BauSIM2022 in Weimar
20.-22. September 2022
Bauhaus-Universität Weimar
ibpsa

•Replacing time-consuming Building Performance Simulations with real-time Surrogate Models and their application in early-stage design space exploration

Max Zorn[1], Natasha Catunda[2], Luisa Claus[2], Natalia Kobylinska[2], Manuel Frey[2], Thomas Wortmann[1]

[1] *Chair for Computing in Architecture, Institute for Computational Design and Construction, University of Stuttgart, DE, E-Mail: max-benjamin.zorn@icd.uni-stuttgart.de*

[2] *Digital Planning, Gruner AG, Switzerland, CH, E-Mail: natasha.catunda@gruner.ch*

Abstract

Sustainable design requires holistic decision making already in the feasibility stage. One critical aspect of a buildings' sustainability is its operational energy consumption, but energy simulations typically are too time-consuming for early, fast-paced design phases. Data-based, Machine Learning models--so called surrogates--can replace time-consuming simulations with real-time estimates. This paper investigates the accuracy of different surrogate models for energy performance by comparing different types of models and numbers of samples. The paper also presents an integrated dashboard for holistic decision making as an application of the developed surrogate.

Introduction

Early-stage design decisions have the biggest impact on building performance and thus on the environmental impact of a building (WP-1202, 2004). It is crucial for designers and other stakeholders to include performative criteria in the assessment of design variants from the earliest stages on, to tackle the challenges imposed on the building sector by the climate crisis. Computational design (CD), more specifically generative modeling, allows designers to create many digital building variants, formally exploring the project's design space in real time. Many Key Performance Indicators (KPIs) such as Gross Floor Area (GFA) or Window-to-Wall Ratio (WWR) are either already embedded in these digital models or can be extracted with little to no computing cost. However, other criteria like building performance must be simulated. Integrating building performance simulation (BPS) in design space exploration (DSE) allows designers to get feedback on their design decision but potentially long runtimes impose a barrier to the performative evaluation of early-stage design variants.

These long runtimes can be overcome by utilizing Machine Learning (ML) techniques, particularly by algorithms capable of predicting. Such an emulator, called surrogate model (SM), can be used to replace BPS, outputting precise enough predictions while drastically reducing computing time. However, a SM's accuracy depends on the size and quality of the data set it has been trained on, which generally needs to be sampled from the design space to be explored. As computing time is shifted from during DSE to before, this approach allows real-time performance assessment.

Real-time building analysis in early-stage design is also a key parameter to co-design strategies, like the Integrated Project Delivery (IPD) approach. IPD aims to effectively enhance collaboration between stakeholders at all stages of the design to overcome the fragmentation of planning and realization processes and is seen by many as a paradigmatic change in planning. According to the American institute of Architects the participation of all stakeholders improves building processes by utilizing diverse expertise already during feasibility studies (AIA, 2007). However, this collaboration can only work if early-stage design tools allow real-time assessment. System response time influences both productivity and satisfaction of users during computational tasks, especially during creative workflows (Brown, 2020). Further, during conceptual phases such as early-stage DSE, fast response times are valued higher by users than high accuracy simulation results. Computing times as short as 1.5 seconds negatively influence engineering tasks as shown by (Simpson et al., 2007) and as not even modern cloud computing platforms like Pollination can provide BPS within such short times (Pollination, 2022), integrating SM into this process is a key aspect in making IPD reality.

While contemporary CD software provides various ways of visualizing geometry and geometric analyses, other KPI data must be visualized independently. To effectively communicate between different stakeholders, a dashboard displaying performative data is needed. Such a concise data visualization allows for faster and more precise DSE (Brown, 2020). Again, traditional BPS tools are too slow to allow the integration of performance assessments in such interactive design tools, leading to the integration of real-time prediction with SMs. We'll present a dashboard capable of visualizing indicators such as floor area, cost, daylight quality, operating energy, and CO2 emissions from construction and operation according to the standard for sustainable buildings in Switzerland (*Standard Nachhaltiges Bauen Schweitz*) SNBS (NNBS, 2014).

This research investigates the performance of various regression-based SMs, predicting energy demands of early-stage design variants in a given design space. Furthermore, the application of a SM is demonstrated by its interaction with generative CD inside Rhino / Grasshopper, where the results are visualized in an interactive, real-time dashboard, following the IPD approach. The subsequent chapters present background, the generative CD model including BPS, the set-up and evaluation of the SM, and the demonstration based on the integrated dashboard.

Background

While Surrogate Models are increasingly popular for architectural design optimization (Wortmann et al., 2015), they are also very applicable to early stage design, particularly to overcome long runtimes of BPS during explorative design phases. Tseranidis et al. (2016) investigate SMs for civil and architectural design and compare the performance of various predictive algorithms applied to a case study design, predicting both structural and energy performance. They provide a framework to test different SM algorithms and choose the suitable one for a given design task. They did not integrate their models into early-stage DSE tools. Geyer and Schlüter (2014) present SMs that are automatically generated to increase accessibility for designers and apply them to a variety of early-stage design as well as retrofitting problems. While they trained models to predict energy demands, those were only able to either predict heating or cooling. Seyedzadeh et al. (2020) use Gradient Boosted Regression Trees to accurately estimate building energy consumption to calculate Building Emission Rates and to then find optimal solutions for retrofitting. Rastogi et al. (2021) present testing strategies to evaluate the feasibility of regression based SMs to substitute BPS. Besides presenting a testing methodology, they conclude that non-linear models are better suitable for SMs replacing BPS than linear models. Westermann and Evins (2019) provide a literature review of 57 studies utilizing SMs in building design. 29% of the reviewed publications applied SMs during early-stage design phases, where rapid feedback was one of the main reasons. However, none of the reviewed publications used floor count, nor building shape as input variables for a SM predicting energy demand.

Generative Model

For efficient DSE, a generative parametric model is needed. Such a model can output many building variants from a set of input parameters. We thus built a definition in Grasshopper that can generate different parametrized volumetric models with inscribed, distinct building programs. The cubature is based on closed polylines, drawn by the designer. The parametric inputs used for generating the variants are:

- Building footprint as a polyline object
- Floor count as integer
- WWR [%] North façade
- WWR [%] East and West façades
- WWR [%] South façade

Geometrical Analysis

The generated variants are used to conduct diverse analyses. For example, based on the cubature, urban quality can be verified, and shading can be analyzed as shown in Fig.1. Further, the building component groups are automatically extracted from the 3D-model. Those building components are based on the eBKPh structure (CRB Schweizerische Zentralstelle für Baurationalisierung [SN], 2020), including outside walls, internal walls, slabs, roofs, and windows.

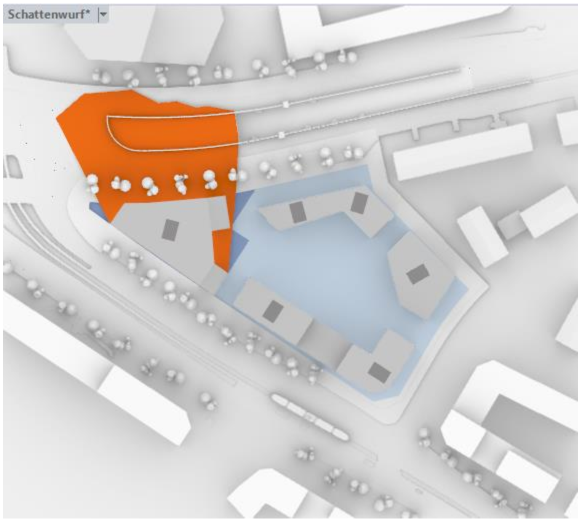

Figure 1: Visualization of the shading analysis in Rhino (orange), indicating the area a building casts shadows on for more than 3 hours a day.

BPS

A BPS model is then generated from the volumes, considering the different building usages and standard values from the *Schweizerischer Ingenieur- und Architektenverein* (SIA, 2021), such as U-Values, schedules, internal loads, ventilation, etc. The standard values correspond to the energy loads and systems of a building with average energy consumption. The BPS model also allows the input of SIA target values, which correspond to the energy loads and systems of an energy efficient building. We use Honeybee, a plug-in for Rhino / Grasshopper, that connects the parametric model with EnergyPlus, to run the simulations. The case study site is located in Zurich, Switzerland. Based on the complexity and size of the evaluated building variants, simulation times varied from around one to six minutes on an AMD Ryzen Threadripper 3970X, running on a single core at 4.2GHz. As these runtimes well exceed the 1.5 seconds delay that already disrupt creative workflows, a much faster solution for energy prediction must be found.

Surrogate Modeling

Introduction

Replacing BPS with SMs is a multi-step process including data generation and preparation, model training, evaluation of the model's accuracy and finally, integration of the SM into the design software. As this research aims to demonstrate how surrogate models can increase interactivity when exploring a given design space, training data must be generated directly from the variants and the input variables must allow broad DSE.

To get an overview on the performance of different SMs predicting heating and cooling loads on our data set, three regression algorithms were trained and evaluated on our data set. The non-linear Random Forest (RF) algorithm, the neural-network based Multilayer Perceptron (MLP), and the Linear Regression (LIN) algorithm.

Feature selection:

Machine-learning algorithms can find relationships between input variables (features) and output variables. They thereby function as black boxes, meaning they consider only the provided data. In case of predicting energy consumption, any descriptive variable of the parametric model can theoretically be used as a feature, even if they're abstract values such as WWR (Seyedzadeh et al., 2020). For the given DSE tool, we identified five key features: I) the variant's global shape, outputted by the generative design algorithm and represented by an index i, II) the number of floors, III-V) WWR of the North, East / West, and South facades, respectively and are thus the same as the input variables for the parametric geometry generator.

Data generation:

As mentioned before, BPS has high computational cost and long runtimes that may take minutes to hours. However, ML algorithms must be trained on reasonably large data sets, to prevent under-fitting, i.e., the model not being able to successfully identify relations within the data and thus not being able to predict precisely. As the proposed predictive model is deployed to reduce computing time when exploring the design space, it would be counter-productive to create an excessive data set that takes weeks or months to simulate. Therefore, the design space must be sampled to create a large enough subset. Latin Hypercube Sampling was used to get a near-random distribution of input variables. To assess the influence of increasing data set sizes on the accuracy of the SM, we sampled more than 2500 variants using the above-described simulation model. From the simulation results, heating and cooling loads are extracted and used as the labels for the SM.

Data preparation:

Both features and labels of the generated sample data must be normalized in order to be used to train a Supervised Learning algorithm:

$$z = \frac{x-\mu}{\sigma} \quad \textbf{(1)}$$

where μ is the mean of all samples:

$$\mu = \frac{1}{N}\sum_{i=1}^{N}(x_i) \quad \textbf{(2)}$$

and σ is the standard deviation:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2} \quad \textbf{(3)}$$

This prevents features with high variance to dominate the objective function, which would lead to the surrogate model being unable to learn from other features and therefore being unable to predict any new values (Westermann and Evins, 2019).

Results

Evaluation of ML Model:

A trained SM's accuracy can be represented by e.g., the root-mean-square error (RMSE) which is calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y})^{2}} \quad \textbf{(4)}$$

where y = simulated value; $\hat{y}$ = predicted value; N = data set size.

In order to not having to split the set into three partitions to train, test, and evaluate the model, cross-validation can be applied. The dataset must be split into two sets, a training partition, randomly created from 80% of the data and a testing partition containing the remaining 20% of the data for final evaluation. During cross-validation, the training set is split into k subsets called "folds". For each fold, the model is trained on k-1 folds and evaluated on the remaining subset, which is used as a testing set to compute indicators like e.g., accuracy. The final model results from the averages of the k-fold values and is evaluated on the testing set, based on RMSE. One benefit of this method is, that cross-validation is scored based on the coefficient of determination $\mathcal{R}^2 \epsilon (-\infty, 1]$, indicating the generalizability of a regression model. The closer $\mathcal{R}^2$ is to one, the better the model.

$$\mathcal{R}^2 = 1 - \frac{u}{v} \quad \textbf{(5)}$$

where u = residual sum of squares; v = total sum of squares. Fig.2 shows the CV scores of the three algorithms trained on our data set.

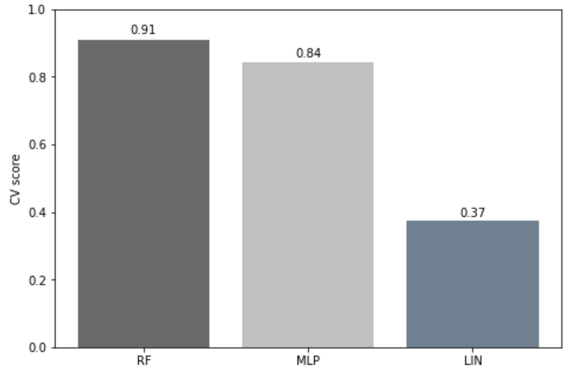

Figure 2: CV scores (R^2) of RF, MLP, and LIN on the training set.

Choice of regression algorithm:

The choice of the regression algorithm has a big effect on the performance of the surrogate model. (Rastogi et al., 2021) evaluated both linear and non-linear regression models predicting energy consumption based on RMSE. They have shown that non-linear models outperform linear ones and are therefore better suitable for this task. Fig.3 shows the RMSE of the RF, MLP, and LIN SMs trained on our dataset. Again, the two non-linear algorithms outperform the linear one, with RF being the best performing and therefore chosen algorithm to be integrated for demonstration. RF regression works by combining the results from a serious of decision tree predictors. Each tree in this forest is depending on the values of a random, independently sampled vector having the same distribution for all trees in the forest (Breiman, 2001).

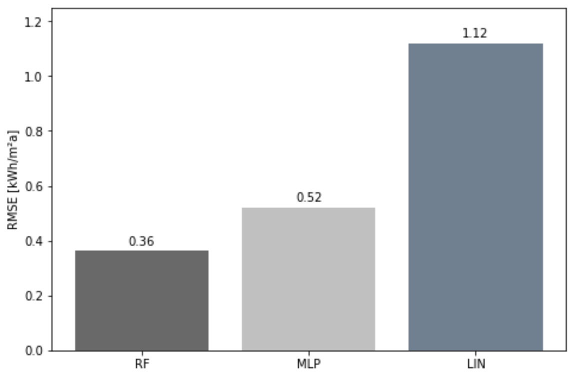

Figure 3: RMSE of RF, MLP, and LIN on the testing set.

Model accuracy and generalizability:

A SM's accuracy is highly dependent on the training data set size. Fig. 4 shows, how the RMSE of the SM trained on our data set increases rapidly at first and then starts to flatten out. While small data sets create high errors, the curve starts to flatten at around N=600. This tradeoff between long runtimes during data generation and model accuracy must be evaluated on a project-to-project basis. However, even with a relatively small data set size of N < 1000 samples, RMSE $\varepsilon \approx 0.7 kWh/m^2$ which corresponds to a relative error of around 10% which is more than good enough to guide early stage design decisions (Geyer and Schlüter, 2014).

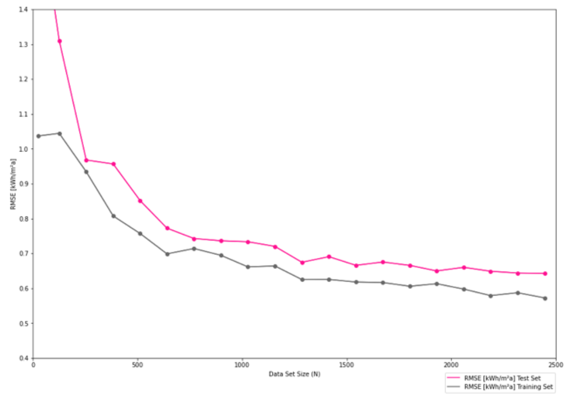

Figure 4: RMSE of testing (pink) and training data (gray) in relation to data set size.

Integration into Grasshopper:

To integrate the trained model into a Rhino / Grasshopper workflow and thus allowing the SM to predict energy consumption for design variants while modeling, two strategies were investigated. As Grasshopper cannot run CPython code natively, one must rely on plug ins. *GH_CPython* (AbdelRahman, 2017) provides a CPython editor and lets you import and use CPython modules inside Grasshopper. The *Hops* component, which is available for Rhino 7.4 and higher, lets users connect Grasshopper to a CPython Server. The big advantage of this approach is, that CPython code is solved outside of the Grasshopper environment, e.g., as a console application, which speeds up the process a lot. As a result, executing the CPython code inside Visual Studio Code and obtaining the results in Grasshopper through *Hops* takes around 50ms, while running the code directly in Grasshopper using *GH_CPython* takes more than a second. Compared to the roughly 5.5 minutes it took on average to simulate a building variant, the SM integrated through *Hops* is more than 6.500 times faster as the BPS, or 330 times faster when using *GH_CPython*.

Integration into Dashboard:

The real-time dashboard displayed in Fig. 5 aims to answer the questions of different stakeholders simultaneously and presents the geometry-based and performative analyses as described so far. Based on the building program, number and size of apartments as well as the needed amount of parking lots can be displayed. The extracted building components are used to calculate the embodied carbon impact as well as costs and rate of return. The analysis results are further used to calculate a couple of indicators from Swiss sustainable labels, such as *Minergie* (Minergie, 2008) and *Standard Nachhaltiges Bauen Schweiz* (SNBS) (NNBS, 2014). All 45 SNBS indicators can be calculated and a secondary dashboard allows the user to specify which qualitative indicators should be displayed. Those indicators calculated directly inside Grasshopper are automatically shown in the dashboard. To indicate *Minergie* standards, benchmarks of embodied and operational energy, as well as CO2 are integrated.

Discussion and Outlook:

This paper has shown how implementing Surrogate Models during design space exploration allows real-time evaluation of building performance metrics such as heating and cooling loads. The non-linear Random Forest regression algorithm has proven to be the most accurate on our data set, followed by the Neural-Network based Multilayer Perceptron algorithm. Linear Regression is not suitable for the presented use case. SMs rely on a reasonably sized training data sets, which must be generated for each design space specifically. While the simulations do not have to be supervised during data sampling and therefore data generation does not add work for the designer (simulations can be run over night or over the weekend), the computational cost of BPS can still lead to long runtimes, potentially taking days to compute the data set. However, the presented regression-based model already achieved acceptable mean errors on comparably small training data sets with $N_{training} \geq 500$ samples but accuracy and generalizability increased further with larger training set sizes. $N_{training} \geq 1500$ samples lead to well generalizable models with relative mean errors of less than 5%.

The largest benefit of integrating prediction models in an interactive early-stage design exploration workshop is the reduction of live computing time. Predicting energy demands with the SM integrated via Grasshopper's native *Hops* component is up to more than 6.500 times faster than running the BPS, allowing real-time evaluation, and therefore an integration of performative criteria into interactive DSE tools.

Connecting generative parametric models with visualization tools, displaying KPI data, is a powerful approach for guiding early-stage design decisions to find more sustainable building designs. With the presented dashboard, this paper has demonstrated how SMs allow a new way of early-stage design assessment, laying the foundation for new forms of data-driven, multidisciplinary decision making in conceptual design phases that enable both real-time feedback and rigorous data analysis.

However, the presented dashboard has only integrated one type of SM, predicting heating and cooling loads and was not yet tested in a real-world interactive design workshop with multiple stakeholders present. Further research could conduct user surveys and investigate how other performative criteria, such as daylight or airflow, can be assessed in real-time through SMs. Future research could also investigate how the presented approach can be used to integrate SMs in various other stages of a building's life-cycle, to not only provide better designs but to also close the Performance Gap (Frei et al., 2018) to effectively and efficiently reduce the carbon footprint of the built environment.

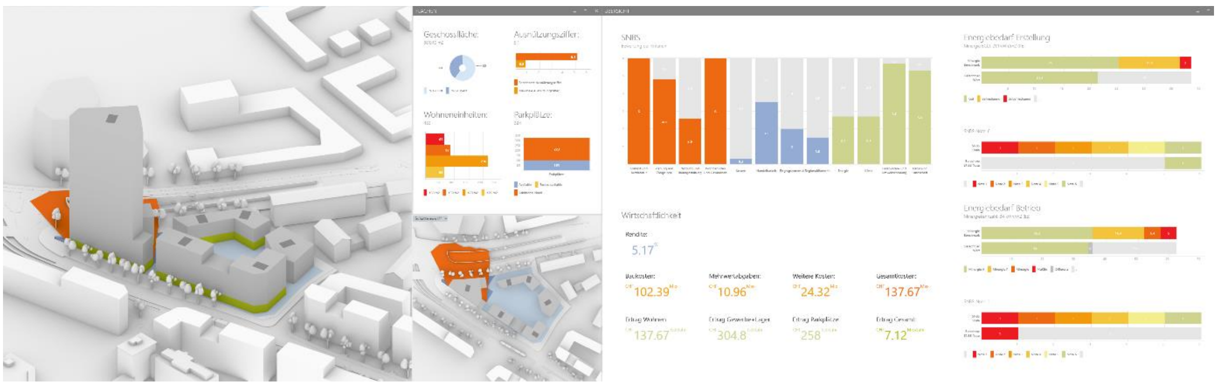

Figure 5: The dashboard concurrently showing the shading visualization and the key figures on the left, different cost and revenue structures as well as a SNBS-Overview in the middle, and Minergie benchmarks for selected indicators on the right.

Acknowledgement

This research was partially supported by the Deutsche Forschungsgemeinschaft (DFG; German Research Foundation) under Germany's Excellence Strategy – EXC 2120/1 - 390831618.

References

AbdelRahman, M., 2017. Mahmoudabdelrahman/Gh_Cpython: Alpha Release Of Gh_Cpython Plugin. Zenodo. https://doi.org/10.5281/ZENODO.888148

Breiman, L., 2001. Random Forest. Machine Learning 45, 5–32. https://doi.org/10.1023/A:1010933404324

Brown, N.C., 2020. Design performance and designer preference in an interactive, data-driven conceptual building design scenario. Design Studies 68, 1–33. https://doi.org/10.1016/j.destud.2020.01.001

Construction Users Roundtable, 2004. Collaboration, Integrated Information, and the Project Lifecycle in Building Design, Construction and Operation. (CURT Publication No. WP-1202). Available at: <https://kcuc.org/wpcontent/uploads/2013/11/Collaboration-Integrated-Information-and-the-Project-Lifecycle.pdf> [Accessed on 23 March 2022].

CRB Schweizerische Zentralstelle für Baurationalisierung. 2020. Elementbasierter Baukostenplan Hochbau eBKP-H (SN 506 511).

Frei, B., Sagerschnig, C., Gyalistras, D., 2018. ParkGap – Performance Gap Gebäude. Bestandsaufnahme und Handlungsempfehlungen für den Gebäudepark der Schweiz. Bundesamt für Energie BFE.

Geyer, P., Schlüter, A., 2014. Automated metamodel generation for Design Space Exploration and decision-making–A novel method supporting performance-oriented building design and retrofitting. Appl Energy 119, 537–556.

Minergie, 2008. The MINERGIE®-Standard for Buildings. Ver. Minergie Bern Switz. ISO 690

NNBS, 2014. Standard Nachhaltiges Bauen Schweiz – Hochbau SNBS. NNBS.

Pollination.cloud. 2022. Pollination. [online] Available at: <https://www.pollination.cloud/> [Accessed 10 March 2022].

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., 2011. Scikit-learn: Machine Learning in Python. MACHINE LEARNING IN PYTHON 6.

Rastogi, P., Emtiyaz Khan, M., Andersen, M., 2021. Evaluating the suitability of regression-based emulators of building performance in practice: a test suite. Journal of Building Performance Simulation 1–19. https://doi.org/10.1080/19401493.2021.1969430

Schweizerischer Ingenieur- und Architektenverein. 2021. Raumnutzungsdaten für die Energie- und Gebäudetechnik (SIA 2024).

Seyedzadeh, S., Pour Rahimian, F., Oliver, S., Rodriguez, S., Glesk, I., 2020. Machine learning modelling for predicting non-domestic buildings energy performance: A model to support deep energy retrofit decision-making. Applied Energy 279, 115908. https://doi.org/10.1016/j.apenergy.2020.115908

Simpson, T.W., Barron, K., Rothrock, L., Frecker, M., Barton, R.R., Ligetti, C., 2007. Impact of response delay and training on user performance with text-based and graphical user interfaces for engineering design. Res Eng Design 18, 49–65. https://doi.org/10.1007/s00163-007-0033-y

The American Institute of Architects. 2007. Integrated Project Delivery: A Guide. [online] Available at: <https://info.aia.org/siteobjects/files/ipd_guide_2007.pdf> [Accessed 10 March 2022].

Tseranidis, S., Brown, N.C., Mueller, C.T., 2016. Data-driven approximation algorithms for rapid performance evaluation and optimization of civil structures. Automation in Construction 72, 279–293.

Westermann, P., Evins, R., 2019. Surrogate modelling for sustainable building design – A review. Energy and Buildings 198, 170–186. https://doi.org/10.1016/j.enbuild.2019.05.057

Wortmann, T., Costa, A., Nannicini, G., Schroepfer, T., 2015. Advantages of Surrogate Models for Architectural Design Optimization. AIEDAM 29, 471–481.