Computational Cost Reduction of a Simulation-Based Optimization Process Through Machine Learning Methods: Neural Networks vs. Random Forest

Iuri Praça Verginio – Federal University of Viçosa, Brazil – iuri.verginio@ufv.br
Rafael de Paula Garcia – Federal University of Viçosa, Brazil – rafael.pgarcia@ufv.br
Mario Alves da Silva – Federal University of Viçosa, Brazil – mario.a.silva@ufv.br
Joyce Correna Carlo – Federal University of Viçosa, Brazil – joycecarlo@ufv.br

Abstract

Simulation-based optimization (SBO) processes are computationally expensive and the combination with machine learning (ML) methods appears as an alternative capable of reducing computational time consumption without losing the robustness of the solutions. This study compares neural network and random forest algorithms as approaches to replace simulations during the SBO processes. The main objective is to define the best machine learning algorithm and the most reliable ratio between simulations and predictions. The problem was implemented in the Grasshopper + Rhino platform and aimed to minimize the annual energy consumption with artificial conditioning in an office building. Comparing the convergence and reliability of the hybrid processes, the results show that the neural network achieved the best results. The results also show that for this particular/specific problem, the ideal budget comprises 80% of simulations and 20% of predictions, maintaining the results' reliability and reducing the computational cost.

1. Introduction

With the recent technological advances in architectural research, the use of tools capable of producing evaluations and analyses of the performance of buildings has grown, in addition to producing completely new typologies, even before their execution. Parametric modeling, simulations, and optimization are some of these tools. Parametric modeling allows the creation of different building typologies from changes in parameters associated with their characteristics (Farouk et al., 2019). When coupled with simulation tools and optimization techniques, optimal solutions can be obtained satisfying pre-established performance conditions of a building. The coupling among parametric modeling, simulation, and optimization is often called simulation-based optimization (SBO).

In an SBO problem, parametric modeling works to modify the solutions, according to some intelligence implemented by the optimization algorithm, in search of optimal regions of the feasibility space. This search is guided by a fitness function that usually depends on the objective function of the problem and that also requires simulations to be evaluated.

However, SBO processes demand a vast amount of computational time mainly due to the simulations involved. To mitigate this issue, Machine Learning (ML) techniques have been employed since they can predict fitness function values associated with new solutions through real simulation data and produce evaluations faster than traditional methods (Seyedzadeh et al., 2019; Melo et al., 2014). This makes it possible to replace some simulations with predictions acquired by machine learning, training it with the results of simulations previously produced during the SBO process.

The use of machine learning techniques for prediction shows promise in optimization problems due to the intrinsic nature of optimization algorithms, which often generate and evaluate multiple solutions while searching for the optimal solution. This process results in the creation of a database containing solutions previously evaluated by the exact function of the problem. This database represents a valuable resource for training machine learning

Part of
Pernigotto, G., Ballarini, I., Patuzzi, F., Prada, A., Corrado, V., & Gasparella, A. (Eds.). 2025. Building simulation applications BSA 2024. bu,press.
https://doi.org/10.13124/9788860462022

techniques, allowing the prediction of new solutions without the need to evaluate each one for its exact function throughout the optimization process. This approach offers the potential to significantly reduce the time and resources required to solve optimization problems that demand high computational costs.

In the past years, several studies have shown the capability of Artificial Neural Networks (ANN) and Random Forest (RF) to aid in solving architecture and engineering problems. One in particular (Bui, Nguyen, Ngo, and Nguyen-Xuan, 2020) estimated the amount of energy used for various activities within a building, such as heating and cooling (energy consumption), using the hybridization of the ANN model with the firefly optimization algorithm (EFA). The performance of EFA-ANN was validated by comparing the obtained results with other methods, such as iteratively reweighted least squares (IRLS), ensemble model, smart artificial firefly colony algorithm-based support vector regression (SAFCA-SVR), extreme learning machine (ELM), which presented best results, and the lowest Root Mean Square Error (RMSE) values. Further, (Zekić-Sušac et al., 2021) used Random Forest, ANN and classification and regression tree (CART) to predict the cost of energy consumed in public buildings. The results have shown that the approach integrating random forest with the Boruta algorithm has produced a higher accuracy.

The neural network was created to mimic concepts from the neurobiological field, and it works through 3 main elements: inputs, hidden layers, and outputs. The inputs correspond to the parameters of the problem being analyzed. The layers are formed by nodes, structures where each input value is associated with a weight through mathematical functions and sent to the next layer (Gurney, 1997). Finally, outputs represent the predicted value of the response variable.

In contrast, Random Forest is a collection of randomized decision trees (Kam Ho, 1995). These decision trees are a machine learning technique that works as a tree structure by repeatedly dividing the given data into smaller subsets until only one data remains in each subset. The inner and final sets are known as nodes and leaf nodes. Then for the final results, it calculates the average predicted values of all independent trees.

Both methods need training and parameter tuning to improve the quality of their responses. Therefore, it is necessary to separate the database into two parts, one for training and the other for testing. The algorithm is exposed to the training part, where it learns patterns that will be applied in the test part to predict the response variable (Mahesh, 2019). The choice of these sets impacts the final quality of predictions. Typically, 80% of the data are used for training and the remaining 20% for testing, as such separation ratio has been theoretically proved to deliver good results (Gholamy et al., 2018). Then, the response variable predicted by the method is compared with the actual values from the database, and depending on the performance of the method, its parameters are adjusted.

The main parameters for the neural network are the number of iterations, nodes, and layers. For the random forest they are the numbers of independent trees and its depth. As suggested by (Karsoliya, 2012), the parameters are obtained by performing robust tests with database samples using different configurations in order to obtain better results.

Therefore, this study aims to evaluate and compare two ML methods, Neural Network and Random Forest, when coupled to an SBO process to indicate which one offers the best performance in predicting the energy consumption of an office building and also discover which is the best percentage of simulations replaced by prediction without losing the quality of the results.

2. Methodology and Simulation

2.1 The Simulation-Based Optimization Problem

A simulation-based optimization problem was selected based on the research of previous authors (Wetter & Wright, 2004), and it seeks to minimize the primary energy consumption based on the annual thermal loads of a single thermal zone that represents an office building (Fig.1). The building model combined the East and West offices (grey shade in Fig.1) into a single thermal zone and added the corridor as internal mass.

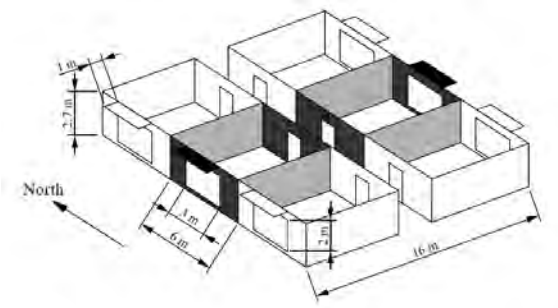

Fig. 1 – 3D Model of the office building. Source: Wetter, M., Wright, J., 2004

The SBO problem quantifies the impact of four parameters: building orientation (180° to -180°); transmittance of shading elements (0.2 to 0.8); and width of the openings to the east and west (0.1 to 5.9 m).

We used the Grasshopper for Rhino SBO implementation from (Waibel et al., 2019). The implementation uses EnergyPlus for the building performance simulation. We analyzed the problem based on TMY2 weather data for Seattle, Washington, USA.

We used the RBFOpt mono-objective optimization algorithm (Costa and Nannicini, 2018). The single objective method is an optimization approach that focuses on minimizing or maximizing a single objecttive value. It is available in the Opossum optimization engine, as it stands out from previous work (Waibel et al., 2019). As for this paper, the minimizing method was chosen, and its parameters were used by default. A maximum of 10000 iterations and evaluations, with local search available and 2000 maximum cycles. Additionally, the generation of 700 solutions served as a stopping criterion for each run of the SBO.

2.2 The Machine Learning Methods

Due to the nature of how these ML techniques work and the need for a previous database to be created, it requires a certain number of solutions through simulations to be made. The size and quality of this database directly impacts the performance of the ML.

In this work, the first solutions generated by the optimization algorithm and evaluated by the exact function of the problem constitute the database used to train the machine learning techniques. To answer one of the questions in this research about the best proportion of solutions evaluated by the exact function compared to those that will be predicted by the ML technique, 5 divisions of the database were proposed to be investigated. We started from 50% simulated and used as a database, and the others 50% were predicted. Then in each split 10% was added to the simulated percentage until the last split, which was 90% simulated and 10% predicted.

The Python language was used to code the algorithm of both ML techniques. To implement the neural network and random forest in the Grasshopper environment, the GHPythonRemote plugin was used (Cuvilliers and Mueller, 2022). This plugin allows the connection of external Python instances to Grasshopper, enabling the use of several code libraries that once were not available. For this problem, the Python programming library Scikit-learn (Sklearn) was used on both ANN and RF.

In the case of ANN, the MLPRegressor parameters were kept as default, with the exception of the number of hidden layers and neurons, which were set to 30, and the maximum iterations equal to 4000. Increasing the number of layers can enhance the model's capacity, yet this can only be done to a certain extent, since rather than extracting meaningful patterns, the model may start to 'overfit' the data. For RF, the RandomForestRegressor from Sklearn, the only specified parameters were the maximum depth of the tree, set to 15, and the number of trees in the forest, set to 100. This follows the same concept of the neurons and layers from ANN, where higher numbers can increase the model's quality but up to a certain threshold, to avoid overfitting.

Consequently, the workflow of the SBO processes coupled with ML techniques is presented as in Figure 2. The optimizer engine generates new solutions, by changing the values of the variables, and their Fitness Function values F(x) are obtained through simulations by EnergyPlus. These solutions are stored in a database until the desired number of simulations is reached, as defined by the simulation/prediction ratio used. This process is represented by green arrows. Then, all the values of the variables and the respective answers found by simulation up to now, serve as a database to feed the machine learning technique. From now on, the ML is the one that will provide F(x) to the optimizer

engine, with the workflow represented in red and black arrows in Fig. 2.

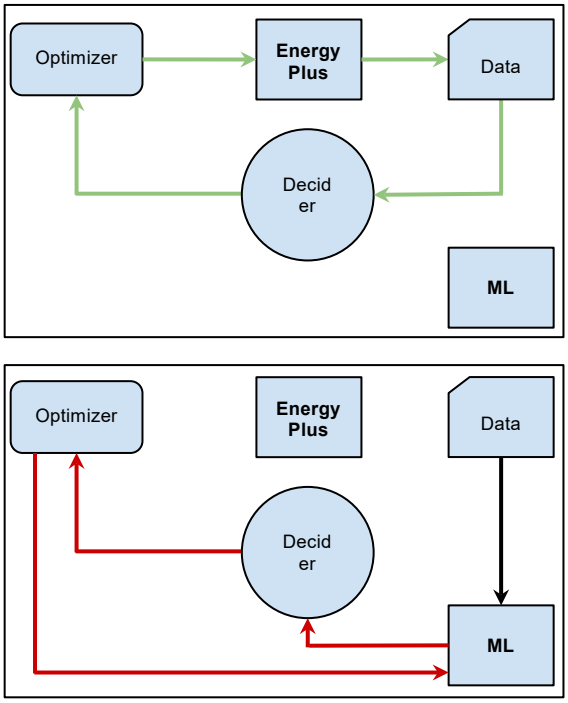

Fig. 2 – Workflow on the cases without ML technique (green arrows), and with hybrid application (red arrows)

2.3 The Analysis

To produce statistics, 25 runs were done for each percentage ratio for each of the ML methods. The coupling process of the simulation with the optimization happens in the evaluation of the fitness function of the solutions, composed of the building energy consumption.

A base case was used for comparisons disregarding the ML techniques and obtaining the values of all solutions by simulation. Adding this case to the other 5 that mix simulation and prediction, a total of 6 x 25 runs of SBO process were carried out, evaluating 700 x 25 x 5 solutions in each of the ML techniques.

After that, in order to compare the methods, we first assessed the performance of both methods by tracing the average of the convergence of the solutions found. Then, we evaluated the root mean squared error (RMSE) values of some results produced by the ML technique, selected according to a random sampling of 30 solutions. Finally, we compared the optimal results obtained at the end of each run through boxplots.

3. Results and Discussions

Undoubtedly, in terms of computational resource expenditure, solutions whose fitness is predicted require much less computational time. On average, for the problem in this paper, the computational time for predicting the fitness of a solution is 75% faster than by simulation, as the first takes around 818 milliseconds and the second 3.4 seconds.

Figure 3 presents the convergence values for NN and RF application. Both exhibit similar behavior, by repeating the values at the end of the process and all percentage combinations do not differ significantly, since all reached the value of 133 kWh/m². Even so, the combination of 90% of simulated cases with 10% predicted by ML, obtained the best response with a consumption metric of 133.19 kwh/m² for NN and 133.11 kwh/m² for RF. Furthermore, in both applications, the 60/40 and 80/20 processes are the closest to the 100% simulated results, represented in red.

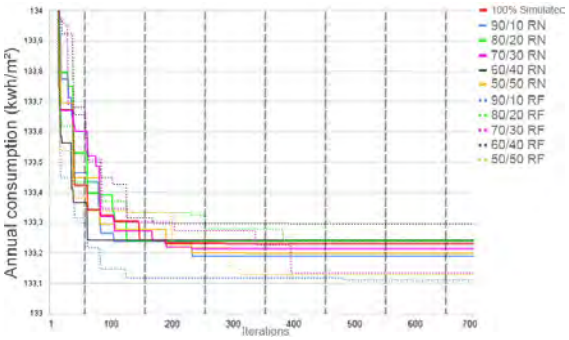

Fig. 3 – Convergence graphs for NN and RF results

When evaluating the difference between the responses acquired by prediction and the same by simulation, it can be seen in the Figure 4 that the values of the RMSE are generally low, not exceeding the margin of 1.65 kwh/m² on the percentage 50/50 for NN and 1.62 kWh/m² for RF. The 80/20 process that obtained the lowest RMSE value is also the one that came closest to the real-case convergence curve (Figure 3).

In practical terms, it is safe to say that all percentages had an acceptable performance regarding the quality of the responses found through the prediction with the ML techniques.

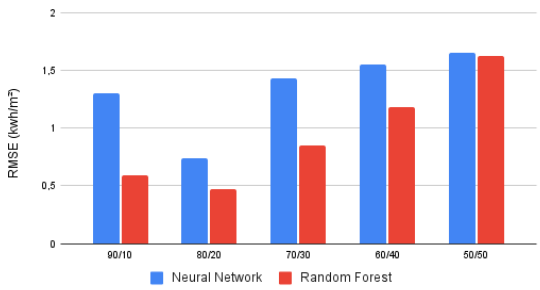

Fig 4 – RMSE values for each percentage in both techniques results

The figure below presents the best results acquired in all 25 runs of each of the percentages. Through the boxplot, it is noticed that the percentage 60/40 stands out, obtaining the lowest median for neural networks, whose results are presented with a black outlayer. Although, the minimum value of 129.886 kwh/m² that was acquired through the neural network initially qualifies it as the best result among all processes. However, when performing more detailed analysis and simulating with the same parameters supplied to the neural network, it was noticed that the response obtained by simulation was greater than that predicted by the model. Consequently, the value of 129.886 kwh/m² cannot be considered the lowest among all the percentages. Meanwhile, it is possible to observe that the 90/10 process obtained the lowest median for Random Forest, followed by 80/20, which also had the lowest dispersion. As for the other processes, in addition to a higher median value, they present a greater dispersion.

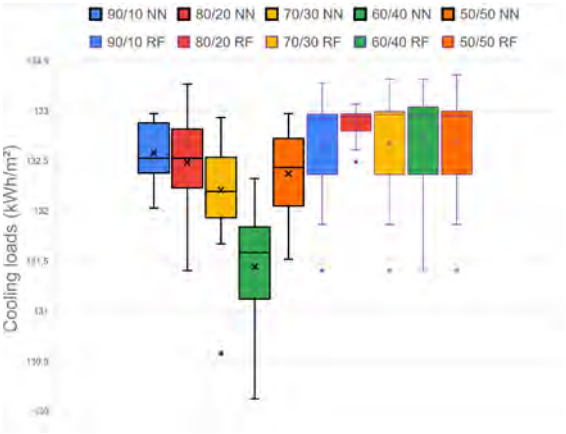

Fig. 5 – Boxplot of best results for each percentage with both techniques

4. Conclusion

This study aimed to evaluate two methods of ML and also discover the best percentage of simulations replaced by prediction without compromising the quality of the results. The results presented here, for both ANN and RF, demonstrate a significant reduction in the computational cost without affecting the optimization process's performance. For all the percentages used, the RMSE values varied between 0.74 kwh/m² to 1.65 kwh/m² for NN and 0,59 kwh/m² to 1,62 kwh/m² for RF.

As for the convergence from both methods, it is shown that the neural network, in all its processes, converged close to 200 solutions (Figure 3). In the RF, however, this convergence takes more time to happen, indicating that the neural network processes could have been stopped much earlier, further minimising the computational cost and outperforming the random forest application.

Additionally, the optimal values presented in the convergence graph are similar to the curve of the results of the 100% simulated case. For the artificial neural network, the configuration closest to this was 80/20, where 80% of the solutions are simulated and 20% are predicted. The same happens in random forest application, where 80/20 not only is the closest to the 100% simulated case but also has the lowest RMSE value of 0.47 kwh/m² and the second lowest median of all processes. Therefore, this is indicative that 80/20 is the best percentage, for both machine learning techniques presented in this study.

In conclusion, both techniques presented a reduction in computational cost, obtaining the best results in the 80/20 division and low RMSE values. However, the neural network proved to be more suitable for this problem, considering that it converged faster, which would allow us to reduce the number of solutions needed for the problem. Still, further research should apply these same comparisons to SBO problems with multiple objectives. This will help to reassert the best technique choice for a hybrid method and significantly reduce the computational cost spent on solving SBO problems.

Acknowledgement

The authors acknowledge the support of the Brazilian funding agency FAPEMIG (project number 206442 - UFV/PIBIC/FAPEMIG 2022 - 2023).

References

Bui, D. K., Nguyen, T. N., Ngo, T. D., Nguyen-Xuan, H. (2020). "An artificial neural network (ANN) expert system enhanced with the electromagnetism-based firefly algorithm (EFA) for predicting the energy consumption in buildings". *Energy*, 190. https://doi.org/10.1016/j.energy.2019.116370

Costa, A., Nannicini, G. (2018). "RBFOpt: an open-source library for black-box optimization with costly function evaluations". *Mathematical Programming Computation*, 10, 597-629. https://doi.org/10.1007/s12532-018-0144-7

Crawley, D. B., Pedersen, C. O., Lawrie, L. K., Winkelmann, F.C. (2000). "Energyplus: energy simulation program". *ASHRAE Journal*, 42, 49-56.

Cuvilliers, P., Mueller, C. (2022). ghpythonremote. GitHub, (version 1.4.6) [Computer software]. https://github.com/pilcru/ghpythonremote

Farouk, A., Eldaly H., & Dewidar, K. (2019). "Parametric design as a tool for performative architecture". *Journal of Al Azhar University Engineering Sector*, 50, (14), n. 50, 148-157.

Gholamy, A., Kreinovich, V., Kosheleva, O. (2018). "Why 70/30 or 80/20 Relation Between Training and Testing Sets: A Pedagogical Explanation". *International Journal of Intelligent Technologies and Applied Statistics*, 11(2), 105-111. https://doi.org/10.6148/IJITAS.201806_11(2).0003

Gurney, K. (1997). An Introduction to Neural Networks (1st ed.). CRC Press. https://doi.org/10.1201/9781315273570

Kam Ho, T. (1995). "Random decision forests". *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1, 278-282. https://doi.org/10.1109/ICDAR.1995.598994

Karsoliya, S. (2012). "Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture". *International Journal of Engineering Trends and Technology* (IJETT), 6, 714-717.

Mahesh, B. (2019). "Machine learning algorithms-a review". *International Journal of Science and Research* (IJSR), 9, 381-386. 10.21275/ART20203995

Melo, A. P., Cóstola, D., Lamberts, R., Hensen, J. L. M. (2014). "Development of surrogate models using artificial neural networks for building shell energy labelling". *Energy Policy*, 69, 457–466. https://doi.org/10.1016/j.enpol.2014.02.001

Nannicini, G. (2021). "On the implementation of a global optimization method for mixed-variable problems". *Open Journal of Mathematical Optimization*, 2, 1-25. 10.5802/ojmo.3

Seyedzadeh, S., Pour R. F., Rastogi, P., Glesk, I. (2019). "Tuning machine learning models for prediction of building energy loads". *Sustainable Cities and Society*, 47, 101484. https://doi.org/10.1016/j.scs.2019.101484

Waibel, C., Wortmann, T., Evins, R., Carmeliet, J. (2019). "Building energy optimization: An extensive benchmark of global search algorithms". *Energy and Buildings*, 187, 218–240. https://doi.org/10.1016/j.enbuild.2019.01.048

Wetter, M., Wright, J. (2004). "A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization". *Building and Environment*, 39, n. 8, p. 989-999, 2004.

Zekić-Sušac, M., Has, A., Knežević, M. (2021). "Predicting energy cost of public buildings by artificial neural networks, CART, and random forest". *Neurocomputing*, 439. 223-233. https://doi.org/10.1016/j.neucom.2020.01.124.