Mobile Networks and Applications (2022) 27:1977–1986
https://doi.org/10.1007/s11036-022-01968-w

Metamodel Development to Predict Thermal Loads for Single-family Residential Buildings

Marcelo Salles Olinger[1] · Gustavo Medeiros de Araújo[1] · Moisés Lima Dutra[1] · Hugo A. M. da Silva[1] · Laércio Pioli Júnior[1] · Douglas D. J. de Macedo[1]

Accepted: 15 November 2021 / Published online: 25 May 2022

Abstract
Several equitable approaches have been proposed to reduce world energy consumption against a backdrop of a growing global climate crisis. Among these, we can mention the attempts to improve the energy use of household appliances and utilities, such as air conditioners. One of the strategies used to reduce these devices' unnecessary energy consumption is estimating the thermal variation in the environments, especially still during their design phase. One of the most advanced methods for this estimation uses computer simulations, which require a high level of technical knowledge. For that, a relatively simple alternative is the creation of metamodels. This work compares two machine learning approaches for developing a metamodel capable of estimating the thermal load in single-family buildings. The metamodels evaluated were the Artificial Neural Networks and the Gradient Boosting Machine. The results obtained made it possible to observe a better performance in the Gradient Boosting Machine approach indicators in relation to Artificial Neural Networks. The negative point is that Gradient Boosting Machine requires a relatively long training time, making its use in routine projects less feasible.

Keywords Energy consumption · Thermal cooling load · Predictive models · Artificial neural networks · Gradient boosting machine · Building simulation

1 Introduction

According to the International Energy Agency (IEA) [6], in 2017 the building sector represented more than 30% of the total final energy consumption in the world. The [1] report points out that the energy demand for air cooling in buildings has tripled in 1990 and 2016. It is estimated that if no change in the current scenario happens, this demand will triple by the year 2050, representing 37% of the increase in the building's electricity consumption. The potential increasing demand for artificial cooling in hot climate countries is even more significant. According to [1], among the 2.8 billion people living in the hottest parts of the world, only 8% have an air conditioning system. In Brazil, in 2016, the portion of artificial cooling at peak loads of electricity grids corresponded to 7.6% of the total, and the projection is that this portion will represent 30.8% of the peak load by the year 2050 [1].

To ensure the user's thermal comfort without significant energy consumption, it is crucial to understand how thermal variations occur in a building. Analysis during the early design stages of a building indicates fundamental decisions for thermal performance. In the early-design phase, the optimization potential is significant, and at this stage, any building's comfort estimating and energy performance can be reflected in decision-making [3, 13].

The most advanced method of estimating thermal performance in buildings is through computer simulations. It uses models based on physical heat transfer equations, following the principles of energy conservation. However, this process requires the technical knowledge of an expert, as dynamic thermal simulations require detailed models and face several problems, mainly associated with the information needed for input data for the simulated model [4]. In the Brazilian context, the analysis of the building's thermal performance through computer simulations is a relevant measure, as the lack of accessible data related to energy consumption patterns makes it difficult to conduct analyses from databases

✉ Marcelo Salles Olinger
marcelo.olinger@gmail.com

[1] Energy Efficiency in Buildings Laboratory - Civil Engineering Graduate Program, Federal University of Santa Catarina, Santa Catarina, Florianópolis, Brazil

[2]. An alternative to overcome these issues is the development of models from computer simulations, the metamodels. Through metamodels, it is possible to obtain energy performance results close to those of complex simulations.

Metamodels for energy efficiency in buildings can be developed using different methods [8]. The most appropriate solution depends on the context and purposes of each application. Versage [16] was able to estimate thermal loads for commercial buildings through different metamodeling methods. Melo et al. [7] developed an Artificial Neural Networks (ANN) model to estimate the cooling degrees hours and heating and cooling thermal loads in residential buildings. The development of a Support Vector Machine (SVM) metamodel capable of estimating thermal comfort in commercial buildings was proposed by [12].

Energy consumption for cooling in buildings is expressive worldwide, and the expectation is that the energy demand will continue to grow in the coming decades, especially in countries with hot climates. In this context, the use of constructive techniques that minimize the need for air conditioning, through specific building components or geometry design, are presented as a solution to mitigate energy consumption. However, the thermal performance of buildings present complex thermophysical phenomena, making thermal comfort estimates to be considered preferentially from the early-design phases. Seeking a tool capable of helping the designers quickly and simply, the possibility of using metamodels arises as a good option.

Therefore, this work presents a comparison between two machine learning approaches for the development of a metamodel capable of estimating the thermal load in single-family buildings.

2 Review

Designers find difficulties in using building performance simulation tools, which may not be compatible with their needs and working methods. Picco et al. [10] proposes to simplify the description of the building and convert a detailed model into a simplified model, with only a limited number of entries. Despite margins of error of around 15% in estimating annual heating and cooling thermal loads, it is observed that model simplifications can help in early design stages when certain characteristics in the building design are not well-defined. In addition to simplifications of models based on physical equations, it is possible to develop models based on statistical functions, which deduce these behaviors. Statistical models work only with inputs and outputs, without correlating cause and effect, although they are more agile. To adapt to the main functionalities of both models, metamodels are introduced.

Automatic learning-based metamodels are mathematical functions that, applied to a significant amount of data, can identify hidden patterns and predict future functionality. According to [18], the most used machine learning methods for predicting the building's energy performance are ANN and SVM. These models effectively solve nonlinear problems and provide highly accurate predictions as long as the defined model definitions and established parameters are properly defined. ANN models have been used to analyze various types of building energy consumption under various conditions, such as heating and cooling loads, electricity consumption, sub-level component operation and optimization, and estimation of usage parameters. The use of SVM has been growing in research and industry. In many cases, the SVM are superior in performances compared to the ANN, even with a small amount of data training.

Yigit [17] developed a metamodel using Gradient Boosting Machine (GBM), through the Sklearn library, to estimate cooling and heating loads in residential buildings in Turkey. According to the author, GBM models are based on decision tree models and have already demonstrated high performance and flexibility in several areas. These models can operate with a database of several variables, discrete and/or continuous, without the need for data pre-processing. The GBM method enables robust metamodels regardless of the number of parameters or type of parameters. Including parameters with low influence on the dependent variables does not compromise the model's accuracy either. The importance of choosing hyperparameters to ensure better performance of the models is highlighted. The metamodel developed in the work of [17] was applied in an optimization algorithm to find the most appropriate types of envelopes to minimize energy consumption. The results demonstrate that the use of the optimization system could bring design solutions with greater energy efficiency and low cost. Additionally, the increased cost of building envelope can reduce energy consumption by up to 10%. However, the author comments that the implementation of GBM in energy optimization buildings problems is still limited.

Versage [16] developed a metamodel to estimate the annual integrated cooling load to evaluate the energy performance of artificially conditioned buildings through the individual performance of their thermal zones. A database of approximately 1.29 million simulated cases was developed, with varied building parameters, for the climate of the Brazilian city Florianópolis. A data sample was adopted for the elaboration of metamodels with the techniques of multiple linear regression, multivariate adaptive regression splines, Gaussian process, SVM, random forest, and ANN. To evaluate and compare the metamodels, four performance indicators were chosen: training time, coefficient of determination (R^2), root-mean-square deviation (RMSE) and Normalized Root Mean Square Error (NRMSE). The

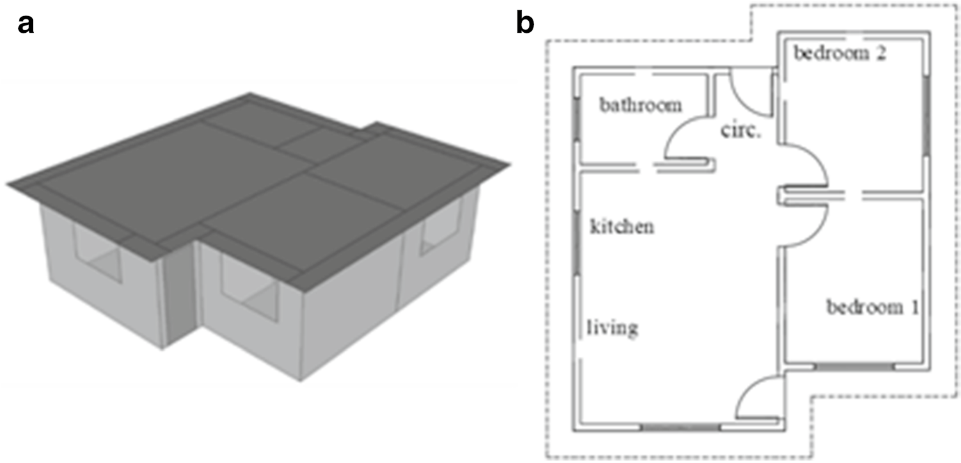

Fig. 1 Simulations' base model. Adapted from [15]

ANN metamodel presented the best performance among those tested. The ANN trained with 1% of the database cases and 72 nodes in the inner layer presented the best overall performance. It was able to reproduce results with errors smaller than 10% for 99.2% of the cases. On the other hand, the metamodel built from SVM had the worst performance. However, the author highlights that other configurations and data processing could change the performance of the evaluated metamodels.

According to the literature, it is observed that different machine learning methods can be used to develop metamodels related to energy efficiency in buildings. There is no rule for choosing the method, and the comparison between the performance of metamodels developed by different machine learning approaches may depend on the configuration of the hyperparameters. Furthermore, they may also depend on specific characteristics of the database used, such as the input and output parameters considered. Therefore, it may be pertinent to consider using more than one type of machine learning technique to search for the most suitable method.

3 Method

This work used a database composed of buildings thermal analyses to develop metamodels using two approaches: ANN e GBM. We compared the performance of the metamodels concerning both approaches and could observe specific issues in the development of each of them. The model training was conducted using 19 parallel processes with a 3.30 GHz Intel Xeon processor. The code used to develop the metamodel is available at the author's \texttt{github} profile.[1]

3.1 The Database

The database (link)[2] used for this work is composed of simulations from the computer program EnergyPlus [5]. The simulation output data is the thermal load necessary to maintain the air temperature in the occupied thermal zones of the building between 21˚C and 23˚C. The simulations were based on a model (Fig. 1) of a single-family building of social housing [15], which had several building characteristics related to the geometry of the envelope and the windows, as well as constructive components varied.

Table 1 presents the parameters of quantitative values, with the maximum and minimum values considered in the sampling. Table 2 presents the variations of the qualitative parameters. Although qualitative, the variables are represented in the database by integer numbers, ranging from 1 to 6 for the wall, 1 to 4 for the roof, and 0 or 1 for blinds and geometry mirroring.

The 46 696 cases in the database were sampled using Sobol's method [14], a quasi-random sampling method that ensures better distribution of cases in hyperspace. The input variables distributions were uniform. Figure 2 shows the independent variable's occurrence distribution as well as the dependent variable. The only input parameter that does not have a uniform distribution is the Mean Annual Temperature. Although the average temperature has been uniformly sampled for values between $10.8^{\circ}C$ and $28.2^{\circ}C$, from the database of weather files in Brazil, there are no cities with average annual temperatures between $10.8^{\circ}C$ and $13.6^{\circ}C$. There are only three cities with average annual temperatures between $13.6^{\circ}C$ and $15.3^{\circ}C$. As a result, the sampled cases with temperatures in these ranges were simulated

[1] https://github.com/marcelosalles/disciplina-machine-learning.git

[2] https://drive.google.com/uc?export=download&id=1XROGI9ZaUX711MjWiyEMv4U84lZFUcGW

Table 1 Quantitative parameters considered in the database

Parameter	Minimum value	Maximum value
Area [m^2]	48	152
Ratio between axes [-]	0.47	2.03
Ceiling height [m]	2.48	3.52
Azimuth [°]	0	360
Wall absorptance [-]	0.19	0.81
Roof absorptance [-]	0.19	0.81
Living room WWR* [-]	0.19	0.81
Bedroom WWR* [-]	0.19	0.81
Glass transmittance [W/m^2K]	2.74	5.76
Solar heat gain coefficient [-]	0.21	0.88
Window opening factor [-]	0.39	0.91
Average annual temperature [°C]	10.8	28.2

* WWR = Window-to-wall ratio

using weather files with mean annual temperatures closer to the sampled value. The output parameter, air conditioning thermal load, had a non-uniform distribution, which varied between 0 and 235kWh/m^{2}year, with an average equal to 53kWh/m^{2}year and median equal to 43kWh/m^{2}year.

3.2 ANN Development

The ANN metamodel was developed using the Sklearn library [9], available for [11]. Quantitative variables were standardized by calculating the z-score, according to Eq. 1, so that all parameters had the same order of magnitude.

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

Where Z is equal to the new value considered for the variable x; μ is the mean of the values of the considered variable; and σ is the standard deviation value of the considered variable. The qualitative variables were transformed into dummy variables, so each column of qualitative value parameters in the dataframe was replaced by new columns. These new columns represent the different values considered for the qualitative values parameter. Thus, for each line of the dataframe, the value 1 is assigned to the column referring to the value considered for the qualitative parameter in that line. On the other hand, the value 0 is assigned to the other columns, referring to the other possible values for that parameter. The number of new columns introduced to replace the original parameter column depends on the number of unique values. The number of columns entered is equal to the number of unique parameter values minus 1. The approach to indicate that a case has the value related to the variable without the corresponding column considers the value 0 in all other columns referring to that parameter. It is also essential to observe the hyperparameters (parameters related to the machine learning process) chosen in developing the metamodel, which influence the results' accuracy. Therefore, we conducted a grid search to find the best combination of hyperparameters.

The varied hyperparameters were:

– `hidden_layer_sizes`: the number of hidden layers and number of nodes in those layers;
– `activation`: the activation function used;
– `batch_size`: the number of cases used in each iteration in the stochastic optimization process;
– `learning_rate_init`: the learning rate;
– `tol`: minimum value of reduction of the loss function over a specified number of iterations (for this study, the specified number of iterations was equal to 10).

The values applied in grid to find the optimal combination of hyperparameters are presented in Table 3

Before the metamodel training step, the sample was divided into a data frame for training, with 90% of cases, and a data frame for testing, with the remaining 10% of cases.

Table 2 Database qualitative parameters

Parameter	Values
Wall	Concrete Concrete + EPS Steel frame Solid brick (10 cm) Solid brick (20 cm) Hollow brick
Roof	Fiber cement tile + concrete slab Fiber cement tile + glass wool (2.5 cm) + concrete slab Fiber cement tile + glass wool (5.0 cm) + concrete slab Fiber cement tile + concrete slab + plaster
Blinds	Yes / No
Mirrored geometry	Yes / No

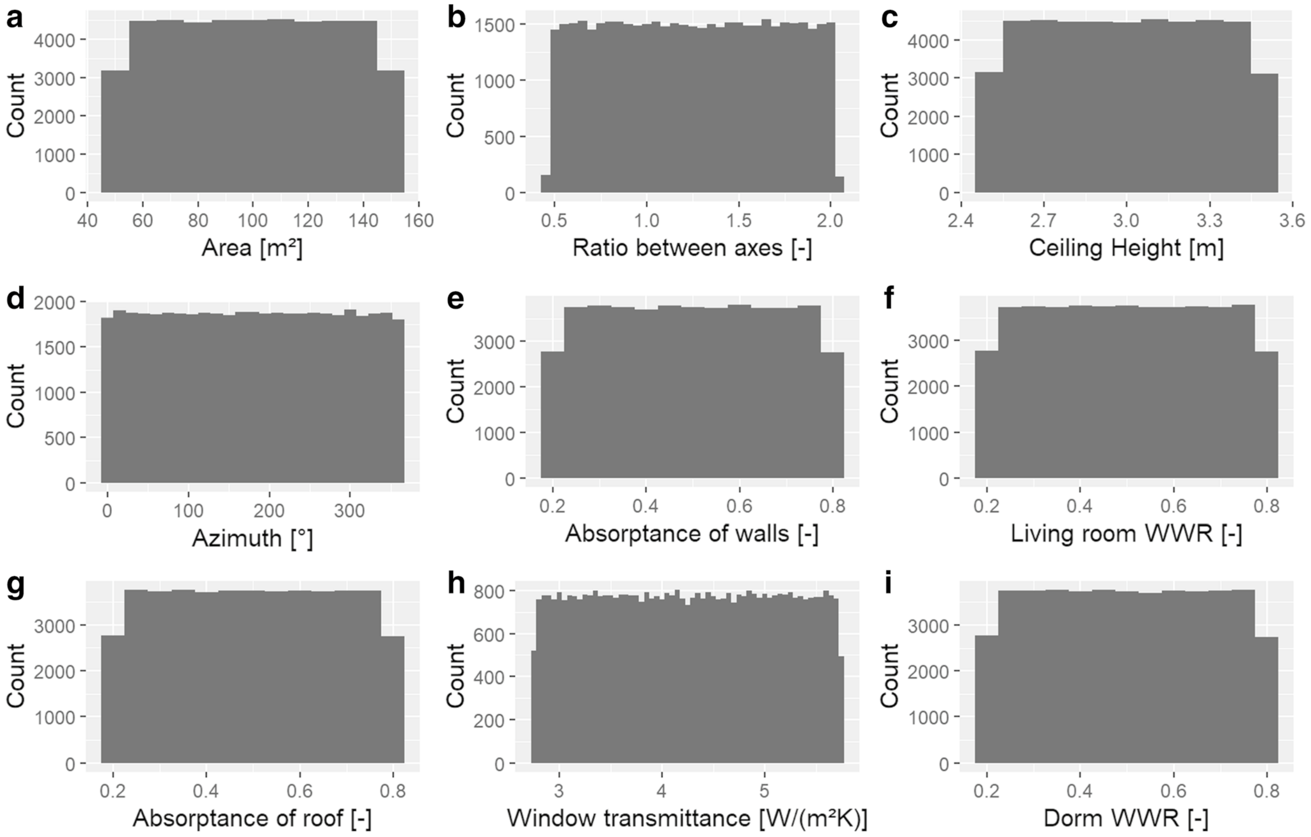

Fig. 2 Input Occurrence distributions parameters

The performance indicator used to compare the different models generated by the grid was the mean of the cross-validation score. We chose the ANN model with the best score in the training stage to have its performance analyzed with the training and test samples. The accuracy indicators of the final metamodel were: the R^2, the RMSE, the mean absolute error (MAE) and the 95th percentile of the absolute error (AE95).

3.3 GBM Development

The GBM metamodel was also developed using the Sklearn library [9], available for [11]. As this approach does not require data pre-processing, there was no standardization or transformation of quantitative variables, and qualitative variables were not transformed into dummy variables. Hyperparameters were also defined from a search for a grid. The varied hyperparameters were:

- `n_estimators`: the number of estimators, or trees;
- `max_depth`: the maximum number of nodes in each tree;
- `min_samples_split`: the minimum number of cases required in a leave of a node;
- `loss`: the loss function. `ls` refers to the least squares regression, `lad` refers to the minimum absolute deviation function, and `huber` refers to a method that combines the previous two;
- `learning_rate`: the learning rate.

The values applied in the grid to find the optimal combination of hyperparameters for GBM are presented in Table 4.

The sample was also divided into a data frame for training, with 90% of cases, and a data frame for testing, with the remaining 10% of cases. As for the ANN, the performance indicator used to compare the different models generated by the grid was the mean of the cross-validation score. The accuracy indicators of the final metamodel were the R^2, the RMSE, the MAE and the AE95.

4 Results

4.1 ANN Results

Figure 3 shows the input data distribution after applying the z-score standardization. It is possible to observe that the quantitative input parameters started to present values

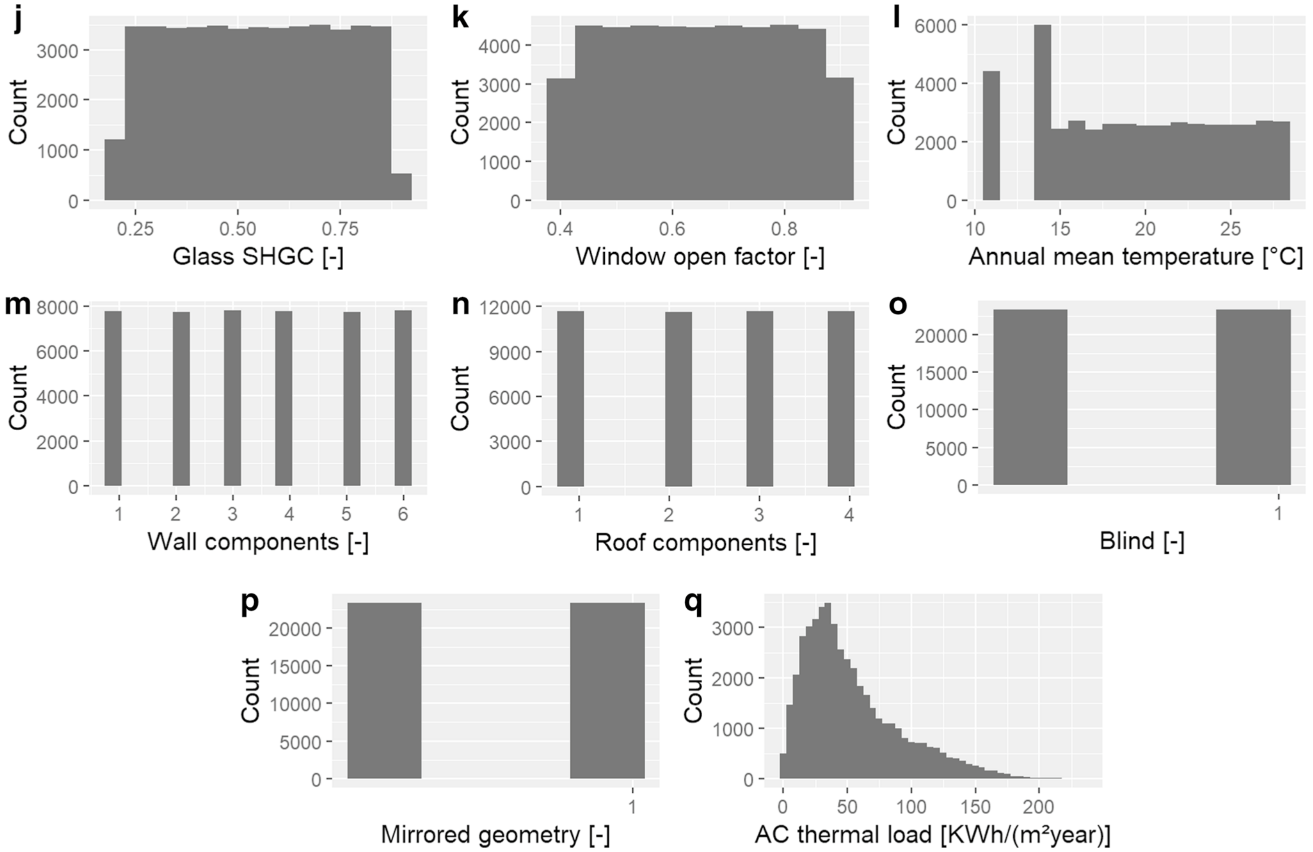

Fig. 2 (continued)

of the same order of magnitude, with an mean equal to zero.

The parameters *Wall*, *Roof*, *Blinds* and *Mirrored geometry* were transformed into dummy variables. Therefore, the parameter related to wall components was replaced by five columns. On the other hand, the parameter related to roof components was replaced by three columns. Finally, the parameters related to the blinds and the mirror geometry were kept with one column. With the combination of the grid hyperparameters, 162 models were trained. The total training time was 2 hours and 17 minutes. The chosen model obtained a value of 0.943 for the mean of the cross-validation score and had the training process interrupted after 310 iterations. The hyperparameters of the chosen model are presented in Table 5.

Figure 4 presents the scatter plot comparing the predicted cases with the simulated cases for the training and test samples. The dotted line corresponds to the line `y = x`, so the accuracy of the results is related to the proximity of the points to the line. Similar trends can be observed in both graphs, but the test sample presents a proportionally more significant number of points away from the dotted line.

The performance indicators of the metamodel are presented in Table 6.

Table 3 Hyperparameters considered in the search for the ANN grid

Hyperparameter	Values
`hidden_layer_sizes`	32 and 64, 64 and 128, 32, 64 and 128
`activation`	`logistic`, `relu`
`batch_size`	32, 64, 128
`learning_rate_init`	0.005, 0.01, 0.05
`tol`	0.0001, 0.0005, 0.00005

Table 4 Hyperparameters considered in the GBM search for the grid

Hyperparameter	Values
`n_estimators`	400, 4 000 and 40 000
`max_depth`	3, 5 and 7
`min_samples_split`	2, 50 and 100
`loss`	`ls`, `lad` and `huber`
`learning_rate`	0.005, 0.01, and 0.05

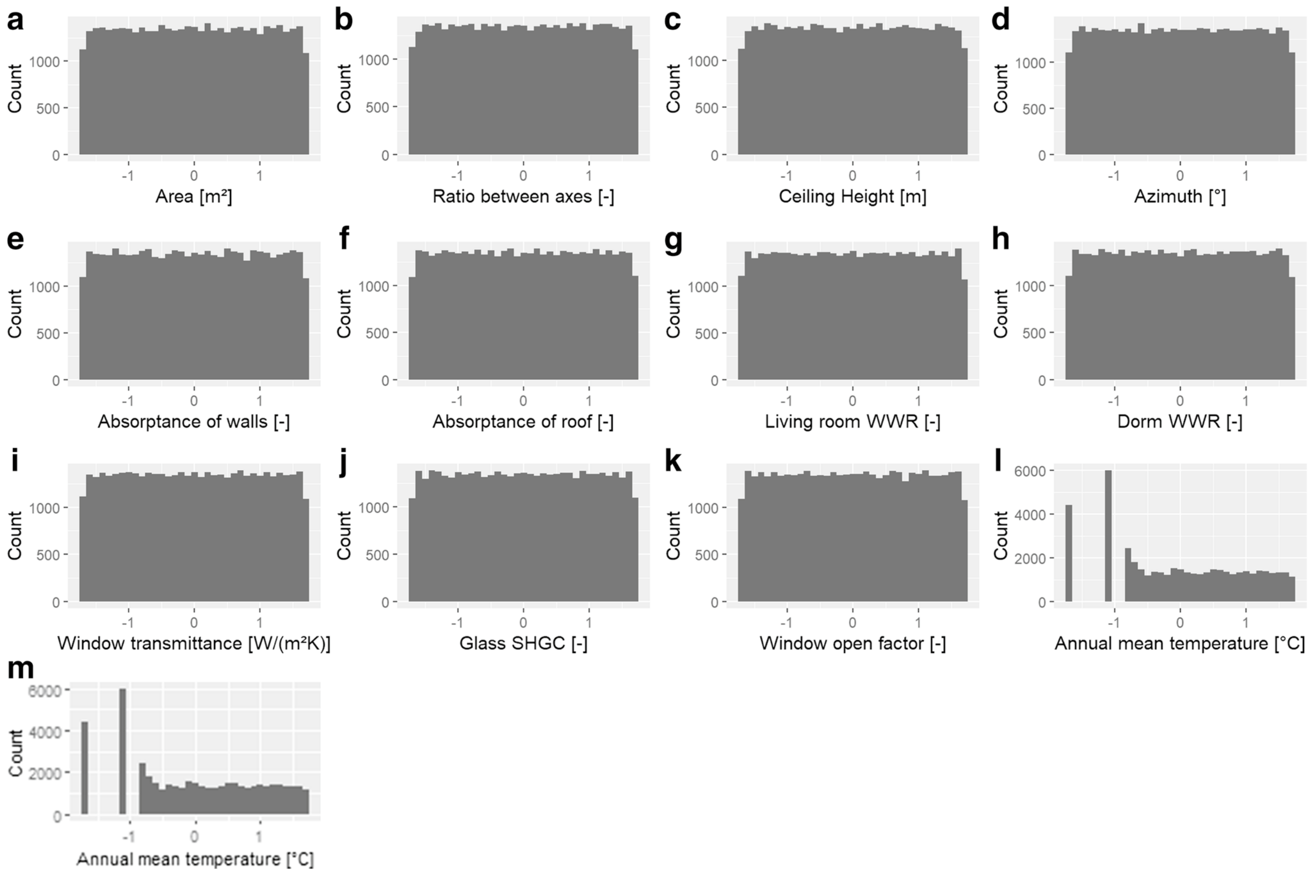

Fig. 3 Standardized input parameters distributions occurrence

4.2 GBM Results

With the combination of the grid hyperparameters, 243 models were trained by the GBM. The total training time was 17 hours and 1 minute. The chosen model had the value of 0.991 for the mean of the cross-validation score. The hyperparameters of the chosen model are presented in Table 7.

Figure 5 presents the scatter plot comparing the predicted cases with the simulated cases for the training and test samples.

The performance indicators of the final metamodel are presented in Table 8.

Table 5 Final ANN template hyperparameters

Hyperparameter	Value
`hidden_layer_sizes`	64 and 128
`activation`	`relu`
`batch_size`	128
`learning_rate_init`	0.01
`tol`	0.0001

5 Discussion

Despite significant differences between the performance indicators of the evaluated metamodels, both machine learning approaches have characteristics that can be considered adequate. The choice of the approach depends on how the metamodel development proposal is determined. The first difference that stands out is the need for pre-processing the input data for ANN while for GBM there is no such need. Defining how the pre-processing of each input variable of the metamodel could take a significant amount of time, as the ideal processing method varies according to the variable type. In this work, we used only standardization by z-score for quantitative variables and transforming qualitative variables into dummy variables. However, other types of data pre-processing can be applied to different types of variables.

Another aspect that may be relevant is the training time of the metamodels. The ANN required significantly less training time (2 hours and 17 minutes) compared to GBM (17 hours and 1 minute).

Considering that we conducted the training of the metamodels on a machine with above-average processing and

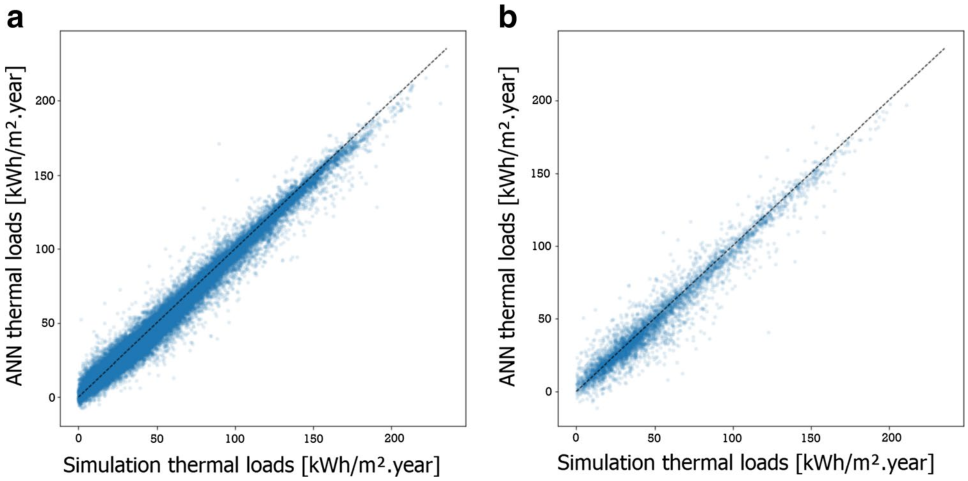

Fig. 4 Comparison between predictions and simulated values for (a) training and (b) testing samples from ANN

parallelization capacity, the need for agility in the development of the metamodel could make the GBM application unfeasible, as the training can take more than 12 hours to complete. This fact indicates a trend that ANN can be a more coherent metamodel to use in a professional routine since GBM training takes more than 12 hours to complete, and the difference between indicators' results is short.

Regarding the metamodels' performance, the GBM presented indicators considerably better than those of the ANN. Both presented acceptable R^2 values, both for the training sample and for the test sample. The difference between the scatter plots of Figs. 4 and 5 shows how the GBM metamodel has better accuracy, with points that are closer to the dotted line, along the entire range of thermal load values considered. In the case of ANN, the model still seems to have a bias in the ranges of higher thermal load values, causing the metamodel to underestimate the output value in the predictions. Despite the differences, in both metamodels the values of MAE and RMSE have a good order of magnitude in relation to the estimated output parameter. The AE95 also remains in the same order of magnitude as the MAE and RMSE, which indicates that the accuracy of the metamodel is not significantly compromised for at least 95% of the cases in the test sample.

Both analyzed metamodels present data overfitting, as the indicators presented lower accuracy values when measured from the test sample concerning the training sample. The reasons for overfitting differ between machine learning methods. In the case of ANN, a number of nodes greater than necessary in the hidden layers can be the cause, as well as the variable `tol`, which interferes in the number of iterations that occur in the training process. In the case of GBM, the increase in the number of estimators can cause the overfit phenomenon. After a certain number of estimators, there is little improvement in the model's performance during the training process, at which point it is possible for overfitting to occur. Since the criterion for choosing the best model within the grid does not consider the test sample, there may be models with the mean value of the cross-validation score close to those of the chosen models that could present superior performance for the test sample. The differences between the indicators obtained by the training and test samples were proportionally more significant in GBM. We observed that their values were greater than twice the values of the training sample for the RMSE, MAE, and AE95. However, the performance of GBM was significantly superior compared to ANN, since the performance indicators of the test sample obtained from it were significantly higher in accuracy than those of the test sample using ANN.

Table 6 Performance indicators of the final ANN model for training and testing samples

Indicator	Training	Test
R^2	0.975	0.945
RMSE	5.97	8.76
MAE	4.29	5.92
AE95	12.05	18.85

6 Conclusion

The development of metamodels can help building design, facilitating the estimation of thermal load demand. The metamodels presented in this work were developed

Table 7 Hyperparameters of the final GBM model

Hyperparameter	Value
`n_estimators`	40 000
`max_depth`	5
`min_samples_split`	100
`loss`	`huber`
`learning_rate`	0.05

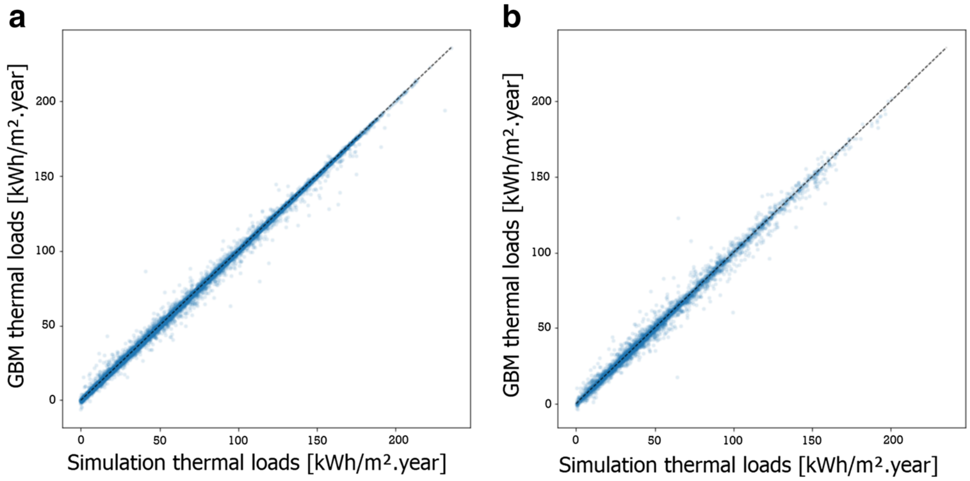

Fig. 5 Comparison between predictions and simulated values for the GBM (a) training and (b) test samples

employing the GBM and ANN methods, implemented with the Python Sklearn library.

The two machine learning approaches presented in this work have specific characteristics that may suit the application context. The ANN requires data pre-processing and can make its application more complex compared to GBM, which does not require this pre-processing. On the other hand, training GBM models are significantly more computationally expensive than training ANN models.

The use of a grid allowed finding the best combination of hyperparameters for the development of metamodels. However, this method defines the final model without considering the test sample, resulting in overfitting the data. A more rigorous adjustment of the hyperparameters that influence overfitting could be explored in future works, guaranteeing more robust metamodels for unseen cases.

According to the performance indicators evaluated (R^2, RMSE, MAE and AE95) the metamodels were able to adequately estimate the thermal loads. However, the GBM metamodel showed considerably higher indicators than the ANN metamodel, both for the training sample and for the test sample. Therefore, the GBM metamodel was the most suitable for the application proposed in this work.

Although the performance of the GBM model got higher performance indicators compared to the ANN model in this study, other configurations of the hyperparameters and pre-processing of the input and output data could result in a better performance for de ANN model. Since there are no definitive answers to the best configuration for these machine learning methods, the results are limited to the optimization methods applied and the computational power available.

Table 8 Performance indicators of the final GBM model for the training and test samples

Indicator	Training	Test
R^2	0.999	0.991
RMSE	1.17	3.48
MAE	0.48	1.97
AE95	1.52	6.89

References

1. (2018a) The Future of Cooling. Tech. rep., International Energy Agency
2. Alves T, Machado L, de Souza RG, de Wilde P (2018) Assessing the energy saving potential of an existing high-rise office building stock. Energy Build 173:547–561. https://doi.org/10.1016/J.ENBUILD.2018.05.044, https://www.sciencedirect.com/science/article/pii/S0378778817335752
3. Belleri A, Lollini R, Dutton SM (2014) Natural ventilation design: An analysis of predicted and measured performance. Build Environ 81:123–138. https://doi.org/10.1016/J.BUILDENV.2014.06.009, https://www.sciencedirect.com/science/article/pii/S0360132314001954
4. Corgnati SP, Fabrizio E, Filippi M, Monetti V (2013) Reference buildings for cost optimal analysis: Method of definition and application. Applied Energy 102:983–993. https://doi.org/10.1016/J.APENERGY.2012.06.001, https://www.sciencedirect.com/science/article/pii/S0306261912004394
5. DOE (2018) EnergyPlus 8.9. https://energyplus.net/
6. IEA (2018b) World Energy Outlook 2018: The Future is Electrifying. Oecd/Iea. https://www.iea.org/workshops/world-energy-outlook-2018-the-future-is-electrifying.html
7. Melo A, Fossati M, Versage R, Sorgato M, Scalco V, Lamberts R (2016) Development and analysis of a metamodel to represent the thermal behavior of naturally ventilated and artificially air-conditioned residential buildings. Energy Build 112:209–221. https://doi.org/10.1016/J.ENBUILD.2015.12.021, https://www.sciencedirect.com/science/article/pii/S0378778815304588?via%3Dihub
8. Østergård T, Jensen RL, Maagaard SE (2018) A comparison of six metamodeling techniques applied to building performance simulations. Appl Energy 211:89–103. https://doi.org/10.1016/J.APENERGY.2017.10.102, https://www.sciencedirect.com/science/article/pii/S0306261917315489
9. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V,

Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830

10. Picco M, Lollini R, Marengo M (2014) Towards energy performance evaluation in early stage building design: A simplification methodology for commercial building models. Energy Build 76:497–505. https://doi.org/10.1016/j.enbuild.2014.03.016
11. Python (2018) Python software foundation. https://www.python.org/
12. Rackes A, Melo AP, Lamberts R (2016) Naturally comfortable and sustainable: Informed design guidance and performance labeling for passive commercial buildings in hot climates. Appl Energy 174:256–274. https://doi.org/10.1016/J.APENERGY.2016.04.081, https://www.sciencedirect.com/science/article/pii/S0306261916305426
13. Roetzel A, Tsangrassoulis A, Dietrich U (2014) Impact of building design and occupancy on office comfort and energy performance in different climates. Build Environ 71:165–175. https://doi.org/10.1016/J.BUILDENV.2013.10.001, https://www.sciencedirect.com/science/article/pii/S0360132313002862
14. Sobol' IM (1993) Sensitivity estimates for nonlinear mathematical models and their Monte Carlo estimation. Math Model Comput Exp 1(4):407–417
15. Triana MA, Lamberts R, Sassi P (2018) Should we consider climate change for brazilian social housing? assessment of energy efficiency adaptation measures. Energy Build 158:1379–1392. https://doi.org/10.1016/j.enbuild.2017.11.003, https://www.sciencedirect.com/science/article/pii/S0378778817315578
16. Versage R (2015) Metamodelo para estimar a carga térmicade edificaċȯes condicionadas artificialmente. PhD thesis , Universidade Federal de Santa Catarina
17. Yigit S (2021) A machine-learning-based method for thermal design optimization of residential buildings in highly urbanized areas of Turkey. J Build Eng 38:102225. https://doi.org/10.1016/j.jobe.2021.102225, https://www.sciencedirect.com/science/article/pii/S2352710221000814
18. Zhao H X, Magoulès F (2012) A review on the prediction of building energy consumption. Renew Sustain Energy Rev 16(6):3586–3592. https://doi.org/10.1016/j.rser.2012.02.049

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.