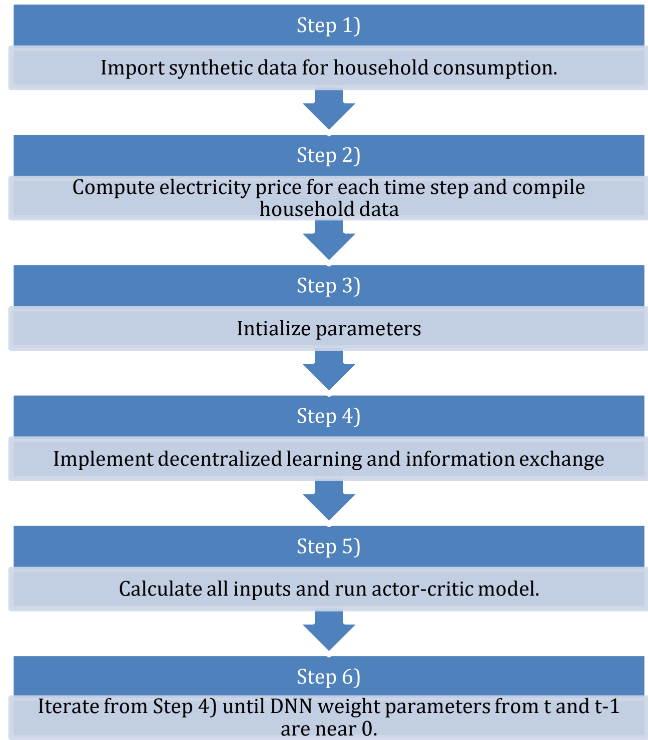

Figure 3: Psuedo code for implementation.

The input data for this paper involved reproducing results with a Markov Decision Process implementation breaking down and scheduling each of the controllable appliances into interruptible and non-interruptible schedules for each hour of the day [2]. Given the complexity of this implementation, simplifications were made to produce the synthetic data imported in Step 1. This synthetic data was the output of the researched raw data as explained in the ANNEX. The synthetic data is hourly consumption data for baseloads as well as the minimum and maximum controllable load for each household. The baseload ($P_{base,n,t}$)

[1] This paper also considers distribution network power flow to ensure solution is within the feasible action space, but this is not included in the implementation.

is the non-controllable load that will be consumed regardless. The maximum controllable load ($P^{max}{}_{DR,n,t}$) is the typical consumption of controllable loads (i.e. customers will experience the most amount of comfort). The minimum controllable load ($P^{min}{}_{DR,n,t}$) indicates a shift that customers would be willing to endure since they would receive a cost benefit by reducing consumption. Electricity prices (ρ_t) are computed based on the time of day and used to create the household data state, $s_{n,t}$.

$$s_{n,t} = \left(\rho_t, P_{base,n,t}, P^{min}{}_{DR,n,t}, P^{max}{}_{DR,n,t}\right) \quad (1)$$

$n: household\ number$
$t: time\ step$
$\rho_t: electricity\ price\ \left[\frac{\$}{kW}\right]$
$P_{base,n,t}: baseload\ [kW]$
$P^{min}{}_{DR,n,t}: minimum\ controllable\ load\ [kW]$
$P^{max}{}_{DR,n,t}: maximum\ controllable\ load\ [kW]$

Once parameters have been initialized, decentralized learning is a unique aspect through implementing federated learning. Federated learning is used to protect customer household data [3]. This solution only implements forward propagation to ensure user data is protected. The set of households are divided into (i) groups for secure aggregation. The load aggregator generates and sends random household data ($X_{j,t}$) that is unique to each group. The household data state is aggregated in each group and merged with the random data sent from the load aggregator for the actor ($\lambda_{j,t}$) and critic models ($\gamma_{j,t}$). The load aggregator then removes the random data so it is left with the real anonymized system data. Next, the bias term for the actor-critic DNN is calculated as the aggregated consumption on the system for the actor ($\bar{\lambda}_t$) and critic ($\bar{\gamma}_t$).

$$\lambda_{j,t} = \sum_{n=j+(x-1)(j-1)}^{ij} s_{n,t} + X_{j,t} \quad (2)$$

$$\gamma_{j,t} = \sum_{n=j+(i-1)(j-1)}^{xj} s_{n,t} + X_{j,t} \quad (3)$$

$j: group\ number$
$i: number\ of\ groups$
$\lambda_{j,t}: aggregated\ state\ for\ actor$
$\gamma_{j,n,t}: aggregated\ state\ for\ critic$
$X_{j,t}: random\ data$

$$\bar{\lambda}_t = \sum \gamma_{:,t} \quad (4)$$

$$\bar{\gamma}_t = \sum \gamma_{:,t} \quad (5)$$

$\bar{\lambda}_t: system\ aggregated\ bias\ term\ for\ actor$
$\bar{\gamma}_t: system\ aggregated\ bias\ term\ for\ critic$

The other unique feature of this paper is the implementation of the actor-critic model for reinforcement learning. Each household contains its own actor critic model. An actor critic model contains 2 DNNs; one called the actor and the other called a critic. In this example, an action is how much of the controllable load will be turned on in the range of $P^{min}{}_{DR,n,t}$ and $P^{max}{}_{DR,n,t}$ (represented as $P^{*}{}_{DR,n,t}$), and the state is the total amount of electricity used by the household $P^{*}{}_{DR,n,t} + P_{base,n,t}$.

As the names imply, the actor is responsible for exploring and trying new actions whereas the critic evaluates and criticizes the actor's decisions. The dynamics of the system that require training give the uncertainty due to price and discomfort, and the agent learns with the policy (i.e. probability distribution) how its actions affect the environment. The actor uses a policy based on the inputs to select some action as the output. This action is then applied to the environment to derive the reward for taking that action as well as computing the resulting state. The state and reward are fed to the critic to calculate the value function of taking that action at the given state. Both the actor and critic rely on each other's findings as they each continue to learn. Given that our environment has a continuous action space and the actor does not know how its decisions are perceived in the environment, this method fits this problem well. Figure 4 summarizes the way the actor and critic interact with the environment. This architecture will learn the optimal weights of the DNN that will compute the optimal DR schedule. The actual structure of the actor and critic DNN is shown in Figure 5 and Figure 6 respectively, with similar inputs; household data ($s_{n,t}$) and the corresponding aggregated system term for the actor ($\bar{\lambda}_t$) or critic ($\bar{\gamma}_t$). Both the actor and critic have three hidden layers with six nodes in the first layer, eighteen nodes in the second layer and six nodes in the third layer. The output layer of the actor is a policy with three action possibilities and the output layer of the critic outputs one value function result.

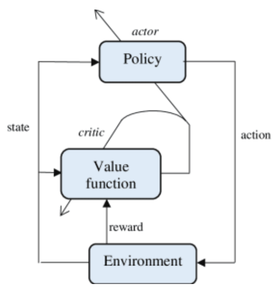

Figure 4: Actor-critic architecture [4].

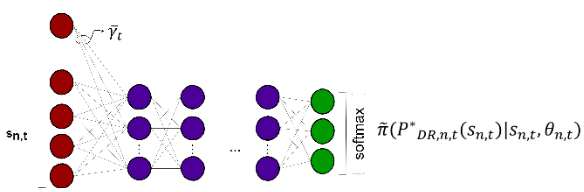

Figure 5: Actor DNN structure.

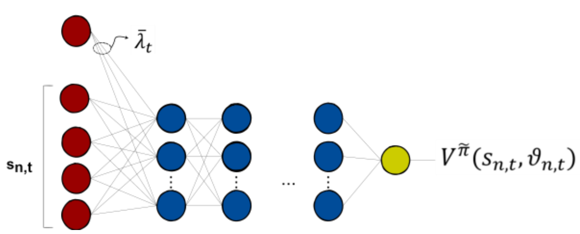

Figure 6: Critic DNN structure.

A key part to learning is the reward function ($c_{n,t}$) which looks at the cost of electricity but also adds a factor for any discomfort (d_n) that customers would feel based on the optimal state $P^*{}_{DR,n,t}$ the DNN chose. As learning is happening at each time step, the temporal difference (TD) error (δ_n) is necessary to compare the weight values of the actor ($\theta_{n,t}$) and critic ($\vartheta_{n,t}$) DNN to the previous time step. There is a discount factor (β) to ensure the most recent value function has a larger impact on the TD error. Once the TD error approaches zero, which is when the exit condition is reached in Step 6. If the end condition is not met and we iterate over another time step, the DNN weights for both the actor and critic are updated using gradient descent.

$$c_{n,t}(s_{n,t}) = \rho_t\left(P^*{}_{DR,n,t}(s_{n,t}) + P_{base,n,t}\right) + d_n\left(P^*{}_{DR,n,t}(s_{n,t}), P^{max}{}_{DR,n,t}\right) \quad (6)$$

c_n : $cost\ of\ household$ [\$]
$P^*{}_{DR,n,t}$: $optimal\ controllable\ load$ [kW]
d_n: $cost\ of\ discomfort\ based\ on\ time\ step\ t$ [\$][2]

$$\delta_n(\vartheta_{n,t-1}) = c_n(s_{n,t-1}) + \beta V^{\tilde{\pi}(\theta_{n,t-1})}(s_{n,t},\vartheta_{n,t-1}) - V^{\tilde{\pi}(\theta_{n,t-1})}(s_{n,t-1},\vartheta_{n,t-1}) \quad (7)$$

δ_n: $TD\ error\ for\ household$
β: $discount\ factor$
$V^{\tilde{\pi}(\theta_{n,t})}$: $value\ function\ of\ critic$
$\theta_{n,t}$: $actor\ DNN\ weights$
$\vartheta_{n,t}$: $critic\ DNN\ weights$

The UBC research team has answered this same problem with other models like Q-Learning and Double Q-Learning. When the team compared those models to actor-critic, none of the models reduced expected daily cost and converged to a solution as quickly demonstrating that this model has better performance for this problem.

[2] The paper describes this piecewise function.

EXPERIMENT

This section will review the experiment conducted and some details about the code shared on GitHub[3].

The synthetic dataset was generated for 32 households hourly electricity consumption data for 100 days. This dataset did not consider complexities included in the paper's input data like how some controllable loads were interruptible and should have had additional constraints in the formulation of the problem. Electricity prices were the same as in the paper, seen in Figure 7. It was assumed that all weekdays and weekends had this price structure, regardless if it was a holiday or not. Additionally, all customers as instructed carried out all DNN DR decisions.

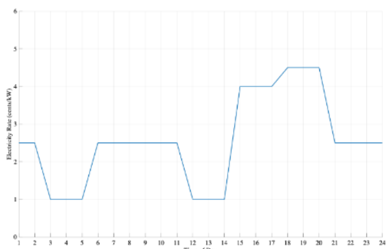

Figure 7: Electricity price based on time of day.

The data import structure was an important design exercise. It was critical to understand how the minimum controllable load, ${P^{min}}_{DR,n,t}$ and maximum controllable load, ${P^{max}}_{DR,n,t}$ relate to each other, and how to build the household state, $s_{n,t}$.

This experiment setup was performed on a MacBook Pro with a 2.5 GHz Quad-Core i7 Intel processor and Intel Iris Pro 1536 MB. While there was enough data generated to run through all 32 households, this experiment performed the actor-critic method for 1 household. This resulted in a little over 1.2 million floating point operators (FLOPs) instead of over 40 million FLOPs if 32 households had been implemented.

It was critical to understand the principle of federated learning and secure aggregation in order to implement this data privacy method. Resources that explained this concept which were critical were [3], [5]. Similarly for the actor-critic method, the following resources helped significantly to understand the method and implementation ideas [6]–[13]. Since vast majority of these resources reference the actor-critic as a game, it was extremely useful to consider this problem as a game to determine. The game is that the distribution network has a set of rules we need to meet[4] and the customer is trying to reduce their electricity bills while still maintaining their comfort.

There were several tricks used for the implementation of the model. The initial stages of actor critic have large oscillations that do not make sense in reality. To address this, in the implemented code there is a bound set to ensure the electricity used by a household does not exceed the maximum controllable load, ${P^{max}}_{DR,n,t}$ as this is the maximum amount that a household would consume. This regularization also helps the model learn by building this intuitive constraint into the implementation. To ensure the actor in particular is exploratory to learn the environment, instead of using softmax to select the highest probability, a categorical probability distribution yields discrete action spaces [9]. This implementation was particularly useful with the *tensorflow_probability* library. To avoid training two networks, used common input and hidden layers but different output layers. In training these models, playing with the number of actions for the actor and learning rate helped to fine tune the model but it was found that three actions and a learning rate of 0.0003 worked well.

[3] https://github.com/AnjaliW/EE8223-Final-Project

[4] This was not implemented but reflected in the problem statement.

RESULTS

This section presents the results from the implementation shown in the GitHub code[5].

Once the DNN weights begin to converge, Figure 8 shows one household's consumption pattern if there was no DR load scheduling compared to the result of the DR with load scheduling. It is clear that the discomfort of the customer is taken into account as the peaks are flattened and there is a clear reduction of load which yields cost savings. When compared to the paper results in Figure 9, the related curves are "Without Load Scheduling (Scenario One)" and "With Load Scheduling without Supply Limit (Scenario Two)". The synthetic data generated as an input has a larger morning peak, and but the general shapes are similar. It is evident that Figure 8 sheds significantly more load yielding on average a reduction of 53% load reduction whereas the paper found 33%. This difference is likely given that the number in the paper includes an optimization for the distribution network's power flow whereas the implementation in this paper does not consider this as a constraint on the feasible action space.

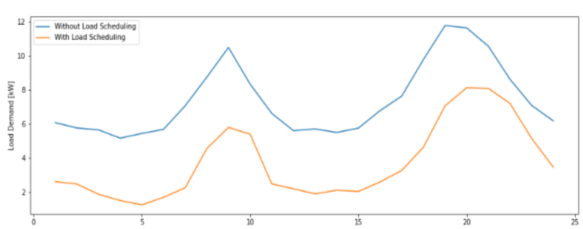

Figure 8: Household consumption pattern on a day with and without load scheduling (i.e. DR).

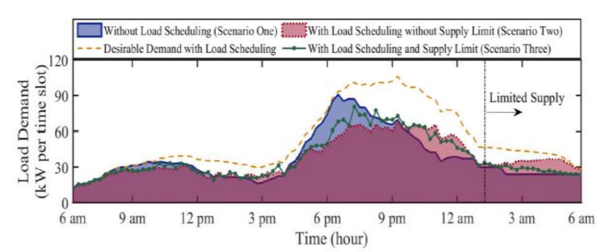

Figure 9: Paper's result system consumption pattern on a day with and without load scheduling (i.e. DR).

Figure 10 shows the daily cost for the household according to the learning that took place. As shown, the weights of the DNN converge around 24 days showing about 41% reduction in cost compared to the earlier training days. This is further observed in Figure 11 when the expected daily total cost is compared on days with and without load scheduling. This chart demonstrates how bounding the maximum controllable load ${P^{max}}_{DR,n,t}$ is not oscillating like it was while the model's parameters were being tuned. This shows how the expected daily cost stabilizes at around $2.63 per day with load scheduling compared to $4.81, again showing about 45% in cost savings. The general trend of training the model is similar to the results from the paper shown in Figure 12, which converged to a solution at 30 days.

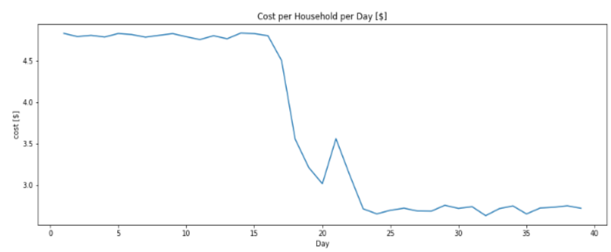

Figure 10: Daily cost for household.

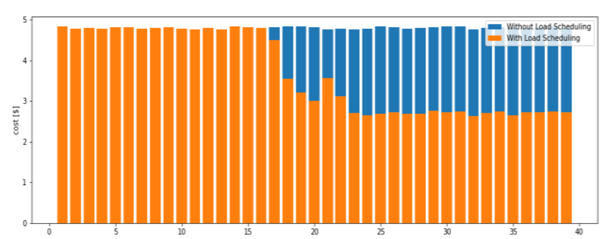

Figure 11: Expected daily total cost with and without load scheduling.

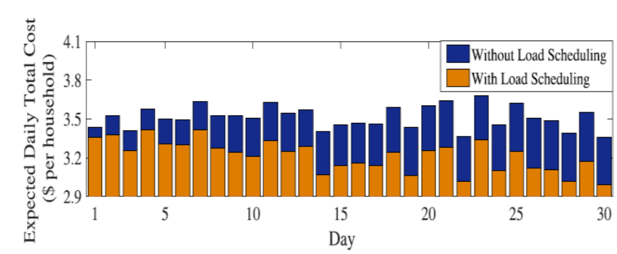

Figure 12: Paper's expected daily total cost with and without load scheduling.

5 https://github.com/AnjaliW/EE8223-Final-Project

Lastly, the value function output from the critic is shown in Figure 13 over all 100 days. The trend of the value function in this implementation closely replicates the paper's result shown in Figure 14. The value function shows that the actor has been making good decisions as the oscillations tapper off.

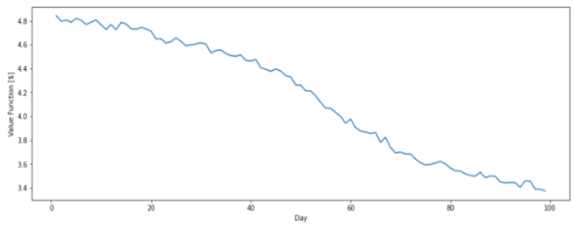

Figure 13: Value function implementation from critic.

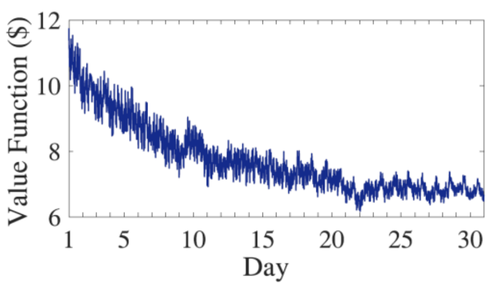

Figure 14: Paper's resulting value function.

CONCLUSION

An actor critic model with federated learning was implemented for 1 out of 32 households. This household experienced over 41% in cost reduction and achieved 53% in consumption reduction to support distribution network needs. These results cannot be compared at face value to the results of the paper seeing 13% in cost reduction and 33% in consumption reduction as the paper considered the additional constraint of the feasible action space given the distribution network constraints which was beyond the requirements for this project. Nonetheless, the trends observed in the resulting figures show a similar pattern.

This paper and implementation focuses a lot on the role of customers in DR, an important factor if these programs are implemented in a real world setting. Federated learning was an important concept to ensure protection of customer data. Then implicating customer preferences like discomfort is crucial to effectively integrating the customer into the transition towards grid modernization. These unique customer preferences are important to learn in a way that makes the customer feel safe given the uncertainty in consumption and action possibilities.

Further work could study the characteristics and unique aspects of different loads to ensure customers are still satisfied with how their controllable loads are controlled. Creating a custom customer experience is an aspect that aggregators will likely have to take on. In addition, dynamic electricity price structure could be elaborated to determine how renewable energy and decarbonisation targets can be met by further incentivizing customers to adapt to these clean generation sources. Depending on what distribution networks are generating from renewable energy, price structures can be dynamically set. Another aspect could be to consider the economics of the capital expenditures distribution networks would have to spend in order to upgrade their system if they were not able to use customers' controllable loads to better manage their networks and extend lifetime of network assets. This cost benefit analysis would be of great use to customers and distribution networks alike.

ANNEX

The following section describes how the synthetic data for the implementation code (InputData_HouseholdStates.xls) was calculated using Raw_Data_Collection.xls.

Five household types were arbitrarily selected to build the set of households (see 1-Home Type Profile on spreadsheet). A description of these households can be found in Table 1. It was assumed that the days were hot and during summer when the AC would be on consistently, all households had an electric vehicle (EV) and none of the houses had a pool (all of which have a significant impact on electricity profile). In terms of occupancy, assume all adults except for 2 adults in a multigenerational home go to work during business work hours, and all kids go to school during the same work hours.

Table 1: House profile assumption summary.

Profile #	Description	# of Adults	# of Kids	Total # Occupants	% of Homes
1	Traditional family	2	2	4	40
2	Multi-generational	4	2	6	5
3	Couple	2	0	2	25
4	Single adult	1	0	1	20
5	Single parent household	1	2	3	10

All homes have controllable and non-controllable loads (see 2-Load Consumption Research tab on spreadsheet). The controllable loads are air conditioner, refrigerator, washing machine, dishwasher, EV, electric water heater, and stove-oven. Average hours of operation for these loads were found and normalized across different days based on the daily operation as seen in Table 2. In addition, each house has baseload from loads like lighting, phantom loads (i.e. electricity that is consumed just because things are plugged in but not necessarily actively being used) and other common loads. The average household demand for these baseloads are summarized in Table 3.

Table 2: Controllable load operation details.

Controllable load	Average daily electricity demand [kWh/day]	Operation Details
AC Central [14]	84	On the whole day
Washing Machine [14]	0.67	Non-interruptible, 1 time event in day
Dishwasher [14]	1.73	Non-interruptible 1 time event in day
Fridge [14]	5	On the whole day
EWH – 2 person household [14]	11.67	On the whole day
EWH – 4 person household [14]	17.47	On the whole day
Stove-Oven [14]	20.83	Non-interruptible 1 time event in day
EV [15]	9.4	Interruptible during particular times

Table 3: Baseload operation details.

Load	Average hourly demand [kWh/hr]
Phantom Loads [16]	0.048
Lighting [17]	8.65
Common Loads [17]	0.36

Based on the research and on assumptions shown in Table 4, load demand patterns were normalized over an average day for all controllable loads and base loads (see 3-Consumption Estimator tab on spreadsheet including the explanation for each hour's assumed operation for each load). All houses except house profile #2 have the exact same baseline, and the rest have a customized amount of controllable load operation schedule. Note that these values represent the most comfortable setting and operation of these loads. All these values are compiled by house profile type (see the following tabs on the spreadsheet: Type1_4Occupants, Types2_6Occupants, Type3_2Occupants, Type4_1Occupant, and Type5_3Occupants) for each hour in 1 day.

Table 4: House profile assumption summary.

Profile #	Range of controllable loads	Range of baseload	Explanation
1	Standard	Standard	Traditional home with significant controllable load during periods away from home or while sleeping. With kids at home, can't always have maximum amount of controllable load. Since most people are away during the days, baseload is standard.
2	Minimal	Minimal	With someone at home during all hours of the day, minimal controllable load during the day but more potential at night. With kids at home, can't always have maximum amount of controllable load. Since grandparents are home, minimal changes on baseload throughout day.
3	Maximum	Standard	Traditional home with significant controllable load during periods away from home or while sleeping. No kids either so maximum controllable load potential. Since most people are away during the days, baseload is standard.
4	Maximum	Standard	Traditional home with significant controllable load during periods away from home or while sleeping. No kids either so maximum controllable load potential. Since most people are away during the days, baseload is standard.
5	Standard	Standard	Traditional home with significant controllable load during periods away from home or while sleeping. With kids at home, can't always have maximum amount of controllable load. Since most people are away during the days, baseload is standard.

Lastly, to calculate a minimum controllable load that is still possible but would induce discomfort for the household is summarized (see P_ctrl_min tab on spreadsheet). This table varies the minimum acceptable amount for the controllable load logged to be reduced at varying hours of the day based on what seemed reasonable.

This spreadsheet (Raw_Data_Calculation) was then fed into MATLAB to create the data file the implementation of the paper would use. See ML Inputs.m for the file that reads in all this data, produces a load curve for each day with some added noise as seen in Figure 15 to output the file InputData_HouseholdStates that is used for the paper implementation. Household data was computed for 100 days for 32 households as shown in Figure 16.

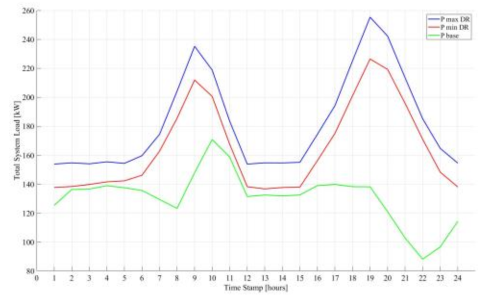

Figure 15: One sample day plotted for aggregated synthetic 32 household data.

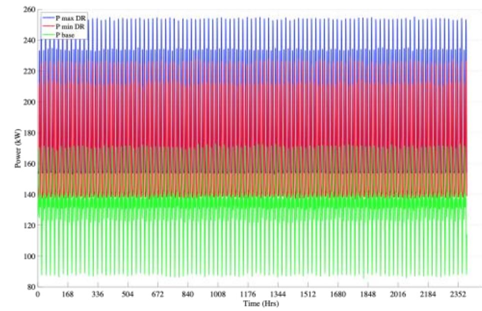

Figure 16: Input file for implementation for hourly electricity consumption across 32 households for 100 days.

BIBLIOGRAPHY

[1] "Figure 7. System load curve before and after demand response.," *ResearchGate*. https://www.researchgate.net/figure/System-load-curve-before-and-after-demand-response_fig3_339008018 (accessed Nov. 18, 2021).

[2] S. Bahrami, V. W. S. Wong, and J. Huang, "An Online Learning Algorithm for Demand Response in Smart Grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4712–4725, Sep. 2018, doi: 10.1109/TSG.2017.2667599.

[3] Google Cloud Tech, *What is Federated Learning?*, (Feb. 05, 2021). Accessed: Dec. 12, 2021. [Online Video]. Available: https://www.youtube.com/watch?v=X8YYWunttOY

[4] I. Schallheim, "Implementation and Optimization of an Open Loop Gait Controller on a Mono Pedal Robot," 2018.

[5] M. Habib ur Rehman, K. Salah, E. Damiani, and D. Svetinovic, "Towards Blockchain-Based Reputation-Aware Federated Learning," Feb. 2020. doi: 10.1109/INFOCOMWKSHPS50562.2020.9163027.

[6] "6.6 Actor-Critic Methods." http://incompleteideas.net/book/first/ebook/node66.html (accessed Dec. 12, 2021).

[7] Serrano.Academy, *A friendly introduction to deep reinforcement learning, Q-networks and policy gradients*, (May 24, 2021). Accessed: Dec. 12, 2021. [Online Video]. Available: https://www.youtube.com/watch?v=SgC6AZss478

[8] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients," *IEEE Trans. Syst., Man, Cybern. C*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012, doi: 10.1109/TSMCC.2012.2218595.

[9] Machine Learning with Phil, *Everything You Need To Master Actor Critic Methods | Tensorflow 2 Tutorial*, (Sep. 30, 2020). Accessed: Dec. 12, 2021. [Online Video]. Available: https://www.youtube.com/watch?v=LawaN3BdI00

[10] "Policies and Value Functions - Good Actions for a Reinforcement Learning Agent - YouTube." https://www.youtube.com/watch?v=eMxOGwbdqKY&list=WL&index=32&ab_channel=deeplizard (accessed Dec. 12, 2021).

[11] "Chapter 12. Reinforcement learning with actor-critic methods · Deep Learning and the Game of Go." https://livebook.manning.com/book/deep-learning-and-the-game-of-go/chapter-12/ (accessed Dec. 12, 2021).

[12] "Actor Critic Algorithms - YouTube." https://www.youtube.com/watch?v=w_3mmm0P0j8&t=221s&ab_channel=SirajRaval (accessed Dec. 12, 2021).

[13] K. Team, "Keras documentation: Actor Critic Method." https://keras.io/examples/rl/actor_critic_cartpole/ (accessed Dec. 13, 2021).

[14] "Burlington Hydro - Appliance Usage." https://www.burlingtonhydro.com/powertoconserve/residential/appliance-usage.html (accessed Dec. 12, 2021).

[15] Virta, "EV Charging - How much electricity does an electric car use?" https://www.virta.global/blog/ev-charging-101-how-much-electricity-does-an-electric-car-use (accessed Dec. 12, 2021).

[16] Toronto and Region Conservation, "Evaluation of a Residential Base Laod Monitoring and Management System." [Online]. Available: https://sustainabletechnologies.ca/app/uploads/2013/01/Energy-Mgmt-2012_web.pdf

[17] A. Parekh, P. Wang, and T. Strack, "Survey Results of User-Dependent Electricity Loads in Canadian Homes," p. 12.