Efficient and Accurate Peer-to-Peer Training of Machine Learning Based Home Thermal Models

Karim Bouboush*
karim.boubouh@um6p.ma
UM6P
Benguerir, Morocco

Robert Basmadjian
robert.basmadjian@tu-clausthal.de
Clausthal University of Technology
Clausthal-Zellerfeld, Germany

Omid Ardakanian
ardakanian@ualberta.ca
University of Alberta
Edmonton, Canada

Alexandre Maurer
alexandre.maurer@um6p.ma
UM6P
Benguerir, Morocco

Rachid Guerraoui
rachid.guerraoui@epfl.ch
EPFL
Lausanne, Switzerland

ABSTRACT

The integration of smart thermostats in home automation systems has created an opportunity to optimize space heating and cooling through the use of machine learning, for example for thermal model identification. Nonetheless, its full potential remains untapped due to the lack of a suitable learning scheme. Traditional centralized learning (CL) and federated learning (FL) schemes could pose privacy and security concerns, and result in a generic model that does not adequately represent thermal requirements and characteristics of each individual home. To overcome these limitations, in this paper we embrace the novel peer-to-peer learning scheme for on-device training of home thermal models. Specifically, we adapt the personalized peer-to-peer algorithm proposed in recent work (called P3) to efficiently train personalized thermal models on resource-constrained devices. Our preliminary experiments with data from 1,000 homes, using the LSTM model, demonstrate that the adapted P3 algorithm produces accurate and personalized thermal models while being extremely energy-efficient, consuming respectively 600 and 40 times less energy than the CL and FL schemes. This result suggests that the P3 algorithm offers a privacy-conscious, accurate, and energy-efficient solution for training thermal models for the many homes in the building stock.

CCS CONCEPTS

• **Computing methodologies → Machine learning**.

KEYWORDS

Smart thermostats, peer-to-peer machine learning, energy-efficiency, thermal models, personalized models.

ACM Reference Format:
Karim Boubouh, Robert Basmadjian, Omid Ardakanian, Alexandre Maurer, and Rachid Guerraoui. 2023. Efficient and Accurate Peer-to-Peer Training of Machine Learning Based Home Thermal Models. In *The 14th ACM International Conference on Future Energy Systems (e-Energy '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3575813.3597453

e-Energy '23, June 20–23, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0032-3/23/06...$15.00
https://doi.org/10.1145/3575813.3597453

1 INTRODUCTION

Smart thermostats have brought home automation systems to the next level of intelligence thanks to their advanced sensing, communication, and control capabilities. Recent advancements have led to the availability of sophisticated smart thermostats [2, 3, 13, 25, 29] capable of running machine learning (ML) inference, fine-tuning, and even on-device training of thermal models for the optimal control of a home's thermal environment – a task that was deemed impossible with programmable thermostats. Extensive research has been conducted on developing thermal models for homes, ranging from white-box modeling approaches that adhere to the laws of physics [9, 18], to gray-box modeling approaches that estimate the parameters of a low-order resistance-capacitance (RC) network from the available sensor data [11, 15, 19, 26], to black-box models using neural networks [21, 22, 28] such as Long Short-Term Memory (LSTM) and time-series statistical models [23], such as ARMA, ARIMA, ARIMAX, and SARIMAX, which merely rely on the sensor data generated by smart thermostats [22, 23, 28].

In this paper, we train thermal models based on LSTM to accurately predict the room temperature. These models are trained using three main schemes, namely centralized learning (CL), federated learning (FL), and peer-to-peer (P2P) learning. In CL, sensor data of each home are sent to a central server and used to train a global thermal model. This raises privacy concerns since the data collected by smart thermostats may contain private information (e.g., occupancy pattern [16]). Furthermore, the central server is potentially a single point of failure (SPoF). In FL, the central server does not have access to the homes' raw data. Instead, it aggregates the model updates computed locally by the participating homes. However, an honest-but-curious server can still infer private information from a client's updates [12]. Moreover, the aggregation server is still a SPoF. A recently developed alternative is peer-to-peer learning, where the central server is eliminated [7, 27], thereby alleviating the SPoF issue and mitigating privacy attacks by an honest-but-curious server. In this approach, every client (i.e., home) trains the respective thermal model locally, then improves it through collaboration with its neighboring peers.

We embrace the P2P learning scheme by adapting the personalized peer-to-peer (P3) algorithm of [5], and execute it on commodity devices such as mobile phones. To minimize the time and energy required to train the thermal models, we take advantage of two abstraction techniques: spatial abstraction via clustering and temporal abstraction via data down-sampling. The empirical results show that, among the three learning schemes, P3 is more accurate and energy efficient, with 600 and 40 times less energy consumption than CL and FL, respectively. Our contribution is twofold:

- We customize the P3 algorithm using a new aggregation rule based on the similarity between the neighboring homes to train accurate and personalized home thermal models.
- We demonstrate that training thermal models using temporal and spatial abstractions for the three schemes results better accuracy and reduction in energy use. However, P3 is extremely energy efficient compared to CL and FL even without abstraction.

The rest of the paper is organized as follows. In Section 2, we give an overview of the related work. Section 3 briefly presents the P3 algorithm and how it is customized for the thermal modeling task. In Section 4, we describe our spatial and temporal abstraction techniques. Section 5 compares the three learning schemes from the perspectives of energy and performance. Section 6 concludes the paper and presents some directions for future work.

2 RELATED WORK

Several attempts have been made in the past to develop a thermal model for homes and buildings, and to predict the room temperature for the optimal operation of the HVAC system. These techniques can be classified into three categories:

White-box models. They use detailed information about the building structure and geometry to write complex mathematical equations based on the laws of thermodynamics [18]. These models are used in popular building performance simulation software, such as EnergyPlus [10].

Grey-box models. They involve mapping the building's envelope and its interior walls into a number of temperature-uniform lumps in a Resistance–Capacitance (RC) network [4]. Then, they use the available smart thermostat data to estimate the parameters of this model [14, 15, 19].

Black-box models. They use the sensor data emitted by the building's smart thermostat to build a model to predict the room temperature. This model can be a time-series model, such as autoregressive model with exogenous inputs (ARX) [23] and autoregressive "moving-average" model with exogenous inputs (ARMAX) [24], or a neural network model, such as multilayer perceptron (MLP) [21, 22] or long short-term memory (LSTM) [21]. We pursue this line of work in this paper.

3 PEER-TO-PEER LEARNING OF THERMAL MODELS

We consider a set of homes that wish to train an ML model for predicting the room temperature using the data collected by their smart thermostat sensors, while taking into account the thermal models of other homes over a peer-to-peer network. This network graph represents a semantic overlay on top of the communication layer, defining a gateway between pairs of homes sharing similar characteristics, without revealing their identity or sharing their private data. Each home aims to learn a personalized thermal model that reflects its own data, while incorporating the collective knowledge of similar homes to improve its thermal model in a privacy-preserving fashion. Hereafter, we use "home" to refer to both the physical home and its device used for communication and model training.

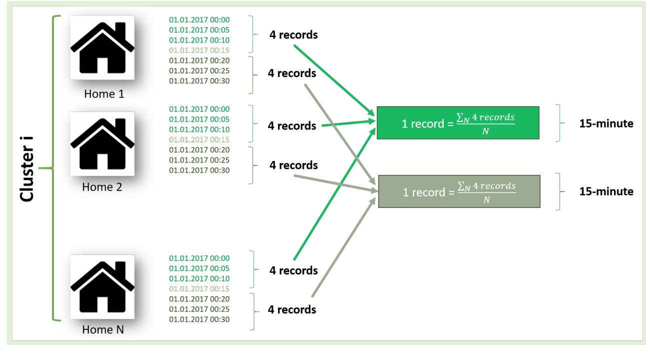


Figure 1: Graphical presentation of temporal abstraction using 15-minute intervals. Similar homes are grouped into clusters, and their data are aggregated and down-sampled in the case of CL, or down-sampled in the case of FL and P3.

To train personalized thermal models in a peer-to-peer fashion, we adopt the P3 algorithm proposed in [5], which consists of two training phases: a local learning phase, where each home learns a thermal model locally, and a collaboration phase, where homes collaborate to enhance their locally trained models. However, instead of the generic version presented in [5], we use an adapted version of P3 that allows for homes with varying computational capabilities to participate in the collaborative training using adequate sample sizes s_i, thereby democratizing the participation in training of thermal models for all homes. To further enhance the collaboration, we propose a new aggregation rule defined in Equation 1 to obtain weighted model updates θ_i^t for each home i at round t based on the similarity between neighboring homes:

$$\theta_i^t = \frac{1}{|V_i^t|} \sum_{j \in V_i^t} \frac{W_{ij}}{D_{ii}} w_j \tag{1}$$

where V_i^t is the set of received and accepted updates from neighbors, W_{ij} is the weighted edge between home i and its neighbor j with $D_{ii} = \sum_{j=1}^{d(i)} W_{ij}$, such that $d(i)$ denotes the degree of node i.

4 METHODOLOGY

We now present our methodology for efficiently training thermal models using the three learning schemes: CL, FL, and P3.

4.1 Temporal Abstraction

Most smart thermostats are equipped with multiple sensors that continuously record quantities related to the indoor and outdoor environments, often at high-frequency (e.g., five minute intervals) for real-time monitoring. However, the fluctuation in indoor temperature over such short intervals is generally small, and can be neglected when designing temperature prediction models. Under this assumption, we suggest using numerical abstraction [8] to derive approximations of the home's environment in larger time intervals, in order to reduce the size of data for fast and energy-efficient thermal model training. Furthermore, the proposed temporal abstraction allows homes to share less detailed sensor data, thus protecting their privacy. We downsample successive home records to larger time intervals of 15, 30, and 60 minutes. Figure 1 illustrates this temporal abstraction using 15-minute intervals.

4.2 Clustering of Homes

Buildings have different thermal properties that depend on various factors, such as location, floor area, and age. The training of a single thermal model, representative of all homes, is therefore not a viable solution. To address this problem, we allow similar homes to train their thermal models together. Hossain et al. [15] have shown that homes in the same climate zone with comparable floor area and age exhibit similar thermal behavior. Assuming we have access to this metadata (location, floor area, and age of homes), we performed clustering using k-means on normalized values of "floor area" and "age" properties to group similar homes into a small number of clusters, k, that is chosen using the elbow method [17]. Homes within each cluster can either send their data to the cloud for centralized model training, jointly train a representative model using FL, or collaboratively learn a personalized model by connecting to each other using the P3 algorithm.

5 EVALUATION

In this section, we first introduce the implementation details, dataset and configured temporal abstraction scenarios used in our experiments. Then, we evaluate the P3 algorithm and compare it to FL and CL from the perspective of energy consumption and accuracy.

5.1 Experimental Setup

5.1.1 Implementation Details. In our experiments, we use two hardware configurations. A workstation representing the central server for both CL and FL, and mid-range Android mobile devices to represent the resource-constrained device (i.e., limited compute power and energy supply) that homes can afford when training the thermal model, in the case of FL and P3. Table 1 presents hardware and software specifications of both configurations. The experiments involving the server were conducted by setting the frequency of the server's CPU to the highest, lowest, and middle values. Our experiments confirmed that using the middle frequency results in the most energy-efficient training. Thus, all experiments were run using the middle frequency value of the processor (i.e., 1.7 GHz).

The source code of our implementation for both Linux servers and Android mobile devices can be found at [1].

5.1.2 Smart Thermostat Dataset. To evaluate the proposed approaches, we considered a dataset of smart thermostats for 1,000 homes in 4 US states: California, New York, Texas, and Illinois. The dataset is collected within the context of the Ecobee's "Donate Your Data" dataset initiative [20]. It contains time-series data with 5-minute resolution collected over 12 months in 2017, spanning all 4 seasons. The size of the dataset is 34 GB. It includes the following features: "Floor area" (Square foot), "Age" (Scalar, year of construction), "Indoor temperature" (Fahrenheit), "Outdoor temperature" (Fahrenheit), "Indoor humidity" (%RH), "Outdoor humidity" (%RH), "Indoor cool setpoint" (Fahrenheit), "Indoor heat setpoint" (Fahrenheit), and "Operating mode of HVAC" (Boolean).

Platform	OS	CPU	Frequency	RAM
Linux Server	Ubuntu 20.04 LTS	Intel Xeon W-2123	Min 1.2 GHz Max 3.6 GHz	32GB DDR4
Android Device	Android 11	Qualcomm SDM710	Min 1.7 GHz Max 2.2 GHz	4GB DDR4

Table 1: Hardware and software characteristics of the considered server and Android device used in our experiments.

5.1.3 Temporal Abstraction Scenarios. We set up four different scenarios in our experiments to train thermal models using the LSTM model. In the first three temporal abstraction scenarios, namely "1 Hour", "30 Min", and "15 Min", the records of homes within the same time interval (i.e., 15, 30, or 60 minutes) are averaged to a single record for each timestamp; then, the resolution of the resulting records is reduced to the target time interval. This down-sampling technique greatly reduces the size of the training data, enabling even less powerful devices to participate in model training and save network bandwidth in the case of exchanging raw data. The "5 Min" scenario corresponds to the default interval of 5 minutes in FL and P3, whereas for CL we average the homes' records without down-sampling. Finally, in the default "RAW" scenario for CL, the received homes' raw data are concatenated without any abstraction.

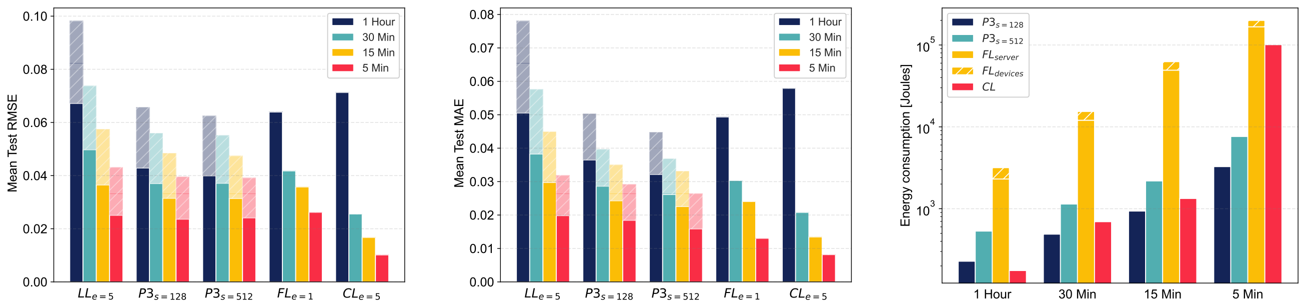
5.1.4 ML Model and Training Parameters. To evaluate our model training methodology, we considered a powerful LSTM model for room temperature prediction. To train each thermal model, we used 85% of the dataset; the remaining 15% was dedicated to testing. More precisely, for each season of the year (i.e., 3 months), we considered two and a half months for training and the remaining 15 days were used to test the learned models. In the experiments conducted under the CL setup, the thermal model was trained for 5 epochs ($e = 5$). In the FL setup, a global model was trained for 100 rounds, with each home training the received model for one epoch ($e = 1$) before sending back its updated model to the server for aggregation. As for P3, each home trained a local model for a single epoch ($e = 1$) using its local data before joining the collaborative training, where the homes trained their personalized thermal models on a small number of samples. These samples were configured with batch sizes s of 128 (the default value), 256, 512, and 1024 samples.

5.2 Accuracy and Efficiency of P3

We present the performance and energy consumption results of CL, FL and P3 in this section. We considered our proposed clustering methodology for CL and FL. For P3, we generated a network graph of homes by constructing a sparse similarity matrix W_{ij}, resulting in each home having 10 neighbors on average.

5.2.1 Model Performance. Figures 2a and 2b show the accuracy[1] of the trained LSTM models under the studied learning schemes, using our proposed temporal abstraction scenarios. To highlight the

[1]Mathematical formulas of RMSE and MAE are given in Section A.3 of [6].



(a) Accuracy (in terms of RMSE) of CL, FL and P3, using different temporal abstractions.

(b) Accuracy (in terms of MAE) of CL, FL and P3, using different temporal abstractions.

(c) Energy consumption of training LSTM using temporal abstraction in CL, FL and P3.

Figure 2: Performance and energy consumption of training an LSTM thermal model in CL, FL and P3, using different temporal abstractions. In (c), we use a logarithmic scale, as the energy use of CL and FL is significantly higher than P3.

significance of the collaborative learning phase of P3 in improving the performance of the trained models, we have included the case of Local Learning (LL), where each home trains its thermal model locally for five complete epochs ($e = 5$) without getting involved in any collaboration with other homes. Since P3 allows each home to learn its own personalized model, the bars with solid color represent the best accuracy value that homes achieved on average, and the dashed bars represent the standard deviation of the accuracy values.

Figures 2a and 2b show that when homes train their model locally, they typically learn a much worse thermal model compared to the other learning schemes, regardless of the temporal abstraction we use. The large dashed bars in LL indicate that some homes can still learn accurate thermal models if they have sufficient training data and computational resource. However, the majority of homes fail to learn a sufficiently accurate thermal model. The P3 algorithm enables these homes to collaborate and achieve a better thermal model by exchanging model updates, using a small enough number of samples that even homes with newly installed smart thermostats can obtain after a few hours of operation. Depending on the chosen sample size, P3 can perform more computation and result in better thermal models, as seen with $P3_{s=128}$ and $P3_{s=512}$. Furthermore, using larger abstraction intervals (e.g., "30 Min" scenario) can greatly reduce the computational and energy requirement for thermal model training of homes using resource-constrained devices, at the cost of a small reduction in model accuracy, as depicted in both figures.

Except for the "1 Hour" scenario, the performance of P3 was on average slightly worse than CL and FL. This is not surprising for CL, given that the server receives data from all homes, and trains a single model accordingly. It is important to mention that we have trained the thermal models for CL using the "RAW" scenario and achieved the worst performance (see Figure 3a in the appendix). In comparison, homes in FL update the global model using the full dataset for each round, while P3 uses a limited number of samples (to account for homes using weak devices for training) and aims to reduce the training energy use. Nevertheless, many homes in P3 performed better than FL using much less training data, while other homes (around 25 out of 207 homes, according to our experiments) performed much worse, resulting in high standard deviation.

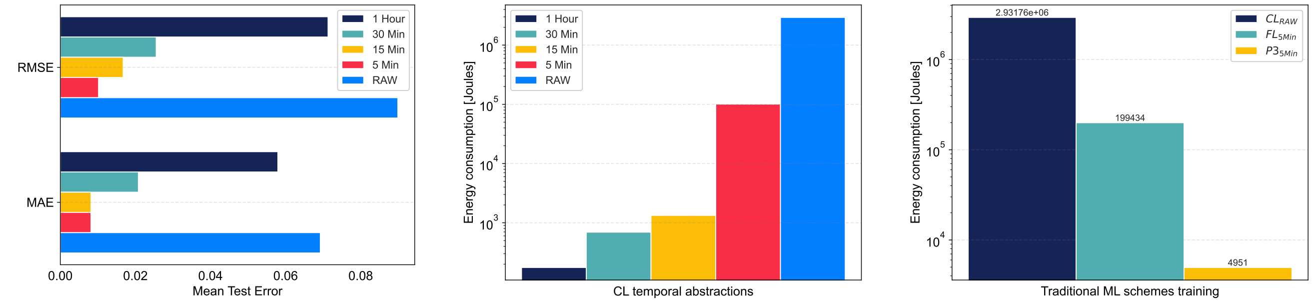
5.2.2 Energy Consumption. We evaluated the energy consumption of the LSTM model training for each one of the three temporal scenarios ("1 Hour", "30 Min", and "15" Min) and the "5 Min" scenario, which represents the default interval for FL and P3, and the averaging scenario for CL. For P3, both the default ($s = 128$), and the optimal ($s = 512$) batch sizes are presented, while for FL, we differentiate between the energy consumed by the server (FL_{server}) and mobile devices ($FL_{devices}$). It is worth mentioning that, despite the small bars of the energy consumed by the devices for FL in Figure 2c (due to the logarithmic scale of the Y-axis), the energy consumption is still significant. For example, in the "15 Min" scenario, the energy consumed by FL_{server} was 49208 Joules, while $FL_{devices}$ consumed 13301 Joules. In the same setting, $P3_{s=512}$ consumed only 2182 Joules. Figure 2c gives the energy consumption (in Joules) of the different experiments conducted. When comparing P3 and CL, except for the "1 Hour" scenario (where CL has a slight edge over P3 for a batch size of 128), in all other scenarios, P3 trains the thermal model for each home more efficiently than CL (both for batch sizes 128 and 512), especially for the "5 Min" scenario, where the difference is quite significant. Comparing P3 and FL, for all considered scenarios, P3 is extremely energy efficient. It is worth underlying the enormous energy consumption of the mobile devices in FL (dashed boxes in Figure 2c, compared to P3.

6 CONCLUSION

We adopted the P2P learning scheme and customized the personalized peer-to-peer (P3) algorithm of [5] for training home thermal models by proposing a new aggregation rule based on the similarities between the neighboring homes, which turned out to be crucial for obtaining a personalized thermal model for each home. Furthermore, we employed temporal and spatial abstraction techniques to reduce the runtime and energy use of training the models. Our experiments showed that P3 enables model training in an extremely energy-efficient manner, compared to CL and FL, while maintaining high accuracy. Thus, it offers a private and practical approach for training thermal models that can be incorporated in the HVAC control loop. In future work, we plan to leverage heterogeneity of smart thermostats to make possible adaptive contribution to model training, depending on their hardware capabilities.

REFERENCES

[1] 2023. Thermal P3. https://github.com/karimboubouh/Thermal-P3.
[2] Sonny Ali and Zia Yusuf. 2018. Mapping the smart-home market.
[3] Onur Ayan and Belgin Turkay. 2018. Smart thermostats for home automation systems and energy savings from smart thermostats. In *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*. IEEE, 1–6.
[4] Peder Bacher and Henrik Madsen. 2011. Identifying suitable models for the heat dynamics of buildings. *Energy and Buildings* 43, 7 (2011), 1511–1522.
[5] Robert Basmadjian, Karim Boubouh, Amine Boussetta, Rachid Guerraoui, and Alexandre Maurer. 2022. On the advantages of P2P ML on mobile devices. In *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*. 338–353.
[6] Robert Basmadjian, Amirhossein Shaafieyoun, and Sahib Julka. 2021. Day-Ahead Forecasting of the Percentage of Renewables Based on Time-Series Statistical Methods. *Energies* 14, 21 (2021). https://doi.org/10.3390/en14217443
[7] Karim Boubouh, Amine Boussetta, Yahya Benkaouz, and Rachid Guerraoui. 2020. Robust P2P Personalized Learning. In *2020 International Symposium on Reliable Distributed Systems (SRDS)*. IEEE, 299–308.
[8] Patrick Cousot and Radhia Cousot. 2014. Abstract interpretation: past, present and future. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. 1–10.
[9] Drury B Crawley, Linda K Lawrie, Curtis O Pedersen, and Frederick C Winkelmann. 2000. Energy plus: energy simulation program. *ASHRAE journal* 42, 4 (2000), 49–56.
[10] Drury B Crawley, Linda K Lawrie, Frederick C Winkelmann, Walter F Buhl, Y Joe Huang, Curtis O Pedersen, Richard K Strand, Richard J Liesen, Daniel E Fisher, Michael J Witte, et al. 2001. EnergyPlus: creating a new-generation building energy simulation program. *Energy and buildings* 33, 4 (2001), 319–331.
[11] L. Di Natale, B. Svetozarevic, P. Heer, and C.N. Jones. 2022. Physically Consistent Neural Networks for building thermal modeling: Theory and analysis. *Applied Energy* 325 (2022), 119806.
[12] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems* 33 (2020), 16937–16947.
[13] Grant Hernandez, Orlando Arias, Daniel Buentello, and Yier Jin. 2014. Smart nest thermostat: A smart spy in your home. *Black Hat USA* 2015 (2014).
[14] Md Monir Hossain, Tianyu Zhang, and Omid Ardakanian. 2019. Evaluating the Feasibility of Reusing Pre-Trained Thermal Models in the Residential Sector. In *Proceedings of the 1st ACM International Workshop on Urban Building Energy Sensing, Controls, Big Data Analysis, and Visualization (UrbSys'19)*. ACM, 23–32.
[15] Md Monir Hossain, Tianyu Zhang, and Omid Ardakanian. 2021. Identifying grey-box thermal models with Bayesian neural networks. *Energy and Buildings* 238 (2021), 110836.
[16] Brent Huchuk, Scott Sanner, and William O'Brien. 2019. Comparison of machine learning models for occupancy prediction in residential buildings using connected thermostat data. *Building and Environment* 160 (2019), 106177.
[17] Kalpana D Joshi and PS Nalwade. 2013. Modified k-means for better initial cluster centres. *International Journal of Computer Science and Mobile Computing* 2, 7 (2013), 219–223.
[18] Mohd Ehmer Khan and Farmeena Khan. 2012. A comparative study of white box, black box and grey box testing techniques. *International Journal of Advanced Computer Science and Applications* 3, 6 (2012).
[19] Julien Leprince, Henrik Madsen, Clayton Miller, Jaume Palmer Real, Rik van der Vlist, Kaustav Basu, and Wim Zeiler. 2022. Fifty shades of grey: Automated stochastic model identification of building heat dynamics. *Energy and Buildings* 266 (2022), 112095.
[20] Na Luo and Tianzhen Hong. 2022. Ecobee Donate Your Data 1,000 homes in 2017. (3 2022). https://doi.org/10.25584/ecobee/1854924
[21] Miguel Martínez Comesaña, Lara Febrero-Garrido, Francisco Troncoso-Pastoriza, and Javier Martínez-Torres. 2020. Prediction of building's thermal performance using LSTM and MLP neural networks. *Applied Sciences* 10, 21 (2020), 7439.
[22] Leopold Mba, Pierre Meukam, and Alexis Kemajou. 2016. Application of artificial neural network for predicting hourly indoor air temperature and relative humidity in modern building in humid region. *Energy and Buildings* 121 (2016), 32–42.
[23] Giorgio Mustafaraj, Gordon Lowry, and Jie Chen. 2011. Prediction of room temperature and relative humidity by autoregressive linear and nonlinear neural network models for an open office. *Energy and Buildings* 43, 6 (2011), 1452–1460.
[24] SL Patil, HJ Tantau, and VM Salokhe. 2008. Modelling of tropical greenhouse temperature by auto regressive and neural network models. *Biosystems engineering* 99, 3 (2008), 423–431.
[25] Quoc Bao Huy Tran and Sun-Tae Chung. 2020. Smart Thermostat based on Machine Learning and Rule Engine. *Journal of Korea Multimedia Society* 23, 2 (2020), 155–165.
[26] Charalampos Vallianos, Andreas Athienitis, and Benoit Delcroix. 2022. Automatic generation of multi-zone RC models using smart thermostat data from homes. *Energy and Buildings* 277 (2022), 112571.
[27] Paul Vanhaesebrouck, Aurélien Bellet, and Marc Tommasi. 2017. Decentralized Collaborative Learning of Personalized Models over Networks. In *Artificial Intelligence and Statistics (AISTATS)*.
[28] Chengliang Xu, Huanxin Chen, Jiangyu Wang, Yabin Guo, and Yue Yuan. 2019. Improving prediction performance for indoor temperature in public buildings based on a novel deep learning method. *Building and Environment* 148 (2019), 128–135.
[29] Danilo Yu, Abdolreza Abhari, Alan S Fung, Kaamran Raahemifar, and Farahnaz Mohammadi. 2018. Predicting indoor temperature from smart thermostat and weather forecast data. In *Proceedings of the Communications and Networking Symposium*. 1–12.



(a) CL model accuracy in terms of RMSE and MAE, using different temporal abstractions. **(b) Energy consumption of model training in CL, using different temporal abstractions.** **(c) Energy consumption of model training in CL, FL, and P3, without temporal abstraction.**

Figure 3: Accuracy and energy consumption of thermal model training using our proposed temporal abstraction (1 Hour, 30 Min, and 15 Min) compared to traditional training on raw data under CL, FL, and P3.

A IMPACT OF TEMPORAL ABSTRACTION

In this set of experiments, we evaluated the effectiveness of our proposed temporal abstraction in terms of model performance and energy required to train the thermal models. Figures 3a and 3b report the RMSE and MAE accuracy, and the energy consumption of the trained model in CL, using the suggested temporal abstraction scenarios of "1 Hour", "30 Min", "15 Min" and "5 Min", compared to the "RAW" scenario, representing the traditional way of training a model. All abstraction scenarios performed better than "RAW" in terms of RMSE and MAE, and consumed significantly less energy than the "RAW" scenario. The "15 Min" and "30 Min" scenarios resulted in balanced trade-off between accuracy and energy consumption, allowing for accurate and energy efficient thermal model training. Figure 3c presents the total energy consumed when traditionally training (i.e., without temporal abstraction) the thermal models using the three learning schemes of CL, FL and P3. Note that traditional training corresponds to the "RAW" scenario for CL and the "5 Min" scenario for FL and P3, where training is done using the whole dataset, with no temporal abstraction. It appeared that CL required a lot of energy, whereas the P3 algorithm (even without temporal abstraction) could train personalized thermal models in an extremely energy-efficient way, thanks to the usage of low-power mobile devices, and no server involvement. Furthermore, the execution time required to train the thermal models can be significantly reduced when using temporal abstraction. For instance, in the case of P3, model training with 1 Hour abstraction required 93% less time compared to training without abstraction, as described in Table 2.

Temporal abstraction	1 Hour	30 Min	15 Min	5 Min
Execution time (seconds)	629 (7%)	1343 (15%)	2518 (28%)	8821 (100%)

Table 2: Execution time of P3, running for 300 rounds under different temporal abstraction scenarios.

B IMPACT OF HOMES CLUSTERING

Clustering homes into groups of similar homes enables the training of a single thermal model representative of all homes within each cluster. Figure 4 shows the accuracy in terms of RMSE of thermal models trained with and without considering home clustering. Except for the case of the "RAW" scenario in CL where clustering resulted in data redundancy (i.e., overfitting), all experiments in both CL and FL resulted in better models when clustering is considered. In the case of P3, although it is possible to maintain connection with all homes of the cluster, it can be costly in terms of network bandwidth as well as update computation. Thus, we connect homes with a small set of similar neighbors (around 10 homes in our experiments) that is enough to learn a personalized thermal model accurately and efficiently.

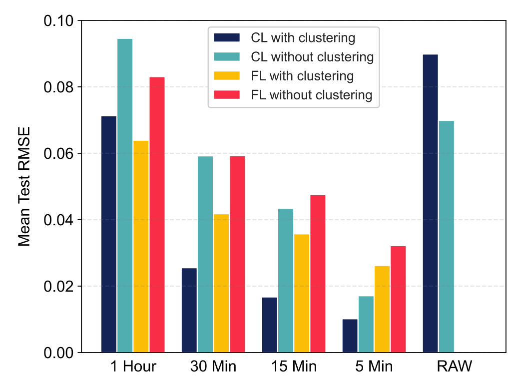


Figure 4: Accuracy in terms of RMSE for CL and FL when training a thermal model using data from all homes, compared to training a thermal model for each cluster of homes.