



HAL
open science

End-to-end deep meta modelling to calibrate and optimize energy consumption and comfort

Max Cohen, Sylvain Le Corff, Maurice Charbit, Marius Preda, Gilles Nozière

► **To cite this version:**

Max Cohen, Sylvain Le Corff, Maurice Charbit, Marius Preda, Gilles Nozière. End-to-end deep meta modelling to calibrate and optimize energy consumption and comfort. 2021. hal-02873577v2

HAL Id: hal-02873577

<https://hal.science/hal-02873577v2>

Preprint submitted on 30 Jan 2021 (v2), last revised 4 Nov 2021 (v3)

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.

End-to-end deep meta modelling to calibrate and optimize energy consumption and comfort.

Max Cohen¹, Sylvain Le Corff¹, Maurice Charbit², Alain Champagne², Marius Preda³, and Gilles Nozière²

¹Samovar, Télécom SudParis, Département CITI, TIPIC, Institut Polytechnique de Paris.

²Oze Énergies.

³Samovar, Télécom SudParis, Département ARTEMIS, ARMEDIA, Institut Polytechnique de Paris.

Abstract

In this paper, we propose a new end-to-end methodology to optimize the energy performance as well as comfort and air quality in large buildings without any renovation work. We introduce a metamodel based on recurrent neural networks and trained to predict the behavior of a general class of buildings using a database sampled from a simulation program. This metamodel is then deployed in different frameworks and its parameters are calibrated using the specific data of two real buildings. Parameters are estimated by comparing the predictions of the metamodel with real data obtained from sensors using the CMA-ES algorithm, a derivative free optimization procedure. Then, energy consumptions are optimized while maintaining a target thermal comfort and air quality, using the NSGA-II multi-objective optimization procedure. The numerical experiments illustrate how this metamodel ensures a significant gain in energy efficiency, up to almost 10%, while being computationally much more appealing than numerical models and flexible enough to be adapted to several types of buildings.

Keyword: Recurrent neural networks; surrogate models; Building Energy Model; Building energy optimization.

1 Introduction

In 2009, the building industry accounted for over 40% of the total French energy consumption, as well as almost a quarter of greenhouse emissions. In the ecological context where energy waste cannot be ignored, improving the energy efficiency of the building industry is an important step. This translated in an objective of 38% reduction in the consumption of the building industry for 2020, through the renovation of 400 000 apartments per year, raised to 500 000 six years later. However, despite setting higher and higher objectives toward renovating the French building park, the actions carried out to date still fall short in terms of results, as stated by the Ademe (Agency for the environment and energy). According to the National Low-Carbon Strategy (SNBC), the average number of yearly renovations is expected to be around 370 000 for the period 2015-2030.

The aim of this paper is to provide optimal building management settings governing Heating, Ventilation and Air-conditioning (HVAC) in order to improve thermal comfort and optimize energy

consumption, without costly, invasive or time consuming renovation works. Global energy demand for heating, ventilation and air-conditioning in commercial or public buildings has been increasing rapidly for the past few decades. This rising demand is at the root of the complex problem of simultaneously maintaining a satisfactory comfort in buildings (air and indoor temperature quality) and reducing the environmental impact. This makes the analysis of building energy performance a challenging multi-criteria problem.

This analysis is decomposed into three steps: (i) designing a model to predict future energy loads and indoor temperature based on the HVAC system and weather data, (ii) calibrating the unknown parameters of this model with real data obtained from sensors in each building and (iii) optimizing the HVAC settings to minimize the total energy consumption in future periods while maintaining a given thermal comfort. Step (i) can be achieved with simulators that describe heat transfers between the building and its environment such as EnergyPlus, TRNSYS or DOE-2. They predict the indoor temperatures and consumptions from the description of the building, HVAC settings and weather data. However, their accuracy relies heavily on the precision of the building description: window to wall ratio, exposition, thermal conductivity or heat capacity among others. Because our strategy does not involve renovation works on the building site and use a simplified building description we usually can only provide rough estimates of these input parameters. The calibration procedure in step (ii) aims at obtaining a parameter estimate associated with accurate indoor temperatures and consumptions predictions. We measure the discrepancy between model predictions and real data with a loss function that is minimized in an iterative process. Because these methods require tremendous number of calls to the modelling function, the calibration step quickly becomes excessively time consuming. One popular alternative in the building optimization literature consists in replacing such simulators by surrogate models.

Metamodelling approaches substitute the physical simulator with a much faster surrogate model during calibration and optimization tasks. This surrogate model is trained on a dataset of simulations conducted by the simulator, that aims at exhaustively capturing building behaviors for various building geometries and management settings. The most recent statistical models make this step more and more relevant in our context, as their accuracy reaches new heights while their computational time is constantly reduced. Yet, most approaches only implement the most simple architectures for building optimization.

In this paper, we first propose to train a metamodel based on Recurrent Neural Networks (RNN). We propose a comparison of several approaches which illustrates that sequence to sequence models, such as RNN, can yield a significant increase in performances with respect to the alternatives previously considered in such settings. In addition, we propose to compare such RNN to attention-based models which are the go-to architectures in other fields to predict sequences with complex dependencies. This study allows to introduce a new metamodel, outperforming standard alternatives used in building optimization. This metamodel, which depends on a few physical parameters, is then calibrated using the CMA-ES optimization procedure and real data to provide accurate predictions for various types of buildings. Two real buildings were used to illustrate the flexibility of this approach. The final step of our end-to-end methodology consists in optimizing energy loads, while maintaining a given level of comfort. This multi-criteria optimization problem requires determining an optimal compromise between consumption and comfort, as improving one objective results in degrading the other. This optimization is achieved through an iterative algorithm, justifying our choice of replacing simulators with a faster surrogate model. Following this methodology, we were able to train and calibrate our metamodel and to reduce the hourly consumption of two buildings by 5% and 10%.

2 Related works

2.1 Physical simulator

Modelling building behavior is traditionally achieved through models based on physical equations that describe heat transfer between the building and its environment. The most common softwares available, EnergyPlus, TRNSYS or DOE-2, are used to simulate the system behavior based on a schematic view of the building. EnergyPlus was used for instance in [Shabunko et al., 2018] to build three types of typical designs and to benchmark the energy performance of 400 residential buildings. In [Zhao et al., 2016], the authors proposed a predictive control framework based on Matlab and EnergyPlus in order to optimize energy consumptions while meeting the individual thermal comfort preference. In [Magnier and Haghighat, 2010], the authors highlight the performances of TRNSYS as a physical simulator, as well as its limits in terms of computational speed: the authors argue that a full optimization process would take as much as ten years, had they not replaced TRNSYS with a surrogate model during optimization. The authors of [Bre et al., 2016] studied the optimization of a single-family house using a combination of Energy-Plus and the NSGA-II optimization algorithm, and discussed sensitivity analysis using the Morris screening method. Likewise, the authors of [Recht et al., 2014] performed sensitivity and uncertainty analysis on another building simulator known as COMFIE, and displayed its modelling performances on a passive building.

In these papers, a schematic building is used in the simulation program and considered as a baseline for energy loads. This demonstrates the capability of such approaches in producing benchmarks but no comparison to real data obtained from sensors are provided which makes such results not suitable in a decision process involving data from sensors in real buildings.

2.2 Surrogate models

The building optimization literature has seen an increasing number of surrogate approaches, as recent sophisticated statistical models provide appealing solutions to be used in this context. In [Bre et al., 2020, Reynolds et al., 2018], statistical models are trained on a dataset sampled from EnergyPlus, allowing significant computational savings during optimization. In [Bre et al., 2020], the authors proposed to combine NSGA-II with an artificial neural network metamodel, here a Feed Forward Network (FFN), in order to optimize the consumption of a 83 m² house. Optimization was also conducted with the original building simulator, EnergyPlus, in order to compare both results and ensure that the FNN could be used as a substitute during optimization. Similarly, [Reynolds et al., 2018] proposed a FFN based meta modelling approach to reduce up to 25% the energy consumption in a small office building. EnergyPlus was used to sample a dataset for various zones of the building. The metamodel was tested using a 4-week long EnergyPlus simulation with variable set point temperatures and using an alternative weather file. An example of recurrent neural architecture as a surrogate model can be found in [Ohta et al., 2020], where the authors focused on an air-conditioning optimization problem using time series.

If these articles justify the use of metamodels, the question of which type of model to choose remains. In an in-depth review, [Roman et al., 2020] compares standard statistical models, such as polynomial regression, multivariate adaptive regression splines, Gaussian processes or Decision Trees, in the context of building performance simulation. Artificial Neural Networks models stand out as a particularly relevant alternative, but are often presented in their most simple, time independent form, such as the FFN used in [Bre et al., 2020]. Although they may yield accurate predictions in some frameworks, these neural networks are not adapted to time series problems,

and are usually substituted for their sequential counterparts, such as recurrent or convolutional based approaches. Therefore, designing metamodels for building calibration and optimization is likely to benefit from such recurrent and attention-based models.

Recurrent Neural Network (RNN) were first introduced as a more suited architecture for dealing with time varying input patterns [Mozer, 1989]. By replacing buffer based approaches with an updated context state, RNN are able to solve time series problems with short time dependencies, but are lackluster in problems requiring long term memory due to vanishing and exploding gradient [Bengio et al., 1994]. Long Short Term Memory proposed in [Hochreiter and Schmidhuber, 1997] aim at bridging that gap by enforcing error flow throughout time in the network. The LSTM architecture was modified in [Cho et al., 2014] in order to simplify its implementation and improve computation times, resulting in a novel model called Gated Recurrent Unit (GRU). In parallel to these advances on recurrent architectures, Convolutional Neural Networks (CNN), rendered popular by [Krizhevsky et al., 2012] for image classification, have been adapted to time series problem. The approaches proposed in [Józefowicz et al., 2016, Kim et al., 2016] outperformed traditional Natural Language Processing (NLP) models by replacing the embedding layer with a character-level convolutional layer.

Recurrent and convolutional approaches coincide in that temporally close time steps data are matched together. In 2017, [Vaswani et al., 2017] proposed an attention based approach to solving NLP tasks that consider the entire input sequence in parallel. The Transformer model is based on a self-attention mechanism, that computes an attention value for every element of a sequence with respect to all others to model their dependency. This attention mechanism allows to understand at each time step which input elements are crucial to predicting the new state. This makes these networks more interpretable than their most widely-used recurrent counterparts such as LSTM or GRU networks and motivate a keen interest for such approach to predict complex time series.

2.3 Calibration

Approaches based on building behavior simulation rely on many unknown physical parameters. Instead of costly campaigns to measure these parameters, that would have to be reiterated for each new building, they may be estimated using an automatic calibration procedure by minimizing a cost function which associates, with each set of parameters, the discrepancy between the simulations and the true energy loads and temperatures, see [Coakley et al., 2014, Le Corff et al., 2018]. As shown in [Nagpal et al., 2019], calibration yields sufficiently accurate results for a variety of different buildings. This ensures limited additional costs to fit a trained metamodel to new buildings. In many related works, this problem cannot be solved since no real data are used in the calibration step, see [Bre et al., 2020, Reynolds et al., 2018], i.e. the calibration is performed based only on simulated data. Optimization is thus conducted without justifying that the theoretical energy savings could be applied to any real building.

The calibration task revolves around a non differentiable optimization problem, which is often tackled by using genetic optimization methods. In [Aird et al., 2016], the authors demonstrate the use of the Non-dominated Sorting Genetic Algorithm II (NSGA-II) to select a set of estimated parameters that jointly minimize the coefficient of variation of the root mean square error, and the normalized mean bias error. All criteria can instead be combined in a single calibration error, in order to turn to single objective differentiation free algorithms that offer a single best candidate, avoiding the need for further selection processes. In [Le Corff et al., 2018], the CMA-ES algorithm ([Igel et al., 2007]) minimizes a combination of heating and cooling errors.

2.4 Our approach

In this paper, we propose an end-to-end methodology, from dataset sampling to metamodel calibration and optimization using data obtained from wireless sensors set in large buildings. We introduce a new metamodel to predict building behaviors after a comprehensive study of several approaches from traditional RNN to a model based on a Transformer architecture [Vaswani et al., 2017].

Once the metamodel is trained using a dataset built using TRNSYS, all the parameters of real buildings and of their Building Management System (BMS) are estimated using real measurements with the CMA-ES algorithm [Igel et al., 2007]. A multi-objective methodology to improve energy efficiency and maintain thermal comfort is then implemented by acting only on the BMS. The NSGA-II approach is used to obtain the Pareto optimal parameters. The performance of this metamodel are compared both in terms of accuracy and computational efficiency with TRNSYS. The paper is organized as follows. Section 3 provides all the deep learning architectures used in this paper to build a metamodel. It also describes the data and variables used in our metamodel. Section 4 illustrates the performance of our metamodel in the calibration and optimization processes for three real buildings. The numerical experiments illustrate how the same metamodel ensures a significant gain in energy in various settings in comparison to the considered alternatives.

3 Meta modelling

3.1 Notations

Let $(Y_k)_{k \geq 0}$ be the state of the building i.e. the inside temperatures and the consumptions of the building management system. The index k denotes time where, in the setting of this paper, data are collected every hour. The aim of the metamodel introduced in this paper is to provide a numerically efficient solution to predict $(Y_k)_{k \geq 0}$ from several sets of input variables.

The parameter θ_{build} represents all unknown parameters regarding the geometrical description of the buildings (windows area ratio, etc.), as well as parameters related to heat transfer (heat capacity, infiltration rate, etc.). Input time series are divided in three variables $(W_k, O_k, I_k)_{k \geq 0}$. The input W_k contains weather data at time k , such as outdoor temperature, relative humidity or irradiance values; O_k is a single value time series representing whether the building is occupied at time step k ; I_k stores all information relative to the building usage: activation hours of the AC and ventilation systems, comfort and reduced temperatures (thermal objectives when the building is occupied or empty). In this section, we describe how a simulator may be used to train the metamodel which aims at mimicking its outputs for various choices of θ_{build} , $(I_k)_{k \geq 0}$, $(O_k)_{k \geq 0}$ and of meteorological data $(W_k)_{k \geq 0}$. The appendix displays a complete list of the variables contained in θ_{build} , $(I_k)_{k \geq 0}$, $(O_k)_{k \geq 0}$ and in $(W_k)_{k \geq 0}$ for the numerical experiments of this paper.

3.2 Proposed benchmarks

In most recent works, a great deal of research activities focused on FFN as surrogate models, see [Bre et al., 2020, Magnier and Haghghat, 2010, Reynolds et al., 2018]. Although they may lead to interesting performance during the training phase, these fully connected architectures are not well suited for time series prediction, in particular for long time spans. We seized this opportunity to explore other approaches that have proven to be more relevant for solving time series tasks in the past few years. Therefore, we decided to evaluate the go-to architectures for time series:

a bidirectional LSTM, a bidirectional GRU (BiGRU), a hybrid model mixing both convolutional and GRU layers (ConvGru), and a Feed Forward Network (FFN) as used in previous works. In addition to those models, a Transformer model, which introduces an attention mechanism to model dependencies, is also considered. These models have been implemented using the deep learning framework PyTorch.

3.3 Our proposed metamodel

Our metamodel is a function $f_{\theta_{\text{meta}}} : (h_{k-1}, u_k) \mapsto y_k$ with parameters θ_{meta} that maps, at each time step k , the building state $u_k \equiv (W_k, O_k, I_k, \theta_{\text{build}})$ and a hidden state h_{k-1} depending on the past values (u_1, \dots, u_{k-1}) , to a prediction of its indoor temperature and consumptions y_k . The model is trained to produce accurate predictions by tuning its parameters θ_{meta} , usually referred to as weights, through an iterative back propagation algorithm, where predictions y_k are compared to the ground truth Y_k .

We use as a backbone a many to many RNN architecture, and denote by h_k^ℓ and x_k^ℓ the hidden state and input of layer $1 \leq \ell \leq L$ at time step k , in particular $x_k^0 \equiv u_k$. The hidden state is traditionally initialized as the zero vector, $h_0^\ell \equiv 0$ for all $1 \leq \ell \leq L$.

In the original and most simple definition of a RNN, the hidden state is computed recursively as $h_k^\ell = \tanh(W_{ih}^\ell x_k^\ell + W_{hh}^\ell h_{k-1}^\ell + b_h^\ell)$, where W_{ih} , W_{hh} and b_h are the weight matrices and bias learned during training, and initialized with random values. Our metamodel is based on a LSTM architecture and replaces the update of the hidden state by the following state equations:

$$\begin{aligned} \Gamma_{u,k}^\ell &= \sigma(W_{iu}^\ell x_k^\ell + W_{hu}^\ell h_{k-1}^\ell + b_u^\ell), \\ \Gamma_{f,k}^\ell &= \sigma(W_{if}^\ell x_k^\ell + W_{hf}^\ell h_{k-1}^\ell + b_f^\ell), \\ \Gamma_{o,k}^\ell &= \sigma(W_{io}^\ell x_k^\ell + W_{ho}^\ell h_{k-1}^\ell + b_o^\ell), \\ \tilde{c}_k &= \tanh(W_{ic}^\ell x_k^\ell + W_{hc}^\ell h_{k-1}^\ell + b_c^\ell), \\ c_k^\ell &= \Gamma_{f,k}^\ell * c_{k-1}^\ell + \Gamma_{u,k}^\ell * \tilde{c}_k. \\ h_k^\ell &= \Gamma_{o,k}^\ell * \tanh c_k^\ell. \end{aligned}$$

An additional fully connected layer is added on top of the RNN architecture:

$$y_k = \sigma(W_y h_k^L + b_y),$$

where σ is the sigmoid activation function $\sigma : x \mapsto (1 + e^{-x})^{-1}$.

The architecture is represented in Figure 1. The parameters to be estimated during the training phase of the metamodel are

$$\theta_{\text{meta}} = \left\{ (W_{iu}^\ell, W_{hu}^\ell, W_{if}^\ell, W_{hf}^\ell, W_{io}^\ell, W_{ho}^\ell, W_{ic}^\ell, W_{hc}^\ell, W_y, b_u^\ell, b_f^\ell, b_o^\ell, b_c^\ell, b_y)_{1 \leq \ell \leq L} \right\}.$$

3.4 Dataset sampling

The training dataset is sampled by exploring the input space of the simulator. We chose TRNSYS in this paper but any simulator can be used to train the metamodel. We define ranges for each input parameters in θ_{build} , $(I_k)_{k \geq 0}$ and $(O_k)_{k \geq 0}$ with the help of energy managers, such as highest and lowest scheduled temperature, or the most early and late hour of arrival of occupants, see the

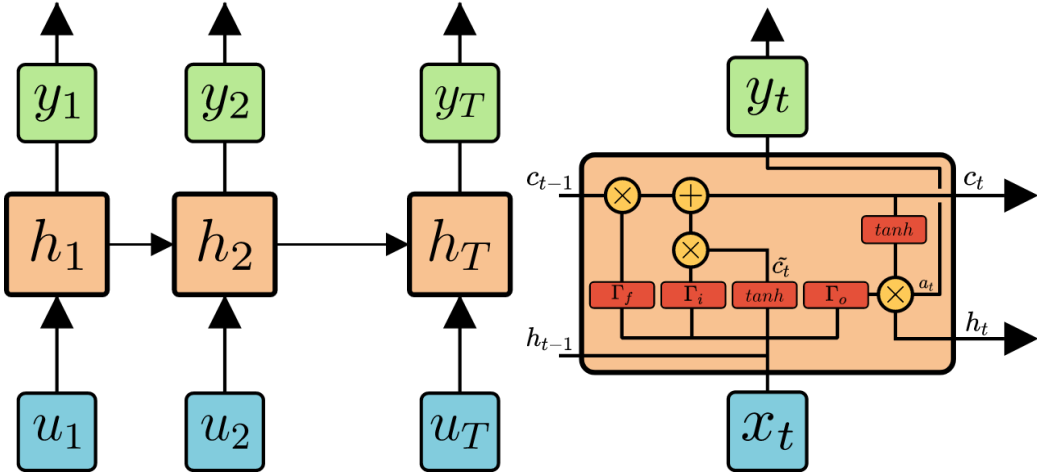


Figure 1: Our metamodel architecture (left), and a detailed LSTM cell (right).

appendices for a complete list of these ranges. Because our dataset encompasses multiple buildings, these ranges are not centered around a specific set of parameters, but rather cover all possible values across our cluster of buildings. In addition, real weather data $(W_k)_{k \geq 0}$ acquired between June and August 2020 were used to obtain a dataset consistent with the real buildings. In our numerical experiments, we chose a uniform sampling method over the ranges for each variable. This allows us to easily split the dataset uniformly into training and testing sets, which is crucial to validate the metamodel.

The input vector u_k contains 27 variables at each time step: 7 variables from θ_{build} , 7 from W_k , 1 from O_k and 12 from I_k . A total of 15,000 training examples were sampled, an example being a month i.e. 672 hours. During the training phase, the parameters of each metamodel described in Section 3 are estimated based on this dataset (called θ_{meta} in the detailed case of the RNN approach). The metamodels compared in this section are defined with a latent dimension of $d_{\text{emb}} = 64$ and a total of $L = 4$ layers. Hyper parameters, such as learning rate, dropout, number of epochs or batch size, were chosen empirically.

3.5 Training and validation

During training, for each example, we compute the Mean Squared Error (MSE) loss, and combine consumption and temperature errors:

$$\text{MSE}_T^{\theta_{\text{meta}}} = \frac{1}{N} \sum_{k=1}^N (\hat{T}_k^{\theta_{\text{meta}}} - T_k)^2 \quad \text{and} \quad \text{MSE}_Q^{\theta_{\text{meta}}} = \frac{1}{N} \sum_{k=1}^N (\hat{Q}_k^{\theta_{\text{meta}}} - Q_k)^2$$

$$\text{loss}(\theta_{\text{meta}}) = \beta \text{MSE}_T^{\theta_{\text{meta}}} + (1 - \beta) \text{MSE}_Q^{\theta_{\text{meta}}},$$

where N is the number of data in each example, T_k and Q_k are the ground truth at time k , and $\hat{T}_k^{\theta_{\text{meta}}}$ and $\hat{Q}_k^{\theta_{\text{meta}}}$ are the predictions given by the metamodel with the current value θ_{meta} of the metamodel for temperature and consumption respectively. In the experiments below, as the inside temperatures and all consumptions are normalized, we chose the non informative value $\beta = 0.5$.

Table 1: Metrics (means and standard deviations) of the metamodels on the validation dataset. The best mean values are displayed in bold (the lowest losses and mean squared errors). Time is the computation time to run a single simulation through the network, and was estimated by averaging 100 inferences.

	BiGRU	Transformer	Ours	ConvGru	FFN
Loss ($\times 10^{-4}$)	2.05 (1.61)	2.72 (2.88)	1.65 (1.47)	2.26 (2.04)	90.5 (41.1)
MSE _T ($\times 10^{-5}$)	1.00 (1.53)	1.49 (2.75)	0.820 (1.44)	1.20 (2.05)	39.5 (39.7)
MSE _Q ($\times 10^{-4}$)	3.84 (2.05)	4.12 (3.01)	2.75 (1.72)	3.53 (1.77)	172 (60.2)
MSE _T ^{occ} ($\times 10^{-5}$)	4.60 (7.16)	6.56 (12.2)	3.79 (6.81)	5.89 (9.91)	176 (189)
MSE _Q ^{occ} ($\times 10^{-4}$)	1.45 (1.00)	2.07 (1.80)	1.22 (0.878)	1.85 (1.23)	113 (50.8)
Δ_Q^{Tot} ($\times 10^{-3}$)	4.03 (11.7)	20.1 (12.4)	2.46 (12.0)	14.9 (16.2)	1.82 (68.8)
Time ($\times 10^{-2}s$)	6.46	4.52	6.51	6.77	0.341

We chose the Adam optimizer [Kingma and Ba, 2015] and all simulations were computed on a single 1080TI GPU card. Table 1 displays the mean values and standard deviations of the loss function on the validation dataset after training. The table also displays the mean squared error MSE_T (resp. MSE_Q) on the temperatures (resp. consumptions) only, as well as these same metrics computed only during occupation time: MSE_T^{occ} and MSE_Q^{occ}. For a global consumption evaluation, we compute the absolute relative error on the cumulative consumptions Δ_Q^{Tot} .

4 Energy Optimization in real buildings

The experiments conducted in our paper to analyze the performance of the trained metamodel focused on the optimization of two buildings in the Parisian region. Each one is represented by a single thermal zone.

- Stanley is a 18,512 m^2 building. It is delimited by four vertical walls of dimension 2,314 m^2 , 1,917 m^2 , 2,123 m^2 and 1,725 m^2 , as well as a roof and ground of dimension 2,304 m^2 . The main insulator is a 10 *cm* layer of polystyrene. It was built in 1983.
- Livingstone is a 13,594 m^2 building insulated with a 8 *cm* layer of polyurethane, including 4 vertical walls with respective areas 1,678 m^2 , 1,274 m^2 , 1,281 m^2 and 1,252 m^2 , a horizontal roof and a horizontal ground of dimension 4,653 m^2 and 4,286 m^2 . It was built in 2006.

Based on a commonly used rule, it is assumed that 2/3 of the full area is occupied by people. Assuming that each occupant requires 12 m^2 , this allows to set the initial values for the number of occupants and the number of PCs (set to 1.2 times this value) in the building during occupancy hours. These values are assumed to be known and fixed and used to sample the training dataset.

4.1 Calibration

During the training phase, metamodel parameters are estimated by minimizing the loss function on the simulated dataset which corresponds to various choices of θ_{build} and $(I_k, O_k, W_k)_{k \geq 0}$, associated with building behaviors $(Y_k)_{k \geq 0}$. Because this dataset is sampled from a simulation model, we train

the metamodel ignoring real building related noise and measurement errors. Additionally, both the BEM and our surrogate model require θ_{build} that cannot be properly identified for each building especially without renovation work. By comparing the metamodel predictions to real historic data during the calibration phase, we search for a set of building related parameters that best match reality.

During this step, the weights θ_{meta} of the metamodel are frozen, meaning that we no longer update each weight matrix of the neural network. Using the coefficient of determination as a cost function, we can compute, for each given set of input parameters θ_{build} and $(I_k, O_k, W_k)_{k \geq 0}$, the difference between estimated and real historical data. Because this is a non differentiable problem, the cost function cannot be minimized using a stochastic gradient descent algorithm as in the training step; instead we use the CMA-ES algorithm [Hansen et al., 2003], an evolutionary algorithm designed to solve constrained non-convex optimisation problems. In our experiments, the variables we adjust are constrained in a realistic range of values, advised by energy managers. In our experiments, the variables we adjust for fitting are constrained by the same ranges defined in the data sampling section. The algorithm is implemented by the author of the paper in the pycma library¹.

Following traditional methodology in building calibration, we measure the performances of the calibrated model with the Mean Bias Error (MBE) and Coefficient of variation of the Root Mean Square Error ($Cv(\text{RMSE})$) criteria. For any sequence $(z_k)_{1 \leq k \leq M}$ associated with predictions $(\hat{z}_k)_{1 \leq k \leq M}$, these quantities are defined as follows:

$$\begin{aligned} \text{MBE}(\%) &= 100 \frac{\sum_{k=1}^M (z_k - \hat{z}_k)}{\sum_{k=1}^M z_k}, & Cv(\text{RMSE})(\%) &= 100 \frac{\text{RMSE}}{\bar{z}}, \\ \text{RMSE} &= \left(\frac{\sum_{k=1}^M (z_k - \hat{z}_k)^2}{M} \right)^{1/2}, & \bar{z} &= \frac{1}{M} \sum_{k=1}^M z_k \end{aligned}$$

where M is the number of data in each example. In a detailed review of calibration methods, the authors of [Fabrizio and Monetti, 2015] have gathered the international recommended ranges regarding these criteria, when validating a calibrated model. Regardless of the simulation program, the $Cv(\text{RMSE})$ should fall within $\pm 20\%$, and the MBE $\pm 5\%$ when considering hourly calibrations. As shown in Table 2, our results for both consumptions and indoor temperatures calibration are well within these guidelines.

Calibration was run for both the metamodel and the original BEM (TRNSYS) for a maximum of 3 hours. As shown in Table 2, we can achieve satisfactory results for Stanley in this timespan, as both model converge to close values for both the $Cv(\text{RMSE})$ and MBE. Figure 3 displays both models calibration results, compared to real data. On the other hand, TRNSYS calibration of Livingstone is sensibly below the results obtained with the metamodel, as calibration did not converge in the available time, see Figure 2. The calibration of the metamodel reached convergence but with a tremendous number of epochs, that would have required to run TRNSYS for about 10 hours in order to get similar performances. As a comparison, we calibrated the metamodel for the same number of epochs as TRNSYS, and obtained similar results. This experiment comforts the idea that TRNSYS and the metamodel behave similarly after the calibration step, but the much shorter computation time of the metamodel allows us to better calibrate complex buildings, such

¹<https://github.com/CMA-ES/pycma>

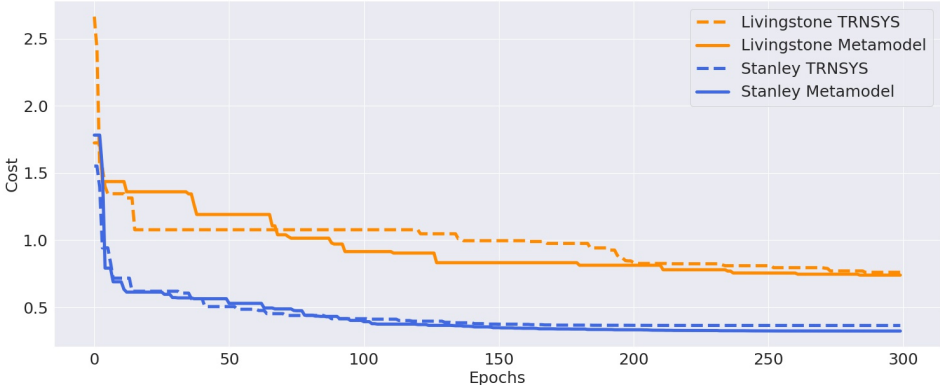


Figure 2: Calibration cost evolution for the metamodel and TRNSYS, on Livingstone and Stanley. Both models were calibrated for 300 epochs, which is enough to reach convergence for Stanley, but not Livingstone.

as Livingstone. See Figure 4 for a visualization of the TRNSYS and metamodel calibration after one hour.

Table 2: Calibration metrics for Stanley and Livingstone buildings.

	MBE_Q	$Cv(RMSE)_Q$	MBE_T	$Cv(RMSE)_T$	Iterations	Computational time
Stanley						
Metamodel	-0.627	11.0	0.134	1.20	300	2mn
TRNSYS	-0.409	12.1	-0.264	1.24	300	3h
Livingstone						
Metamodel	-0.690	14.2	-0.0551	1.29	10000	1h
Metamodel	-0.574	14.2	-0.413	1.95	300	2mn
TRNSYS	-1.08	15.8	0.156	1.96	300	3h

Validation The metamodel will assist the decision process for building management by simulating thermal behavior of future weeks. Because the calibration process requires real data, the metamodel is calibrated on several past weeks, in order to capture the real building behavior in a situation as close as possible to the future period we aim to match. We validate the calibration phase using two successive weeks, by applying the calibrated settings to the two following weeks, with fresh weather data, and compare the results to the true observed values. The results are displayed in see Figure 5 for Stanley building.

4.2 Optimization

Once the metamodel is calibrated, we can use it as an accurate simulator for how the building will react to changes in its usage. After a successful calibration, all building related variables

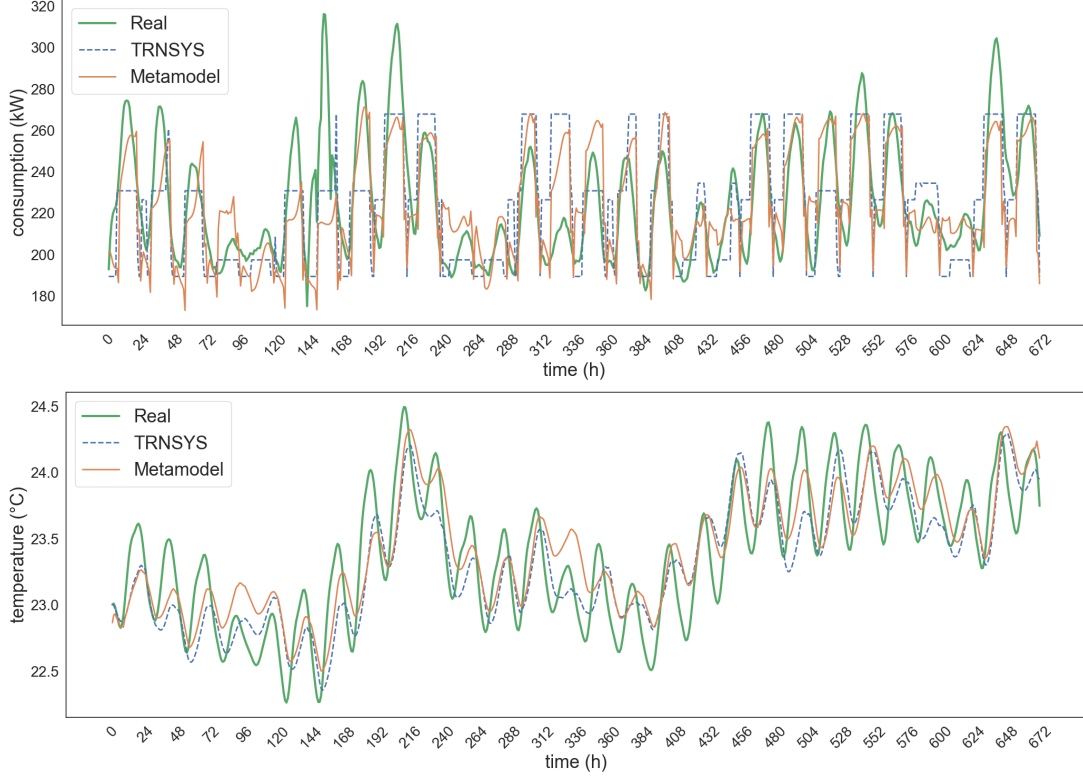


Figure 3: Consumption and temperature simulations after calibration, for both the metamodel and TRNSYS, for Stanley.

contained in θ_{build} are correctly estimated. The parameters I_k associated with the HVAC system can be optimized for a given set of weather data W_k . The optimization task consists in finding a set of usage related parameters that reduce consumption while keeping the same level of comfort. Optimizing energy consumption requires minimizing two conflicting objectives, making it impossible to find a solution that optimizes both objectives simultaneously. Instead, we search for optimal compromises between energy consumption and comfort, in the form of a Pareto front. Indeed, for any such optimal compromise, we can always get a higher level of comfort, for the price of a higher consumption. The consumption criteria is the energy load during the month; the comfort criteria is the gap between indoor temperature and a constant reference temperature T^* :

$$\text{Comf} = \frac{1}{N_{\text{Occ}}} \left(\sum_{k=1}^{N_{\text{Occ}}} \mathbf{1}_{k \in \text{Occ}} (\hat{T}_k - T^*)^2 \right)^{1/2} \quad \text{and} \quad \bar{Q} = \frac{1}{N} \sum_{k=1}^N \hat{Q}_k,$$

where $T^* = 22.5^\circ\text{C}$, N^{opt} is the number of hours to be considered in the optimization process and Occ is a subset of daytime hours specifying at which hours the target temperature has to be reached in the building. Following recent works in building energy optimization, we search for a set of optimal parameters using NSGA-II ([Deb et al., 2000]), another evolutionary algorithm, but

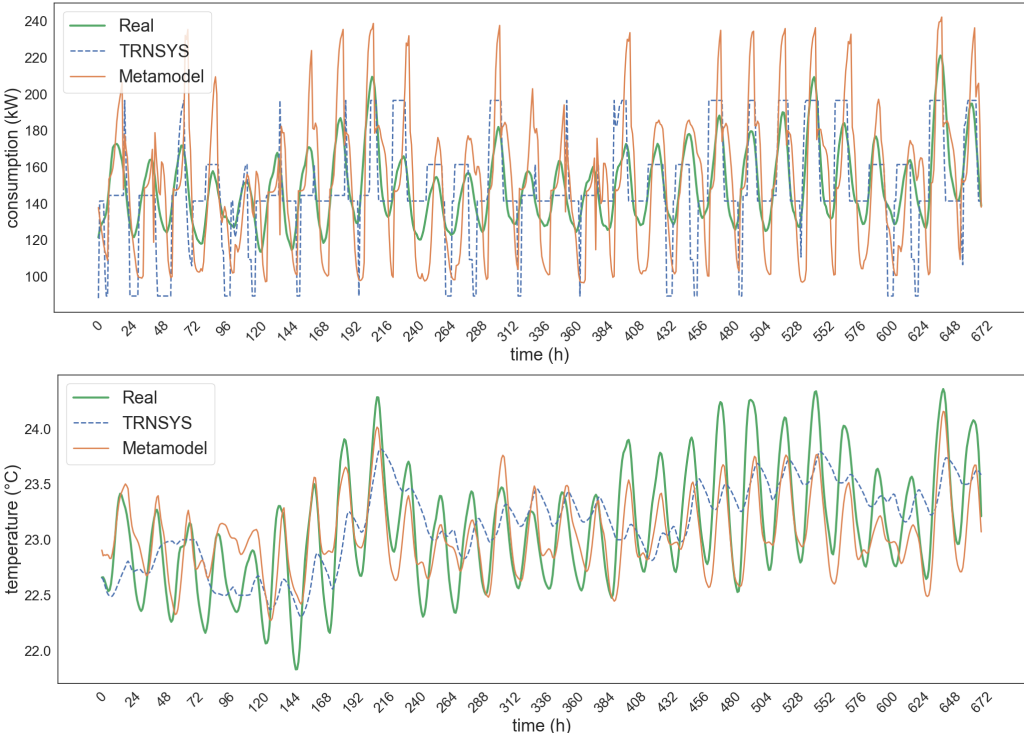


Figure 4: Consumption and temperature simulations after calibration, for both the metamodel and TRNSYS, for Livingstone.

adapted to multi objective problems. An implementation can be found in the Pygmo² library. In the absence of a stopping condition, we simply run the optimization for a set 3000 epochs (2 hours). The result can be viewed as a Pareto front which is given in Figure 6 for the second month used in the calibration process. As observed during calibration, this process can take a colossal number of epochs before achieving satisfactory results, once again justifying the use of a much faster metamodel. The predicted time series associated with the BMS parameters selected in Figure 6 are given in Figure 7. The relative gain, as well as the expected energy savings for both building are available in Table 3.

Table 3: Energy gain after optimization. Relative gain represents the energy load reduction between calibration and optimization steps, when maintaining the initial level of comfort. We then apply this coefficient to the real monthly consumption to obtain the reduction forecast in MWh. We also provide a more interesting reduction obtained by reducing the comfort criteria by 0.5° C.

	relative gain (%)	prevision (MWh)	relative gain / 0.5° C (%)	prevision / 0.5° C (MWh)
Stanley	5.32	8.05	10.5	15.9
Livingstone	9.92	9.96	17.3	17.3

²<https://esa.github.io/pygmo2/>

5 Conclusion

In this paper, we proposed an end-to-end meta modelling methodology to optimize building energy loads with a small computational budget. A calibration step allows to tune the proposed metamodel to ensure compatibility between simulations and real building observations. This metamodel can also be used to optimize the BMS of real buildings with a multi-criteria optimization process. We experimented with various deep learning architectures more suited to recurrent problems than Feed Forward Networks. Results show that a wide variety of models display encouraging results on our sampled dataset, while largely outperforming FFN.

The main impact of our study is to provide a fast process which is able to calibrate a model and to use the calibration results to optimize the settings of the BMS. We illustrated how other available thermal exchange solvers cannot reach optimality in a reasonable time.

References

- [Aird et al., 2016] Aird, G., Coakley, D., and Kerrigan, R. (2016). Application of an optimisation approach for the calibration 1 of high-fidelity building energy models to support model-2 predictive control (mpc) of hvac systems.
- [Bengio et al., 1994] Bengio, Y., Simard, P. Y., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5 2:157–66.
- [Bre et al., 2020] Bre, F., Roman, N. D., and Fachinotti, V. D. (2020). An efficient metamodel-based method to carry out multi-objective building performance optimizations.
- [Bre et al., 2016] Bre, F., Silva, A. S., Ghisi, E., and Fachinotti, V. D. (2016). Residential building design optimisation using sensitivity analysis and genetic algorithm.
- [Cho et al., 2014] Cho, K., van Merriënboer, B., Gulcehre, C., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation.
- [Coakley et al., 2014] Coakley, D., Raftery, P., and Keane, M. M. (2014). A review of methods to match building energy simulation models to measured data.
- [Deb et al., 2000] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: Nsga-ii. In *PPSN*.
- [Fabrizio and Monetti, 2015] Fabrizio, E. and Monetti, V. (2015). Methodologies and advancements in the calibration of building energy models.
- [Hansen et al., 2003] Hansen, N., Müller, S., and Koumoutsakos, P. (2003). Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11:1–18.
- [Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9:1735–1780.
- [Igel et al., 2007] Igel, C., Hansen, N., and Roth, S. (2007). Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 11:1–28.

- [Józefowicz et al., 2016] Józefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the limits of language modeling. *ArXiv*, abs/1602.02410.
- [Kim et al., 2016] Kim, Y., Jernite, Y., Sontag, D. A., and Rush, A. M. (2016). Character-aware neural language models. In *AAAI*.
- [Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.
- [Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NIPS*.
- [Le Corff et al., 2018] Le Corff, S., Champagne, A., Charbit, M., Noziere, G., and Moulines, E. (2018). Optimizing thermal comfort and energy consumption in a large building without renovation works. *2018 IEEE Data Science Workshop (DSW)*, pages 41–45.
- [Magnier and Haghghat, 2010] Magnier, L. and Haghghat, F. (2010). Multiobjective optimization of building design using trnsys simulations, genetic algorithm, and artificial neural network.
- [Mozer, 1989] Mozer, M. C. (1989). A focused backpropagation algorithm for temporal pattern recognition. *Complex Systems*, 3.
- [Nagpal et al., 2019] Nagpal, S., Mueller, C. T., Aijazi, A. N., and Reinhart, C. F. (2019). A methodology for auto-calibrating urban building energy models using surrogate modeling techniques.
- [Ohta et al., 2020] Ohta, Y., Sasakawa, T., and Sato, H. (2020). Evolutionary air-conditioning optimization using an lstm-based surrogate evaluator. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8.
- [Recht et al., 2014] Recht, T., Munaretto, F., Schalbart, P., and Peuportier, B. (2014). Analyse de la fiabilité de comfie par comparaison à des mesures. application à un bâtiment passif.
- [Reynolds et al., 2018] Reynolds, J., Rezgui, Y., Kwan, A., and Piriou, S. (2018). A zone-level, building energy optimisation combining an artificial neural network, a genetic algorithm, and model predictive control.
- [Roman et al., 2020] Roman, N. D., Bre, F., Fachinotti, V. D., and Lamberts, R. (2020). Application and characterization of metamodels based on artificial neural networks for building performance simulation: a systematic review. *Energy and Buildings*, page 109972.
- [Shabunko et al., 2018] Shabunko, V., Lim, C., and Mathew, S. (2018). EnergyPlus models for the benchmarking of residential buildings in Brunei Darussalam. *Energy and Buildings*, 169:507–516.
- [Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *NIPS*.
- [Zhao et al., 2016] Zhao, J., Lam, K. P., Ydstie, B. E., and Loftness, V. (2016). Occupant-oriented mixed-mode EnergyPlus predictive control simulation. *Energy and Buildings*, 117:362–371.

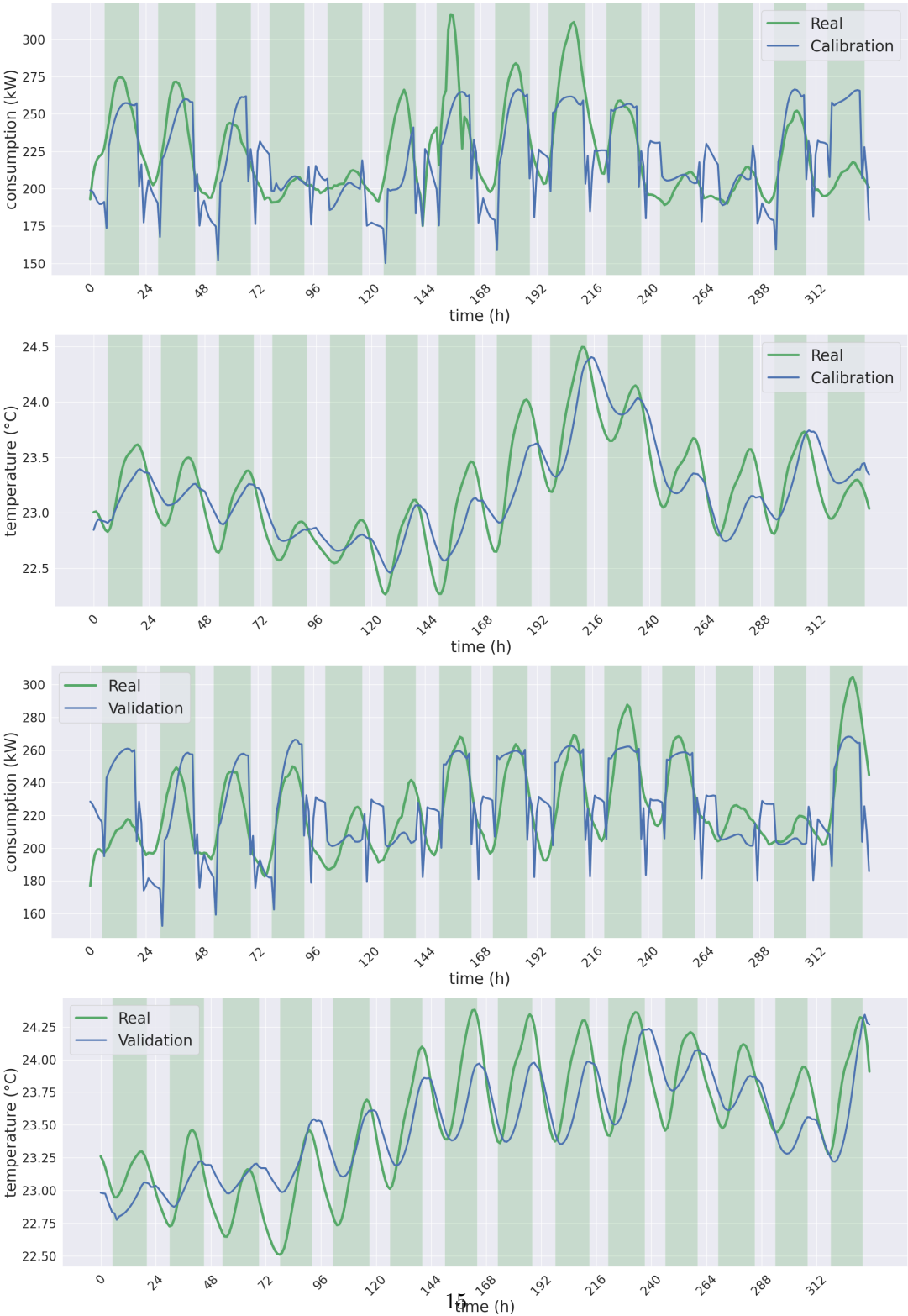


Figure 5: Consumption and temperature simulations after calibration on two weeks (top), simulation on the two following weeks for the same parameters (bottom). Stanley building.

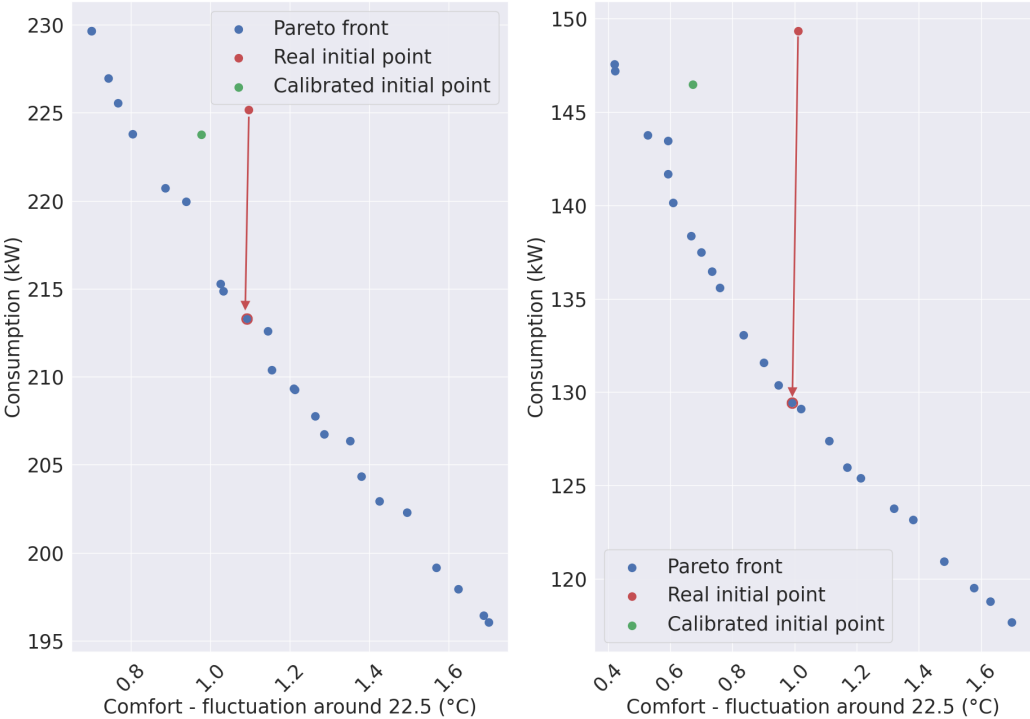


Figure 6: Pareto front after optimization for the Stanley (left) and Livingstone (right) building. We select the point of closest equivalent comfort, corresponding to a 5.3% (Stanley) and 9.9% (Livingstone) reduction in consumption.

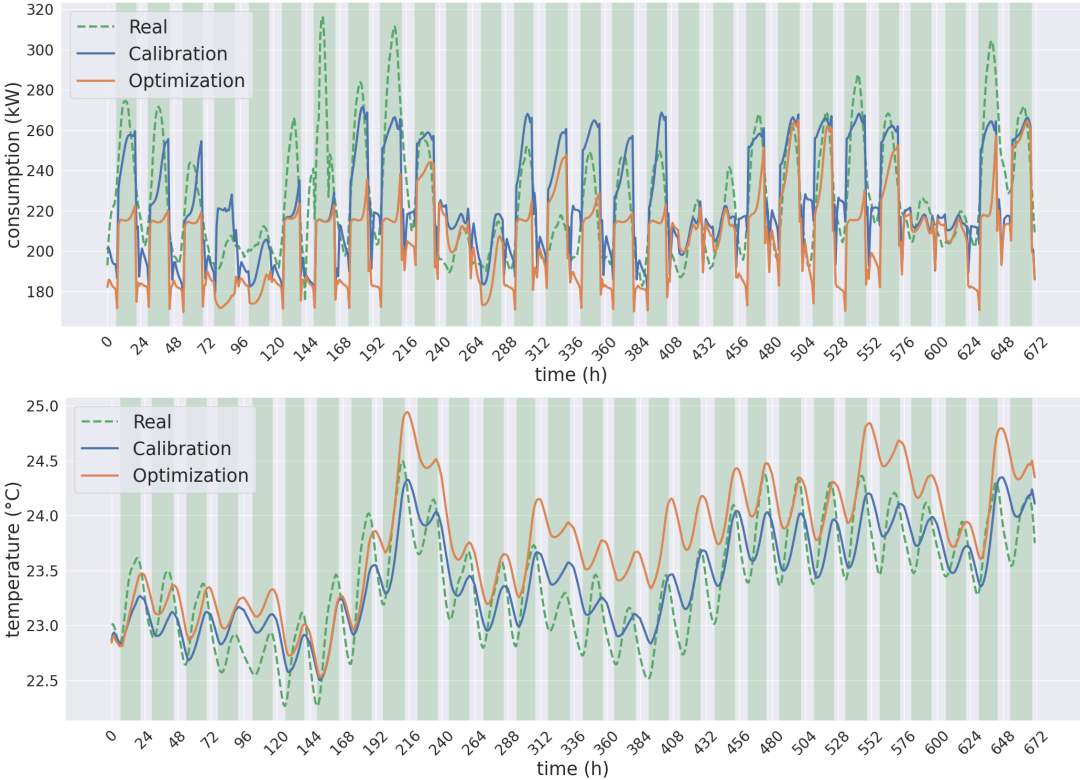


Figure 7: Consumption and temperature simulations after optimization (metamodel) for the Stanley building.

A Additional illustrations

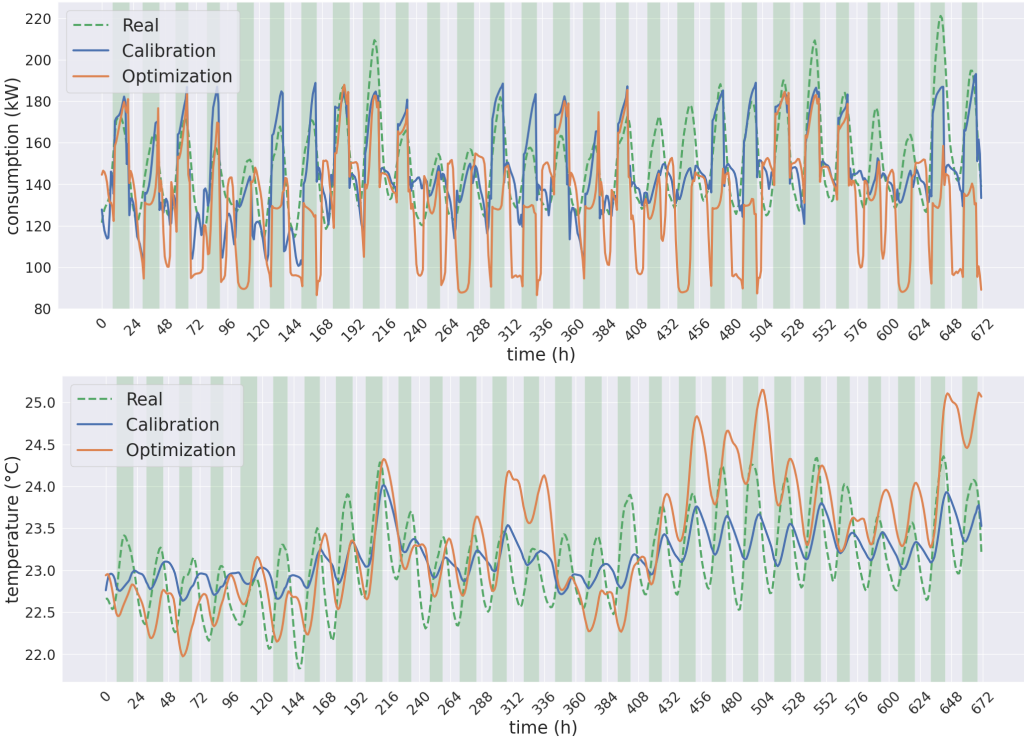


Figure 8: Indoor temperature and consumption for Real data, Calibration and Optimization for Livingstone.

B Ranges used to train the metamodel

Variable	Minimum	Maximum	Step
airchange_infiltration_vol_per_h (m^3h^{-1})	0.1	0.5	0.1
capacitance_kJ_perdegreK_perm3 ($\text{kJK}^{-1}\text{m}^{-3}$)	50	300	10
power_VCV_kW_heat (kW)	0	1000	100
power_VCV_kW_clim (kW)	0	1000	100
nb_occupants	1000	2000	200
nb_PCs	1000	2000	200
percent_light_night	0	70	10
percent_PCs_night	0	70	10
facade_1_thickness_2 (m)	0.05	0.15	0.05
facade_2_thickness_2 (m)	0.05	0.15	0.05
facade_3_thickness_2 (m)	0.05	0.15	0.05
facade_4_thickness_2 (m)	0.05	0.15	0.05
roof_1_thickness_3 (m)	0.05	0.15	0.05
facade_1_window_area_percent	40	50	5
facade_2_window_area_percent	40	50	5
facade_3_window_area_percent	40	50	5
facade_4_window_area_percent	40	50	5

Table 4: θ_{buid} ranges.

Variable	Minimum	Maximum	Step
start_clim_day (h)	7	9	1
end_clim_day (h)	18	20	1
t_clim_red_day (°C)	24	30	0.5
t_clim_conf_day (°C)	20	24	0.5
start_heat_day (h)	6	8	1
end_heat_day (h)	17	19	1
t_heat_red_day (°C)	17	22	0.5
t_heat_conf_day (°C)	22	24	0.5
start_ventilation_day (h)	7	9	1
end_ventilation_day (h)	18	20	1
t_ventilation_day (°C)	18	26	0.5
vol_ventilation_day	0.7	1.7	0.3

Table 5: I_k ranges. Each parameter can hold a different value for each day of the week. For ease of reading, we replaced them by a single line, as the ranges are the same for every day.

Variable	Minimum	Maximum	Step
start_occupation_monday (h)	7	9	1
start_occupation_tuesday (h)	7	9	1
start_occupation_wednesday (h)	7	9	1
start_occupation_thursday (h)	7	9	1
start_occupation_friday (h)	7	9	1
end_occupation_monday (h)	17	20	1
end_occupation_tuesday (h)	17	20	1
end_occupation_wednesday (h)	17	20	1
end_occupation_thursday (h)	17	20	1
end_occupation_friday (h)	17	20	1

Table 6: O_k ranges.

Variable	Description
DNI	Direct Normal Irradiance
IBEAM_H	Direct Horizontal Irradiance
IBEAM_N	Direct Normal Irradiance
IDIFF_H	Diffuse Horizontal Irradiance
IGLOB_H	Global Horizontal Irradiance
RHUM	Outdoor Relative Humidity
TAMB	Outdoor temperature

Table 7: Weather data as contained in W_k .

variable	description
Q_AC_OFFICE	AC consumption
Q_HEAT_OFFICE	Heat consumption
Q_PEOPLE	Heating power due to human activities in the building
Q_EQP	Consumption of equipment, such as computers, elevators, fridges
Q_LIGHT	Consumption of lights
Q_AHU_C	Consumption of AHU when cooling outside air
Q_AHU_H	Consumption of AHU when heating outside air
T_INT_OFFICE	Indoor temperature

Table 8: BEM output variables at each time step.