



HAL
open science

Machine learning driven parameter identification for grey-box thermal modelling for buildings

Quang Hùng Nguyen, Benoit Delinchant, Florent Chatelain

► **To cite this version:**

Quang Hùng Nguyen, Benoit Delinchant, Florent Chatelain. Machine learning driven parameter identification for grey-box thermal modelling for buildings. Building Simulation 2023 - 18th International IBPSA Conference and Exhibition, Sep 2023, Shanghai, China. hal-04244889

HAL Id: hal-04244889

<https://hal.science/hal-04244889>

Submitted on 16 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Machine learning driven parameter identification for grey-box thermal modelling for buildings

Quang Hùng Nguyen^{1,2}, Benoit Delinchant¹, Florent Chatelain²

¹Univ. Grenoble Alpes, CNRS, Grenoble INP*, G2Elab, F-38000 Grenoble, France

²Université Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France

*Institute of Engineering Univ. Grenoble Alpes

Abstract

District-scale energy management solutions require having building models that are both fast and capable of adapting to the reality. This paper presents an approach to improve the speed and flexibility of parametrized models in building thermal modelling, with a case study on the grey-box modelling of an individual house. It consists of performing machine learning (ML) on the parameter identification step, taking into account external variables in the process. The ML model learns how the identified values of the parameters vary according to the changing conditions and uncertainties affecting the building. The trained model is then used to frequently update the RC model with appropriate parameter values during simulation on new data. During validation, this ML-enhanced modelling method provides prediction accuracy levels plausibly comparable to the classical method, with large gains in execution speed, giving it the potential to be deployed to energy controllers of buildings which are less powerful than personal computers.

Highlights

- Perform machine learning on the parameter identification step.
- Fast ML-driven identification on new data, suitable for MPC applications.
- Plausibly comparable prediction accuracy to classical grey-box modelling.

Introduction

The widespread adoption of distributed renewable energies generation and the rise of collective self-consumption practice have generated a push to develop energy flexibility solutions at the district scale. This requires having modelling methods that are not only capable of adapting to changing conditions and uncertainties affecting the building but also faster than detailed physics simulations. The grey-box modelling, also known as resistive-capacitive (RC) modelling, is suitable for this task because it retains the major physical dynamics and can be constructed in a data-driven manner. It relies on the mathematical analogy between heat transfer and electrical transfer to represent the modelled building by a thermal network consisting of thermal resistances, capacitances, and sources. The air within a thermal zone is considered uniform in temperature and is represented by a node. Different models can be named according to the number of resistances and capacitances

(xRyC, e.g. 3R2C), while the order of the model refers to the number of thermal masses represented – which corresponds to the number of capacitances. RC models can be constructed following a **forward approach**, where the network topology and parameter are derived from the building geometry and from properties of construction materials. The **inverse approach** consists of selecting a generic structure then performing parameter identification to find the set of RC parameters that reproduce best the thermal dynamics observed in the data. The structure can be either single zone or multiple zoned, the latter consists of multiple single zone models connected together (Belić, Slišković, and Hocenski 2021). The complexity of zone model can range from 1R1C (1 resistance 1 capacitance) up to 6R4C (Li et al. 2021). The choice of appropriate model order in the single zone case has been discussed by (Reynders, Diriken, and Saelens 2014) : the authors tested multiple RC models from 1st-order to 5th-order on detached single-family houses representative of the Belgian building stock and concluded that models of 2nd to 4th-order gave good results within 1°C RMSE while 5th-order model was less robust due to overfitting.

The inverse approach is data-driven and is applied by the majority of studies involving RC models (Li et al. 2021). It is of greater practical interest because the thermal performance of the building can degrade in long term, or when the building gets renovated, the initial model becomes obsolete. Moreover, in medium term, because of the reduced-physics nature of RC models, external factors that are not taken into account induce biases on the identified parameters of the RC model. These factors can include occupational gains, ventilation rate and angle of solar irradiation. As they change during the year, the induced biases also change, making the initial model lose accuracy when used in another moment of the year. It is therefore necessary to readjust RC parameters regularly by repeating the identification step. To our knowledge, there is no other research work which addresses repeated parameter identification outside of our research team. One issue of repeating parameter identification is the elevated computation cost due to the optimization process. It involves an optimization problem that is often non-convex, for which there are two common solutions: either using a global optimization algorithm or having a multi-start strategy, which is to initialize a gradient method with many initial guesses. (Nguyen-Hong et al. 2017) proposed a meta-optimization approach to reduce the size of the parameter identification problem by fixing RC parameters

with high observed variance to their theoretical values, thus reduces the total computation time.

In this paper, we present a new approach to improve the speed of readjusting RC parameters: perform machine learning (ML) on the identification step in order to update the RC parameters faster and more frequently according to changing conditions in the operational phase of the model. This also adds modelling flexibility by taking into account external factors such as occupation, wind speed, and the moment in the year. Since the ML models are lightweight, they can be deployed to less powerful microprocessors of the energy controllers of the buildings to perform on-the-fly parameter identification and prediction, once the training is done on more powerful machines. This is suitable for distributed architecture of energy flexibility controllers and helps decreasing energy use and investment costs.

Previous studies also treated the coupling of machine learning with a physics-based model (Ellis 2021). Their approach is usually to model the prediction errors of the physics-based component while our approach seeks to maintain the importance of the physics component by modelling the bias induced on the identified parameters.

Method

This part begins by recalling the classic RC modelling workflow then introduces the new modelling method expanded with ML-driven identification.

Classic RC modelling

In the classic RC method used to predict indoor temperature T_{int} , after the model structure is chosen, parameter identification is performed to find the appropriate values for the parameters of the RC model, denoted θ_{RC} , which reproduces the dynamics of T_{int} given the observed data on the input variables of the RC model, as shown in Figure 1.

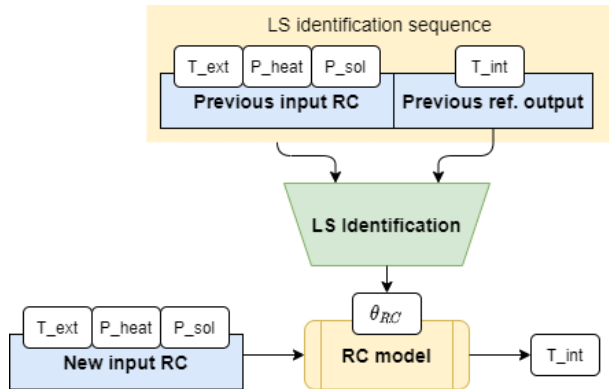


Figure 1 - Classical RC modelling scheme using least-squares (LS) identification

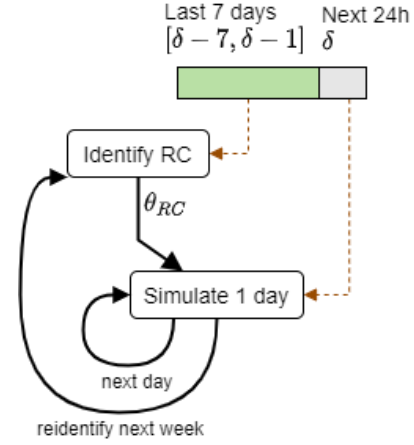


Figure 2 – Classical RC modelling workflow

Expanded method with ML-driven identification

The new method is a two-step process and introduces a metamodel, or identifying model, which will determine the suitable RC parameters to be used according to changing situations. This work will use machine learning (ML) models as the metamodel. Once the metamodel is trained on the results of classic identification, it will replace the role of the optimization process. More specifically, in the first step, a piecewise classic identification by least-squares optimization is performed on a historical data set to produce the θ_{RC} dictionary. This θ_{RC} dictionary is then combined with the external variables to constitute the training data for the metamodel. To increase the robustness of the metamodel, identifications with poor results from the first step can be excluded from the θ_{RC} dictionary.

Let $\theta_{RC,\mu}^{(\delta)}$ be the parameter set of the RC model obtained by method μ and used to model the building during the day δ (μ can be either LS – least squares identification or ML – machine learning model). These parameters can be obtained by identification on the period $t \in T = [t_1, t_2](t_1, t_2 < \delta)$ in the observed data:

$$\begin{aligned} \theta_{RC,LS}^{\delta} &= \operatorname{argmin}(T_{int,pred} - T_{in,ref}) \\ &= f(In_{RC}(t), Out_{RC}(t), v_{ext}(t)) \end{aligned} \quad (1)$$

We want to learn the following function to substitute the identification step in order to better adapt θ_{RC} to the operating conditions:

$$\theta_{RC,ML}^{\delta} = \hat{g}(In_{RC}(t), v_{ext}(t), t), t \in T \quad (2)$$

Figure 3 show the additional information taken into account by the ML-enhanced RC method and Figure 4 summarizes the workflow of the expanded modelling method.

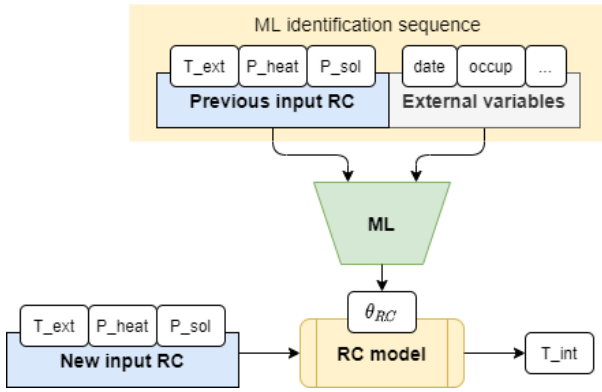


Figure 3 - ML enhanced RC modelling scheme

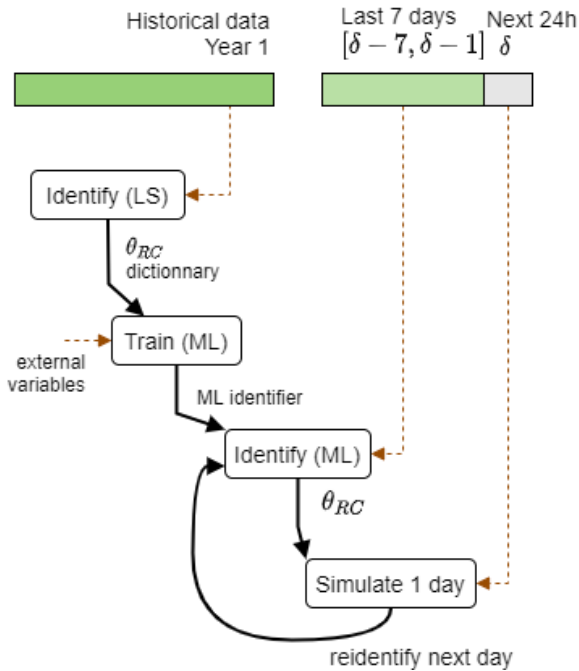


Figure 4 - The proposed ML enhanced modelling workflow

To test the ability of the metamodel to provide appropriate RC parameters according to the operating conditions, it will be compared to classic methods which are purely based on least-squares optimization.

Case study

Subject and reference data

The building to be modelled is a two-story energy positive house in the Parisian suburbs, France.



Figure 5: A 3D render of the studied house

An existing detailed dynamic thermal model of the house in EnergyPlus software has been adapted for this study and its simulation outputs are considered to be the reference data for our study. Usage and HVAC setpoints are scheduled by interval, with the exception of shading by windows which stays closed all the time.

The model is used to produce 2 sets of data, each of 1 year in length:

- Training data (**Year 1**): generated with the climate where the studied house is located (Paris-Orly typical meteorological year (TMY) 1983-1998).
- Validation data (**Year 2**): used to test the extrapolation ability of the ML models, generated using more recent weather data than Year 1 (Paris-Orly TMY 2007-2021, obtained from the <https://www.climate.onebuilding.org/> website (Lawrie and Crawley 2022)).

Some descriptive statistics of the used weather files are shown in Table 1:

Table 1 Characteristics of the used weather files

	Year 1	Year 2
avg. T_{ext} (°C)	11.1	12.2
std. T_{ext} (°C)	6.9	6.9
avg. GHI (W/m ²)	122	135

RC structure and model

For this case study, the single-zone 2R2C structure is chosen for its simplicity while still able to reproduce most of the dynamics of indoor temperature.

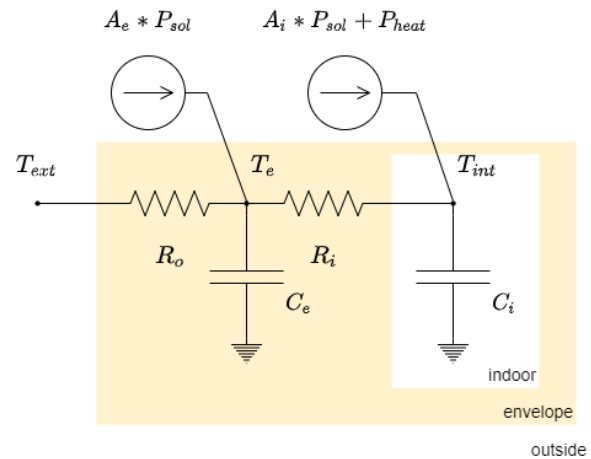


Figure 6- 2R2C thermal network model used

The model has 2 state variables, corresponding to the thermal masses modelled: thermal envelope temperature T_e and indoor temperature T_{int} . The 3 input variables are outdoor temperature T_{ext} , heating power P_{heat} , and solar irradiation P_{sol} . To facilitate parameter identification, other heat fluxes P_{misc} due to occupants' heat gain, electric appliances usage, and ventilation are also given to the model.

The system equations associated to this model are:

$$\begin{cases} C_i dT_{int} = \frac{1}{R_i} (T_e - T_{int}) dt + (P_{heat} + P_{misc}) dt \\ \quad + A_i P_{sol} dt \\ C_e dT_e = \frac{1}{R_i} (T_{int} - T_e) dt + \frac{1}{R_o} (T_{ext} - T_e) dt \\ \quad + A_e P_{sol} dt \end{cases} \quad (4)$$

Model integration is done with the Euler scheme, with a timestep of 2 minutes.

Parameter identification

Identification is done on sliding observation windows of 7 days width, with the stride length of 1 day, with the objective of minimizing squared identification errors on T_{int} over the whole observation window. The parameter vector to be optimized, θ_{optim} , consists of θ_{RC} and the starting condition of state variables, which in this case is $T_{e,init}$ because its initial value has significant influence on the estimated value of RC parameters.

$$\theta_{optim} = [R_i, R_o, C_i, C_e, A_i, A_e, T_{e,init}] \quad (5)$$

The gradient method used is Trust Region Reflective algorithm implemented in the *scipy.optimize* Python package (Virtanen et al. 2020).

The initial guess of θ_{RC} are the same for all identifications and are chosen based on the typical order of magnitude of each parameter. Our tests have shown that this gives overall better results than reusing the previous identification result, $\theta_{RC}^{(\delta-1)}$, for the next day δ . Meanwhile, the initial guess for $T_{e,init}$ (initial envelope temperature at the beginning of the identification sequence) is the average of the initial indoor temperature $T_{int,init}$ and outdoor temperature $T_{ext,init}$. Bounds are applied to keep parameters values within reasonable physical values.

Table 2 – Parameter bounds and initial guesses for optimization

Parameter	Min	Max	Initial guess
R_i (J/K)	5×10^{-4}	5×10^{-2}	1×10^{-2}
R_o (J/K)	5×10^{-4}	5×10^{-2}	1×10^{-2}
C_i (K/W)	1×10^5	5×10^7	1×10^6
C_e (K/W)	1×10^6	5×10^8	1×10^7
A_i (m ²)	0.01	10	3
A_e (m ²)	1	100	10
$T_{e,init}$ (°C)	Between $T_{ext,init}$ and $T_{int,init}$		$(T_{ext,init} + T_{int,init})/2$

Metamodels

2 classes of machine learning model are studied: ridge linear regression (referred in the following sections by the short name “**Ridge**”) and multilayer perceptron (**MLP**). These models are chosen because their simplicity is suitable to the small quantity of data of the learning task. Their input features are a combination of:

- classic variables on $[\delta - 7, \delta - 1]$: same variables as input of RC model, T_{ext}, P_{heat} and P_{sol} . See explanation in the following paragraph.

- external variables or not (respectively marked “**_ev1**” or “**_ev0**” in the short names which will appear later in this paper) on $[\delta - 7, \delta - 1]$, except occupation which is assumed to be known in advance on $[\delta - 6, \delta]$: wind speed and occupation.
- contextual variables : moment in the year, coded by a float on $[-1, +1]$, with -1 being on the moment of new year, close to the middle of winter, and +1 being close to the middle of summer in June.

For the classic variables, instead of giving the whole 3 time series, we decided to give descriptive statistics (mean and hourly total variation). Variants of the input are:

- **cvDay** = give daily statistics of classic variables
- **cvWeek** = give 1 statistics for whole sequence of measures $[\delta - 7, \delta - 1]$
- **cv71** = give daily statistics only for the day before ($\delta-1$) and the same day of last week ($\delta-7$), in order to account for the weekly cycle on input.

The **target variables** are the 6 parameters for our 2R2C model, provided by the identification on Year 1 for the ML model to train on:

$$y = [R_i, R_o, C_i, C_e, A_i, A_e] \quad (6)$$

Prediction loop

Prediction is done in a day-by-day manner, reinitializing $T_{int,RC}$ to match $T_{int,ref}$ each beginning of day and producing 1 day of prediction on each iteration.

We assume that we can anticipate in advance occupation of day δ in the form of occupation schedule in the prediction phase with the ML model.

Since our focus is on evaluating the prediction of temperature, other inputs which are weather data and P_{heat} during the day ahead are considered to be known in advance and are given to RC model.

Prediction of day d is done from $\theta_{RC,d}$ which got identified from measures of $[\delta - 7, \delta - 1]$ inclusive.

Validation

The following methods of estimating and updating θ_{RC} over the prediction loop are compared:

- LS, θ_{RC} static all year based on Year 1 (**LSreuse_year**)
- LS, reuse θ_{RC} from Year 1, piecewise static by week (**LSreuse_week**, chosen to be the baseline)
- LS, reidentify θ_{RC} on-line, piecewise static by week (**LSnew_week**)
- Machine learning methods:
 - Ridge, update daily (**ridge_day**)
 - MLP, update daily (**MLP_day**)

The validation target is indoor temperature prediction accuracy, using mean absolute error (MAE) as the metric. It is the error of the predicted indoor temperature T_{int} from the RC simulation with respect to the reference data.

For **reuseLS_year**, reuse identification results of Year 1 to calculate a single parameter set for prediction in Year2 by averaging:

$$\theta_{RC}^{Year2} = \frac{1}{\text{number of days}} \sum_{\delta} \theta_{RC,LS}^{\delta,Year1} \quad (3)$$

For **reuseLS_week**, reuse the parameter set of the same week in Year 1 to predict in Year 2. Thus each 7 consecutive days in Year2 share the same θ_{RC} .

For **newLS_week**, reidentify θ_{RC} weekly using Year 2 data to apply to the next 7 consecutive days as shown in Figure 4.

Results and discussion

Identification for train set

To increase the amount of data for training, identification is done with sliding window at a step size of 1 day instead of 7 days. The width of the window is 7 days.

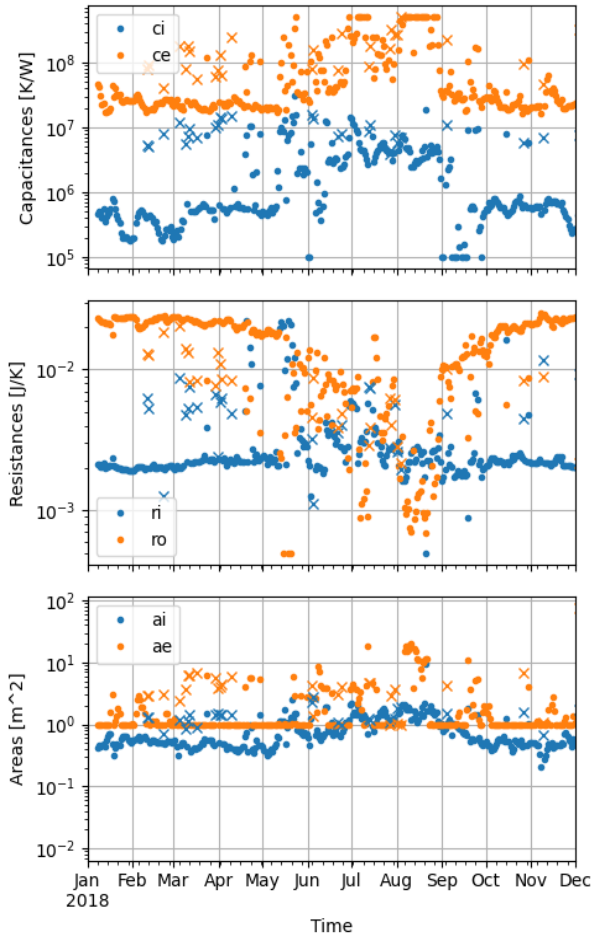


Figure 7 - Identified RC parameters from sliding 7 days windows, presented by pairs of capacitances, resistances and surface areas. Values from rejected identifications are marked as “x”

Parameter sets having unsatisfactory R2 scores of identification, decided to be $R2 \leq 0.7$, are excluded from the subsequent training of the metamodels in order to improve their robustness.

Training of ML models

The training is done in Python with the scikit-learn package (Pedregosa et al. 2011). The 4 target variables R_i, R_o, C_i, C_e are log-transformed, considering their behavior in Figure 7. The input features are standardized.

Since MLP results depend on random initialization, multiple trainings are performed for this type of model with the seed of the pseudorandom generator going from 1 to 15, then the model with the best test score on Year 1 gets selected for subsequent steps.

To train the MLP, the following parameters are seen to produce best results:

- Regularization coefficient alpha = 1.
- Single hidden layer of 8 neurons.
- ReLU activation function. On the other hand, this was seen to produce more aberrant values in θ_{RC} on the validation set. These values were then clipped to be within acceptable bounds for θ_{RC} , as specified in the classic identification step (Table 2), before being used for prediction.

For Ridge regression, setting alpha = 1 was seen to produce better results.

Of the two external variables tested, wind speed is found to be of little importance and was removed from subsequent tests.

Prediction accuracy and speed

The LSnew_week method took over 4 minutes to reidentify 52 θ_{RC} , while the same process for the ML models took negligible time.

All the predicted $T_{int,RC}$ sequences, which are 1 day long, are concatenated then compared with the reference indoor temperature $T_{int,ref}$. Statistics on absolute error (AE) over the whole simulated year are calculated for each tested method: mean absolute error (MAE), standard deviation and maximum AE. Table 3 shows the results for some of the tested methods.

Table 3 - Prediction accuracy of different methods. All units are in °C

	MAE	std. AE	max AE
Ridge_day_cvWeek_ev1	0.89	0.81	4.86
Ridge_day_cvDay_ev1	0.88	0.69	4.36
MLP_day_cvWeek_ev1	1.25	1.48	10.76
LSnew_week	0.91	0.83	5.36
LSreuse_week (baseline)	0.87	0.83	5.89
LSreuse_year	2.79	2.27	8.85

Contrary to expectation, the LSnew_week method (updating the θ_{RC} with newly identified values) performed worse than baseline. This could be explained by the instability of the classic identification step which induces some randomness in the prediction results. Therefore, a performance better than that of LSnew_week but poorer than the baseline method (which is between 0.87 and 0.91) can be considered plausibly desirable, which is the case for the Ridge_day methods shown in Table 3.

The prediction from some of the methods are shown in Figure 8 in comparison with the reference T_{int} over 2 days in March. In this example, reusing θ_{RC} from the previous year delivers the worst result. Reidentifying by least-squares optimization gave a better prediction but fails to reproduce the thermal dynamics of the building on the first day shown. The new method with metamodel

(Ridge_day_cvWeek_ev1) produced the best prediction over the two days shown. On the other hand, the new methods are seen to fail on some other days of the year.

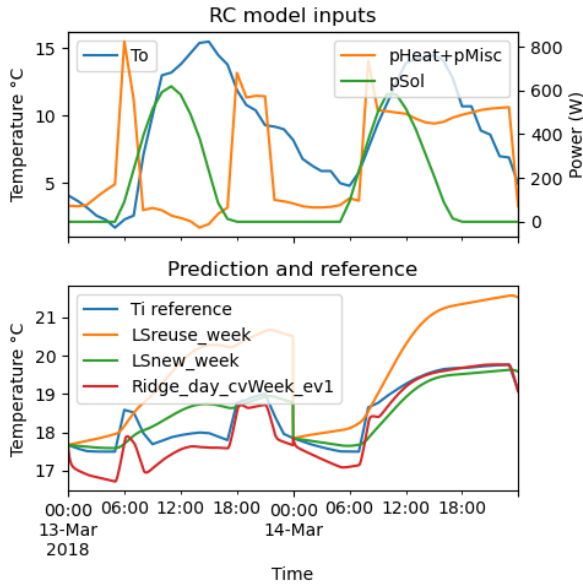


Figure 8 - Predicted and actual indoor air temperature over 2 days of March on validation set. The predictions are reset to actual temperature at midnight because prediction is done in a day by day manner

Using neural networks as the metamodel yielded highly inconsistent results, the extreme case being the MLP with fewer input features (MLP_day_cv71_ev0) attaining a lower MAE than the most informed MLP, as shown in Table 4. This can be due to the random weight initialization which is inherent to neural networks. This might also be explained by the small quantity or richness of training data. The MLP model is therefore not suitable for our purposes.

Table 4 - inconsistent results of MLP methods. All units are in °C

	MAE	std. AE	max AE
MLP_day_cvWeek_ev1	1.25	1.48	10.76
MLP_day_cvDay_ev1	1.06	1.02	12.51
MLP_day_cv71_ev0	0.86	0.70	7.14
LSnew_week	0.91	0.83	5.36
LSreuse_week (baseline)	0.87	0.83	5.89

Although the ML-driven methods did not perform better than the baseline considering MAE, the Ridge methods achieved smaller standard deviation of error and lower maximum error. In other words, the Ridge methods appeared to produce more consistent predictions.

At current configurations, the benefit of taking into account external variables and updating θ_{RC} more frequently are not significant. The identification sequence length, currently chosen to be 7 previous days, can be reconsidered. Performing identification on shorter sequences, e.g. 2 days or even 1 day, can allow capturing faster dynamics in θ_{RC} , notably the usage difference between weekdays and weekends, leading to more prediction accuracy gains from the ML-driven identification. On the other hands, identifying on shorter

windows risks producing less stable identification results. New tests should be performed to examine if this hypothetical benefit outweighs the increased risk.

Limits of the study

The ML identifier depends on the results of classical identification for its training thus it is limited by the weaknesses of this step (e.g. poor choice of identification sequence length). Furthermore, the ML model still does not incorporate knowledge of building physics and of RC model dynamics, limiting its potential.

The reference data also presents some limits: the building usage lacks randomness compared to reality because it is based on a weekly schedule which does not change between Year 1 and Year 2. Moreover, the weather file used are typical meteorological year data instead of normal, non-averaged weather data.

Conclusion

We developed and tested a method to expand grey-box thermal modelling of buildings by modelling the medium term evolution of the RC parameters in a 2 step approach: perform classic parameter identification by optimization then construct a model to predict the appropriate RC parameters, effectively learning the identification task while taking into account external factors. 2 classes of identifying models were used – linear ridge regression and multi-layer perceptron, then compared to more naïve methods of readjusting RC parameters. Firstly, the test results quantified the merit of going from constant RC parameters to piecewise-static for year-long simulations (2.79°C MAE versus <1°C MAE). 2R2C models given by the new ML-driven identification method plausibly attained comparable prediction accuracy as classical identification method (0.89°C for ML-driven versus 0.91°C for classic identification).

More tests should be done to see if these conclusions can still apply to more complex RC model structures, using more realistic reference data. The presented method can also be applied to grey-box modelling of complex systems in other scientific domains where simplifying modelling assumptions and external disturbances are involved.

Nomenclature

θ_{RC} – parameters of the RC model

$\theta_{RC,\mu}^{(\delta)}$ – θ_{RC} obtained by method μ and used to simulate day δ

θ_{optim} – parameters vector to be optimized

LS – least-squares optimization

T_{int} – indoor temperature

T_{ext} – outdoor temperature

T_e – wall (envelope) temperature

T_{int} – indoor temperature

C_i – thermal capacity of indoor air

C_e – thermal capacity of wall (envelope)

R_i, R_o – thermal resistance of wall

A_i, A_e – indoor and outdoor area receiving solar irradiation

P_{sol} – solar irradiation, assimilated with global horizontal irradiation (GHI)

MAE – mean absolute error

RMSE – root-mean-square error

Acknowledgements

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme.

References

Belić, Filip, Dražen Slišković, and Željko Hocenski. 2021. "Detailed Thermodynamic Modeling of Multi-Zone Buildings with Resistive-Capacitive Method." *Energies* 14 (21): 7051. <https://doi.org/10.3390/en14217051>.

Ellis, Matthew J. 2021. "Machine Learning Enhanced Grey-Box Modeling for Building Thermal Modeling." In *2021 American Control Conference (ACC)*, 3927–32. <https://doi.org/10.23919/ACC50511.2021.9482715>.

Lawrie, Linda, and Drury Crawley. 2022. "Development of Global Typical Meteorological Years (TMYx)." 2022. <https://www.climate.onebuilding.org/>.

Li, Yanfei, Zheng O'Neill, Liang Zhang, Jianli Chen, Piljae Im, and Jason DeGraw. 2021. "Grey-Box Modeling and Application for Building Energy Simulations - A Critical Review." *Renewable and Sustainable Energy Reviews* 146 (August): 111174. <https://doi.org/10.1016/j.rser.2021.111174>.

Nguyen-Hong, Quan, Audrey Le Mounier, Van Binh DINH, Benoit Delinchant, Stéphane Ploix, and Wurtz Frederic. 2017. "Meta-Optimization and Scattering Parameters Analysis for Improving On Site Building Model Identification for Optimal Operation." In *Building Simulation 2017*. San Francisco, United States: International building performance simulation association. <https://hal.archives-ouvertes.fr/hal-01395599>.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12 (85): 2825–30.

Rackauckas, Christopher, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. 2020. "Universal Differential Equations for Scientific Machine Learning." *ArXiv:2001.04385 [Cs, Math, q-Bio, Stat]*, August. <http://arxiv.org/abs/2001.04385>.

Reynders, G., J. Diriken, and D. Saelens. 2014. "Quality of Grey-Box Models and Identified Parameters as Function of the Accuracy of Input and Observation

Signals." *Energy and Buildings* 82 (October): 263–74. <https://doi.org/10.1016/j.enbuild.2014.07.025>.

Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, et al. 2020. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python." *Nature Methods* 17 (3): 261–72. <https://doi.org/10.1038/s41592-019-0686-2>.