**UNIVERSIDADE DE LISBOA**
**INSTITUTO SUPERIOR TÉCNICO**

# Surrogate Models for Efficient Implementation of Building Performance Analyses and Optimizations

**Gonçalo Roque Araújo**

**Supervisor**: **Doctor** Paulo Manuel Cadete Ferrão

**Co-Supervisors**: **Doctor** Maria da Glória de Almeida Gomes

**Doctor** Manuel de Arriaga Brito Correia Guedes

Thesis specifically prepared to obtain the PhD Degree in:

**Sustainable Energy Systems**

**Draft**

**Fevereiro 2025**

# Acknowledgments

First of all, I would like to thank my supervisors, Professors Paulo Manuel Cadete Ferrão, Maria da Glória de Almeida Gomes, and Manuel de Arriaga Brito Correia Guedes, for guiding this thesis and for their availability, support, and above all, their enthusiasm for this research. This work would not have been possible without their contributions, and much of this research's merit is theirs as well. They provided me guidance in the development of the research methods, as well as in interpreting research solutions, which yielded high-value research outputs. Their confidence and feedback also helped me to grow as a person and researcher.

I also sincerely appreciate the numerous opportunities provided throughout my academic years, both in theoretical contributions and practical collaborations. In particular, I owe a special thanks to Professor António Menezes Leitão for his friendship, and all the time he dedicated to guide me throughout my academic journey, from my Master's to my PhD. To Professor Ricardo Gomes, for providing valuable feedback and collaboration on numerous contributions throughout my PhD. To Professor Luís Santos, for providing me with research guidance, insights, suggestions, and collaborations.

I also want to give special thanks to my family who had a critical role in this long journey and made this thesis possible. In particular, I would like to thank my parents and brother for believing in me and supporting me throughout my life. To my wife, Inês, for always being there for me from the beginning to the end of this thesis, for tirelessly believing in me, even when I sometimes did not, and for constantly making me want to be a better person and researcher. To my friends Gonçalo and João, for their friendship and for showing constant enthusiasm in my research.

I am also extremely grateful to the members and ex-members of the Algorithmic Design for Architecture research group, headed by Professor António Menezes Leitão, for all their feedback, fellowship, knowledge sharing, and collaboration, particularly to Inês Pereira and Renata Castelo Branco. I greatly value their multiple research contributions and their support.

Lastly, institutional acknowledgments: This research was funded by Fundação para a Ciência e Tecnologia (FCT) through IN+ UIDP/EEA/50009/2020 — IST-ID and CERIS UIDB/04625/2020, and a Ph.D. grant under the contract of FCT 2021.04849.BD. This research was also funded by Project C-TECH—Climate Driven Technologies for Low Carbon Cities, grant number POCI-01-0247-FEDER-045919, LISBOA-01-0247- FEDER-045919, co-financed by the ERDF—European Regional Development Fund—through the Operational Program for Competitiveness and Internationalization—COMPETE 2020, the Lisbon Portugal Regional Operational Program—LISBOA 2020, and the FCT under the MIT Portugal Program.

**Título:** Modelos preditivos para uma implementação eficiente de análises e otimizações para o desempenho de edifícios
**Nome:** Gonçalo Roque Araújo
**Doutoramento em:** Sistemas Sustentáveis de Energia
**Orientador:** Doutor Paulo Manuel Cadete Ferrão
**Co-orientadores:** Doutora Maria da Glória de Almeida Gomes;
Doutor Manuel de Arriaga Brito Correia Guedes.


**Resumo**

Atualmente, o vasto conjunto de ferramentas de simulação e design disponíveis traz inúmeros benefícios para as práticas da engenharia, construção e arquitetura. Uma destas vantagens é a capacidade de reproduzir o ambiente construído de uma zona. Contudo, a maioria destas ferramentas exige modelos com uma descrição apropriada, e uma correta leitura dos resultados obtidos. A descrição destes modelos requer conhecimento especializado, e interpretações incorretas dos resultados podem induzir em erro potenciais soluções. Ainda, o tempo computacional aumenta significativamente quando se realizam um vasto número de simulações ou em modelos complexos. Para superar estes obstáculos, o estado da arte remete para o potencial do desenvolvimento de modelos preditivos com recurso a técnicas de Inteligência Artificial.

Estes modelos conseguem prever métricas de simulação e desempenho de edifícios de forma precisa, mais rápida e menos complexa. Isto resulta numa implementação mais eficiente de análises e otimizações do desempenho dos edifícios. No entanto, estes modelos geralmente não são portáveis, a sua precisão é altamente dependente da qualidade dos dados, e o seu desenvolvimento é complexo. Por isso, requerem uma abordagem sistemática para o seu desenvolvimento e utilização. Esta tese propõe uma metodologia para o desenvolvimento de modelos preditivos que consiga tornar mais eficiente a implementação de análises e otimizações de edifícios. A metodologia proposta abrange cinco etapas, desde a seleção das ferramentas até à implementação do modelo preditivo. O processo é ilustrado em quatro casos de estudo que utilizam diferentes bases de dados de edifícios: bases de dados reais, sintéticas e iterativas.

Os resultados das análises e otimizações dos estudos são documentados e discutidos, sendo realizada uma análise qualitativa que relaciona os diferentes tipos de bases de dados com a sua influência no modelo resultante. Quantitativamente, alguns resultados mostram que os modelos desenvolvidos podem aumentar de forma precisa a velocidade de simulação em cerca de 100 vezes e reduzir consideravelmente a complexidade de simulação ou cálculo da métrica respectiva. Por fim, são discutidos os resultados qualitativos e quantitativos, e delineadas e descritas possíveis investigações e aplicações futuras que podem emergir desta investigação.

**Palavras-chave:** Design Algoritmíco; Inteligência Artificial; Modelação Energética de Edifícios; Optimização; Modelos Preditivos.

**Title:** Surrogate Models for Efficient Implementation of Building Performance Analyses and Optimizations


**Abstract**

The large set of available simulation and design tools brings numerous benefits to urban and architectural practices. One of these advantages is the ability to reproduce an area's built environment. However, most of these tools require appropriate simulation inputs and outputs. Describing these inputs is laborious and requires expertise, and inappropriate outputs can mislead planning solutions. In addition, large simulations take considerable time to yield significant results. Particularly when performing multiple simulations and simulating large models. To overcome these obstacles, emergent solutions in research show the potential of developing Surrogate Models with the leverage of Artificial Intelligence or Machine Learning techniques. These models can predict simulation and performance metrics accurately, faster, and with fewer inputs. This results in a more efficient implementation of building performance analysis and optimizations. However, surrogate models usually lack portability, their accuracy is highly dependent on data quality, and their development is complex. Thus, they require a systematic approach for their development and use. This thesis proposes a framework for developing Surrogate Models that can make the implementation of building analyses and optimizations more efficient. The proposed framework encompasses five stages, from tool selection to model deployment, illustrated in 4 case studies that use different types of building information databases: Existing, Synthetic, and Iterative databases.

The results of the studies analyses and optimizations are recorded and discussed, and a qualitative analysis is performed relating the different database types to their influence on the model accuracy, number of input and target features, development speed, scope, qualities, and limitations. In particular, some results show that the developed surrogate models can increase the simulation speed by $\approx 100$ times and considerably reduce the required input features compared to their respective simulation or calculation inputs. Finally, qualitative and quantitative outcomes are discussed, and future research and applications that can emerge from this research are outlined and described.



**Keywords:** Algorithmic Design; Building Performance Optimization; Building Performance Simulation; Machine Learning; Surrogate Models.

**Título:** Modelos preditivos para uma implementação eficiente de análises e otimizações para o desempenho de edifícios


## Resumo Alargado

Atualmente, o vasto conjunto de ferramentas de simulação e design disponíveis traz inúmeros benefícios para as práticas da engenharia, construção e arquitetura. Uma destas vantagens é a capacidade de reproduzir o ambiente construído de uma zona, como o seu desempenho térmico, lumínico, estrutural, entre outros. Contudo, a maioria destas ferramentas exige modelos com uma descrição apropriada, e uma correta leitura dos resultados obtidos. A descrição destes modelos requer conhecimento especializado, e interpretações incorretas dos resultados podem induzir em erro potenciais soluções. Ainda, o tempo computacional aumenta significativamente quando se realizam um vasto número de simulações ou em modelos complexos. Para superar estes obstáculos, o estado da arte remete para o potencial do desenvolvimento de modelos preditivos com recurso a técnicas de Inteligência Artificial.

Estes modelos conseguem prever métricas de simulação e desempenho de edifícios de forma precisa, mais rápida e menos complexa. Isto resulta numa implementação mais eficiente de análises e otimizações do desempenho dos edifícios. No entanto, estes modelos geralmente não são portáveis, a sua precisão é altamente dependente da qualidade dos dados, e o seu desenvolvimento é complexo. Por isso, requerem uma abordagem sistemática para o seu desenvolvimento e utilização, particularmente na análise e previsão de ambientes construídos. Esta tese propõe uma metodologia para o desenvolvimento de modelos preditivos que consiga tornar mais eficiente a implementação de análises e otimizações de edifícios. A metodologia proposta abrange cinco etapas: (1) a seleção das ferramentas, (2) base de dados de edifícios, (3) tipo de modelo, (4) afinação do modelo, (5) implementação do modelo.

Na primeira etapa, selecionam-se as ferramentas necessárias para o desenvolvimento do modelo e/ou da sua base de dados. Isto pode incluir ferramentas de modelação geométrica, simulação e programação. Na segunda etapa (base de dados de edifícios) estabelece-se o tipo de dados mais indicados para treinar o modelo. Em diferentes casos estes modelos podem ser desenvolvidos através de bases de dados reais, sintéticas e iterativas. Na terceira etapa procura-se o tipo de modelo mais indicado para o problema. Este pode ser um problema de regressão ou classificação. Na quarta etapa afina-se o modelo(s) selecionado através de processos de melho-

ria da base de dados assim como de otimização dos Hiperparâmetros do modelo. Finalmente o modelo resultante é implementado para diferentes tipos de utilização.

O processo é ilustrado em quatro casos de estudo que utilizam diferentes bases de dados de edifícios: bases de dados reais, sintéticas e iterativas. Bases de dados reais são compostas por dados empíricos obtidos através de medições ou levantamentos de campo. Bases de dados sintéticas são geradas através de simulações de um amplo domínio de variáveis que representem o problema com eficácia. Finalmente, bases de dados iterativas são desenvolvidas através de processos de otimização baseados em simulação. Usam-se resultados iniciais para construir a base de dados, e acelera-se o processo posteriormente.

No primeiro caso de estudo desenvolve-se um modelo capaz de prever o certificado energético de um edifício com apenas 20 inputs. Este modelo foi desenvolvido com uma base de dados de $\approx 60000$ certificados em Portugal. Este modelo foi utilizado para desenvolver uma aplicação web para a previsão e otimização de um certificado energético, e outra para múltiplos certificados. Resultados obtidos com a otimização demonstra melhorias de 60% e 20% respetivamente. O segundo caso aborda o desenvolvimento de um modelo com 6 inputs capaz de prever uma simulação energética para um edifício situado em Lisboa, Portugal. Para isso desenvolve-se uma base de dados sintéticas corresponde a diferentes arquétipos de edifícios para diferentes épocas construtivas. Este modelo é implementado em uma aplicação web para a simulação e otimização de um edifício, e uma otimização implementada para a simulação e otimização da reabilitação de um bairro em Lisboa, Portugal. Resultados obtidos demonstram 16% e 8% de melhorias respetivamente. O terceiro caso aborda o desenvolvimento de um modelo com 4 inputs capaz de prever a simulação energética de um escritório com um envidraçado termocrómico com diferentes propriedades. Usou-se uma base de dados iterativa para desenvolver o modelo. Neste estudo implementou-se o modelo para otimizar as transmitâncias solares e visíveis, assim como a temperatura de transição deste envidraçado. Obtiveram-se melhorias de até 17% nas necessidades energéticas totais. O quarto caso de estudo usa também uma base de dados iterativa para desenvolver um modelo com 24 inputs capaz de prever a simulação energética de um complexo de 6 edifícios. Implementa-se o modelo para uma otimização dos materiais de construção do projeto.

Quantitativamente, alguns resultados mostram que os modelos desenvolvidos podem aumentar de forma precisa a velocidade de simulação até 100 vezes e reduzir consideravelmente a complexidade de simulação ou cálculo da métrica respetiva. Por fim, são discutidos os resultados qualitativos que comparam as diferentes bases de dados em relação a precisão do modelo,

número de inputs e objetivos, tempo computacional, utilização, qualidades e limitações. Bases de dados reais geram resultados com boa precisão e rápido desenvolvimento por utilizar bases de dados existentes, sendo ideal para operações de edifícios ou cenários que não exigem simulações. Contudo, a sua eficácia depende significativamente da qualidade dos dados. Bases de dados sintéticas geram previsões com menos inputs e objetivos, sendo mais adequado para projetos em estágios iniciais, embora exija mais tempo computacional devido à necessidade exponencial de simulações. A base de dados iterativa destaca-se em problemas de otimização, consegue conter mais inputs e objetivos, oferece um desenvolvimento mais rápido para tarefas com muitas simulações em estágios finais de projetos, mas apresenta maior margem de erro para soluções de baixo desempenho devido ao seu carácter de otimização. Cada uma destas abordagens possui características e limitações específicas que devem ser consideradas para cada caso de estudo. No final, interpretam-se os resultados, abrindo caminho para novas investigações e aplicações futuras na área do desempenho do ambiente construído.

**Palavras-chave:** Design Algoritmíco; Inteligência Artificial; Modelação Energética de Edifícios; Optimização; Modelos Preditivos.

# Contents

# List of Tables

# List of Figures

# Acronyms

**AD** Algorithmic Design. xxv, 2, 11, 16–20, 35, 37–40, 42, 59, 61, 83, 86

**ADA** Algorithmic Design and Analysis. xxv, 2, 3, 5, 11, 17, 20, 40–42, 46, 83, 85

**AEC** Architecture, Engineering, and Construction. 1, 3, 4, 19, 21, 23, 31, 37, 38, 45, 48, 127

**ANN** Artifical Neural Networks. xvii, 26, 30–32, 48, 49, 68, 109, 112, 113, 117, 119

**AOP** Analysis and Optimization Processes. xx, xxi, xxv, 1–6, 11, 12, 16, 23, 24, 31, 32, 35, 38, 40–42, 48, 58–61, 63, 65, 66, 69, 80–83, 92, 93, 106–108, 115–117, 121–129

**BID** Building Information Database. xviii, xx, xxi, xxvi, 2, 3, 5–7, 30–32, 35–48, 52, 54, 59–61, 63, 65, 66, 69, 73, 82, 83, 85–88, 93, 98–100, 106, 108, 109, 112, 116, 117, 121–128

**BIM** Building-Information Modelling. 37

**BPS** Building Performance Simulation. xix, xxv, xxvi, 1–7, 11–20, 22–24, 30–32, 35, 37, 39–46, 59, 83–85, 95, 101, 108, 112, 116, 117, 122–124, 126–128

**CAD** Computer-Aided Design. 1, 35, 37, 39, 59, 61, 83

**DHW** Domestic Hot Waters. 72

**EPC** Energy Performance Certificates. 7, 8, 30, 35, 65, 67–71, 73–79, 82, 122

**ET** Extra Trees. 49, 52, 54, 74, 75, 88

**EUI** Energy Use Intensity. 14, 19, 30, 83, 84, 87–89, 91, 122, 123

**FE** Feature Engineering. xxvi, 27, 28, 35, 36, 42, 56, 57, 65, 93, 126, 127

**GB** Gradient Boosting. 74

**GP** Gaussian Process. 25, 26, 30–32, 48, 49, 56

**H** Hypervolume. xviii, xix, xxv, 22, 23, 48, 58, 59, 108, 113–115, 117, 120

**HVAC** Heating, Ventilation, and Air Conditioning. 19, 66, 68, 72, 79, 81, 96, 98, 99, 101–103, 123

**IBEA** Indicator-Based Evolutionary Algorithm. 113–115

**IDE** Integrated Development Environment. 16–18, 40, 61, 95, 115, 122, 127

**KNN** K-Nearest Neighbors. 26, 31, 48, 49

**LR** Linear Regression. 86, 88

**LR** Logistic Regression. 48

**MAE** Mean Average Error. xvii, xxvi, 27, 88

**MAPE** Mean Absolute Percent Error. 27, 50

**ML** Machine Learning. 2, 12, 40, 42, 100

**MLP** Multi-Layer Perceptron. 49, 54, 57, 74

**MOO** Multi-Objective Optimization. xxv, 21, 22, 39, 46, 69–71, 74, 75, 82, 84, 85, 89–92, 94–96, 98, 99, 101, 106, 108, 109, 111, 112, 114, 122

**MSE** Mean Squared Error. 50

**NSGA** Non-Dominated Sorting Genetic Algorithm. xxi, 47, 59, 85, 89, 90, 99, 101, 112–115

**OMOPSO** Multi-Objective Particle Swarm Optimization. 113, 115

**PMV** Predicted Mean Vote. 15, 30

**$R^2$** Coefficient of determination. xvii, xx, 27, 28, 31, 32, 50–52, 54, 56, 57, 88, 91, 100, 108, 112, 113, 122–125

**RF** Random Forests. 88

**RMSE** Root Mean Squared Error. xvii, xx, 27, 50, 52, 54, 56, 57, 74, 75, 82, 88, 91, 100, 101, 112

**ROI** Return on Investment. 60–62, 69

**SDG** Sustainable Development Goals. 13, 94, 95, 129

**SM** Surrogate Models. xvii, xviii, xxi, xxvi, 2–7, 11, 12, 24, 25, 27, 28, 30–33, 35–37, 39–49, 51, 52, 56, 58–63, 65, 66, 68, 69, 72, 82, 83, 85, 88–93, 95, 98, 100, 101, 106, 108, 109, 112, 113, 116, 117, 121–129

**SMPSO** Speed-constrained Multi-objective Particle Swarm Optimization. 113, 115

**SOO** Single-Objective Optimization. xxi, xxv, 5, 21, 22, 39, 84, 85, 89–92, 108, 122, 123

**SVM** Support Vector Machines. 25, 26, 28, 31, 32, 48, 49

**TCG** Thermochromic Glazing. 66, 94, 95, 98, 99, 101–104, 106, 123

**UBEM** Urban Building Energy Models. 83

**UDI** Useful Daylight Illuminance. xix, xx, 14, 15, 20, 31, 44, 46, 61, 64

**VO2** Vanadium Oxide. 94

## Core Concepts

**Building Performance Simulation:**
In this research, Building Performance Simulation (BPS) refers to numerical calculations performed by an algorithm or tool to assess aspects of building performance. These simulations require specific inputs in different formats (Simulation models) to return the desired outputs. BPS can include energy simulations, daylighting, thermal comfort, and more.

**Analysis and Optimization Processes:**
Analysis and Optimization Processes (AOP) is referred to in this research as building and urban performance analysis processes that require large numbers of iterative evaluations. This can be the case with sensitivity analyses for specific building parameters and building performance optimizations.

**Algorithm Design and Analysis:**
Algorithmic Design and Analysis (ADA) is a merger of two fields: Algorithmic Design (AD) and BPS. AD is the process of describing shapes through algorithms, and ADA is a process to unify both AD and BPS by automating the generation and analysis of building designs.

**Multi-Objective Optimization:**
Multi-Objective Optimization (MOO) generally represents the core of building performance optimization approaches, since a building is a complex system that usually requires trade-off analysis. On the other hand, Single-Objective Optimization (SOO) focuses exclusively on one performance aspect of a building.

**Non-dominated solutions:**
In Multi-Objective Optimization (MOO), non-dominated solutions are known as optimum solutions. These represent solutions that cannot improve more for one objective without harming the other(s).

**Hypervolume:**
Hypervolume (H) is a metric to measure the objective-space (i.e., the domain of obtained solutions) covered by the non-dominated-solutions. This is useful to identify trade-offs involved in the optimization problem (e.g., more efficient buildings often entail higher costs).

**Surrogate Models:**

Surrogate Models (SM) are data-driven simplified representations of computationally expensive calculations. They are used in multiple professional fields including Medicine, Finance, Chemistry, Engineering, and Physics to approximate results faster than conventional methods, with high accuracy.

**Building Information Databases:**

Building Information Database (BID) are referred to in this thesis as the databases with building details used to develop the SM. They can be Existing BID (measurements, datasets, surveys), Synthetic BID (generated with multiple sampling methods through BPS), and Iterative BID (Generated through iterative processes, usually optimization)

**Feature Engineering and Selection:**

Feature Engineering (FE) and Selection is a process to select, clean, generate, and transform features (inputs) of a database to improve the SM performance. There are multiple methods to do this ranging between various degrees of complexity.

**Hyperparameter Optimization:**

Hyperparameter optimization is a process to find the best parameters required to develop the SM. It is also applicable to optimization algorithms. This is usually done by evaluating performance metrics for SM such as accuracy or Mean Average Error (MAE), and with the Hypervolume for optimization algorithms.

# Chapter 1

# Introduction

## 1.1 Context

The emergence of computers and new technologies expanded the capacity of Architecture, Engineering, and Construction (AEC) industry in the representation of their design, execution, construction, and rehabilitation projects. Computer-Aided Design (CAD) emerged and allowed practitioners a more accurate and faster way to represent design plans and construction details. This contributed to a significant growth in the AEC industry and consequently, an increased awareness regarding its environmental impact.

The environmental impact of the AEC industry is directly related to a building's performance, which was determined with numerical simulation approaches where practitioners performed complex and extensive calculations to obtain insights regarding performance aspects of a building such as thermal, structural, daylighting, acoustic, emissions, ventilation, among others. Thus, such calculations were performed by experts usually in the late stages of a design or construction project.

These calculations can now be performed faster with specific tools known as Building Performance Simulation (BPS) tools. These tools allowed practitioners to predict a building-specific behavior by inserting the required inputs and selecting the desired outputs. With these simulations, it was possible to integrate numerical Analysis and Optimization Processes (AOP) in the design and planning process, and not only in the execution project.

Unfortunately, the use of BPS tools with AOP is still out of reach for most practitioners and presents several limitations:

1. BPS tools are not portable and require different model formats and descriptions. This forces practitioners to develop models with different descriptions and inputs for each simulation, which is error-prone.

2. Most BPS are time-consuming and still take a considerable amount of time to perform the calculations for multiple or large models (e.g., it takes hours to simulate a neighborhood). This constitutes a significant limitation, particularly for AOP that require testing multiple variations of a project.

3. The use of BPS tools requires extensive knowledge regarding building physics and performed calculations to understand and process the inputs and results.

4. There is a trade-off between simulation detail and computational cost.

Algorithmic Design and Analysis (ADA) emerged as a way to solve the portability issue of BPS tools. ADA allows practitioners to describe a design project with algorithms, and analyze the design with BPS, all in one script. With this design process, it is possible to swiftly change parameters or design heuristics and obtain the respective model without effort. Additionally, some AD tools are either capable of supporting BPS tools or exporting models that can be read by them. Consequently, with AD it is possible to integrate AOP to automatically explore a design space, a process often referred to as Algorithmic Design and Analysis (ADA)

The use of ADA constituted a trade-off, which allowed practitioners to obtain higher portability between their project design, performance analysis, and optimization, at the cost of learning programming languages that have a high learning curve. However, ADA did not address the remaining limitations of BPS processes, which still require significant computation time, expertise and testing to be successfully applied. Thus, the use of AOP even though possible, presents the following limitations:

5. Because of the iterative nature of these processes, they add another layer of computational cost to the one required by BPS.

6. Setting up and performing the process and learning a programming language adds another layer of complexity.

7. Lacks systematic approaches to different building performance aspects and problems.

8. Lacks friendly interfaces for results post-processing.

Coincidentally, recent advances in research have been demonstrating the advantages of Surrogate Models (SM) developed with Machine Learning (ML) techniques, such as supervised learning, unsupervised learning, reinforcement learning, and deep learning, to help practitioners execute their Analysis and Optimization Processes (AOP). SM predict a target output after being fitted with a database that illustrates its variation according to specific training features. Particularly, SM developed with a project database can help reduce the number of required inputs and features while significantly improving BPS and AOP computational times. This allows for faster, and easier AOP. Applied to building performance, Building Information Database (BID) containing building features and their performance indicators can be used to train SM and overcome the above-mentioned obstacles.

The research available for the application of SM in building performance is limited, however, many studies highlight that these models have the potential to play a crucial part in the future of building performance and achieving sustainable and development goals. Nonetheless, these studies also outline many obstacles that need to be overcome to achieve this goal:

9. SM are usually specific to the problem at hand and lack portability.

10. Their accuracy highly depends on data quality and/or simulation detail.

11. Improving their accuracy is challenging and complex.

12. There are not many SM capable of explaining the uncertainty of their prediction results.

13. The computational cost for generating building databases with BPS (if necessary) can increase exponentially to the point of unfeasibility.

14. It is challenging to develop a SM and select the most suitable model.

In this thesis, a framework for developing SM to improve building performance AOP is developed and evaluated. This framework aims to build a comprehensive approach that can be applied systematically to overcome these challenges and improve the integration of AOP for building and urban projects.

The presented framework encompasses 5 stages, starting by identifying the building performance problem and required toolkit, identifying different types of BID that can be used, selecting the adequate SM, improving the SM performance if necessary, and deploying the model. This process is then implemented and evaluated in different case studies representative of three different types of BID: Existing BID, Synthetic BID, and Iterative BID

## 1.2 Motivation

Buildings in their lifecycle are responsible for $\approx 40$ % of global energy use, 25% water use, and 39% carbon emissions [1, 2, 3]. As such, there is a dire need to renovate the built environment and improve future construction to achieve global sustainable development goals. In the Architecture, Engineering, and Construction (AEC) industry, some practitioners use Analysis and Optimization Processes (AOP) to assess their building or urban projects' future performance. Current state-of-the-art methods to perform these processes allow the user to obtain the best solutions for their projects that guarantee performance standards and goals. However, a building is a complex system entailing multiple performance aspects, and AOP are time-consuming, difficult to employ, and lack portability between tools.

Because of the iterative nature of AOP, multiple evaluations of the problem are required to use these processes. In addition, to perform the required numerical calculations for building performance aspects, BPS tools are generally used. These tools present a trade-off between simulation detail and computational cost. All these factors increment the computational cost and time of AOP. Moreover, because these tools and processes are difficult and not portable, they demand a large interdisciplinary knowledge to correctly asses multiple aspects of building performance, such as being experienced with multiple BPS tools and Algorithmic Design and Analysis (ADA).

Research has been trying to overcome these challenges using Surrogate Models (SM) to replace expensive computational calculations. SM can generate accurate predictions much faster and with fewer inputs, making them a more suitable and easier alternative than BPS to perform building performance AOP. However, there are still significant research gaps to be overcome for SM to be effective in building performance. They are specific to the prediction task, challenging to develop, and their databases often require time-consuming processes to be suitable for training.

The motivation for this thesis emerges from the need to improve the use of AOP in building performance and expand the reach of these processes to all practitioners in the AEC industry. To do this, it is crucial to overcome the challenges associated with BPS, AOP, and particularly, SM for building performance. In the following sections, this research question is presented and discussed, and this thesis's goals are outlined.

## 1.3 Research Question

The proposed PhD research was structured by the following research question:

- How to improve the integration of Analysis and Optimization Processes (AOP) in building and urban projects?

To answer this question one can deconstruct this research question into three main sub-questions regarding AOP:

- How to make them faster?
- How to make them easier?
- How to make them portable?

SM provide an answer to these questions, but not without challenges. Current literature highlights the challenges of developing a SM for building performance AOP: they depend on data quality; their development is complex and significantly undocumented in the research; and still lack portability between different problems. With this in mind, the next Section briefly describes this thesis while outlining its research goals.

## 1.4 Research Methods

To answer the previously stated research questions, this thesis proposes a framework for a systematic approach to SM development in building performance AOP. This framework is implemented and evaluated in case studies concerning different types of databases. With this approach, a researcher or practitioner can develop and implement a SM from start to finish, for their project or practice. The models can be used by any user familiar with the required inputs for the SM and can be implemented for any building performance problem. The methodological advances of this framework encompass:

1. Providing a systematic approach to develop a SM for any building performance problem.
2. Identifying different kinds of databases and providing suitable approaches to enhance the data quality.

3. Providing workflows to select the most suitable SM for the problem at hand.
4. Providing workflows to increase the selected model's accuracy if needed.
5. Providing workflows to implement the models appropriately, depending on the building performance problem scope.

An outline of this thesis is provided in the next Section, with a particular focus on its structure.

## 1.5 Thesis Outline

This thesis is structured in 5 Chapters, including the present. This Section provides a brief outline of each Chapter in the next paragraphs.

In Chapter 2, a large corpus of knowledge is gathered in three separate sections concerning Building Performance Simulation (BPS), Analysis and Optimization Processes (AOP), and Surrogate Models (SM). In Section 2.1 (Building performance simulation), multiple performance aspects of BPS are investigated along with available tools and performance metrics. Afterward, ADA is described, along with its tools and applications for building performance AOP (Section 2.2). Section Surrogate Models documents different model types that can be used, and how they are evaluated, improved, and deployed in building performance problems. Each field's Challenges and opportunities are identified at the end of each Section.

Chapter 3 comprises 5 Sections to describe the proposed framework in practice. Section 3.1 (Tools) suggests a suitable toolkit to apply the framework, and Section 3.2 (Building Information Databases) expands on the different types of Building Information Database (BID) considered for this study. Section 3.3 (Model type) documents the implementation of some SM types, and Section 3.4 (Model tuning) provides an adaptable workflow to improve the SM accuracy. Finally, Section 3.5 (Model deployment) suggests and describes possible SM implementation formats for different problems.

Chapter 4 documents the framework application in case studies for the three building databases researched in this thesis. This part is composed of 4 Sections, each culminating in a core scientific publication representative of this thesis framework.

In Section 4.1 (Optimization of building retrofit using a Surrogate Model developed with an Existing Building Database), a SM is developed to predict the Energy Performance Certificate of a building with an Existing BID. The resulting SM is implemented in a web app to predict, analyze, and optimize any building energy certificate in Portugal.

Section 4.2 (Optimization of Building retrofit and design using a Surrogate Model developed with a Synthetic Building Database) implements the framework for a SM capable of predicting a building energy simulation result of a building in Lisbon. The model is trained using a Synthetic BID, and deployed for a problem to optimize the retrofit of a building block, and for a Single-Objective Optimization (SOO) of building design in a web application.

In Section 4.3 (Optimization of thermochromic glazing using a Surrogate Model developed with an Iterative Building Database) a SM is developed to predict the energy use of an office containing glazing with different phase-change-material properties. The model is trained with an Iterative BID and implemented to optimize the glazing properties that yield minimum energy use.

Finally, Section 4.4 (Optimization of a residential block using a Surrogate Model developed with an Iterative Building Database) develops a SM to predict the total energy use and standard deviation of a mixed-use building block. The model is also implemented for an Iterative BID. However, this study covered stage 4 of the framework (Section 3.4 - Model Tuning). This SM was implemented for an optimization problem concerning the construction of the block.

Chapter 5 documents the main quantitative and qualitative findings. This is done in Section 5.1 (Main Research Findings) by evaluating and comparing the different types of SM developed, their BID, and their respective AOP. Finally, Section 5.2 (Future Work and Applications) suggests directions for related research and possible applications of this framework in a distant but reachable future.

## 1.6 Research Outputs

In this section, research outputs are divided into two groups. The first group encompasses the implementation of the 4 case studies and a brief description of the main published documents. The second group outlines articles and papers published to support the proposed framework. In each group the research outputs are organized chronologically. In total 5 journal articles, 4 conference papers, 2 conference posters, and 1 book chapter were published.

**Conference paper:**

**Status:** Published

Araújo, G. R.; Santos, L.; Leitão, A.; and Gomes, R. (2022). "AD-Based Surrogate Models for Simulation and Optimization of Large Urban Areas." In 27th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA 2022), 689–98. Sidney. doi: `https://doi.org/10.52842/conf.caadria.2022.2.689`

This article comprised the development of a SM to predict BPS results for a building in Lisbon, Portugal. This consisted of simulating multiple building variations for different geometric and material variables. As part of this thesis's research, the model was developed as a case study to illustrate the proposed framework's process for Synthetic BID. The resulting model demonstrated acceptable accuracy with only six input variables and reduced simulation times by $\approx$ 100-fold. The model was implemented to solve an optimization problem that aimed to minimize energy use, its standard deviation among the buildings, and the total rehabilitation cost of a neighborhood of 21 buildings.

**Journal article:**

**Status:** Published

Araújo, G. R.; Teixeira, R.; Gomes, M. G.; and Moret Rodrigues, A. (2022). "Multi-Objective Optimization of Thermochromic Glazing Properties to Enhance Building Energy Performance." Solar Energy 249

(October 2022). Elsevier Ltd: 446–56.

In this article, a SM was developed to predict BPS results for an office space with a Thermochromic Glazing installed. Thermochromic glazing is a system with a Phase-Change-Material coating that varies its thermal properties according to the surface's temperature. The SM was developed with an Iterative BID extracted from simulation-based optimization results. This case study illustrates the proposed framework process for Iterative BID without any model tuning techniques. The model was implemented to find the best glazing properties that yielded minimum lighting, heating, and cooling energy use. The developed model showed acceptable accuracy, significantly reduced the simulation time, and required only four input features. Moreover, the improved optimization process found the inherent trade-offs that occurred when the glazing switched states, between the increased lighting and reduced heating and cooling energy use.

**Journal article:**

**Status:** Published

Araújo, G. R.; Gomes, R.; Gomes, M. G.; Correia Guedes, M.; and Ferrão, P. (2023). "Surrogate Models for Efficient Multi-Objective Optimization of Building Performance." Energies 16: 4030. doi:`https://doi.org/10.3390/en16104030`

This article described the development of a SM capable of predicting BPS results of a fixed geometry 6-building block in Lisbon, Portugal. The model was developed with an Iterative BID extracted from a simulation-based optimization with 24 feature variables: wall, roof, floor, and window construction materials for each building. This case study illustrates the proposed framework process for an Iterative BID with model tuning techniques. The developed model demonstrated outstanding accuracy, significantly reduced the optimization process's simulation time (approximately 80-fold), and required only four inputs per building. The optimization results illustrate the existing trade-offs between construction costs, energy use, and energy use standard deviation between buildings.

**Journal article:**

**Status:** Published

Araújo, G. R.; Gomes, R.; Ferrão, P.; and Gomes, M. G. (2023). "Optimizing Building Retrofit through Data Analytics: A Study of Multi-Objective Optimization and Surrogate Models Derived from Energy Performance Certificates." Energy and Built Environment. doi: `https://doi.org/10.1016/j.enbenv.2023.07.002`

In this article, a SM was developed to predict a building or property Energy Performance Certificates (EPC) and energy use. The model was trained with an Existing BID consisting of $\approx$ 65000 EPC in Portugal and 20 input features. This case study illustrates the proposed framework process for developing a SM with an Existing BID and model tuning techniques. In this case, the resulting model requires fewer inputs than an energy certification process and shows acceptable accuracy. Furthermore, it was implemented in a web app to help users predict and obtain energy information regarding their EPC, and optimize it by considering retrofit variables promoted by the Portuguese government.

Other publications stemmed from this research to provide foundational support for the proposed framework:

**Journal article:**

**Status:** Published

Araújo, G. R.; Pereira, I.; Leitao, A.; and Guedes, M. C. (2021). Conflicts in passive building performance: Retrofit and regulation of informal neighbourhoods. Frontiers of Architectural Research, 10(3), 625-638.

**Journal article:**

**Status:** Published

Aleixo, J.; Araújo, G. R.; and Guedes, M. C. (2021). Comparison of passive design strategies to improve living conditions: a study in Ondjiva, Southern Angola. Renewable Energy and Environmental Sustainability, 6, 21.

**Conference poster:**

**Status:** Published

Araújo, G. R.; Leitão, A., Pereira, I.; Gomes, R., and Ferrão, P. (2021). A non-linear surrogate model of building archetypes to speed up cities' adaptation to the post-carbon age. In Congresso MITPortugal 2021.

**Conference paper:**

**Status:** Published

Araújo, G. R.; Teixeira, H.; Gomes, M. G., and Moret Rodirgues, A. (2022). Otimização de Envidraçados Termocrómicos para um Clima Mediterrânico. Congresso Construção 2022, 239.

**Conference poster:**

**Status:** Published

Araújo, G. R.; Gomes, R., and Ferrão, P. (2022). Surrogate models for time-consuming building performance simulations and optimizations. Congresso MITPortugal 2022.

**Book chapter:**

**Status:** Published

Guedes, M. C.; Araújo, G. R.; and Albuquerque, N. (2023). Thermal Comfort in Informal Settlements: Case Studies in Sub-Saharan Africa. In Climate Change and Sustainable Development (pp. 129-148).

CRC Press.

**Conference paper:**

**Status:** Published

Araújo, G. R.; Gomes, R.; Gomes, M. G., Ferrão, P.; Guedes, M. C. (2024). Análise das necessidades energéticas do parque edificado de lisboa com recurso a técnicas de IA. Congresso Construção 2024.

**Conference paper:**

**Status:** Published

Araújo, G. R.; Gomes, R.; Gomes, M. G., Ferrão, P.; Guedes, M. C. (2024). Técnicas de IA para implementar processos de simulação e optimização do desempenho de edifícios. Congresso Construção 2024.

This thesis also achieved **3rd place** in the 7th edition of the PhD Open Days pitch competition, hosted by Técnico Lisboa in 2021.

# Chapter 2

# State-of-the-Art

## Preamble

Urban growth is steadily increasing, and at the present rate, it is estimated that the population in urban areas will have increased from 55% to 60% by 2030 [1]. This leads to two significant challenges: (1) development and improvement of urban areas with a growing scarcity of resources; and (2) addressing sustainability and climate goals [4, 2]. To tackle these challenges, recent design and construction practices resort to simulation-based approaches to test solutions when they cannot be tested empirically [5, 6, 7]. These approaches can help practitioners with their decisions by extrapolating data from simulations and understanding how these decisions affect the urban metabolism of a city [8, 9]. Particularly, Building Performance Simulation (BPS) approaches can help reduce the built environment's energy use by predicting building performance indicators such as energy use, thermal performance, and daylighting, among others [10, 11, 12].

Despite the advantages of these approaches most have constraints that are difficult to comply with. BPS requires an extensive understanding of the respective model's complexity, inputs, workflows, and outputs [13]. Lacking any of these is a problem that can contribute to critical errors by even the most experienced users [14, 15, 16, 17]. Additionally, simulations are time-consuming when testing large building and urban models [5, 10, 13]. The iterative process typical of Analysis and Optimization Processes (AOP) for building performance combined with the difficulty and diversity of tools and analyses for different simulations, creates a ripple effect on the impact of the problems mentioned above.

To solve some of these problems, Algorithmic Design and Analysis (ADA) emerged to unify and automate building performance AOP. Algorithmic Design (AD) is described as a design system in which an algorithm generates geometry based on specific variables and relationships [18, 19]. It's scalability allows the automatic generation of design solutions and its portability enables the integration of simulation tools and optimization processes [18]. This mechanizes most steps typical of AOP. Albeit, setting up the BPS is still complex, with the added challenge of learning a programming language to automate the processes [20].

Previous attempts to predict faster BPS results and with fewer inputs involved Surrogate Models

(SM) developed with Machine Learning (ML) techniques [21]. These models have been used to predict building simulation outputs such as building carbon emissions [22], energy consumption [23], and day-lighting [24]. ML models are trained with a simulated case-specific database and substantially improve simulation run-time, deliver faster and more accurate results, and promote a smoother integration with current digital design workflows [25, 26]. However, such models are usually case-specific, and significantly depend on data quality, [27, 28].

So far, SM have not been used for their potential to bridge the gap between BPS and AOP with practitioners, thus, state-of-the-art building performance Analysis and Optimization Processes (AOP) are still out of their reach. Chapter 2 of this thesis unveils the main challenges and opportunities associated with BPS, AOP, and SM for building performance.

## 2.1 Building performance simulation

Building Performance Simulation (BPS) has emerged to support building and urban project decisions by automating numerical calculations to deduce building performance metrics. With BPS tools, users can access quantitative insights regarding various aspects of building performance such as energy, daylight, and indoor air quality, among others [29]. These tools can be used from the early to late stages of building projects, and have a long history of use to meet Sustainable Development Goals (SDG) [30].

To understand the scope of the field of BPS it is possible to search by keywords through the Scopus database [31]. Figure 2.1 shows a line plot of the published articles with the keyword *"Building Performance Simulation"* in combination with a stacked area chart of the published articles with the keywords *"Building Performance Simulation"* and *"Energy", "Comfort", "Daylight"*. It is visible that the field has experienced a large growth since 2005 and that energy simulations are responsible for most of the studies published within the scope of BPS.

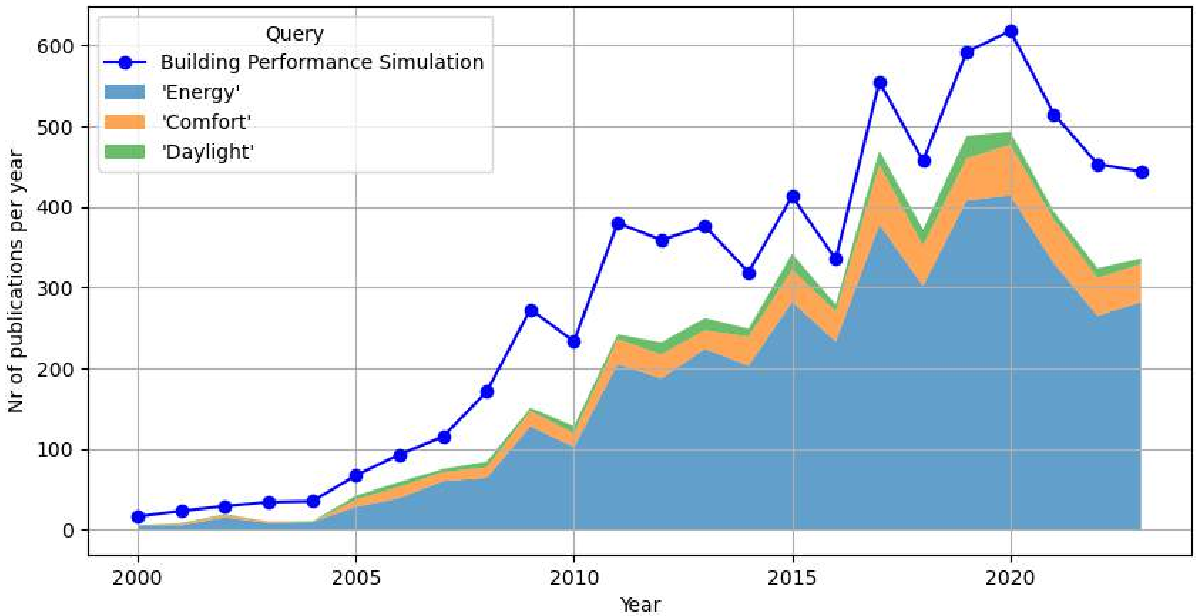


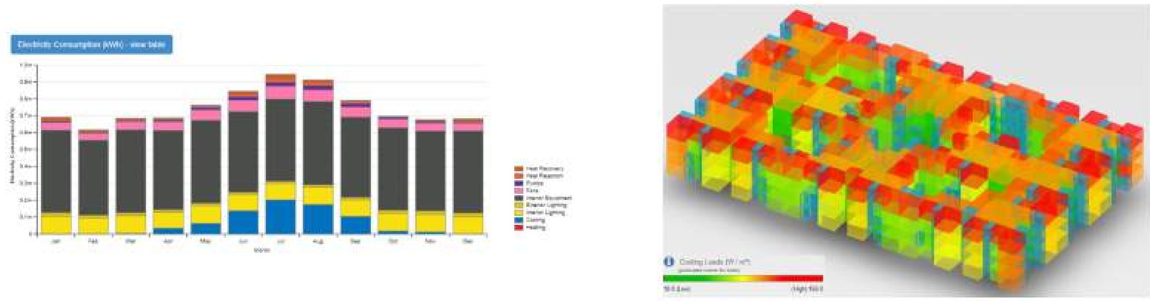

Figure 2.1: Histogram plot of the yearly number of publications containing the keywords *"Building Performance Simulation"* and *"Energy", "Comfort", "Daylight"*.

### 2.1.1 Energy

Multiple BPS tools can be used to calculate the energy performance of buildings. Namely, EnergyPlus [32], Open Studio [33], Design Builder [34], Modelica [35], Revit Insight [36], among others. These tools perform numerical calculations according to user-specified inputs about the system that is being simulated [16]. In the case of buildings, common inputs are weather files, materials, geometric descriptions, electric equipment, occupancy schedules, and surface boundary conditions, among others [37, 38]. The user can specify the period and time step of the calculations, along with different calculation methods [37].

When evaluating the energy performance of buildings using BPS, multiple outputs can be used, including Energy Use Intensity (EUI) [39], which represents the amount of energy consumed per unit of floor area, typically in $\mathrm{kW\,h\,m^{-2}}$. Other used BPS outputs include heating and cooling loads, peak electrical demand, and energy cost [40, 41, 42]. These metrics can be used to assess the energy performance of buildings and evaluate the impact that various parameters have on them [17, 43, 44].

Figure 2.2 shows different output visualizations. In (a), a monthly chart of electricity consumption is returned by a simulation in Openstudio [33]; In (b), the annual cooling loads of an urban area are simulated and visualized by floor in a 3D model simulated with Revit Insight [36]. BPS tools for energy performance have a wide variety of output formats for different time frames.



(a) Monthly Electricity Comsumption simulated in Openstudio [33]

(b) [Annual cooling loads simulated with Revit Insight [36]

Figure 2.2: Energy BPS output visualizations. (a) retrieved from OpenStudio [45], and (b) retrieved from Autodesk-University [46]

### 2.1.2 Daylight

To perform Daylight studies in buildings, some BPS tools that can be used are Radiance [47], Diva for Rhino [48], among others. Required inputs typically encompass the building's geometric description, material properties, location, and climate data such as weather files. These inputs can be used to calculate how natural light interacts with the building and inddor environmental model. BPS tools that provide Daylight simulations typically encompass multiple simulation parameters and sky models, such as the standard sky conditions [49], Climate-based sky [50], and Tregenza sky [51]. The user can define the position of single or multiple sensors in the geometric model where the tool performs the calculations and returns illuminance values in $\mathrm{lx}$ for the specified sensor(s).

With the illuminance values returned by the BPS, other performance metrics of Daylight in buildings can be calculated such as the Daylight Factor [52, 53], Useful Daylight Illuminance (UDI) [54], Annual Sunlight Exposure, and Daylight Autonomy [55]. Moreover, other visual comfort metrics can be calculated like Daylight Glare Probability [56] and Glare Index [57]. These metrics can help measure the building's natural lighting performance. Users can consider different glazing and material parameters [58, 59, 60], as well as experiment with multiple designs and orientations [61, 62, 63].

Figure 2.3 shows the UDI for different facade depths and window heights. Because Daylight BPS typically entails calculations for multiple sensors, outputs are usually visualized in spatial heat maps. With these outputs, users can assess the impact of their design strategies in multiple building areas.

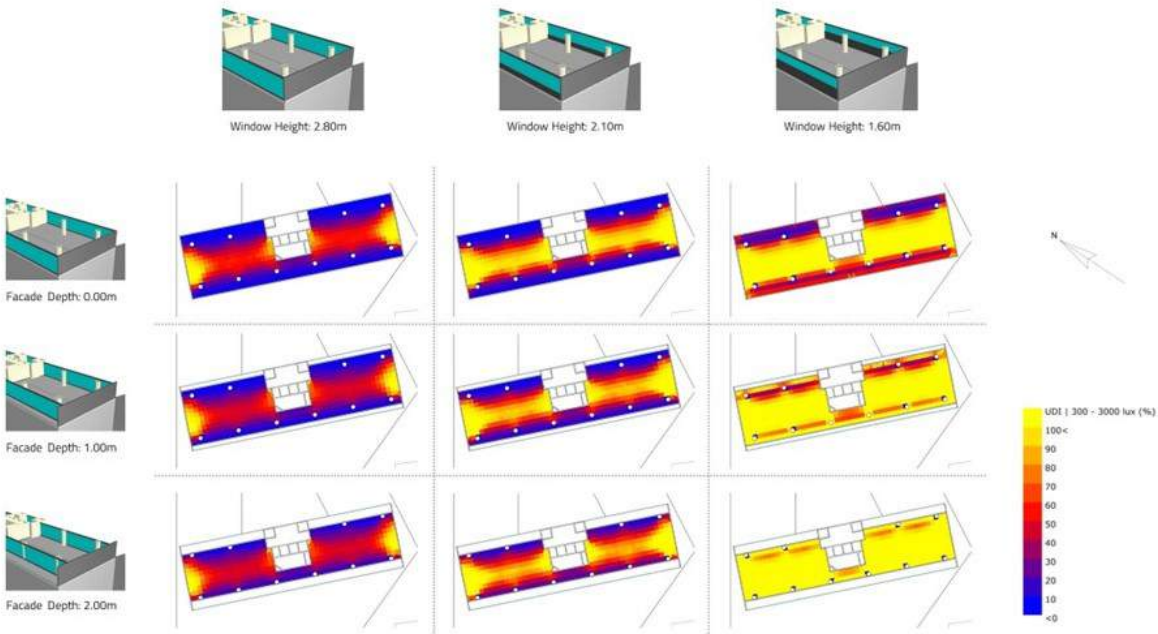



Figure 2.3: Useful Daylight Illuminance (UDI) for different building solutions. ©G. Reis. Retrieved from [64]

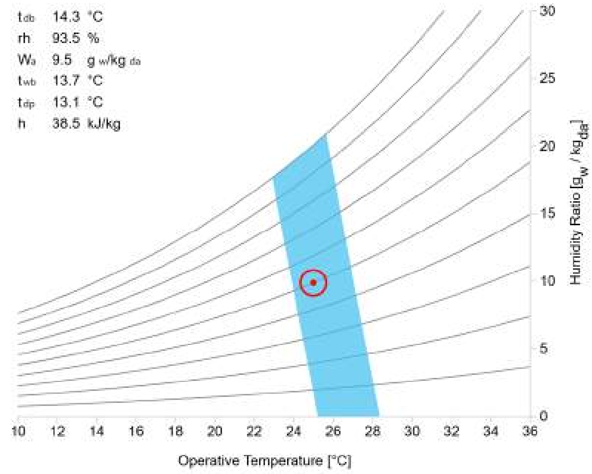
### 2.1.3 Comfort

BPS tools that perform energy simulations (e.g., EnergyPlus) can also assess thermal comfort by calculating outputs like indoor temperature, humidity, and other thermal outputs to calculate comfort metrics [65]. Commonly used metrics are the Predicted Mean Vote model [66], adaptive comfort model [67], Givonni bioclimatic model [68], and thermal autonomy [69], among others. With BPS tools users can calculate these metrics and experiment with different equipment, materials, and designs to explore more comfortable solutions [70, 71].

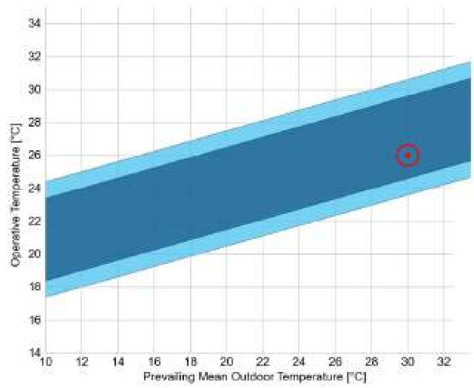
Figure 2.4 shows output formats and visualization examples obtained from Tartarini et al. [72]. In (a) the Predicted Mean Vote (PMV) metric is illustrated for a prevailing indoor temperature [66, 72]; in (b) the ASHRAE 55 standard for Adaptive comfort is visualized [67, 72]. All these results can be analyzed and compared between multiple parameters to find the best solutions for different aspects of building performance [73, 74]. Furthermore, these outputs can help achieve spatial lighting or thermal effects solicited in the project [75, 76].

### 2.1.4 Challenges and Opportunities

Despite its merits, BPS presents several shortcomings concerning the uncertainty of the obtained results and the difficult implementation of BPS tools [15, 17, 74, 77]. Harish and Kumar [74] perform a literature review of modeling methodologies of building energy systems. Results highlight a trade-off between




(a) Predicted Mean Vote chart according to ASHRAE 55 standard [72]. Comfort threshold highlighted in blue




(b) Adaptive Comfort chart according to ASHRAE 55 standard [72]. Comfort thresholds highlighted in blue.

Figure 2.4: Comfort BPS output visualizations. (a) and (b) Retrieved from Tartarini et al. [72].

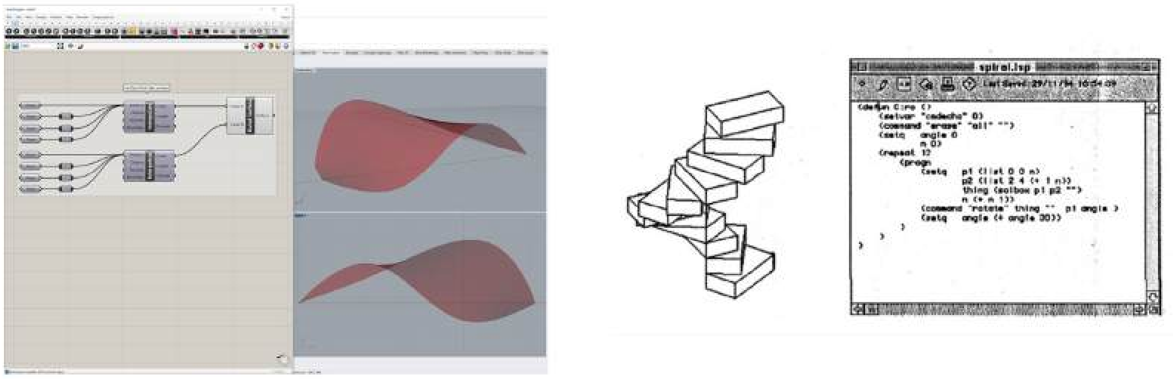
simulation accuracy and computation time. Moreover, the authors mention that more simplified building models are needed. Attia [16] performs a user survey of architects' and engineers' selection and use of BPS and outlines main implementation challenges and opportunities based on previous studies and user survey findings. The main findings show that practitioners require accurate, practical, and portable tools that can be easily integrated with design processes.

According to these studies, integrating simple, accurate, and fast BPS tools with design tools is paramount to overcoming the mentioned obstacles. This allows users to perform accurate parametric analyses concerning building and design alternatives. Some design tools already include BPS tools as plug-ins, such as Sketchup [78], Revit [36], Rhino [79, 80], among others. In particular, these tools also provide the option to algorithmically describe the building project, which can automate design tasks, and test multiple possibilities through BPS. The next Section elaborates on the Algorithmic Design field and its role in integrating BPS in Analysis and Optimization Processes (AOP).

## 2.2 Algorithmic Design and Analysis

Algorithmic Design (AD) is a design process that leverages algorithms and computational methods to incorporate the iterative power of computers in the design workflow [81, 20]. Other similar and related terms are "Parametric Design" and "Generative Design" [20, 82, 83, 84]. Caetano et al. [20] performed a study to discuss the computational terminologies generative, parametric, and algorithmic designs based on an extensive literature review. The authors argue that with AD, one can use parametric and generative design, while the opposite is not verified.

The use of AD has been gaining prominence to develop flexible, accurate, and automated designs [81, 85]. This is mainly done through an Integrated Development Environment (IDE) for a visual or textual programming language [86]. Using visual programming languages consists of a combination of visual components in an IDE illustrating functions that require specific design inputs to run, and return design outputs. Textual programming languages use a textual IDE to write and define functions and operations the computer needs to execute [87].



(a) Script to design a surface in a visual programming language. ©Brendan Harmon. Retrieved from Harmon [88]

(b) Script to design spiraling stairs in a textual programming language. ©Gabi Celani, retrieved from CELANI [89]

Figure 2.5: Visual and Textual programming languages.

Despite its advantages, AD still presents shortcomings. In particular, AD requires users to learn a programming language, and visual programming languages lack scalability, which makes them unsuitable for complex design tasks [90, 87]. Learning a programming language can be difficult, while complex models become slow and can crash when using visual programming languages [86, 91, 92].

AD can also integrate BPS tools to automate simulation tasks while changing its parameters [19]. Aguiar et al. [18] develops a framework entitled Algorithmic Design and Analysis (ADA) for this process. The iterative nature of ADA is suitable for such tasks, thus, analyses and optimizations can be implemented with different BPS tools, all within one AD script [18]. This can help overcome the portability barrier presented by BPS tools (Section 2.1, see Challenges and Opportunities). In this Section, multiple visual and textual AD tools are described, and their use in studies for different design stages is explored.

### 2.2.1 Algorithmic Design Tools

While textual programming tools emerged first, recently visual programming tools have been increasing in popularity, particularly in the architectural field, since they are more intuitive and user-friendly, with a quick initial learning curve [90, 93]. However, a trade-off emerges since using visual components becomes overwhelmingly difficult to comprehend and work with for larger programs [87, 90, 94, 95, 96]. These shortcomings are illustrated in Figure 2.6, where a script to create a script for a complex design and geometry is overly complicated. This makes it difficult to trace design parameters and significantly increases the computation time. Thus, nowadays most design tools integrate an AD IDE with visual and textual programming languages. In addition, visual programming AD tools also generally support components created with textual programming languages [97].

Concerning visual programming AD tools, one early example is Rhino's Grasshopper [79, 80] developed in 2007. Grasshopper is a tool that allows users to program geometric features using numbers or

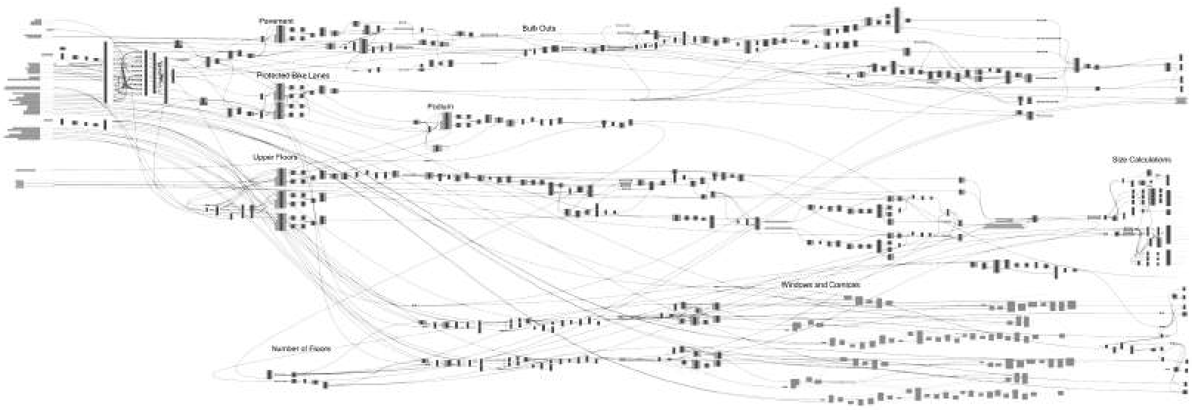


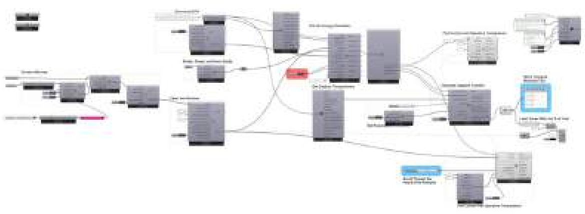
Figure 2.6: Large visual programming language script. This image highlights the complexity and lack of scaling present in visual programming languages. ©James Graef. Retrieved from Graef [98]

sliders and using manually created geometries as input. Moreover, the tool provides plug-in development, real-time feedback, debugging features, and model traceability from the algorithm to the model but not the other way. The outputs are seen in the Rhino model environment while the program is developed in Grashhoper's IDE. Another example released in 2011 is Dynamo [99], which now works as a standalone version or as a plug-in for other 3D modeling software such as Revit. This tool has the same features as Grasshopper with the advantage that the geometry can be seen directly in the IDE.

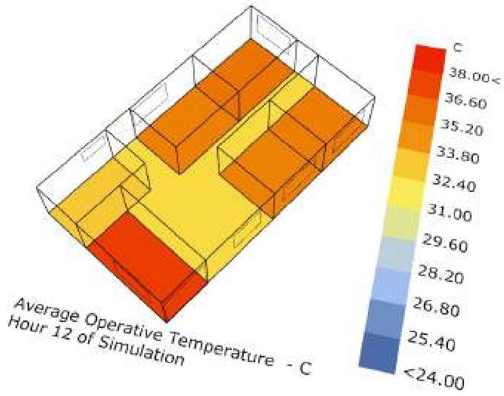
Textual programming languages have emerged earlier. One example is the 1986 AutoLISP implemented in AutoCAD. This tool allowed the user to manipulate AutoCAD functions and automate parts of the design. However, these early tools still lacked traceability and debugging features, which turned their use into a difficult task [87, 91]. Recently, textual programming AD tools like Rhino.Python and Khepri emerged with these features [86, 97], which allowed users to improve their learning curve as they had some degree of code-to-model traceability and error messages. These tools are still difficult to learn despite bringing the advantage of scalability when modeling large and complex models.

Integrating AD with BPS tools is one of its main advantages since the user can not only develop form-finding programs but also perform iterative analyses and optimizations to guide their project solutions. One example is Ladybug Tools [100], a visual programming language plug-in for Grasshopper and Dynamo that allows the integration of BPS by converting an AD model geometry and inputs into the required model format by the BPS tool. This plug-in integrates EnergyPlus for energy simulations, and Radiance for Daylight. Similarly, the Eppy tool integrates Python with EnergyPlus, allowing the user to develop building energy models with any geometry and inputs [101].

These tools allow users to perform parametric exploration of their design and simulation inputs while granting access to an ample spectrum of customized output visualization (Figure 2.7). However, when the computational power required to perform BPS is incremented with scalability issues of complex models, it catalyzes the mentioned shortcomings of AD tools. Examples are an increased difficulty running and editing the model, the necessity of learning a programming language for AD, as well as knowing the required BPS tool's inputs to obtain accurate outputs. Moreover, when adding iterative analyses or optimization processes, a new layer of knowledge is required to perform such tasks.



(a) LadyBug Tools script for comfort simulation.




(b) LadyBug Tools 3D model output visualization for a comfort simulation.

Figure 2.7: Ladybug Tools Energy simulation and resulting output visualization. ©Chris Mackey. Retrieved from Hydra Share [102].

In the following sections, various use cases of AD tools and their integration with BPS will be showcased across different stages of building projects. Specifically, to understand how these tools are used from the early to late stages, as well as their applications in the design and engineering fields. By examining existing research, it is possible to illustrate the flexibility and portability of these tools. This discussion highlights opportunities and potential challenges associated with adopting AD and BPS tools at each building project stage.

### 2.2.2 Building Performance Analysis

Building Performance Analysis consists of assessing the impact of changes in inputs on specific outputs or outcomes [103]. This is commonly applied in the Architecture, Engineering, and Construction (AEC) to evaluate solutions for building designs, construction methodologies and materials, and other building projects. While exploring published research papers in Scopus containing the keywords ”Building Performance” and ”Analysis”, it is seen a rapid increase in publishing after 2005 (Figure 2.8). This coincides with AD integration with BPS and the appearance of tools like Grasshopper (2007) and Dynamo (2011).

Multiple studies have explored AD to find better design solutions for different building performance aspects. Concerning building energy performance, Alghamdi et al. [104] tests the impact of 15 building design parameters on the energy consumption of a classroom in Australia. Results show up to $\approx$ 40% energy reductions with different roof constructions and by changing the Heating, Ventilation, and Air Conditioning (HVAC) setpoints. Elbeltagi et al. [105] develop a framework with Ladybug Tools to perform a sensitivity analysis of early-stage design parameters and visualize the results, and Najjar et al. [106] uses Autodesk Green building Studio to improve the construction of a single-family house's Energy Use Intensity (EUI), which represents the energy use per m$^2$ of the total floor area. In the latter, results show reductions of up to 15% in the EUI after building retrofits.

For daylighting, Henriques et al. [107] use Radiance to find adequate design configurations of a skylight, while Wagdy and Fathy [108] explore different values for glazing ratios, louver sizes, and configurations for multiple daylight metrics such as Spatial Daylight Autonomy, Annual Sunlight Exposure,

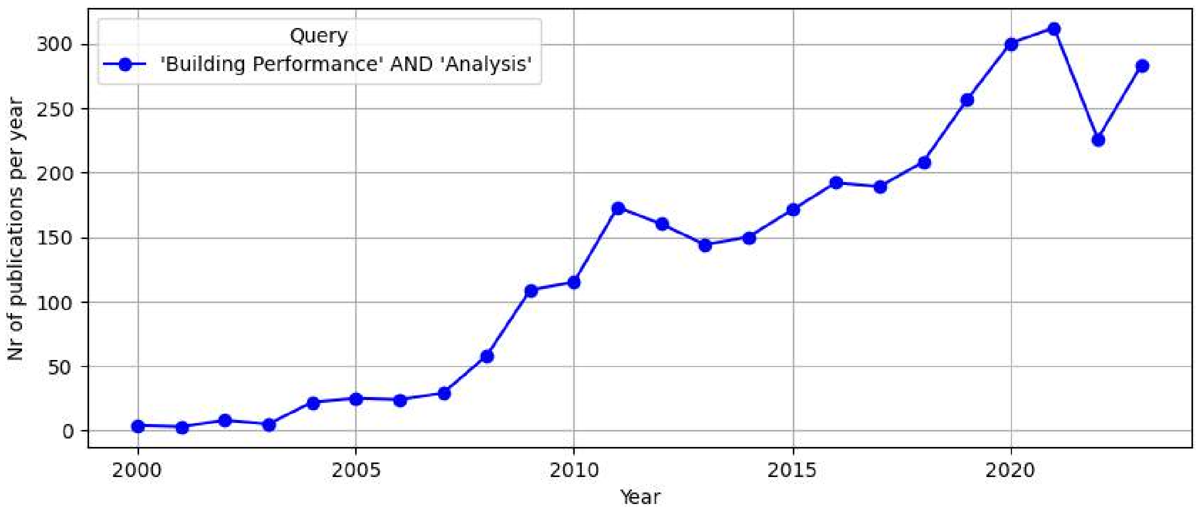



Figure 2.8: Scopus query for the keywords ”Building Performance” and ”Analysis”.

and Daylight Glare Probability. Queiroz et al. [109] estimate Useful Daylight Illuminance (UDI) for different window and room design parameters with both Radiance and EnergyPlus, and Castelo Branco and Leitão [110] search for improved solar panel configurations in a building design with radiation maps. These studies illustrate the potential benefits obtained from integrating BPS with AD tools for Daylight performance [18, 111, 110, 112].

Many building performance analysis studies exist in the literature to seek improvements in thermal comfort. For instance, Chowdhury et al. [113] used Design Builder to test different low-energy cooling strategies for an office building in a subtropical climate. Similarly, Aleixo et al. [114] uses Ladybug Tools and Grasshopper to explore the impact of multiple passive cooling strategies on the thermal comfort and daylight performance of informal settlements in Angola. At a different scale, Ramakrishnan et al. [115] use EnergyPlus and Ladybug Tools to test the properties of a phase change glazing material (such as thickness and transition temperatures) for more comfort in naturally ventilated buildings. Results show that users can identify suitable strategies to improve the building's thermal comfort and that these analyses provide useful feedback at multiple levels [111, 112, 113, 116, 117].

Exploring different project solutions with ADA can provide advantages to practitioners exploring methods to improve a building's performance. This happens due to the interoperability between design and analysis models within the same script. However, The automation and flexibility associated with ADA allows users to go one step further and develop automated optimization processes. The next sections describe the field of building Performance Optimization according to existing literature. Initially, it illustrates common optimization frameworks for building performance. Afterward, optimization results and their visualization are discussed, as well as ways to improve the optimization results. Finally, challenges and opportunities are outlined according to previous studies.

### 2.2.3 Building Performance Optimization

In the literature, when querying the Scopus database for the keywords ”Building Performance” and ”Optimization” an increase in publications with the terms is seen since 2007. Figure 2.9 compares the publications per year with keywords ”Building Performance”, ”Analysis” and ”Optimization” in a stacked area chart. Results show that both terms follow the same trendline of yearly publications, but the number of studies concerning optimization has been increasing.

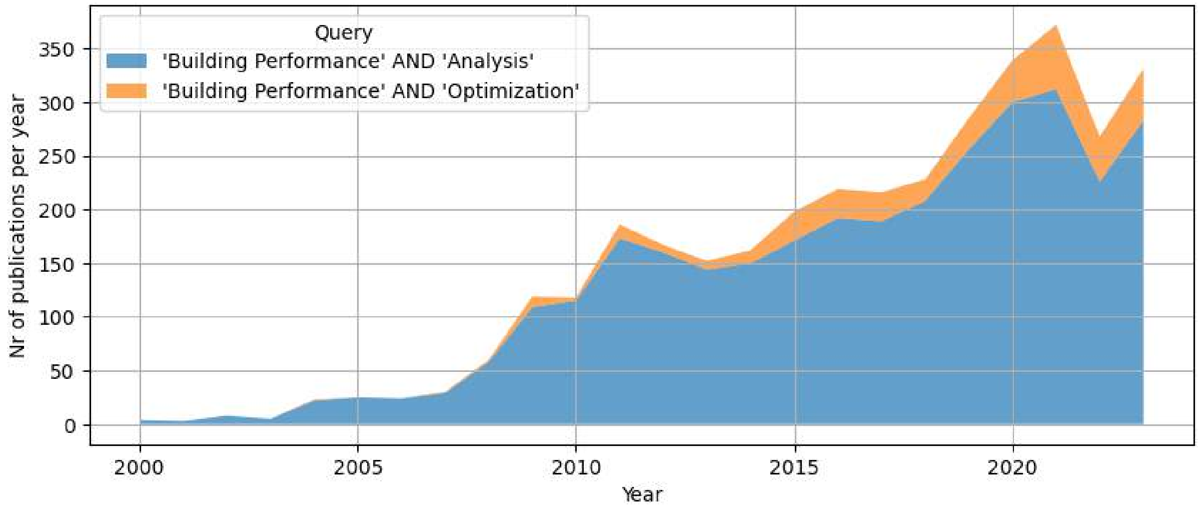



Figure 2.9: Stacked chart of Scopus queries for the keywords ”Building Performance” and ”Analysis” and ”Optimization”.

Automated building performance optimization is a method to select optimal solutions from a set of available combinations for a specific building project, according to performance criteria and complying with constraints ([118, 119, 120]. In mathematics, optimization concerns finding inputs of a function that minimize or maximize the value of an objective function [121]. Within the AEC industry, most building performance optimizations aim to solve single- or multi-objective functions often entitled Single-Objective Optimization (SOO) and Multi-Objective Optimization (MOO) [122, 118, 123, 124]. But because a building is a complex system that encompasses multiple performance aspects, and these are often conflicting, practitioners commonly use Multi-Objective Optimization (MOO) for a more holistic approach [122, 118, 125]. An example of an objective function can be described by the following equation (Equation 2.1):

$$f(x, ..., x_n) = max(o, ..., o_n) \tag{2.1}$$

where the optimum solution of function $f$ with variable(s) $x, ..., x_n$ is the maximum value for objective(s) $o, ..., o_n$. Objectives can comprise multiple aspects of building performance such as thermal comfort models, energy use, construction costs, and daylight performance, among others [126, 127, 128].

Multiple algorithms can be used to search the solution space for the optimum values. In particular, there are deterministic, population-based searches, and hybrid searches, which comprise a combination of deterministic and population-based optimizations [129, 130]. Deterministic approaches use mathematical models that produce the same results when given the same initial conditions and variables.

Examples of this approach encompass linear programming methods, gradient descent, hill climbing searches, and more [123].

Deterministic approaches such as linear programming aim to find the best outcomes of a function represented by linear relationships [131]. Gradient descent is used to find local and global minimums of a differentiable function [132], while Hill climbing searches continuously move towards increasing objective values to discover its peak [133]. Some disadvantages of these methods are that building performance objective functions are often non-linear [118, 134], and that methods like gradient descent and hill climbing are prone to find local minimums instead of global [135, 136].

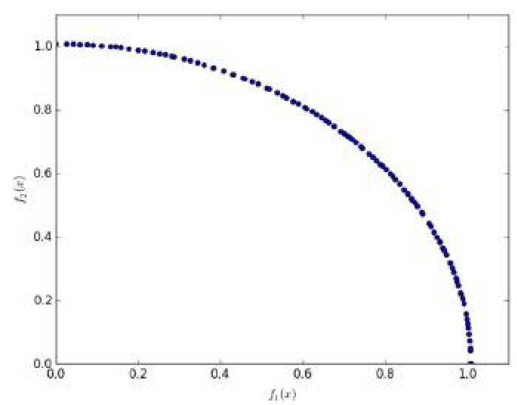
Population-based search approaches use algorithms to search variable archetypes (populations) and find near-optimal solutions. These are often called metaheuristics [137, 122], and are used for optimization problems where the user assumes no prior mathematical knowledge of an objective function [135], often called ”derivative-free” or ”black-box” problems. This is often the case for building performance optimization [24, 122, 118].

In building performance, one commonly used algorithm from the metaheuristics family is the Genetic Algorithm [138, 139, 140]. A randomized algorithm inspired by evolution principles that mimic natural selection processes and supports SOO and MOO [141]. Other examples of commonly used population-based algorithms include metaheuristics inspired by numerous biological patterns such as ant colonies and bird flocks, among others [142, 143, 144]. Some disadvantages are that these algorithms often require many iterations to converge to near-optimal solutions and their results vary widely according to their parameter settings [135].

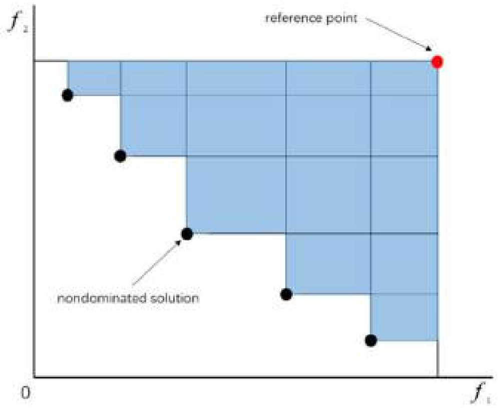
Waibel et al. [138] compile a benchmark of derivative-free SOO algorithms applied to building energy performance in the literature. Results confirm the ”No Free Lunch” theorem proposed by Wolpert and Macready [145] in that no single algorithm managed to outperform all others for different optimization problems. In addition, the authors show the relevance of the algorithms’ parameters (Hyperparameters) in obtaining the best optimization results.

Similarly, Pereira and Leitão [146] compare the performance of SOO and MOO metaheuristics for a skylight truss structural optimization. The authors do this by comparing the resulting Pareto-optimality chart of each tested algorithm. The Pareto-optimality chart was introduced by Censor [147] and denominates MOO optimum solutions as non-dominated solutions. These represent solutions that cannot improve more in one objective without harming the other(s). Pereira and Leitão [146] also compare the algorithms’ obtained Hypervolume (H) to understand how the algorithms searched the solution space. In a MOO, H refers to the measure of the space covered by the set of non-dominated solutions in the objectives [148, 149, 150]. Figure 2.10 (a) illustrates a Pareto-optimality chart for two objective functions $f_1$ and $f_2$, while (b) illustrates the calculation of an algorithm’s non-dominated solutions H.

Both Waibel et al. [138] and Pereira and Leitão [146] highlight the importance of the algorithms’ Hyperparameters in the convergence for optimum solutions. However, finding the best Hyperparameters requires more computational power to test different combinations. In particular, when applied to building performance and integrated with BPS, it becomes time-consuming to search for optimal Hyperparameters while also searching for optimal solutions of the original optimization problem [122, 24]. These and




(a) 2 objective Pareto-optimality for objective functions $f_1$ and $f_2$. Retrieved from [151]




(b) Calculation of the Hypervolume (H) with non-dominated solutions. ©Xingping Sun, retrieved from [152]

Figure 2.10: Pareto-optimality chart and Hypervolume (H) calculation for 2 objectives.

other inherent shortcomings of BPS associated with optimization processes are discussed in the next Section, which explores challenges and opportunities identified in the literature for building performance optimization.

### 2.2.4 Challenges and Opportunities

Even though there have been advances in Analysis and Optimization Processes (AOP) applied to the AEC industry, there are still some hindrances preventing their widespread use. Many studies have dwelled on these aspects but four in particular have extensively reviewed AOP of building performance.

In this first study, Evins [153] presents a review about the application of computational optimization methods in sustainable building design. The paper provides a comprehensive overview of various optimization processes. The author covers 74 works where optimization was implemented for building performance. Results show that most AOP objectives focus on energy, construction costs, life cycle costs, carbon emissions, thermal, daylight, and visual comfort. Besides emphasizing the potential of computational optimization applied to building performance, the author suggests further research to focus on improving usability, interoperability, and computational speed in AOP with the use of meta-models.

Attia et al. [154] develop a study to explore challenges and opportunities for the integration of AOP with BPS in low energy buildings. The article reviews 165 publications of current optimization methods applied to building performance and showcases the results of a survey of 28 optimization practitioners (mainly engineers) on the challenges and opportunities of AOP and BPS. The authors conclude that building performance optimization has a large potential to achieve current development and sustainable goals. Among the findings, participants point to 4 soft obstacles and 7 technical obstacles. Soft obstacles include high expertise requirements, low trust in the results, lack of systematic approaches, and low appreciation among the AEC industry. Technical obstacles include the uncertainty of the BPS model; long computation times; missing information; low portability and environments to integrate bps and optimization; lack of friendly interfaces for results post processing.

In 2014, Nguyen et al. [122] provide an extensive review of AOP integrated with BPS implemented in the field of building performance. The study categorizes and evaluates various optimization techniques, highlighting their use in improving overall building performance. The authors document the strengths and weaknesses of these techniques in dealing with the complex, multi-objective nature of building performance. Main findings indicate that simulation-based optimization is a powerful approach for improving building energy performance. However, the authors outline specific challenges for the efficient integration of AOP with BPS: issues with local minima convergence, selecting the optimization algorithms, integrating multiple objectives, performing computationally expensive simulations and optimizations, and optimum solutions uncertainty. The authors advocate for the use of Surrogate Models (SM) to overcome some of these challenges.

Finally, Tian et al. [155] conduct a survey for 119 participants to assess the challenges and opportunities of adopting AOP with BPS in their projects. In addition, optimization methods for passive buildings performance optimization were reviewed and categorized. Survey results show that most participants agree that AOP for building performance has many advantages such as adjusting design variables, helping in making decisions, finding optimal solutions, and studying multiple performance objectives. However, results also show that the main hindrances for participants to adopt AOP with BPS in their projects are: long calculation times, lack of standard optimization methods, lack of user friendly interface, poor results analysis ability, and uncertainty regarding the value obtained over time spent. The authors suggest that overcoming these barriers might lead to a wider implementation of AOP in building performance.

To conclude, while the adoption of AOP processes brings numerous advantages to the field of building performance, it still has significant shortcomings that prevent its widespread use. In particular, they are difficult to perform, time-consuming, and lack portability among tools. For instance, energy and daylighting simulation of a large urban area requires different tools and model formats and may take up to several hours to days to perform. Some studies suggest using meta-models, or SM, to overcome these major complications. The next Section describes what are SM, how they work, how to improve them, and how to deploy them. Finally, it reviews applications of SM in the building performance field along with their challenges and opportunities.

## 2.3 Surrogate Models

Surrogate Models (SM) are simplified representations of more complex and computationally expensive programs. They are used to make predictions, study systems, and streamline AOP by harnessing data from function variables and results [25, 156, 26, 157]. There are multiple types of SM for different prediction tasks, which can be classification tasks or regression. Classification tasks predict discrete outputs while regression tasks predict continuous outputs. The performance of these models in predicting the desired outputs is often evaluated through different metrics for each type of prediction task. Thus, considerations must be taken to select suitable models with adequate prediction performance.

These models have experienced rapid growth in the literature (Figure 2.11) and have been used in multiple fields of research such as Finance, Medicine, Engineering, and Physics, among others

[158, 159, 160, 161]. However, by taking a look at published papers in Scopus that contain the keywords ”Surrogate Models” and ”Building Performance” it is observed that not many studies contain both keywords, which suggests that studies exploring them have not been sufficiently reported. The following sections document different types of models for regression and classification tasks; research methods to improve SM performance; and showcase their integration and implementation in building performance studies along with its challenges and opportunities.

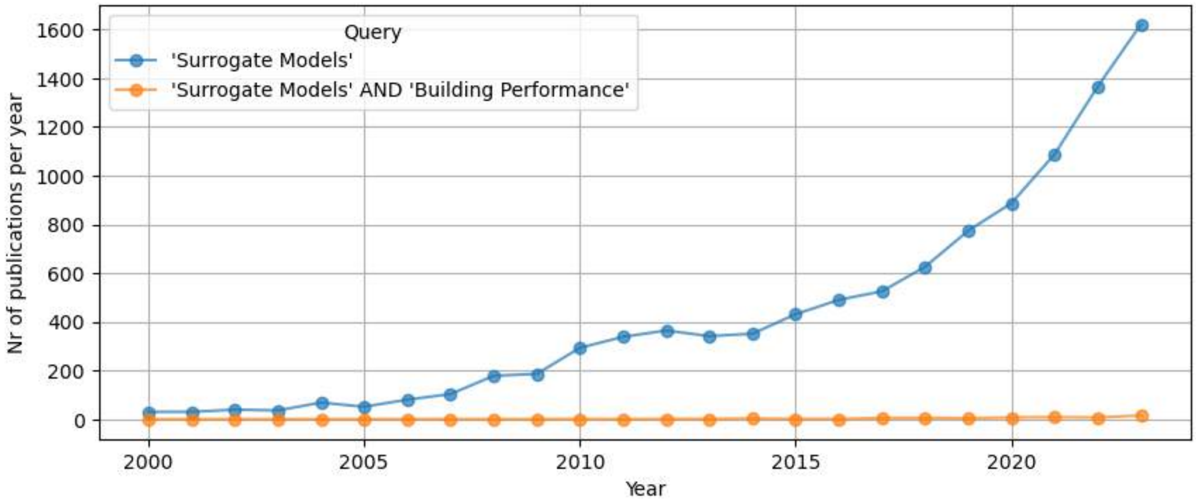



Figure 2.11: Line plot of Scopus queries for the keywords ”Surrogate Models” and ”Building Performance”.

### 2.3.1 Types of Models

According to Forrester et al. [26], there are three stages in developing a SM. Summarily, it begins with data preparation and establishing the modeling approach by considering the features that will be used to predict the target output. Some models require additional data preparation such as scaling the features to adequate values or converting categorical entries into integer numbers. An initial data cleaning should be performed to guarantee that the database contains representative samples of the problem. The second stage comprises model training, where a portion of the database is set aside to train the chosen models. The third stage tests the model performance by comparing the predicted values with the ones contained in the testing sample that are not used to train the model.

Data cleaning techniques can be used to improve the data quality. For instance, outlier and null values can be culled, input data can be scaled, labels can be encoded into integer numbers, and the most representative features of the problem can be selected [26, 157, 162]. Scaling techniques encompass normalizing data samples or converting them into a boundary system [163, 164]. Feature selection is an optimization problem by nature. Thus, this process can encompass a wide range of techniques from correlations to derivative-free optimization problems [165, 166]. Initially, direct correlations between features and target outputs are commonly calculated to identify a starting pool of features.

Multiple model types can be chosen for both regression and classification prediction tasks. For instance, Linear models, Statistical, Polynomial, Gaussian Process (GP), Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Artifical Neural Networks (ANN), and Ensemble methods. Each model type's usability is suitable for different problems and data types and requires different data processing methods [25, 156, 26].

Polynomial models use equations to fit a function to the target features and output [167]. It is suitable for regression problems with simple relations between features that can be approximated using polynomial descriptions. These models often benefit from data processing techniques like data scaling or normalization [26, 168]. One limitation is that it can become computationally expensive when high-order polynomial data is included [169].

GP models use a probabilistic approach to predict the target output. These models can be used for both classification and regression tasks, and are suitable for complex problems with non-linear relationships [170, 171, 172]. Usually, these models require feature scaling to improve numerical convergence [172]. Because it is a probabilistic approach, it also provides uncertainty quantification for predictions [172, 173]. GP models can become computationally expensive with large datasets and are sensitive to outliers and faulty values [174, 175].

SVM are primarily used for classification but can also be adapted for regression tasks. They work by finding an optimal plane that separates data into classes with the largest possible margins [176, 177, 178]. These models are suitable for classification problems that are highly representative of the solution space and for non-linear regression problems. However, they are computationally expensive, which makes them unsuitable for high-dimensional features and large databases [179, 180].

KNN can be used for classification and regression tasks. The model assigns the most frequent value among the "k" nearest samples for classification tasks, and the mean value for regression tasks [181, 182]. These models are powerful for quick and simple pattern recognition but their accuracy can degrade with high-dimensional data. Moreover, it is sensitive to Hyperparameter tuning to find the optimum "k" and distance values [183, 184].

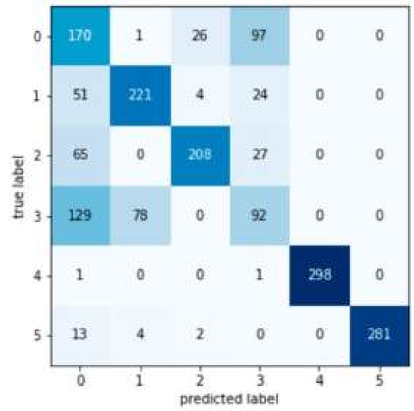
ANN are used for both regression and classification tasks and are versatile tools that excel in pattern recognition. They consist of layers of interconnected nodes that can model complex non-linear problems [185, 186]. They are suitable for large and complex databases and data processing techniques such as handling outliers and missing values, scaling the data, and others can improve their performance [187, 188]. The obscure initialization values, difficulty interpreting the results, and over-fitting to the training sample are some shortcomings of these models [189, 190].

Ensemble models combine multiple predictive models to improve the accuracy and robustness of predictions [191, 192]. Some examples are Random Trees and Gradient Boosting [193, 194]. These combine multiple decision trees to leverage each one's strengths and weaknesses [195]. These models are used when a single model performance is insufficient. They are robust regarding data scaling but models like Gradient Boosting can benefit from scaling or normalization [[196]. These models often sacrifice feature interpretation to improve their prediction accuracy [197].

According to Bishop and Nasrabadi [198], multiple metrics can be used either as loss functions for models that require it or to evaluate their performance. These are used according to the prediction task at hand. Because classification tasks aim at predicting discrete outputs, metrics like accuracy, precision, recall, f1-score, and confusion matrixes are used [199]. Accuracy is the ratio between correctly predicted samples and total samples. Recall refers to the number of correctly predicted samples of one category over its total number of samples. Precision measures the accuracy of correctly predicted categories. Finally, the f1-score is a metric that represents a balance between precision and recall, and confusion matrixes are simply representations of the true over the predicted values for each category. Figure 2.12 illustrates a typical classification report and confusion matrix of a classification model.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 1.00 | 1.00 | 1.00 | 11 |
| 1.0 | 1.00 | 0.62 | 0.76 | 13 |
| 2.0 | 0.55 | 1.00 | 0.71 | 6 |
| micro avg | 0.83 | 0.83 | 0.83 | 30 |
| macro avg | 0.85 | 0.87 | 0.82 | 30 |
| weighted avg | 0.91 | 0.83 | 0.84 | 30 |

(a) Classification report for a classification model predicting 3 categories. ©Muthukrishnan, retrieved from [200]




(b) Confusion matrix of a classification model predicting 6 categories. ©Xingping Sun, ©Zhengwei Qu, retrieved from [201]

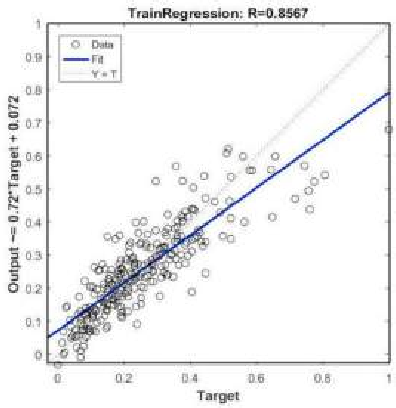
Figure 2.12: Visualizing the predictive performance of a classification model.

Regression tasks require a different approach since they aim to predict continuous output. In this case, commonly used metrics are the Mean Average Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percent Error (MAPE), and Coefficient of determination ($R^2$) [26, 198]. The MAE measures the overall prediction error of the test sample. However, larger error values and outliers will get diluted with this metric. To solve this problem, the RMSE and MAPE can be calculated. Finally, the $R^2$ represents the proportion of variance in the output which is explained by the model's input features. This value typically ranges between 0 and 1, where a higher value indicates that the model explains a greater portion of the output variance. Chicco et al. [202] review these metrics and suggest that the $R^2$ provides better feedback on the performance of regression models.

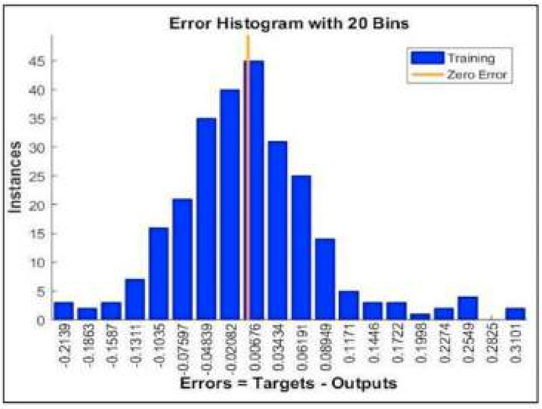
Multiple model types can be evaluated and compared with the described metrics to provide feedback on suitable SM for the problem at hand. After choosing the model type, some techniques can still be applied, if necessary, to improve the model's performance even further. Finally, Table 2.1 summarizes the described model types, their advantages and disadvantages. In the next Subsection, tuning techniques that can be used to do this are explored and documented.

### 2.3.2 Model Tuning

If the model's accuracy is insufficient, some techniques can be used to improve model performance [26, 198]. Among them are Feature Engineering (FE) techniques and Hyperparameter optimization. FE consists of operations performed on the database's features to improve model performance, while Hyperparameter optimization seeks to find the best combination of the model's parameters [204, 205].




(a) Scatter plot of a regression model predicted values (x-axis) and true values (y-axis)




(b) Histogram plot of prediction errors.

Figure 2.13: Visualizing the predictive performance of a Regression model. ©Adil Masood. Retrieved from [203]

Some FE methods are already mentioned in Section 2.3 (See Types of Models), such as feature selection and transformations like scaling, label encoding, and normalization. Feature selection techniques are used when the number of features need to be reduced and/or the models are sensitive to large numbers of features [206, 162, 207]. Feature scaling and transformations are suitable for models sensitive to specific ranges of values, and when there are skewed feature distributions [208, 209, 210].

In addition to these methods, features can also be created through interactions or polynomial relations such as powers and products of features. This is suitable when there are potential non-linear relationships between features and target outputs [207, 210]. In contrast, a reduction in feature dimensions can also be obtained with Principle Component Analysis [211]. This is useful for models that become computationally expensive with an increasing number of features like SVM. Finally, if the database contains unbalanced features that show majority and minority class values, under-sampling or oversampling techniques can be explored according to the database size [212, 213].

Similarly to the optimization algorithms discussed in Section 2.2 (See Building Performance Optimization), SM also have Hyperparameters that guide their fit of a database. These can be tuned to find the best-performing models among the set of Hyperparameters [214, 215]. Some techniques include multiple sensitivity analyses, search methods, and more complex optimization problems [205]. Similarly to equation 2.1 seen in Section 2.2 (See Building Performance Optimization), Hyperparameter optimization can be illustrated by the same equation where the objective variable is represented by the chosen model's performance metric:

$$f(x, ..., x_n) = max(R^2) \tag{2.2}$$

where $x, ..., x_n$ are the SM Hyperparameters, and in this case, the objective is to maximize the $R^2$ score.

Grid-based methods that explore all possible combinations can be used for a small set of Hyperparameters with discrete variables. However, as the number of Hyperparameters and their boundaries increase, the computational cost of the process exponentially increases [216, 217]. Random search

Table 2.1: Summary of different model types - Descriptions, Advantages, and Disadvantages

| Model | Description | Advantages | Disadvantages |
|---|---|---|---|
| **Linear Models** | Fits a linear equation to describe relationships between input features and target outputs. | Simple, interpretable, and computationally efficient. Suitable for linearly separable data. | Performs poorly for non-linear relationships. Sensitive to outliers and noise. |
| **Polynomial** | Fits data using polynomial equations to approximate relationships. | Simple to implement and works for polynomial relationships. Benefits from data scaling. | Computationally expensive for high-degree polynomials. Risk of overfitting with complex data. |
| **Gaussian Process (GP)** | A probabilistic model providing predictions and uncertainty quantification for complex, non-linear relationships. | Provides uncertainty quantification. Suitable for small datasets with complex patterns. | Computationally expensive for large datasets. Sensitive to outliers. |
| **Support Vector Machines (SVM)** | Finds an optimal hyperplane to separate classes (classification) or fit a regression plane. | Effective for high-dimensional spaces. Works well for non-linear data with kernel tricks. | Computationally expensive for large datasets. Requires careful hyperparameter tuning. |
| **k-Nearest Neighbors (KNN)** | Assigns output based on the nearest “k” samples using distance metrics. | Simple and intuitive for pattern recognition. No strong assumptions about data distribution. | Performance degrades with high-dimensional data. Sensitive to hyperparameter selection (e.g., “k”). |
| **Artificial Neural Networks (ANN)** | A network of interconnected nodes capable of modeling complex, non-linear problems. | Highly versatile and powerful for large datasets. Can handle complex non-linear patterns. | Requires large data for training. Prone to overfitting. Results can be hard to interpret. |
| **Ensemble Methods** | Combines multiple models (e.g., Decision Trees, Gradient Boosting) to improve performance. | Improves accuracy and robustness. Handles noisy data well. | Computationally expensive. Sacrifices interpretability for performance. |

algorithms can be leveraged to overcome this shortcoming and still run a computationally inexpensive analysis [217]. Unfortunately, it often fails to find optimal combinations, its performance depends on the dimensions of the Hyperparameters, and the obtained results can vary due to the algorithms' random nature [218]. Finally, Bayesian Optimization and Genetic Algorithms can also be implemented to search large Hyperparameter spaces and have proven effective for complex SM like optimizing an ANN [219, 220, 221]. The optimum results obtained with these methods come at higher computational costs when compared with less complex search approaches. In addition, when using the Genetic Algorithm, the algorithm itself might require tuning [221].

After processing the data, developing the model, and if necessary, tuning it, it can be implemented for specific problems within the building performance field of research. The next Subsection explores practical implementations of SM for building performance. Afterward, challenges and opportunities to these implementations are documented and discussed.

### 2.3.3 Surrogate Models in Building Performance

Despite that the application of SM to building performance studies is still recent, there are still a few studies available in the literature that encompass this application. Some use existing databases, including sensors, measurements, records, and classifications (Existing Building Information Database (BID)). Others generate databases with a static sampling of BPS (Synthetic BID) or with an iterative sampling of BPS, such as optimization algorithms iterations (Iterative BID).

Using existing databases, Zhang et al. [222] evaluate 4 different types of SM in their capacity of predicting the hot water energy use of a building. Real measured data for 2012 is used to train the model while measurements for 2011 were used as validation data. The authors developed hourly and daily models with the hourly model performing better in the validation test. Results show that ANN performed worse due to insufficient training data, but all models achieved acceptable accuracy. The study also highlights the added advantage of GP enabling uncertainty analyses. Some shortcomings pointed out by the authors are that the developed models are restrained to the problem at hand and their performance is highly dependent on data quality.

Likewise, Benavente-Peces and Ibadah [223] test different classification models including GP, ensemble models, decision trees, and K-Neighbors. The study uses open-source databases of buildings' energy usage to predict its Energy Performance Certificates (EPC). The authors highlight the accuracy of the models in predicting the building's energy efficiency class. However, they also advise prioritizing feature selection techniques and prioritizing data quality over data quantity. The main limitations pointed out by the authors are the dependency on the quality of the database and the need for further refinement if systematically implemented.

Magnier and Haghighat [224] trained an ANN to optimize the Predicted Mean Vote (PMV) and Energy Use Intensity (EUI) results of an air-conditioned building's energy simulation. The model was trained with a Synthetic database generated by a Latin Hypercube Sampling Method applied to the optimization variables of the problem [225]. This generated a balanced database that was simulated with a BPS tool. The authors validate their results with a simulation-based optimization and note that despite taking 3 weeks to generate the database, the 10 years that would take to achieve optimum results with simulation are reduced to only 7 minutes.

Similarly, Westermann et al. [226] develop a Convolutional Neural Network to predict hourly data of a building's cooling and heating demand simulation for several locations in Canada. This database was simulated in 15 hours. The authors implement the developed SM to optimize a building's design parameters for several locations. Results show an $R^2$ of 0.92. Moreover, this study showcases the potential of SM in solving most shortcomings of AOP with BPS. However, it is noted that most studies involving SM are specific to the problem at hand and often require retraining for other holistic variables.

With a different approach to the model training database, Wortmann [227] optimizes the daylight of a room with a GP of a Radial Basis Function model [228] capable of predicting the room's Useful Daylight Illuminance (UDI). This model uses results from simulation-based optimization iterations to update the model's accuracy (Iterative BID). In this case, the optimization process still performs simulations but takes fewer iterations to achieve optimum results. Comparing this approach with simulation-based optimization shows faster convergence to optimum results throughout the optimization process. However, the author notes that extensive bench-marking is required to implement these strategies efficiently [24].

In brief, SM developed with existing BID show high accuracy in both regression and classification tasks for building performance. However, they are dependent on the data quality and specific to the problem. Synthetic BID provides good results in replacing BPS for AOP. Yet, simulating the databases is time-consuming, and resulting SM are specific to the prediction task. Finally, Iterative BID obtained remarkable performance in speeding up optimization processes, however, other shortcomings are still experienced such as the lack of portability and required expertise (see Section 2.2, Challenges and Opportunities). In addition, to avoid extensive testing of different models and optimization algorithms, a benchmark of model performance for each optimization algorithm must be documented.

This last Subsection takes a broader look at the implementation of SM in building performance. By identifying and documenting previous significant review studies it is possible to elaborate on the main challenges and opportunities that stem from integrating SM with AOP.

### 2.3.4 Challenges and Opportunities

As seen in Sections 2.1 and 2.2, there are multiple advantages for integrating BPS and AOP in different building project stages. These tools, analyses, and optimizations can help provide a clear path to reducing buildings' life-cycle emissions and achieving Net Zero energy. However, there are still relevant obstacles to overcome if they are to support a wide adoption among not only researchers but also Architecture, Engineering, and Construction (AEC) practitioners. They are computationally expensive and time-consuming, not portable among design and analysis tools, and they require significant expertise for setting up and performing both BPS and AOP.

To overcome these barriers, Literature converges on SM to reduce the computational cost of these processes, reduce their complexity, and streamline their application. The last Subsection demonstrates how SM can be applied in building performance, and their potential in overcoming these problems. This section reviews past research to outline the advantages and challenges emerging from these applications.

Seyedzadeh et al. [27] perform an extensive review on four types of SM applied in building energy performance: ANN, SVM, GP, and clustering algorithms such as KNN. Afterward, the authors suggest a framework for SM selection and highlight the main advantages and drawbacks of implementing these models. Results show that different models are more suitable for different problems and databases. For instance, ANN demonstrated to be effective for short-term load forecasting, while GP is beneficial when dealing with uncertainty in input and prediction values.

All reviewed studies demonstrated reasonable accuracy with data processing techniques and Hyperparameter optimization. Nonetheless, Seyedzadeh et al. [27] note that selecting the best-performing model is challenging, and advise a thorough analysis of the data and prediction task. Moreover, the scalability of models to other problems, handling uncertainty, the computational cost of BPS for data generation, and the complexity of SM development are all challenges to their efficient implementation among experts and practitioners.

In another study, Westermann and Evins [28] gathered a knowledge corpus of 57 publications that developed SM models with Synthetic databases. The authors document the study's subject, model type, and building sampling methodology. The reviewed studies comprised subjects including optimization problems (32%), early design problems (29%), sensitivity analysis (25%), and uncertainty quantification (14%). The most used model types were Linear Regression (33%), GP and ANN (13%), and SVM (10%). Finally, the most used sampling methods in the studies for database generation were static (81%), while iterative sampling methods were only used in 19% of the studies.

With the review results, Westermann and Evins [28] confirm that SM are a powerful method to progress in BPS and AOP research. Furthermore, the authors extract challenges and opportunities of applying SM developed with Synthetic databases for building performance. Some challenges include that most SM developed in the literature are case-specific with low scalability to other buildings or problems; lack portability for different prediction tasks; both model-selection and Hyperparameter optimization are challenging and time-consuming; there is a specific limit to each SM input features capacity until exponentially increasing computational cost; and there is uncertainty regarding the best sampling strategies. Some opportunities include that SM is highly valuable to obtain time savings in AOP compared with traditional BPS; most optimization studies obtained time-reductions up to 80% of simulation time, high accuracies ($R^2 > 0.95$), and their results have shown better optima than when performed fully with BPS.

All these studies reiterate the potential value of SM applied to AOP in the field. This includes significantly speeding up these processes, reducing the number of inputs and complexity while maintaining accuracy, and obtaining better AOP results. However, some research gaps associated with SM are the specificity to the problem being analyzed, and high dependence on data quality, model, and feature selection. Also, their tuning is complex and time-consuming when the model accuracy is sub-par. Solving these research gaps can help improve the integration of SM with building performance AOP.

This thesis proposes a new framework for SM development applied to building performance AOP that may respond to the main research gaps and challenges identified. This includes required tools, building databases (hereafter referred to as Building Information Database (BID)), model selection, tuning, and deployment. The framework is illustrated and evaluated for multiple case studies concerning three different types of BID: Existing BID; Synthetic BID; and Iterative BID.

This framework establishes a comprehensive approach of SM development for building performance, overcoming the identified obstacles up until now in this thesis. Thus, it is illustrated with case studies, from general to specific building performance problems; identifies different kinds of databases and suitable approaches to improve data quality; and provides a structured approach to model tuning to reduce its computational cost. In addition, it suggests possible model deployment methods to improve SM usability according to the problem at hand.

# Chapter 3

# Research Methodologies

## Preamble

This research aims to facilitate the integration of Analysis and Optimization Processes (AOP) with building and urban projects. Both AOP and Building Performance Simulation (BPS) limitations must be tackled to achieve this. On one hand, BPS obstacles preventing the analysis execution hinder the process (Section 2.1, see Challenges and Opportunities). On the other hand, these limitations increase according to the AOP and its scope (Section 2.2, see Challenges and Opportunities).

One common BPS obstacle is the need to integrate it with AOP. Since simulations require extensive knowledge of inputs and outputs, a certain amount of expertise is required to do so [14, 16, 123]. Another limitation is the number of required tools to integrate BPS effectively [5, 10, 18, 76], and its time-consuming nature [122, 125]. To solve these problems, Surrogate Models (SM)s can predict BPS results much faster, with easy-to-grasp and fewer inputs than BPS [22, 24, 73, 27, 28, 229].

This thesis proposes a new framework to develop SM for AOP in building performance and evaluates it with case studies. This framework encompasses 5 stages: Tool Selection, Building Information Database (BID), Model Type, Model Tuning, and Model Deployment (Figure 3.1). Finally, case studies are developed to evaluate the proposed framework for all the proposed BID.

Stage 1 (Tools Selection) documents the main toolset used to develop the SM for the different typologies of problems. These can include Computer-Aided Design (CAD), BPS, Algorithmic Design (AD), and programming tools. The model development toolkit can vary according to the scope of the AOP (e.g., lighting, thermal, energy, among others), to the problem proposed in the case study. Stage 2 (BID) describes the 3 different BID by their types (See Section 2.3, see 2.3.4). These databases can be already available, extracted, and measured (Existing BID); generated based on their building feature boundaries (Synthetic BID); and generated by iterative approaches (Iterative BID). Stage 3 (Model Type) tests and evaluates multiple SM types to identify the most suitable one for the problem. These models can predict continuous or discrete values such as the energy use of a building or Energy Performance Certificates (EPC). Stage 4 (Model Tuning) optimizes the selected SM if needed. This is done with Feature Engineering (FE) and selection, and Hyperparameter optimization. Finally, stage 5 (Model Deployment) integrates the resulting model with the case study and deploys it to suitable interfaces.

This framework is evaluated and compared according to its accuracy, development time, number of features, number of objectives, and adaptability. This is done by comparing the case study SM prediction errors, database compilation time, and the number of required samples to illustrate the problem, as well as by analyzing the suitability of each BID for different problems.

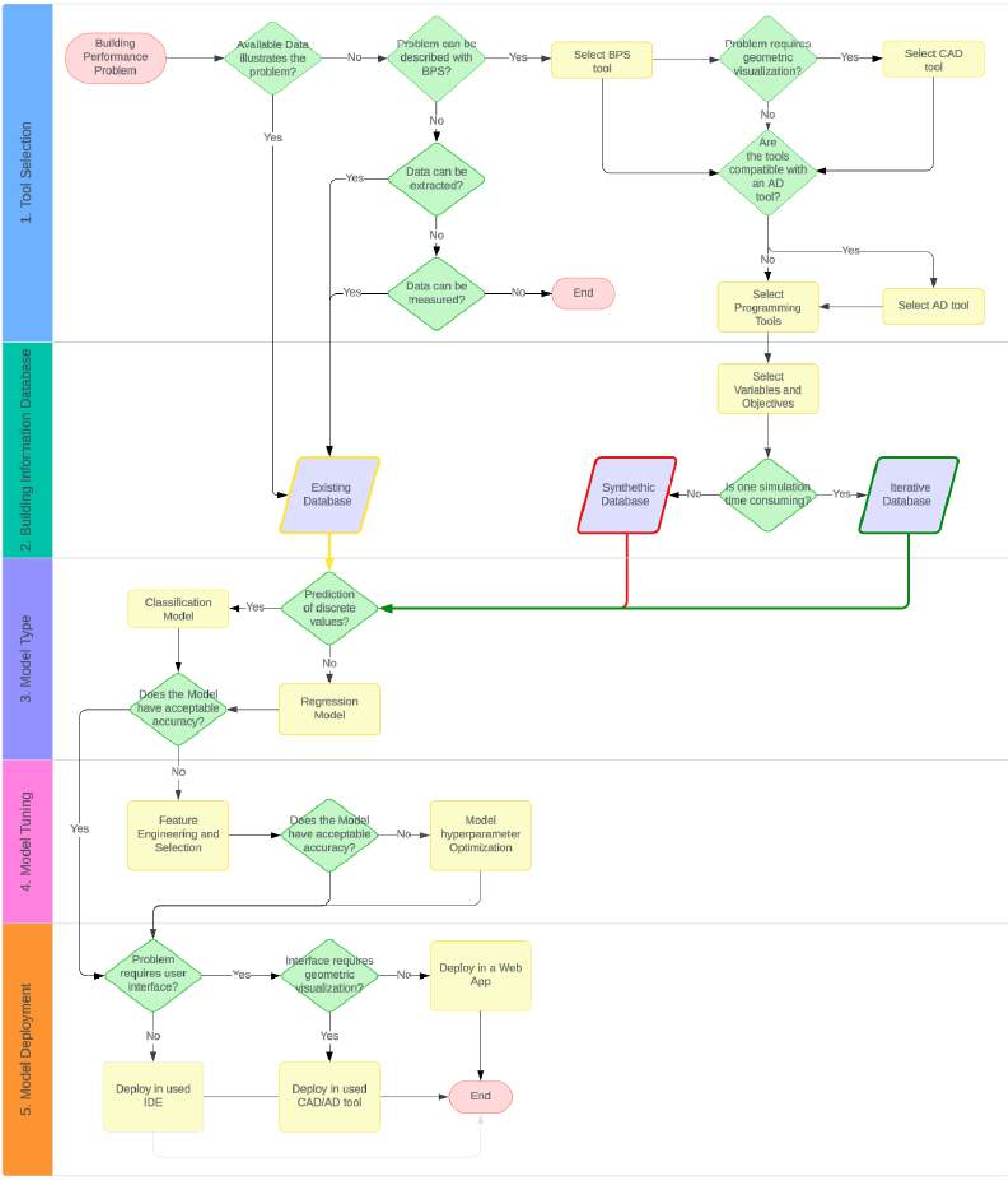



Figure 3.1: Research Process Flowchart.

## 3.1 Tools

This research uses CAD tools as visualization platforms for building databases, simulation results, and geometric representations. AD tools are used to generate building geometry databases and integrate them with BPS tools to obtain a database capable of training a SM. Programming tools are needed to generate building databases and geometry, perform optimizations, train, and deploy the SM.

### 3.1.1 Computer-Aided Design Tools

CAD tools create highly accurate 2D or 3D models throughout the entire design process. As such, they have been the most commonly used tools in the AEC industry in the past years [230]. Although CAD is currently experiencing a paradigm shift toward BIM [231], it is still a core design methodology in many aspects of the AEC industry. In this research, CAD tools are used to visualize simulation results and develop any required geometric model to support BID development (Section 3.2), and model deployment (Section 3.5)

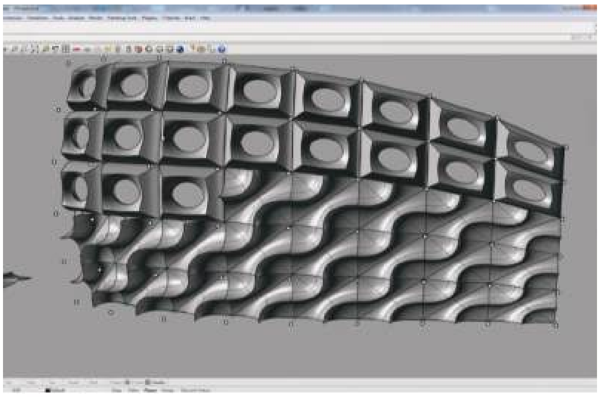
*Rhino 3D* [79] is the main CAD tool used to showcase the evaluated case studies in this research. *Rhino 3D* is known for its flexibility and versatility in creating 3D complex shapes. It supports Nun-Uniform Ration B-Spline curves and mesh modeling (Figure 3.2 (a)). This makes it suitable for design and visualization needs. This tool is often used alongside other compatible plug-in tools for analysis, annotation, and rendering tasks (Figures 3.2 (b, c, d, respectively)). Because of this portability with multiple BPS, AD, and rendering tools, *Rhino 3D* is used in the case studies when needed as it possesses all the required features to support this research.

### 3.1.2 Building Performance Simulation Tools

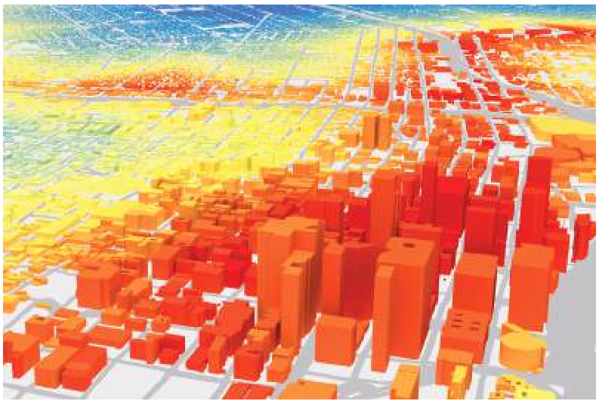
BPS tools can perform extensive and numerical calculations to obtain inputs regarding Building Performance on multiple aspects. In the AEC industry, these tools are often used in every step of a project. Additionally, BPS has increased in frequency of use due to growing concerns about the sustainability and energy efficiency of buildings and communities[29]. In this research, BPS tools will be used to develop BID to train and test the developed SM (Section 3.2).

BPS tools can be segmented according to their performance scope including energy, life cycle assessment, structural, acoustics, fluid dynamics, and natural lighting. This wide variety constitutes a challenge to develop adequate SM. Investigating all these would be unfeasible with the current research time frame. As such, two BPS aspects were selected based on research background and relevance to the United Nations sustainable development goals [2]: Energy and Natural Lighting simulations. To perform BPS for these aspects, the tools used in this research are *EnergyPlus* [32] and *Radiance* [236].

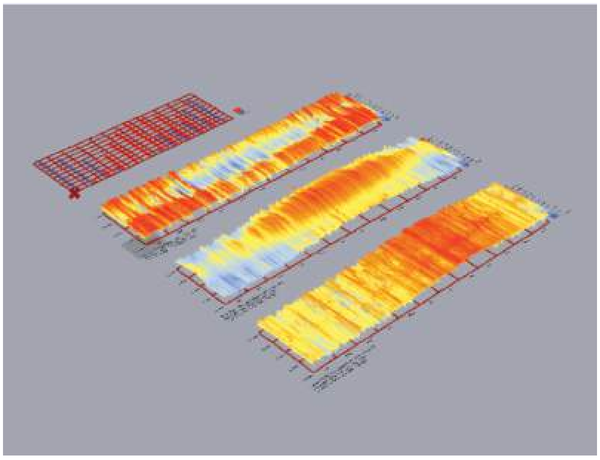
*EnergyPlus* is an open-source BPS tool that models the energy and thermal performance of a building. This tool can provide a detailed report of a building's thermal, energy, and lighting performance, as well as environmental impact. *EnergyPlus* has an extensive community and documentation that can support this work. Additionally, the possibility to be integrated with numerous CAD and AD tools (e.g.



(a) 3D modeling. Retrieved from [232]



(b) Urban analysis. Retrieved from [233]



(c) Documenting and Annotating. Retrieved from [234]

(d) Rendering. Retrieved from [235]

Figure 3.2: Modeling, Analysis, Annotation, and Rendering examples with Rhino.

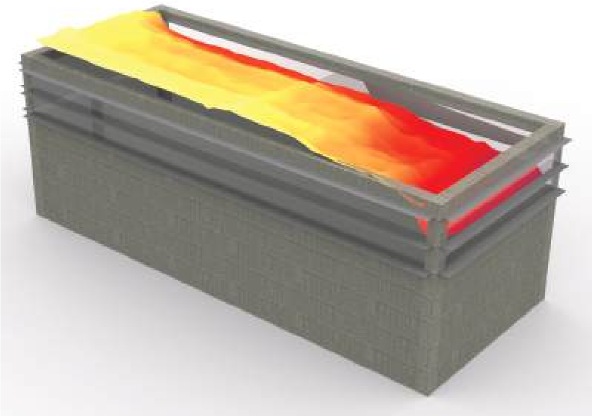
*Rhino*, *Grasshopper*, *Python*, *Revit*, among others) makes it a suitable energy simulation tool to assist with the research process.

*Radiance* is an open-source lighting simulation tool that is used in multiple fields of the AEC industry. This tool can simulate the lighting behavior of a given building. Additionally, *Radiance* can be integrated with other tools to generate, automate, and post-process simulation results.

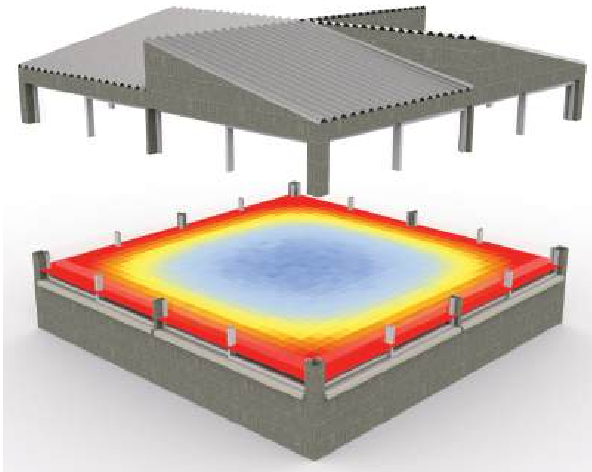
Both these tools can be integrated with *Rhino3D* with the plug-in *Ladybug Tools* [100]. Users can generate *EnergyPlus* and *Radiance* models, automate simulations, and visualize results directly in *Rhino*. This can be used in multiple building AOP for both Energy (Figure 3.3 (a)) and Daylighting (Figure 3.3 (b, c)) simulations and optimizations.

### 3.1.3 Algorithmic Design Tools

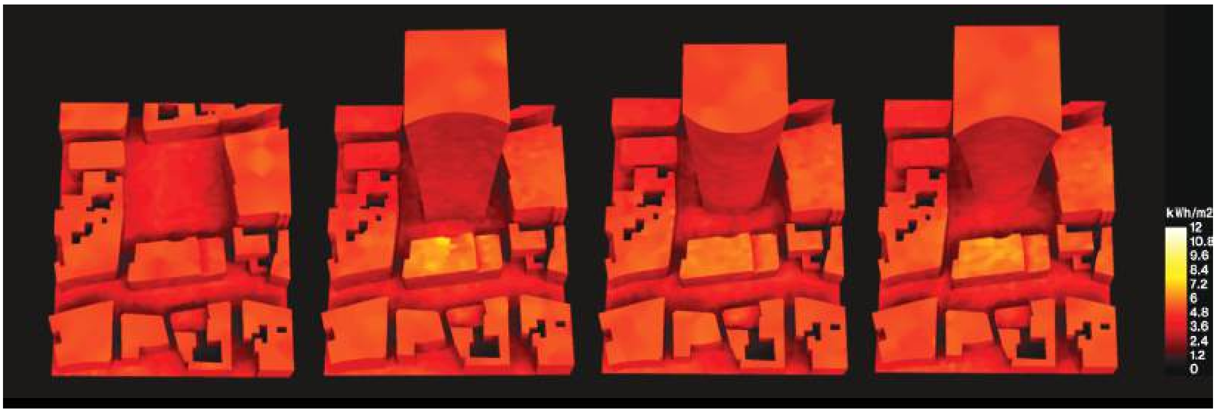
AD tools are capable of describing geometric shapes as an algorithm with multiple variables and relationships by using visual or textual programming languages (See Section 2.2, Algorithmic Design Tools). Many state-of-the-art AOP rely on these tools to be executed [18, 19, 111, 122, 125]. In particular, AD tools are used in this research to automate BID development (Section 3.2) by generating geometries according to the desired features and deploying models (Section 3.5).



(a) Thermal autonomy study.



(b) Useful daylighting illuminance study. Retrieved from [237].

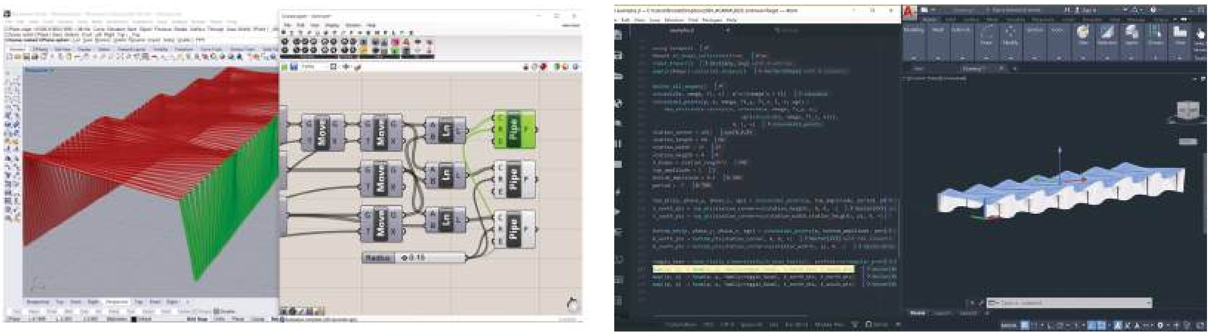



(c) Incident radiation study. Retrieved from [238].

Figure 3.3: EnergyPlus and Radiance integrated with Rhino and Ladybug tools for simulation results post-processing.

One tool that is often used in this research is *Grasshopper* [80], which is already integrated into *Rhino3D*. *Grasshopper* is a visual programming language that consists of functions that receive inputs and return outputs (Figure 3.4 (a)), allowing users to automate their designs and variations. Another AD tool that is used in this research that uses a textual programming language is *Khepri*, which uses *Julia* [239] to develop AD compatible with multiple CAD and BPS tools (Figure 3.4 (b)). Additionally, *Khepri* has a built-in optimization library with multiple algorithms for Single-Objective Optimization (SOO) and Multi-Objective Optimization (MOO).

Finally, in some cases that do not need geometric representation or daylighting simulations for BID development or model deployment, the *Eppy* tool is used [101]. *Eppy* allows users to create, edit, and simulate *EnergyPlus* files with *Python* programming language, which streamlines the entire research framework into one algorithmic description. With *Eppy*, users can develop BID by simulating variations of buildings and urban models and directly train, test, tune, and deploy SM,



(a) Grasshopper visual programming language Sammer et al. [86].

(b) Khepri using Julia textual programming language. Castelo Branco et al. [240]

Figure 3.4: Visual, and Textual programming languages examples in Grasshopper (a) and Khepri (b) tools.

### 3.1.4 Programming Tools

Programming tools are required to use AD and to develop and deploy SM. *Jupyter Notebook* [241] is the main IDE used throughout this research since it supports both *Python* and *Julia* programming languages. Which can also be used with other AD tools.

This research uses *Python* libraries such as *Pandas* [242] and *NumPy* [243]. Additionally, the ML library *SciKit-Learn* [244] is used to develop the SM (Section 3.3). These libraries are used to develop the databases and to train, tune, and deploy the models. SM that do not require geometric modeling can be deployed with *Streamlit* [245], which is a library for *Python* that allows users to create dashboards and interact with the developed models. Thus, in Section 3.5, *Streamlit* provides user interaction with the case study. These tools work in a Integrated Development Environment (IDE), thus, in this research, their use will be described through figures and code snippets.

## 3.2 Building Information Databases

Building Information Database (BID) are referred to in this research as datasets containing multiple building characteristics and respective performance results. Building characteristics can be variables used to train the SM to predict AOP results. In some cases, BID are available from public sources or can be extracted and measured [246]. If the required databases are inaccessible or nonexistent, they can be generated with ADA or optimization processes using BPS tools (See Figure 3.1). Synthetic BID generated with ADA will be used to develop SM suitable for design exploration problems [247], while Iterative BID will be used in more complex and specific AOP, such as design execution, rehabilitation, or construction optimization problems [248, 249].

This Section describes the three different BID that can be used in the development of SM for building performance: Existing (See Existing Building Databases), Synthetic (See Synthethic Building Databases), and Iterative (See Iterative Building Databases). The first BID uses existing, extracted, or measured datasets and tunes them according to the case study. The second BID is generated by using ADA to generate multiple building archetypes that vary according to the desired features and sampling methods. The third BID uses the results from simulation-based optimizations with ADA.

### 3.2.1 Existing Building Databases

Static and Iterative sampling are suitable approaches to generate a BID. However, they can be computationally intensive, and often, existing building databases can provide the information required. These can contain architectural and construction information over the years, such as historical, technical, extracted, and measured data.

Historical databases can be obtained from building repositories and can be valuable resources to offer insight into past projects, building materials, and design trends. Technical databases provide gathered data related to the building's performance such as energy, occupancy, comfort, structural, and acoustic, among others. Extracted databases are obtained from any publicly available website such as Real Estate web pages with house or building listings. Measured data can be obtained by recording different building aspects with field devices that log real-world data.

An example of an existing database is the California housing prices dataset (Figure 3.5) [250] obtained from the open-source platform Kaggle. This database provides $\approx 20000$ house price's ($P$) and their respective feature values for latitude $lat$, longitude $lon$, age $a$, number of rooms $r$, number of bedrooms $br$, number of households $h$, population $p$, median income $i$, and ocean proximity $op$. The following equation can illustrate it:

$$f\left(\begin{bmatrix} lat & lon & a & r & br & h & p & i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ latn & lonn & a_n & r_n & br_n & h_n & p_n & i_n \end{bmatrix}\right) = \begin{bmatrix} P \\ \vdots \\ P_m \end{bmatrix} [€]$$
$$lat \in [32...42], \quad lon \in [-124...-114], \quad a \in [0...52], \quad r \in [0...6000], \quad br \in [0...6000], \quad h \in [0...6000],$$
$$p \in [0...35000], \quad i \in [0...14], \quad op \in \{\mathrm{Near\,Bay, 1h\,ocean, inland, near\,ocean, island}\} \tag{3.1}$$

where $n$ represents the number of samples in the existing BID. A case study developing a SM with this type of BID is presented in Section 4.1.

As seen in Figure 3.5, such databases can provide valuable insights regarding the impact of the described features on a house's value. However, they also frequently have limitations such as outliers, manual input errors, and lack of finer data granularity. In this case, we see values of 6000 total bedrooms and households and even 40000 total rooms. These values represent outliers that can harm the accuracy of a house price predictive model trained with this BID.

Another significant issue is the availability and interoperability of these databases, respectively their unavailability and limited integration with BPS tools. On one hand, most AOP comprise building performance objectives that cannot be measured or extracted when they are not publicly available. On the other hand, if these databases are available, they frequently lack the required quality to train an acceptable predictive model. To solve the first problem, we can use Synthetic BID (See Synthethic Building

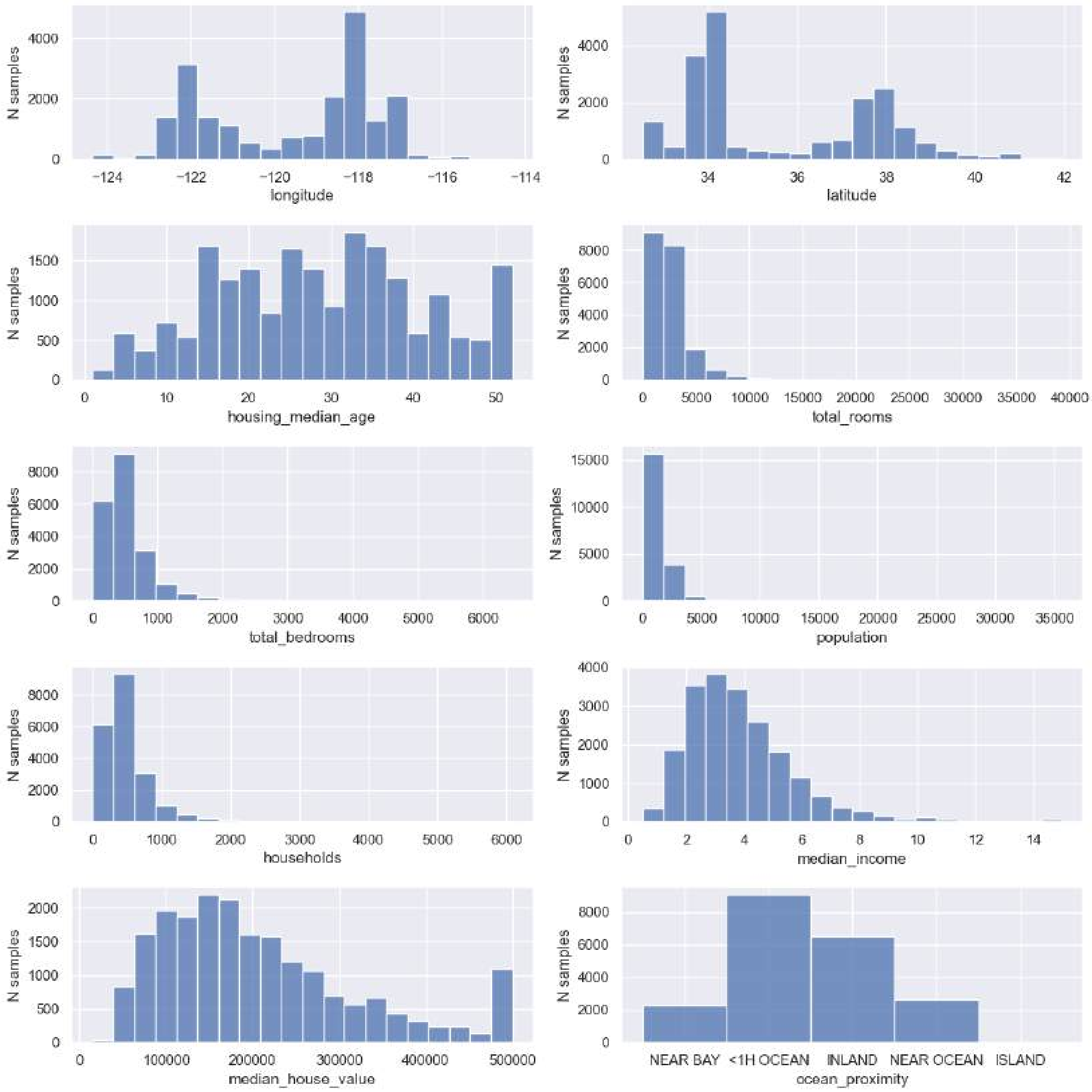



Figure 3.5: Histogram plots of the California Housing prices dataset [250].

Databases) or Iterative BID (See Iterative Building Databases) to train the SM. To solve the second problem, we can use FE and selection techniques described in Section 2.3 (See Model Tuning) to improve the database's quality.

### 3.2.2 Synthethic Building Databases

Existing BID will often not be available or not be suitable to the task at hand, particularly for AOP that comprise design exploration and require the use of BPS tools. As such, ADA approaches represent an efficient method to synthesize the required databases. To do this, a set of archetypes representative of the case study must be generated and simulated with BPS tools. These archetypes can be described by discrete and/or continuous variables (commonly known in the ML field as features) that are arguments of an algorithm developed with an AD tool. Afterward, domains that illustrate a robust variation of the existing features and their relationships are defined. Finally, building samples generated within these domains are simulated to obtain the results. The simulation process of each sample can be defined by the following equation:

$$f(x_1, ..., x_n) = y \tag{3.2}$$

where $\{x_1, ..., x_n\}$ represents the sample's training features and $y$ represents the BPS result for that sample which often is the value to be predicted by the SM (commonly known as the target feature). Consequently, the complete BID simulation results can be represented by the following equation:

$$f\left(\begin{bmatrix} x_{11} & \cdots & x_{n1} \\ \vdots & \ddots & \vdots \\ x_{1m} & \cdots & x_{nm} \end{bmatrix}\right) = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \tag{3.3}$$

where $m$ represents the desired number of samples in the BID.

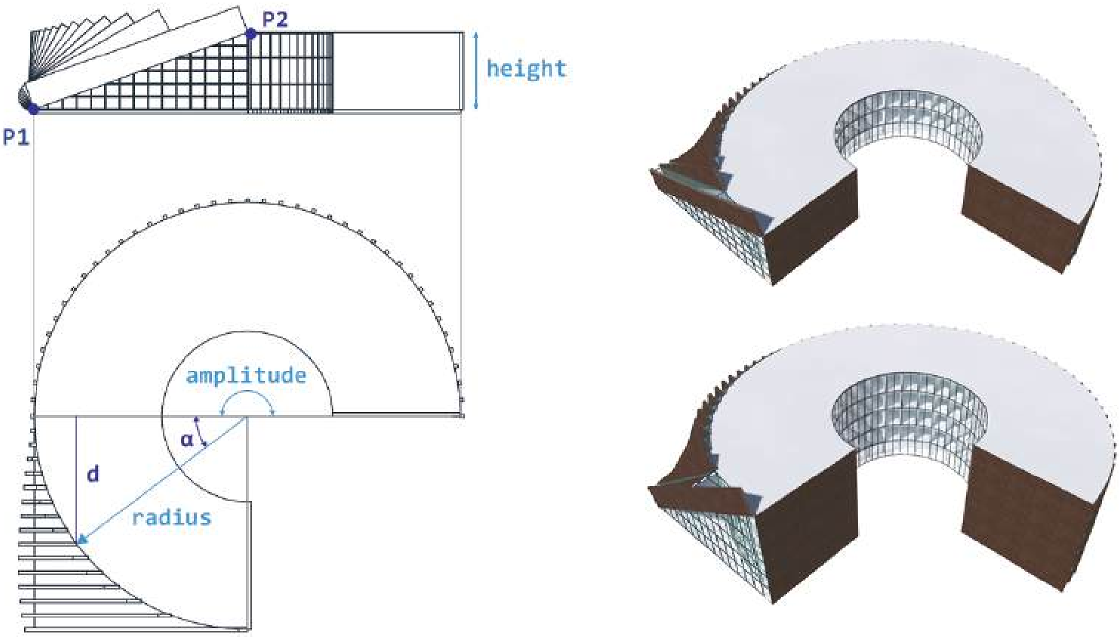
Using this approach for an example of the design case of the Isenberg School of Management by Bjarke Ingels Group (BIG) described by Castelo-Branco and Leitão [251] (Figure 3.6), it is possible to generate a Synthetic BID to train a model that predicts the energy consumption of multiple variations of this building according to a set of selected parameters and their respective domains. In this case, considering the features number of floors $n_f$, number of beams $n_b$ amplitude $\alpha$, inner radius $r_i$, and building width $\Delta r$, Equation 3.4 would represent the BID like so:

$$f\left(\begin{bmatrix} n_f & n_b & \alpha & r_i & \Delta r \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ n_{fm} & n_{bm} & \alpha_m & r_{im} & \Delta r_m \end{bmatrix}\right) = \begin{bmatrix} E \\ \vdots \\ E_m \end{bmatrix} [\mathrm{kW\,h\,m^{-2}}]$$
$$n_f \in [3...8], \quad n_b \in [20...80], \quad \alpha \in [\frac{\pi}{2}...\frac{3\pi}{2}], \quad r_i \in [5.0...30.0][\mathrm{m}], \quad \Delta r \in [10.0...40.0][\mathrm{m}] \tag{3.4}$$

where $E$ represents the energy consumption of a building sample in $\mathrm{kW\,h\,m^{-2}}$ and $m$ the number of samples in the BID. A case study with a SM developed with this type of BID is documented in Section 4.2.

The number of samples $m$ in the dataset is obtained by generating a suitable search space that explores possible combinations amongst the features domain. Thus, an increase in the number of features exponentially increases the number of samples. In this case, a highly detailed BID can be obtained by generating a linear distribution of $r_i$ and $\Delta r$ features with a step size of [1 m], $\alpha$ a step size of $\frac{\pi}{20}$, $n_b$ 1 beam, and $n_f$ 1 floor. Alas, this would generate 3.5 million possible combinations that need to be simulated to develop the BID, rendering the process unfeasible if each simulation took only 20 seconds.

This problem can be overcome through many methods such as generating distributions of combi-




(a) Isenberg School of Management geometric parameters [251].

(b) Isenberg School of Management 3D Model [251].

Figure 3.6: Isenberg School of Management project visualizations and parameters.

nations with different sampling methods. A simple solution is to increase the domains' step sizes and still be able to generate a detailed BID that can be used to train an accurate SM. Increasing $r_i$ and $\Delta r$ step size to [5 m] and $n_b$ step size to 2 would result in $\approx 5000$ simulations, which sees the cumulative simulation time reduce significantly. However, if the BPS uses a large or complex model, the cumulative simulation time may increase and take from several hours to months to complete.

Another solution would be to use the feature boundaries to generate a random normal distribution of samples according to the desired number of samples, and variables mean, standard deviation, minimum, and maximum values. In this case, the developed database is illustrated in Figure 3.7 for 1000 samples with random normal distribution values of $E$. When comparing this synthetic data with the existing housing prices dataset illustrated in Figure 3.5, we see a finer data granularity with fewer outliers and a generally better-detailed database.

Limitations associated with this method lie in the adequate number of samples to be generated, which can exponentially increase and render the process unfeasible when given large numbers of features and wide domains that need to be explored through simulation. Particularly, continuous variables can require a large discrete set of values to efficiently represent their domain (e.g., building areas, window areas, building orientation, among others), and discrete variables can have multiple classes (e.g., construction materials, construction periods, window types, multiple equipment types, among others). Additionally, SM for multiple performance aspects such as $E$ and Useful Daylight Illuminance (UDI) would require the same BID to be simulated with both simulation engines (e.g., EnergyPlus and Radiance). When both the number of features and number of objectives are compromising the computational time of the BID

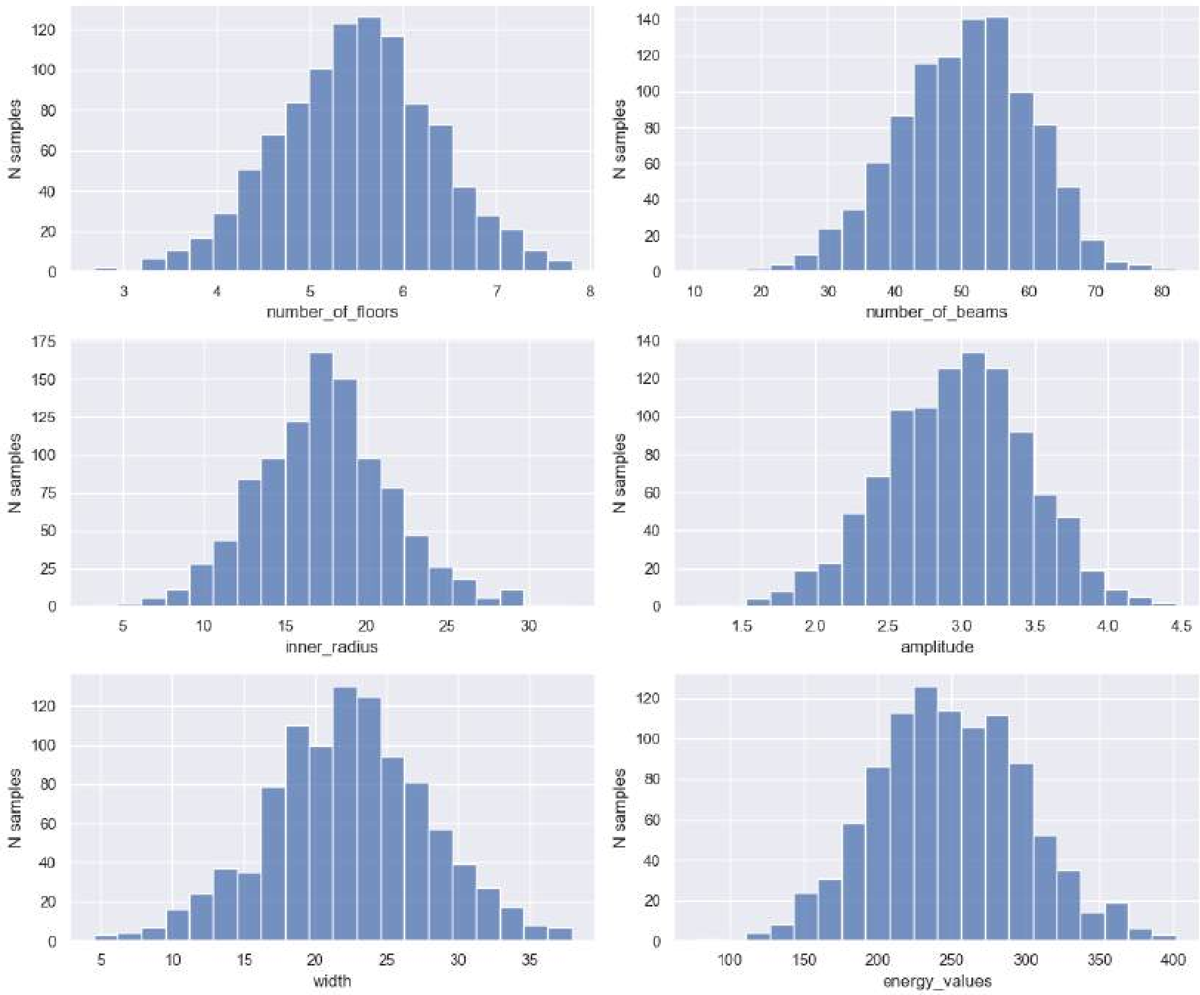



Figure 3.7: Histogram plots of the generated database of Isenberg School of Management with normal distributions of features.

development, the next solution is to test an Iterative BID.

### 3.2.3 Iterative Building Databases

For more specific problems of the AEC industry (e.g., construction materials properties optimization, rehabilitation, construction, design, and execution of specific projects), optimization is a suitable iterative method to generate an Iterative BID because it requires fewer samples to efficiently train the SM and, therefore, less computational time to perform the final optimization [24, 122]. This method consists of developing an optimization problem that requires BPS and using past iterations and respective results obtained by early explorations of the optimization algorithm to generate the BID.

An optimization problem is solved by finding the combination of variables within specified domains that yields the minimum or maximum value of an objective function represented by the following equation:

$$f(x_1, ..., x_i) = min(o_1, ..., o_j) \tag{3.5}$$

where $i$ represents the number of variables and $j$ the number of objectives. Optimization problems typically entail multiple objectives which are often conflictive [122, 118]. Additionally, most building performance indicators are outputs of BPS, or extensive calculations performed by field experts [24]. Because of this, optimization problems that are focused outside the practitioners' realm of knowledge are usually treated as Multi-Objective Optimization (MOO) problems with derivative-free functions [122, 123, 125].

Metaheuristics, a class of optimization algorithms, have been widely used in the field of optimization for building and urban performance optimization with positive results. These algorithms guide their search based on biological heuristics such as evolutionary, and swarms of different kinds, among others [137]. When one algorithm performs a sufficient amount of iterations, it is possible to use them as a BID defined by:

$$f\left(\begin{bmatrix} x_{11} & \cdots & x_{1i} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mi} \end{bmatrix}\right) = \begin{bmatrix} o_{11} & \cdots & o_{1j} \\ \vdots & \ddots & \vdots \\ o_{m1} & \cdots & o_{mj} \end{bmatrix} \tag{3.6}$$

where $m$ represents the specified number of iterations performed by the optimization algorithm and $j$ is the number of objectives. A case study of a SM developed with this type of BID is presented in Sections 4.3 and 4.4 of this thesis.

Expanding on the Isenberg School of Management design case described in Subsection Synthethic Building Databases (Equation 3.4), it is possible to have a more complex problem that can be overcome with an Iterative BID. A performance design problem to minimize a building's energy consumption $E$ in $\mathrm{kW\,h\,m^{-2}}$ and construction cost $C$ in €, and maximize the building's UDI (%) could consider the additional variables beam width $b_w$, 3 beam materials $b$, 3 glazing materials $g$, 3 wall constructions $w$, and orientation $\theta$. In addition, lighting simulation requires an extra set of simulations. Thus, the BID is represented by the following equations:

$$f(n_f, n_b, \alpha, r_i, \Delta r, b_w, b, g, w, \theta) = min(E, C, -UDI) \tag{3.7}$$

$$f\left(\begin{bmatrix} n_f & n_b & \alpha & r_i & \Delta r & b_w & b & g & w & \theta \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ n_{fm} & n_{bm} & \alpha_m & r_{im} & \Delta r_m & b_{wm} & b_m & g_m & w_m & \theta_m \end{bmatrix}\right) = \begin{bmatrix} E & C & UDI \\ \vdots & \vdots & \vdots \\ E_m & C_m & UDI_m \end{bmatrix} \tag{3.8}$$

$$n_f \in [3...8], \quad n_b \in [20...80], \quad \alpha \in [\frac{\pi}{2}...\frac{3\pi}{2}], \quad r_i \in [5.0...30.0][\mathrm{m}], \quad \Delta r \in [10.0...40.0][\mathrm{m}]$$

$$b_w \in [0.2...1.0][\mathrm{m}], \quad b \in [0,1,2], \quad g \in [0,1,2], \quad w \in [0,1,2], \quad \theta \in [0...2\pi[$$

To solve this optimization problem, ADA can be used to automate the *EnergyPlus* and *Radiance* simulations according to the solutions chosen by the optimization algorithm. Any algorithm can be selected or combined with others for the desired number of iterations. A higher number of iterations will result in a more detailed BID. However, because the distribution of the BID feature values in Equation

3.8 can be skewed, it might fail when predicting feature values not as well explored by the algorithm.

An example of this is illustrated in Figure 3.8 where the Non-Dominated Sorting Genetic Algorithm (NSGA)II [141] algorithm was run for 5000 iterations to develop the BID (Equation 3.8)). It is visible that the feature sample distributions converge toward each respective optimum value, which translates into an imbalanced BID when compared with the one generated in Figure 3.7. This can harm the SM performance. Notably, the objective results were generated without simulations, hence they are excluded from the figure.

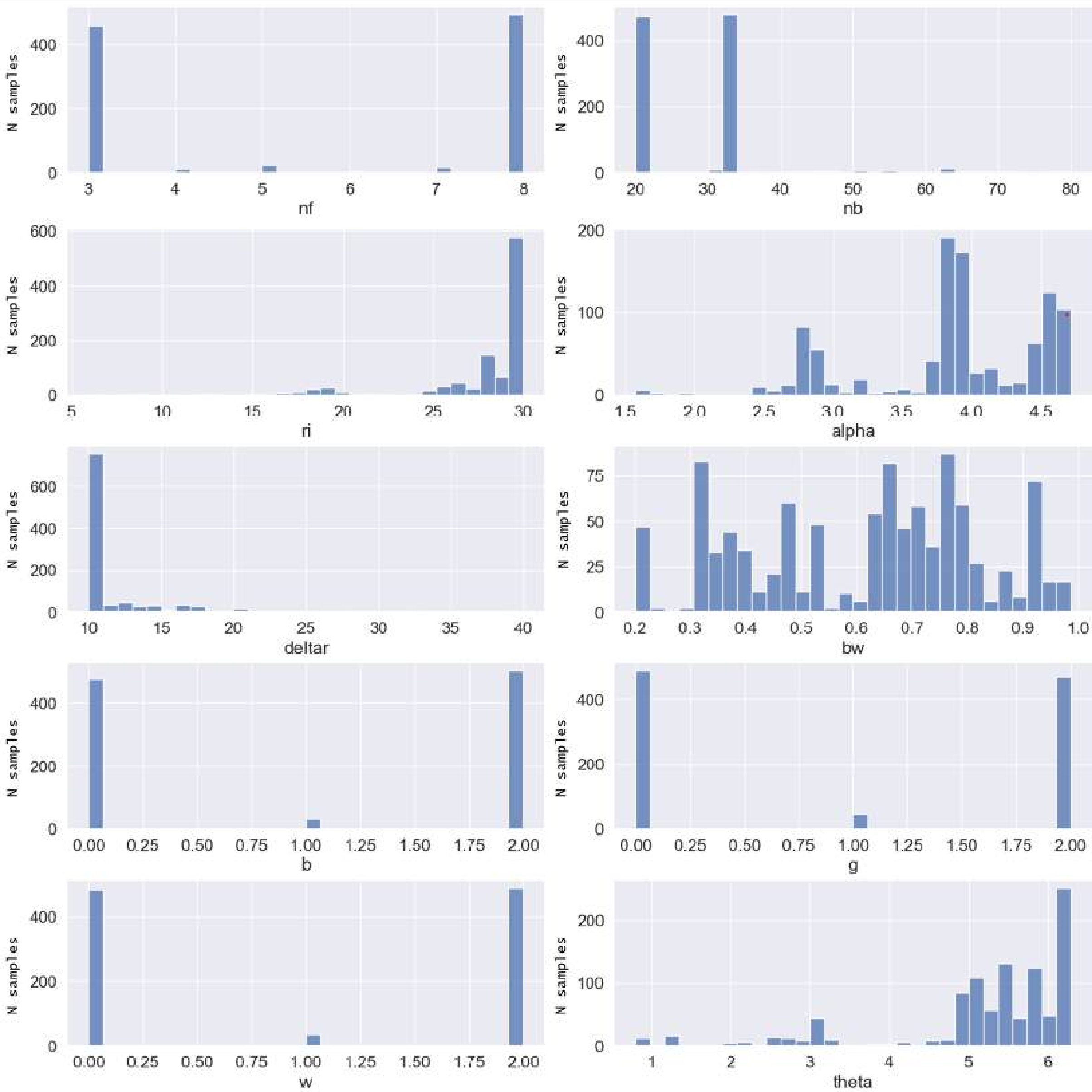



Figure 3.8: Histogram plot of the optimization search space of Isenberg School of Management design variables.

Wolpert and Macready [145] state with the "No Free Lunch" theorem that no algorithm outperforms all others for all problems. This means that one has to either know from experience which algorithms yield better performances for each problem [138] or test multiple metaheuristics to find the best one.

Moreover, these algorithms have parameters that define how they search for the optimal space called Hyperparameters. These can be fine-tuned before deploying the final SM and AOP to provide optimum results [218].

A widely used key performance index for optimization algorithms is the Hypervolume (H) [148, 149, 150]. This metric (Section 2.2, see Building Performance Optimization) describes how widely the algorithm explored the problem solutions by calculating the n-dimensional space that a set of non-dominated solutions contains. Thus, the optimization problem to maximize the optimization algorithm performance can be defined by the following equation:

$$f(h, ..., h_n) = max(H(o_1, ..., o_j)) \tag{3.9}$$

where $n$ represents the number of Hyperparameters of the optimization algorithm and the objective $H$ is the calculated H obtained from the algorithm's non-dominated solutions results. Given that the accuracy of SM can be improved with Hyperparameter optimization as well, this procedure will be covered in more detail in the Model tuning Section (3.4) for the performance of both SM and optimization algorithms.

## 3.3 Model type

This Section follows the development of a BID by using it to test multiple SM and choosing the most adequate one for the model tuning (Section 3.4) and deployment (Section 3.5) stages. To do this, each model must be trained, evaluated, and compared regarding its overall performance in predicting the target feature. The target feature can be any desired output from an AOP.

Because AEC industry problems can often entail multiple objectives and outputs [118, 122], a SM must be produced for each necessary output, which in turn is often a category or a continuous variable. Classification and regression models respectively predict discrete and continuous variables. These models will be discussed throughout this Section regarding their use and performance evaluation, and in the end, the models' performance is compared, and the most suitable one is selected based on the respective evaluation metrics. This model is then tuned and deployed to make predictions on upcoming data.

### 3.3.1 Classification models

SM that need to predict binary or multiple classes are designated classification models. SciKit-Learn [252] provides an extensive library of models suitable for classification tasks. In this research process, the supervised learning models that can be tested are Gaussian Process (GP), Logistic Regression (LR), Support Vector Machines (SVM), K-Nearest Neighbors (KNN), ensembles, and Artifical Neural Networks (ANN) models since these have been extensively used to replace computationally expensive simulations for similar problems (See Section 2.3, Types of Models). Finally, a smaller fraction of the BID will be used to test and evaluate the performance of the SM.

Several performance indicators can be used to evaluate the performance of these models, including precision, recall, F1 score, accuracy, and confusion matrices. The F1 score measures the balance between precision and recall, while accuracy measures the proportion of correct predictions. The confusion matrix provides a detailed breakdown of true positive, true negative, false positive, and false negative predictions. Finally, a comparison is made between the performance of each SM and the best is selected based on the chosen metrics. If the SM is still not performing adequately, it can be tuned to improve its accuracy. The model is then deployed and used to make predictions on new data.

Based on the house prices example described in Subsection Existing Building Databases, it is possible to segment house prices into 5 categories with intervals of ≈100k € evenly distributed among maximum and minimum house prices (See Figure 3.5). The classes are labeled in crescent order, thus higher classes have higher prices than lower classes. Afterward, it is necessary to split the database into training and test samples, which will respectively train and evaluate the performance of the developed model. Finally, we can train the selected models and evaluate them by calculating their precision, recall, f1-score, accuracy, and confusion matrices. In this case, comparing the performance of sample models LogisticRegression, KNeighborsClassifier (Instead of GP classifier since the database is large and it would be computationally intensive), Extra Trees (ET) Classifier, and Multi-Layer Perceptron (MLP) Classifier could be done with Python Code Listing 3.1.

This Code Listing imports the required libraries from Scikit Learn and Pandas, imports the housing prices database, and splits it into training and testing samples with a respective 70/30% ratio, which is an acceptable split of the data for the model development (See Chapter 2, Section 2.3). Afterward, it creates a dictionary of instances of all the selected models and, for each selected model, fits, predicts the test samples, calculates the performance indicators, and stores these results in the created dictionary. The stored results can be used to visualize and compare the classification reports and the confusion matrices in a heatmap of each developed model as seen in Figure 3.9.

By analyzing the classification reports, it is possible to see that model accuracy ranges from 0.35 to 0.67, with the Extra Trees Classifier having the best performance. Additionally, by inspecting the confusion matrices, it is possible to see that all models struggled with predicting "Class 3" and "Class 4", as well as with "Class 2" and "Class 3", mislabelling them often. These results were obtained without any data preprocessing and model tuning techniques. To achieve acceptable model accuracy, one can use multiple techniques such as feature engineering, feature selection, and Hyperparameter optimization, among others. These will be described in more detail in Section 3.4.

These results demonstrate the accuracy of simple SM developed to predict housing price categories. However, if the goal was to predict the exact house price, a regression model would have to be used. The next Subsection covers the same approach for regression models by training the models, predicting test samples, calculating and storing model performance indicators, and comparing their results.

### 3.3.2 Regression models

SM that need to predict continuous values are named regression models, which are used in this study to predict a target variable. The regression models that will be used are Linear Regression (with different polynomial degrees), GP Regressor, SVM, KNN, Ensemble models, and ANN. The goal is to identify the

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, KBinsDiscretizer
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.neural_network import MLPClassifier
# Read CSV file from housing prices database
houses = pd.read_csv("housing.csv").dropna()
# Split the database between features and target prediction output
X = houses.drop("median_house_value",
                axis=1)
y = houses["median_house_value"]
# Segment housing prices into 5 classes
categorizer = KBinsDiscretizer(n_bins=5,
                               encode='ordinal',
                               strategy='uniform')
y_discrete = categorizer.fit_transform(y.values.reshape(-1,1)).ravel()
#Split database into 70% training and 30% testing samples
X_train, X_test, y_train, y_test = train_test_split(X_scale,
                                                    y_discrete,
                                                    train_size=0.7,
                                                    test_size=0.3,
                                                    random_state=42)
# Initialize the models
models = {"Logistic Regression": LogisticRegression(),
          "KNeighbors": KNeighborsClassifier(),
          "Extra Trees": ExtraTreesClassifier(),
          "MLP": MLPClassifier(max_iter=1000)}
# Results storage
results = {}
# Iterate over the models, train, predict, and store results
for name, model in models.items():
    # Fit model
    model.fit(X_train, y_train)
    # Predictions
    y_pred = model.predict(X_test)
    # Calculate confusion matrix
    cm = confusion_matrix(y_test, y_pred)
     # Generate classification report
    report = classification_report(y_test, y_pred, output_dict=True)
    # Store results
    results['Classification_models'][name] = {'Confusion Matrix': cm,
                                             'Classification Report': report}
```

Code Listing 3.1: Code to develop classification models capable of predicting housing price categories and store their results in a dictionary.

best-performing model providing the most accurate predictions.

To evaluate the performance of the regression models, the following metrics will be used: Coefficient of determination ($R^2$), Root Mean Squared Error (RMSE), Mean Squared Error (MSE), Mean Absolute Percent Error (MAPE), and other metrics suggested by the literature (see Section 2.3, Types of Models). After evaluating the performance of each model, the one that produces the lowest errors and highest

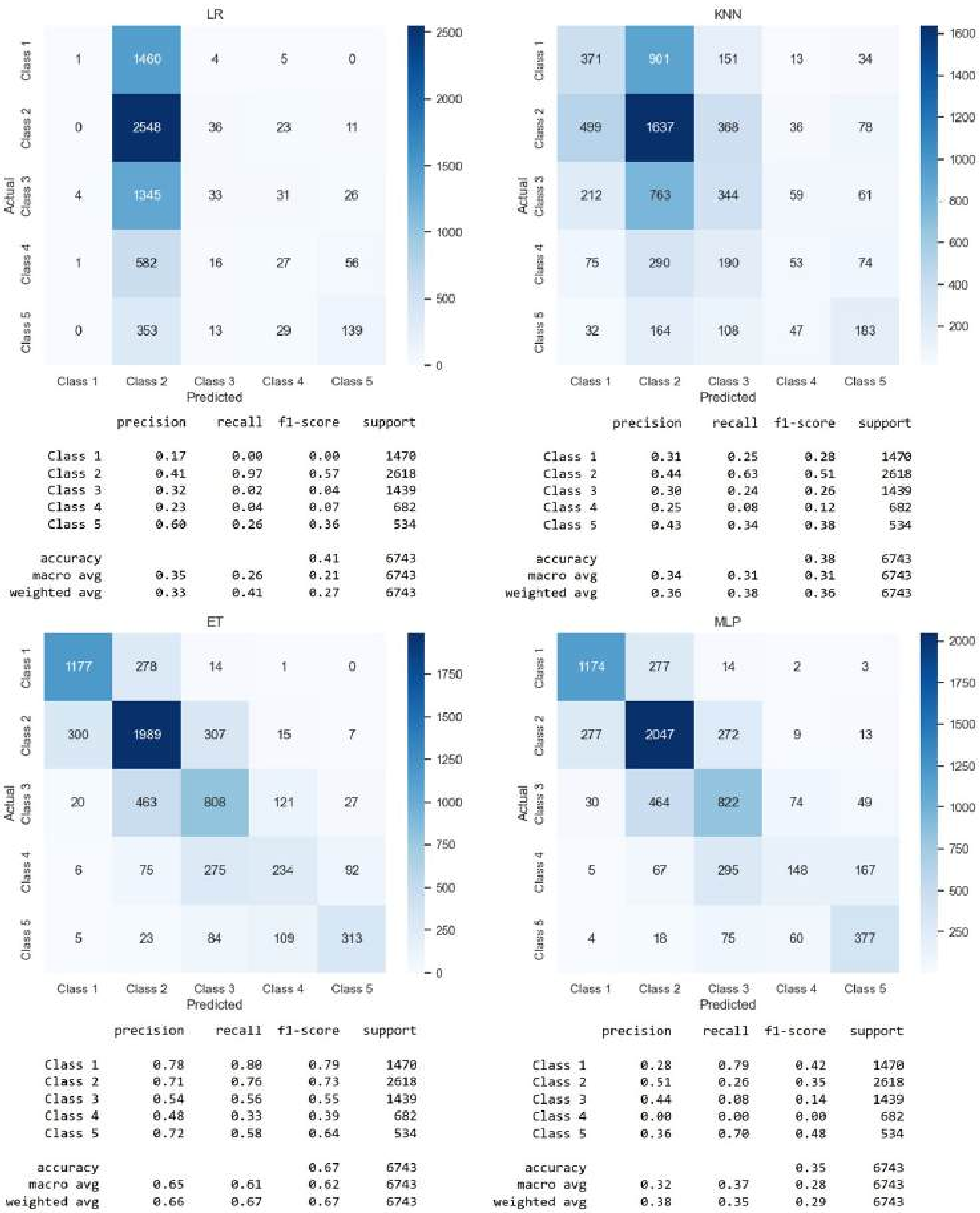



Figure 3.9: Confusion matrices and classification reports of selected models seen in Listing 3.1.

$R^2$ can be selected. Because most building performance metrics fit into this category, they are the most frequently used models in this research.

Applying this to the California housing prices dataset (Subsection Existing Building Databases, Figure 3.5), one can aim to develop a SM that predicts the exact housing price given the database's features. To do this, a few adjustments must be made to the Code Listing 3.1. In this code, the regression models are imported, along with the regression performance metrics. Afterward, the same train/test split of the samples is performed, but for the continuous values of the housing prices. Finally, the models are instantiated and trained iteratively and the performance results are stored in the results dictionary. This process is documented in Code Listing 3.2.

```python
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.neural_network import MLPRegressor
# Read CSV file from housing prices database
houses = pd.read_csv("housing.csv").dropna()
# Split the database between features and target prediction output
X = houses.drop("median_house_value",
                axis=1)
y = houses["median_house_value"]
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    train_size=0.7,
                                                    test_size=0.3,
                                                    random_state=42)
# Initialize the regression models
models = {
    "Linear Regression": LinearRegression(),
    "KNeighbors Regressor": KNeighborsRegressor(),
    "Extra Trees Regressor": ExtraTreesRegressor(random_state=42),
    "MLP Regressor": MLPRegressor(max_iter=1000, random_state=42)
}
# Iterate over the models, train, predict, and store results
for name, model in models.items():
    model.fit(X_train, y_train)  # Fit model
    y_pred = model.predict(X_test)  # Predictions
    rmse = np.sqrt(mean_squared_error(y_test, y_pred)) # Calculate RMSE
    r2 = r2_score(y_test, y_pred)  # Calculate R2
    # Store results
    results['Regression Models'][name] = {'RMSE': rmse,
                                          'R2': r2}
```

Code Listing 3.2: Code to develop Regression models capable of predicting housing prices and store their results in a dictionary.

Obtained results for this example show Coefficient of determination ($R^2$) from $\approx$0.3, to 0.8, and RMSE ranging from $\approx$ 50 to 100 thousand \$ (Figure 3.10). From these results, the ET Regressor outperforms all others in predicting housing prices and should be considered as the best-performing model. However, one can visualize the results with error histograms as seen in Figure 3.11 if a more detailed error distribution analysis is required. The obtained RMSE and $R^2$ of 0.8 is an acceptable and informative model of the housing prices. Although, all the other models have shown insufficient accuracy. Thus, all these models can still be improved with data processing and model tuning techniques. The next Section elaborates on these processes and demonstrates their application in this research.

## 3.4 Model tuning

To improve the performance of predictive models many techniques can be used either to refine the BID and/or the model parameters. To refine the database it is possible to use feature engineering techniques that aim to scale, prune, or change the BID's features. To find the best SM parameters, one can use

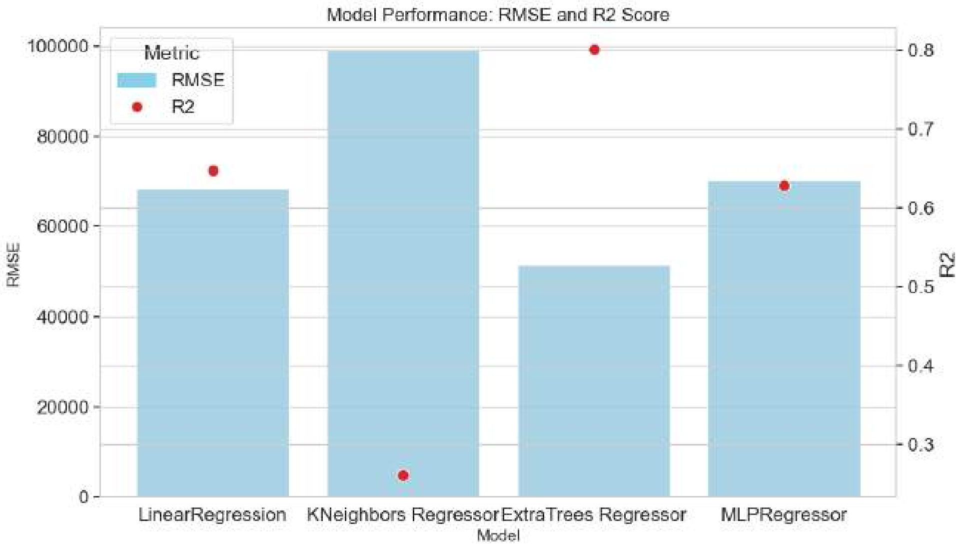



Figure 3.10: $R^2$ score and Root Mean Squared Error of housing prices for selected models seen in Listing 3.2.

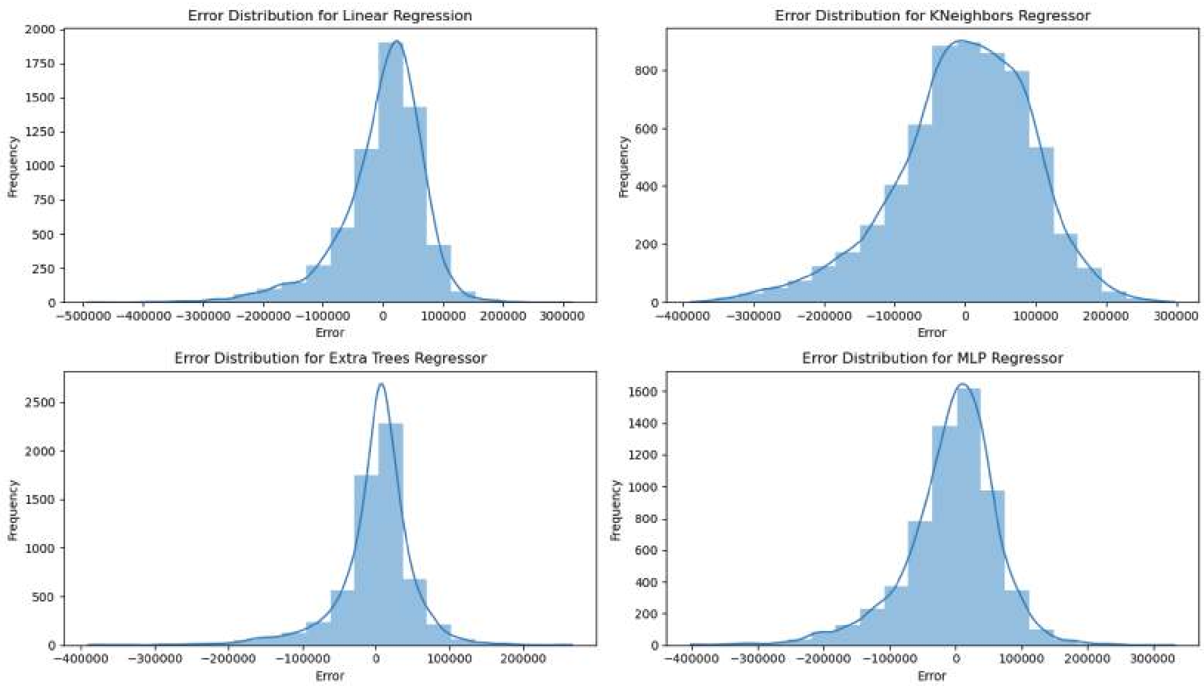



Figure 3.11: Error histogram ($) of housing prices for selected models seen in Code Listing 3.2.

Hyperparameter tuning techniques to explore and find the best combination of model parameters that yield the best predictive results (See Section 2.3, Model Tuning).

This section documents possible tuning strategies that can be used to improve a model's performance. This is done by segmenting and demonstrating the improvements obtained from these strategies when applied to some of the previous models seen in Section 3.3.

### 3.4.1 Feature engineering

To improve the model by refining the BID, one can use feature engineering techniques to tackle data issues such as dimensionality, noise, and feature imbalance. These issues can negatively affect the performance of a model, harming its predictive capabilities. To tackle these problems, one can create features from existing ones to unveil latent patterns and relationships in the data [204, 253], prune the database with sampling techniques [254, 255, 256], and scale the data to suitable values to help identify abnormalities and make the data more suitable to be interpreted by the model [208, 257].

To unveil latent patterns and relationships in the BID it is possible to create features through mathematical transformations, combinations, and derivations from existing variables. These techniques can range from simple multiplications such as finding a simple bounding volume from a building area and height to creating complex interactions of powers of existing features to capture nonlinear data relationships (Polynomial features).

To address noise problems such as outliers and abnormal values it is possible to use multiple techniques within pandas [242] libraries, and to deal with class imbalances one can use the imbalanced learn library [258], which provides techniques such as Random Undersampling [259] and Tomek Links [260]. These techniques are also useful when the BID is large and an obstacle to tuning the model.

Some models cannot handle high dimensional data and usually benefit their prediction scores with Feature Scaling [208]. This can include Normalizing the data [164], scaling the data to have a mean of 0 and standard deviation of 1 (Standard scaling), or scaling the data to a predefined boundary (MinMax scaling) [261].

While analyzing the housing prices distribution plot in Figure 3.5 it is possible to see that the features ”total_rooms”, ”total_bedrooms”, ”households”, and ”population”, have skewed results. Thus, it is possible to remove samples that largely diverge from the observed distribution and scale the data with Standard scaling. This process is illustrated in Figure 3.12.

Building on the regression models developed in Subsection Regression models (Code Listing 3.2), it is possible to try and improve the models' $R^2$ and RMSE with some of the feature engineering techniques described above. As an example, Polynomial features of $2^{nd}$ degree can be applied to the house prices features, which can then be scaled using standard scaling. To do this, it is required to apply the polynomial features transformation to the unsplit database, re-split the database into training/test samples, scale the train samples features, and finally retrain the imported models (documented in Code Listing 3.3).

By comparing the obtained results with the results of the developed models without Feature engineering in Figure 3.13, it is possible to see significant improvements in both $R^2$ and RMSE of the Linear Regression, KNeighbors, and MLP regressor, while the ET regressor did not obtain any improvements. To further improve these models' $R^2$ and RMSE, the next step in this research comprises Hyperparameter tuning, which will be detailed in the next Subsection.

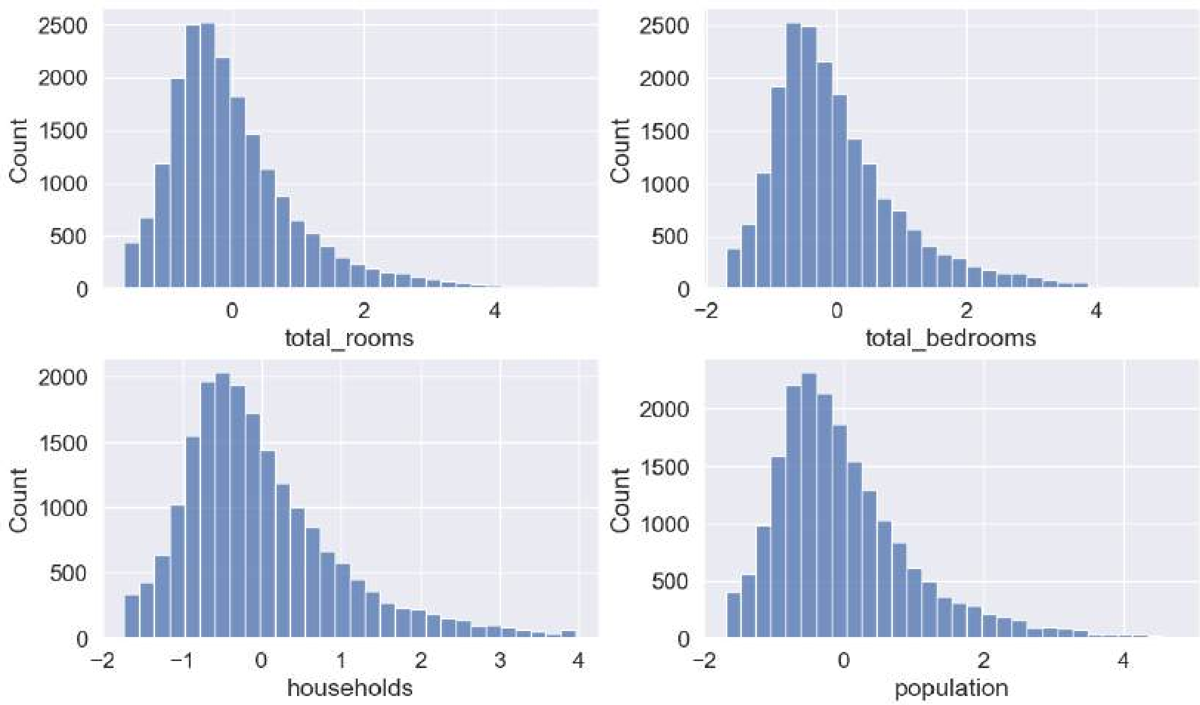



Figure 3.12: Feature distributions seen in Figure 3.5 after removal of abnormal sample values and scaling the data.

```
from sklearn.preprocessing import StandardScaler, PolynomialFeatures
# Initialize models
models = {"Linear Regression": LinearRegression(),
          "KNeighbors": KNeighborsRegressor(),
          "Extra Trees": ExtraTreesRegressor(random_state=42),
          "MLP": MLPRegressor(max_iter=1000, random_state=42)}
# Apply Polynomial Features
poly = PolynomialFeatures(degree=2)
X_poly = poly.fit_transform(X)
# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X_poly,
                                                    y,
                                                    test_size=0.3,
                                                    random_state=42)
# Scale the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
# Train and evaluate models with feature engineering
for name, model in models.items():
    model.fit(X_train_scaled, y_train)
    y_pred = model.predict(X_test_scaled)
    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_test, y_pred)
    results["With FE"][name] = {'RMSE': rmse, 'R2': r2}
```

Code Listing 3.3: Code to apply feature engineering techniques to housing prices database, train the models, and store their results in a dictionary.

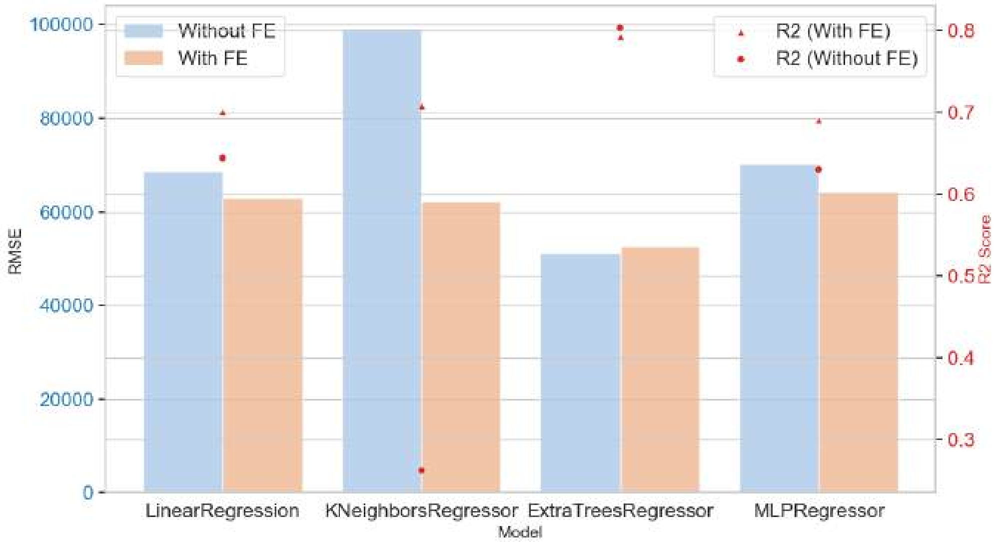



Figure 3.13: Coefficient of determination ($R^2$) and Root Mean Squared Error (RMSE) of housing prices for comparison of the developed models (seen in Subsection Regression models) with and without Feature engineering (FE).

### 3.4.2 Hyperparameter tuning

Hyperparameter tuning aims to find the combination of parameters of a SM that yield the best performance results [205]. In this research, this is achieved by using search algorithms provided in libraries such as scikit learn [252], or by using optimization algorithms such as metaheuristics [122, 141, 142] or a Bayesian optimization approach that uses a Gaussian Process (GP) as the SM for the objective function [219, 220].

Two Hyperparameter search functions available in scikit learn are the grid search cross-validation and the randomized search cross-validation. The grid search comprises an exhaustive search within a grid over specified parameter boundaries. Due to its computationally intensive nature, it is suitable for a limited number of Hyperparameters, with a small search space. For larger Hyperparameter spaces that are computationally intensive, one can use the randomized search cross-validation. This function implements a randomized search over the Hyperparameters, where each setting is sampled from a distribution over their Hyperparameter's boundary.

In Bayesian optimization, the GP serves as a probabilistic model that approximates the performance of a SM with the given Hyperparameters and within their specified boundaries. By using this approach, the optimization process navigates the Hyperparameters solution space until convergence to the best configuration of Hyperparameters that yields the maximum chosen performance metric.

Building on the house prices regression models with Feature Engineering (FE) seen in Code listing 3.3, it is possible to apply Hyperparameter tuning for some models. Considering that the only model that did not respond well to FE was the ExtraTress Regressor, and that the Linear Regression model has no Hyperparameters to tune, it is possible to try and further expand the performance of both MLP and KNeighbors Regressors with a bayesian optimization of their Hyperparameters. This can be achieved by maximizing the $R^2$ score with the following equations:

$$
\begin{gathered}
f(\text{KNN}(n, w, a, l_s, p)) = min(R^2) \\
n \in [1...200], \quad w \in \{\text{uniform}, \text{distance}\}, \quad a \in \{\text{auto}, \text{balltree}, \text{kdtree}, \text{brute}\}, \\
l_s \in [10...50], \quad p \in [1...3]
\end{gathered} \tag{3.10}
$$

$$
\begin{gathered}
f(\text{MLP}(l_1, l_2, l_3)) = min(R^2) \\
l_1 \in [10...200], \quad l_2 \in [10...200], \quad l_3 \in [10...200], \quad l_4 \in [10...200]
\end{gathered} \tag{3.11}
$$

where the Hyperparameters studied for the optimization of the KNeighbors Regressor are $n$ neighbors, weight $w$, algorithm $a$, leaf size $l_s$, and power $p$ and the Hyperparameters for the MLP Regressor are the number of neurons for each layer.

Equation 3.10 can be implemented in Code Listing 3.4 with a function integrated with the Optuna package library [262] that explores the solutions with a Bayesian optimization approach. This function takes an Optuna solution trial and trains an MLP Regressor with an undersampled version of the training database with the respective trial parameters. It then evaluates the model using the respective function and returns the resulting minimum negative $R^2$ (since the goal is to maximize and the direction of the study is set to minimize). After the optimization, the model is retrained with the full training sample and with the best parameters found, and the performance results are stored in the dictionary.

After performing the optimization for both KNeighbors and MLP Regressors, it is possible to visualize the performance improvements by plotting the previously obtained results dictionary in Figure 3.13 in combination with the obtained optimized results as seen in Figure 3.14. The resulting optimization to the MLP Regressor improved its $R^2$ by $\approx$ 14% and decreased its RMSE by $\approx$ 17% when compared with the model that was trained using only FE. However, KNeighbors has only mildly improved its performance with optimization when compared with its FE predecessor results, but the FE processes influenced significantly its performance when compared with KNeighbors without FE and optimization.

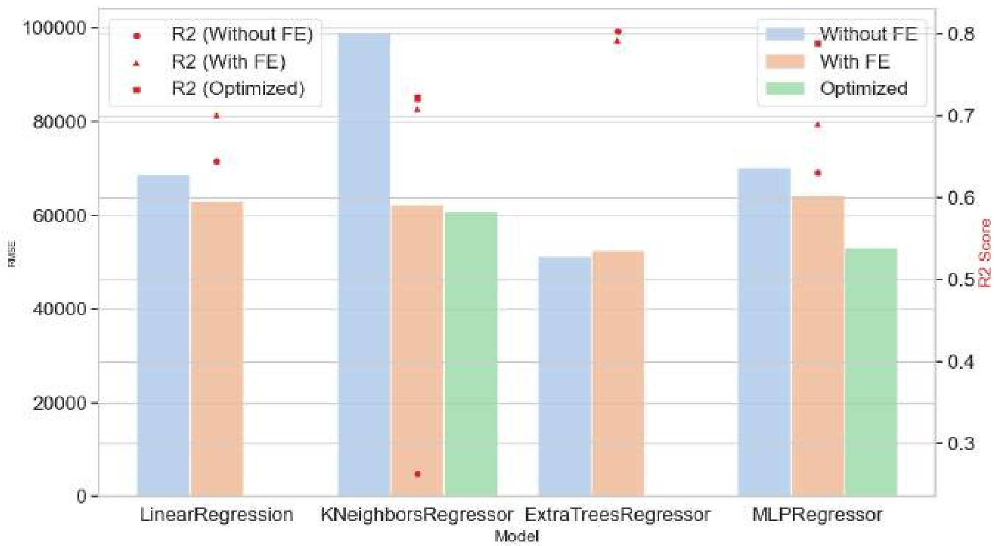



Figure 3.14: Coefficient of determination ($R^2$) and Root Mean Squared Error (RMSE) comparison of the developed models (seen in Subsection Regression models) with and without Feature engineering (FE), and with their Hyperparameters optimized.

```python
import optuna
from imblearn.under_sampling import RandomUnderSampler
#Undersample the data
rus = RandomUnderSampler()
X_train_res, y_train_res = rus.fit_resample(X_train_scaled, y_train)
# Function to calculate RMSE and R2 and evaluate a model
def evaluate_model(model):
    predictions = model.predict(X_test_scaled)
    rmse = mean_squared_error(y_test, predictions, squared=False)
    r2 = r2_score(y_test, predictions)
    return rmse, r2
# Function to optimize MLP Regressor
def optimize_mlp(trial):
    define trials for layers 1,2, and 3
    l1 = trial.suggest_int('n_l1', 10, 200)
    l2 = trial.suggest_int('n_l2', 10, 200)
    l3 = trial.suggest_int('n_l3', 10, 200)
    #Train the model with the respective neurons for each layer
    model = MLPRegressor(hidden_layer_sizes=(l1,
                                             l2,
                                             l3),
    #Reduce the number of iterations for faster optimization
                         max_iter=250,random_state=42)
    model.fit(X_train_res,
              y_train_res)
    rmse, r2 = evaluate_model(model)
    #return value to minimize
    return -r2

study = optuna.create_study(direction='minimize')
study.optimize(optimize_mlp, n_trials=50) #Define number of iterations
mlp_params = study.best_params
best_mlp_model = MLPRegressor(hidden_layer_sizes=(mlp_params['l1'],
                                                  mlp_params['l2'],
                                                  mlp_params['l3']),
                              max_iter=1000, random_state=42)
best_mlp_model.fit(X_train_scaled, y_train)
best_mlp_rmse, best_mlp_r2 = evaluate_model(best_mlp_model)

results["Optimized"]['MLP'] = {'RMSE': best_mlp_rmse,
                               'R2': best_mlp_r2,
                               'params': mlp_params}
```

Code Listing 3.4: Code to optimize the number of neurons of a four-layered MLP and store the results in a dictionary.

This Model Tuning process can be skipped altogether throughout this research's studies if the developed SM already provides acceptable accuracy in Stage 3 (Section 3.3) and can also be used to optimize the performance of an optimization algorithm for building AOP. Differences consist only in the objectives of the optimization, which in this case should target optimization performance metrics such as minimum and maximum values obtained for one objective, and Hypervolume (H) for multiple objectives. These metrics allow the evaluation of the explored solution space, which can be particularly useful to train less biased SM with iterative approaches (See Subsection Iterative Building Databases) and to obtain better results with the optimization process (See Section 2.2, Building Performance Optimization). Building on the example described in Equation 3.9 and Figure 3.8, it is possible to optimize the parameters population size $ps$, crossover $c$, and mutation $m$ of an NSGAII algorithm to yield the maximum H [148, 149, 150] with Equation 3.12.

$$
\begin{gathered}
f(\text{NSGAII}(ps, c, m)) = max(H) \\
p_s \in [ps_{min}..ps_{max}\}, \quad c \in \{c_1, ..., c_i\} \quad m \in \{m_1, ..., m_n\}
\end{gathered} \tag{3.12}
$$

Finally, with the model tuning stage of this research described (Figure 3.1), the next Section comprises the description of the Model deployment stage. This is done by integrating the developed model with optimization case studies regarding building performance and deploying it into a suitable interface, which will be described in the next chapter.

## 3.5 Model deployment

SM such as the ones described in the previous Sections can provide many benefits to the efficient implementation of Analysis and Optimization Processes (AOP) in building performance problems. Particularly, they can predict performance results much faster, and because they require fewer inputs and calculations, they require less expertise to integrate with AOP than conventional methods. Moreover, because they can be deployed in multiple environments and tools, SM have the potential to improve portability issues associated with the different spectrum of building performance indicators (See Chapter 2, Challenges and Opportunities). Thus, Model deployment is a vital stage in this research to ensure a suitable integration of a SM with a building AOP (Figure 3.1).

The research strategy adopted for the developed SM throughout this work is described in this Section, with particular focus on the types of AOP that will be integrated with the SM (Subsection Case studies), and on their respective interactive interface (Subsection Interfaces). First, case studies are segmented and described according to their different types of BID, described in Section 3.2. These consist of different building performance problems that are tackled with the development of a SM following the flowchart described in Figure 3.1. Developed case studies will tackle typical building performance problems such as Design, Construction, and Rehabilitation, which tackle different aspects of a building's life cycle.

Finally, each case study can have a different working environment, such as programming tools, Computer-Aided Design (CAD) tools, BPS tools, and Algorithmic Design (AD) tools. This is expected and faced as an opportunity to test the flexibility of the developed SM in being applied to different user interfaces. Thus, Subsection Interfaces describes different working environments and interfaces that can be used throughout this research.

### 3.5.1 Case studies

The case studies integrated within this research aim to illustrate the process of SM development to solve limitations associated with AOP of building performance. To do this, they are segmented according to the specified BID types (Existing, Synthetic, and Iterative Building Information Database (BID)) defined in Section 3.2. Additionally, during its lifecycle, a building performance problem that is solved with AOP may encompass different expertise fields such as Design, Construction, and Rehabilitation.

Expanding on the BID types described in Section 3.2, it is possible to implement the SM developed in 3.3 and 3.4 for case studies that leverage the advantages of such models to solve building performance problems. For example, a model developed with an Existing database such as the California housing prices in Section 3.2 (See Existing Building Databases) can be used to improve the return on investment of real estate development in California. This can be achieved through a parametric analysis of an investment's Return on Investment (ROI), or a Bayesian optimization process to maximize the ROI with the given variable domains.

A "Construction" optimization problem to maximize ROI of the construction of a building block and respective housing sales can benefit from more user-friendly features such as rooms, bedrooms, and people per house (obtained from dividing the respective features by the total households in the block). Afterward, it is possible to train the best-performing model obtained in Section 3.3 (In this case the Extra trees Regressor) with these new features and predict a housing price. From there, it is possible to approximate the total investment cost $C_i$ assuming an average construction cost per m$^2$ $C$, an average room area $A$, the number of households $n_h$, and number of rooms per household $n_r$ (Equation 3.13). The housing sales price $P$ can be predicted with the developed model, and then multiplied by the number of houses to obtain the investment returns $I$ (3.14). Then, ROI is calculated with Equation 3.15.

Finally, the objective function and its variables can be represented by the Equation 3.16 according to the variables $lat$, $lon$, $n_r$, $n_h$, area population density per house $n_p$, and area median income $m_i$, and where $i$ is the number of iterations performed by the optimization. This yields a single objective optimization where the best solution is the combination of these parameters that provide the highest returns. Notably, this function can be adapted to any performance indicator that depends on housing price to provide an informed investment. Furthermore, it can be complemented with other indicators and respective SM, such as quality of life indicators, energy, and emissions, among others.

$$C_i = n_r n_h A C \tag{3.13}$$

$$I = P n_h \tag{3.14}$$

$$ROI = \frac{I - C_i}{C_i} \tag{3.15}$$

$$f\begin{pmatrix} lat & lon & n_r & n_h & n_p & m_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ lat_i & lon_i & n_{ri} & n_{hi} & n_{pi} & m_{ii} \end{pmatrix} = max \begin{pmatrix} ROI \\ \vdots \\ ROI_i \end{pmatrix} \text{[ratio]} \tag{3.16}$$

$$lat \in [32.0...42.0], \quad lon \in [-124... - 114], \quad n_r \in [3...15],$$

$$n_h \in [10...500], \quad n_p \in [0...10], \quad i \in [1.0...14.0],$$

On the subject of other performance metrics and synergy of performance indicators, it is possible to demonstrate this concept by building on the Isenberg School of Management BID example demonstrated in Section 3.2 (See Synthethic Building Databases and Iterative Building Databases). In these examples, the construction cost $C$, energy use $E$, and Useful Daylight Illuminance (UDI) of a building (In this case the Isenberg School of Management) can be optimized according to its geometric (e.g., number of floors $n_f$, number of beams $n_b$, amplitude $\alpha$, inner radius $r_i$, building width $\Delta r$, and orientation $\theta$) and construction parameters (e.g., beam width $bw$, 3 beam materials $b$, 3 glazing materials $g$, 3 wall constructions $w$). These problems could focus on different variables depending on their problem type, such as a Design problem described in Equations 3.4 or a rehabilitation problem with other relevant variables such as the ones added in Equation 3.8.

### 3.5.2 Interfaces

Once the models are developed and the AOP defined, the last step is to deploy a user interface that enables the input of the SM features, and the control parameters of the AOP. In this research, one of three optional user interfaces will be selected based on its suitability for addressing specific building performance problems. One option involves deploying the model directly within its IDE. Another option entails developing a web app to input parameters and visualize problem results. The last option includes deploying the model in a script within a CAD or AD tool environment.

Deploying the model directly within its development IDE is fast and precise because it enables the visualization results on the fly, as the code is run. To illustrate this, we could define the building performance problem mentioned in (Equation 3.16) by building on the code developed throughout this Chapter (particularly Sections 3.3 and 3.4) with Code Listing 3.5.

In this code, a Bayesian Optimization is applied to search for the combination of variables that yield the maximum ROI, by leveraging on the developed SM to estimate housing price. After running the optimization process, the results can be visualized in any suitable chart or graph, such as a bar chart to evaluate feature importance in driving the house price up, and a scatter chart of latitude and longitude to visualize the best location provided by the optimization (Figure 3.15, left and right respectively).

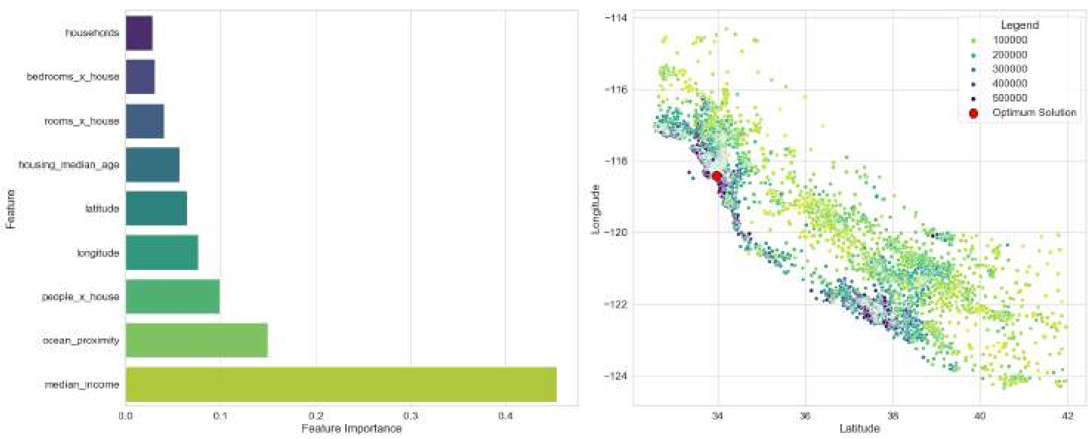



Figure 3.15: Feature importance for prediction of median house value and scatter plot of latitude and longitude with the obtained best solution location.

```python
#Create the features of average rooms, bedrooms, and people per house
X_opt = X.copy()
X_opt["rooms_x_house"] = X_opt["total_rooms"] / X_opt["households"]
X_opt["bedrooms_x_house"] = X_opt["total_bedrooms"] / X_opt["households"]
X_opt["people_x_house"] = X_opt["population"] / X_opt["households"]
X_opt.drop(["total_bedrooms", "total_rooms", "population"], axis=1, inplace=True)

#... Train/Test split and fit new model ...

# Define a model prediction function
def predict_house_price(construction):
    return model.predict(construction)
#Define Objective function
def objective(trial):
    #Define variables
    longitude = trial.suggest_float('longitude', -124, -114) # Example ranges
    latitude = trial.suggest_float('latitude', 32, 42)
    #Include more variables if needed
    #Build features DataFrame for prediction
    features = [latitude, longitude, ..., median_income]
    #Build DataFrame for model prediction
    construction = pd.DataFrame([features]).T
    construction.columns= X_opt.columns
    # Predict the house price using the surrogate model function
    predicted_price = predict_house_price(construction)
    # Calculate construction cost at 25 sqm per room, £5000 per sqm
    construction_cost = rooms_per_house * households * 25 * 5000
    # Calculate ROI
    roi = (predicted_price * households - construction_cost) / construction_cost
    return roi
# Running the optimization
study = optuna.create_study(direction='maximize')
 # Adjust n_trials as needed
study.optimize(objective, n_trials=500)
```

Code Listing 3.5: Code to perform a Real Estate Development optimization to maximize Return on Investment (ROI).

If the model is tweaked to a parametric analysis, it is also possible to develop a web app capable of predicting a ROI based on user investment inputs (Figure 3.16). In this case, the user can use sliders to change inputs such as the number of households, location, number of rooms per house, construction costs, and room area. In return, the bar chart displays changes according to the resulting construction cost and investment return.

Deploying a SM for the design optimization of the Isenberg School of Management, as described in Equation 3.4, requires a suitable environment for its results visualization, such as a Computer-Aided Design (CAD) tool environment. In this study, surrogate models can be integrated within the Rhino [79] and Grasshopper [80] environments, using a Python script embedded into the visual programming language environment. This setup facilitates on-the-fly optimization and result visualization within Rhino. This approach is illustrated in Figure 3.17, where Rhino is employed to display optimal solution results. Additionally, the environment supports the visualization of charts, which enables the presentation of a Pareto

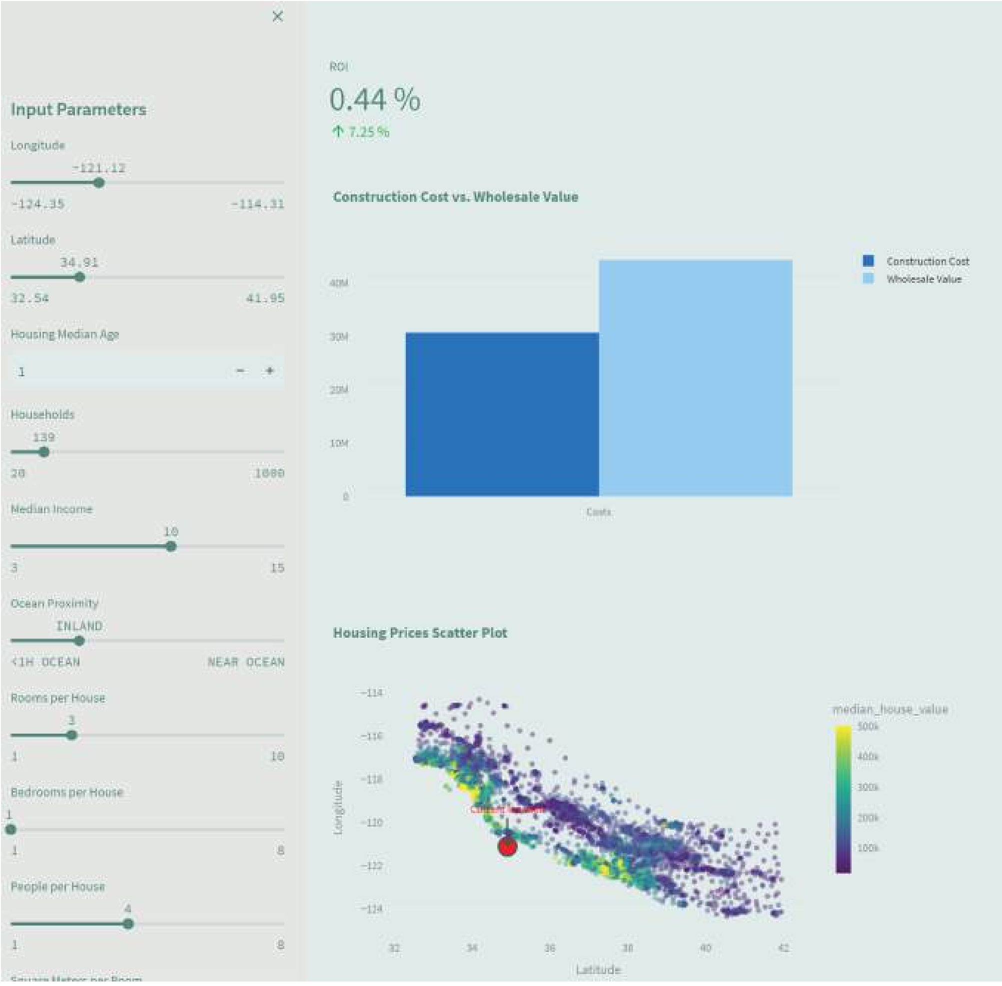
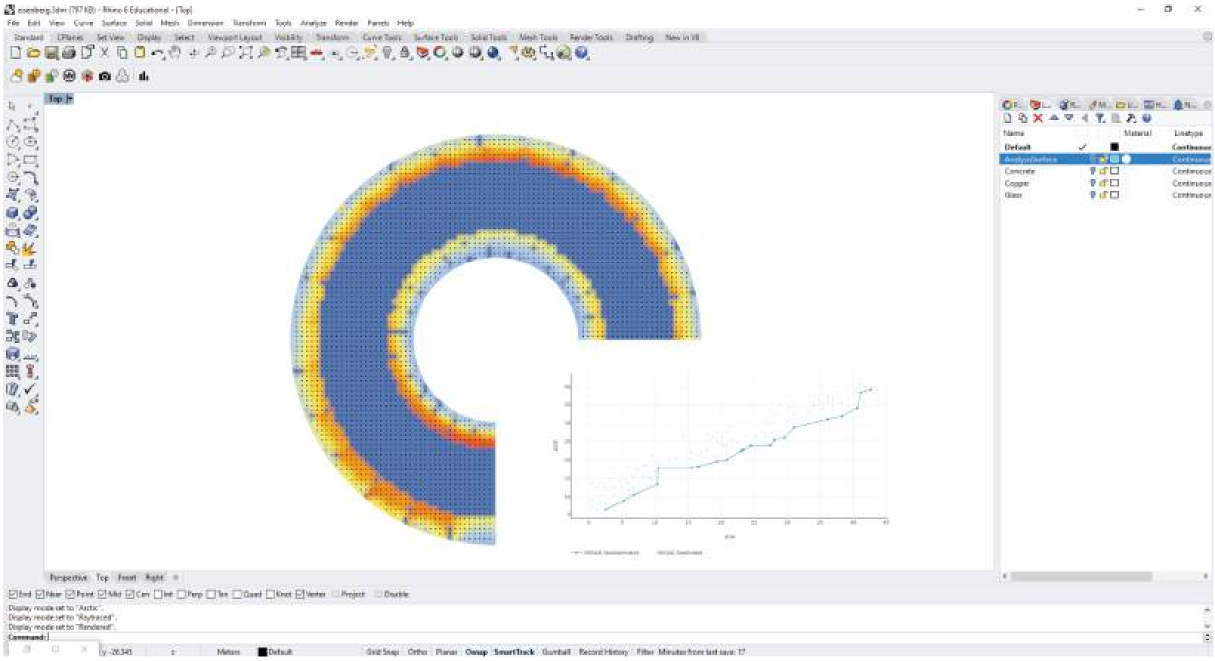


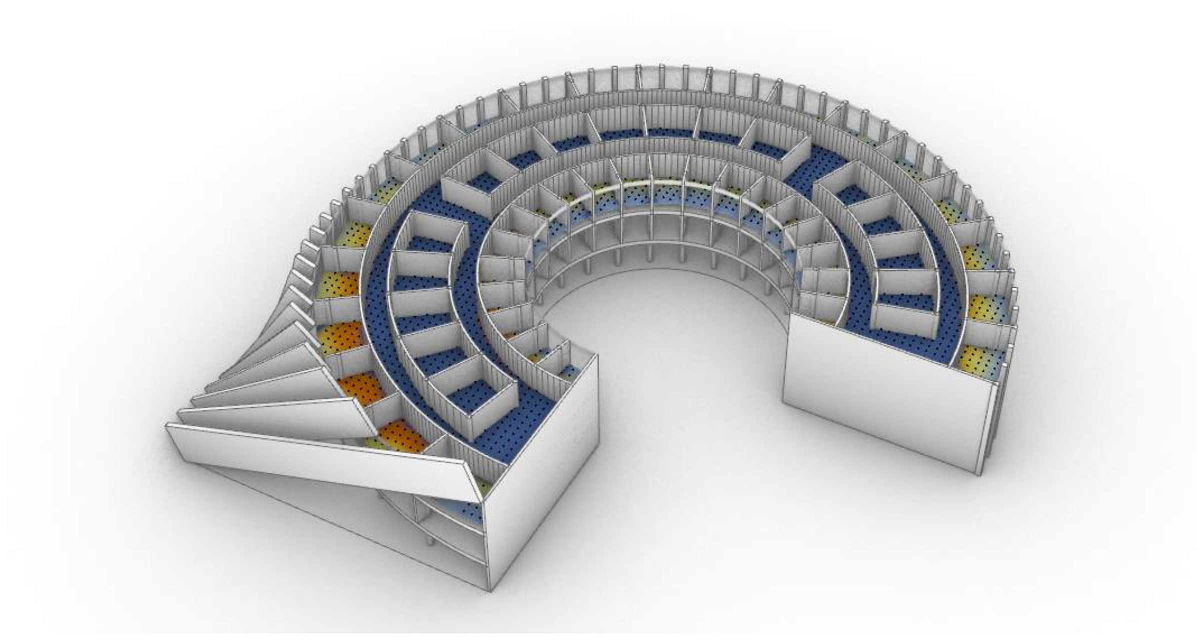

Figure 3.16: Input and results visualization with the developed sample web app.

chart to exhibit the optimization outcomes of a spatial Useful Daylight Illuminance (UDI) optimization.

The Model deployment stage concludes the Research process Part of this research. In the next Part, SM will be leveraged in case studies to solve building performance problems with AOP. This will be done considering the used Building Information Database (BID) and the type and scope of the optimization problem. Results are discussed regarding the improvements provided by the SM to the AOP. Ultimately, the results of this research are synthesized into a qualitative comparative analysis, evaluating the strengths and weaknesses of the respective BID types used in the studies.



(a) Pareto Front and Plan visualization.



(b) Pespective view.

Figure 3.17: Isenberg School of Management Useful Daylight Illuminance (UDI) optimization deployed in Rhino. (a) Visualization of Pareto Front in interface. (b) perspective view of a solution.

# Chapter 4

# Case Studies

## Preamble

The mext Chapter of this thesis delves into implementing the proposed framework. The three Building Information Database (BID) for Surrogate Models (SM) development are explored, one for each case study. Each case study comprises a building performance problem description, a BID documentation, SM development, and if necessary, tuning, and model deployment cases.

The first case study[1] [246] explores the optimization of building retrofit strategies and building design optimization. Here, a SM is developed using an Existing BID containing Energy Performance Certificates (EPC) from Portugal. This model predicts the certificate label and energy use when given specific building characteristics. This study illustrated the proposed framework for Existing BID. Here, the goal was not to speed up simulation processes but to simplify an otherwise complicated energy certification process to promote its use for Analysis and Optimization Processes (AOP). In this case, the model estimated accurate results with only 20 inputs compared to the 88 existing inputs in the EPC database. Moreover, Feature Engineering (FE) and selection techniques are applied to the BID to improve the model's accuracy. The model was implemented in two web apps for single and multiple building AOP, demonstrating its portability.

In the second case study[2] Araujo et al. [247], the focus shifts towards the optimization of both energy performance and cost in building design and rehabilitation problems. Using a Synthetic BID, a SM is developed that, when given building geometric and material features, predicts the energy simulation results of any building in Lisbon, Portugal. This study illustrated the proposed framework for Synthetic BID by generating thousands of building samples representative of Lisbon, Portugal, and simulating them to use the results as a target output for the model. The resulting model was able to speed up the case study simulation time by $\approx$ 100-fold, with only six input features that defined a building's geometry and construction materials. The model was implemented in two different AOP for the urban and building scale, showcasing its portability.

[1] DOI: 10.1016/j.enbenv.2023.07.002
[2] DOI: 10.52842/conf.caadria.2022.2.689

The third case study[3] [248] concerns the optimization of the Thermochromic Glazing (TCG) properties that yield the minimum energy use for lighting and Heating, Ventilation, and Air Conditioning (HVAC) of an office. An Iterative BID is used to develop two SM, one for HVAC and the other for lighting predictions. This study illustrates the proposed framework for Iterative BID by using the proposed optimization variables and objectives as a database. In this case, a simulation-based optimization is performed for 2000 iterations and is then used to develop the BID. The resulting model sped up the simulation from 30 to 0.02 seconds, using only four inputs corresponding to the glazing's properties.

Finally, the fourth case study[4] [249] explores the optimization of building construction materials. Using an iterative BID, a SM is developed to predict the energy use of a mixed-use six-building complex. Similarly to the third case, this study illustrates the proposed framework for Iterative BID but with an additional step in the Model Tuning stage, hyperparameter optimization. This model was able to speed the simulations from 20 to 0.02 seconds, with 24 total inputs, four inputs for each building corresponding to its walls, roofs, floors, and windows materials.

Applying these cases to the proposed framework has the goal of demonstrating its versatility and efficiency in applying SM to improve the integration of AOP in building performance. This allows the development of more general SM, with suitable data quality, in a more streamlined fashion, and with a systematic approach to model tuning. In addition, examples of deployment of the models are illustrated for future use.

[3]DOI: 10.1016/j.solener.2022.11.043
[4]DOI: 10.3390/en16104030

## 4.1 Optimization of building retrofit using a Surrogate Model developed with an Existing Building Database

One of the largest shares of energy consumption is associated with building stock (in Europe, it is responsible for 40% of total energy consumption), and since the lifetime of a building can exceed 100 years, it is important to improve energy regulation and develop instruments that promote the reduction of greenhouse gas emissions without neglecting the thermal comfort and quality of life of its occupants [263]. Pasichnyi et al. [264] and Pérez-Lombard et al. [265] point to regulation, audits, and certification as three basic policy instruments for enhancing energy efficiency in buildings. In this context, building Energy Performance Certificates (EPC) emerged to help achieve energy efficiency in buildings since the early 1990s [264], and their main goals are threefold: (1) to inform stakeholders of the building sector about building energy consumption and performance [266], (2) to decarbonize the building stock, and (3) to enhance investment in more efficient and sustainable solutions and systems, as documented on the Building Directive update from 2018 [267].

Building EPC has been developed as a key policy instrument to improve buildings' energy efficiency, decrease energy consumption, and provide more transparency on energy use in buildings. These are a central element of the Energy Performance Building Directive (EPBD). In 2002, the EU Parliament defined EPC as " a document recognized by a Member State or by a legal person designated by it, which indicates the energy performance of a building or building unit" [268, 267].

Regarding building energy certification, Pérez-Lombard et al. [265] distinguish three main advantages: (i) bench-marking, (ii) rating, and (iii) labeling. Nikolaou et al. [269] add (iv) building stock databases and methods for improving energy efficiency. Pasichnyi et al. [264] reviewed 79 papers focused on EPC applications from data analyses and stated that most EPC data have wider applications than initially intended by the EPC policy instrument. The detailed characterization of the building stock provided by the EPC scheme can be applied in the development of tools and applications enhancing energy efficiency measures.

Alongside, energy-efficient building retrofit is a key aspect of reducing carbon emissions [270], improving public health [271], and creating new jobs [272]. Despite their multiple benefits and efforts to promote building retrofits, the retrofit rates worldwide remain low, usually less than 1% per year [273].

The EPC benchmarking system for buildings and its related renovation policies have contributed significantly to improving energy use in buildings and achieving the European renovation wave goals [3, 274]. However, they still present constraints that prevent citizens and stakeholders from implementing building retrofits, such as a general lack of participation and EPC data errors.

The lack of awareness and engagement with the EPC system is discussed by Watts et al. [275], which surveyed how people perceive EPC, their importance, and if they implement the suggested retrofits. Results show that the EPC scheme is perceived more as mandatory bureaucracy, and most people do not follow retrofit indications. In addition to this, Hardy and Glew [276] showed that 36% to 62% of EPC in the United Kingdom contain errors and that these are mostly caused by EPC assessors. Consequently, the authors consider that new methodologies can be developed to prevent misleading results and reduce input complexity, and conclude that ML has great potential to help tackle these issues. Thus, the use of ML techniques to create models capable of predicting a building EPC or energy needs with easy-to-grasp inputs can help in (1) identifying, controlling, and correcting issued EPC with input errors [277]; and (2) engaging the community with the EPC scheme and its benefits [275].

A household's EPC is issued by qualified experts (architects or engineers) who audit the households and collect information regarding building geometry, constructive solutions, heating, ventilation, and Heating, Ventilation, and Air Conditioning (HVAC) systems, domestic water heating systems (DHW), and energy production systems, to determine the energy performance parameters and calculate energy performance indicators (including the energy label). Moreover, improvement measures and their impact on the energy label and energy consumption are suggested on the EPC by the experts [278].

The EPC considers numerical calculation methods based on the ISO standards (ISO 52000-1 [279], 52003-1 [280], 52010-1 [281], 52016-1 [282], and 52018-1 [283]) for heating, cooling, and domestic hot water needs and provide energy efficiency benchmark by considering reference values [284, 285]. The energy needs indicators calculated for the energy certification process are summarized in Tables 4.1 and 4.2. In most European countries, the labeling of the energy efficiency of a household assumes eight classes (from A +, the most energy efficient to F, the least) [267] and is calculated by defining the parameter $R = Ntc / Nt$ in specific range values (Table 4.2).

Table 4.1: EPCs' energy indicators.

| **Indicators** | **Description** |
|---|---|
| *Nic* | Annual nominal needs of useful energy for heating ($\mathrm{kW\,h\,m^{-2}}$) |
| *Ni* | Annual reference needs of useful energy for heating ($\mathrm{kW\,h\,m^{-2}}$) |
| *Nvc* | Annual nominal needs of useful energy for cooling ($\mathrm{kW\,h\,m^{-2}}$) |
| *Nv* | Annual reference needs of useful energy for cooling ($\mathrm{kW\,h\,m^{-2}}$) |
| *Nac* | Annual nominal needs of useful energy for the production of domestic hot water ($\mathrm{kW\,h\,m^{-2}}$) |
| *Na* | Annual reference needs of useful energy for the production of domestic hot water ($\mathrm{kW\,h\,m^{-2}}$) |
| *Ntc* | Annual total nominal primary energy needs ($\mathrm{kW\,h\,m^{-2}}$) |
| *Nt* | Annual reference primary energy needs ($\mathrm{kW\,h\,m^{-2}}$) |

Most studies regarding EPC surrogate models found in the literature focus on accurately predicting and validating existing EPC. Buratti et al. [286] used an Artifical Neural Networks (ANN) model developed with a database of 6500 EPC received by the Umbria Region (central Italy). The developed Surrogate Models (SM) allowed the authors to evaluate regional building energy consumption. Furthermore, the authors evaluated the accuracy of the model and identified EPC that required data correction. To engage the community with the EPC scheme, Khayatian et al. [229] simplify the EPC inputs parameters and features, by developing a regression model using ANN, with acceptable accuracy and capable of predicting building energy consumption and EPC results with only 12 features.

Such models can also be useful to improve household energy use and perform the best possible retrofits according to their specific needs and capabilities, both by reducing the input complexity and by

Table 4.2: EPCs' energy efficiency labels.

| Energy label | *R = Ntc/Nt* |
|---|---|
| A+ | $R \leq 0.25$ |
| A | $0.25 < R \leq 0.5$ |
| B | $0.5 < R \leq 0.75$ |
| B- | $0.75 < R \leq 1$ |
| C | $1 < R \leq 1.5$ |
| D | $1.5 < R \leq 2$ |
| E | $2 < R \leq 2.5$ |
| F | $2.5 < R \leq 3$ |

allowing the application of optimization techniques (2). In this sense, Fan & Xia [287] developed two optimization models to find the best building retrofit that yielded maximum energy savings and minimum investment payback time. This is done considering the South African building EPC scheme and its tax incentive initiative program.

In this study, an existing Building Information Database (BID) of Portugal's building EPC is used to develop a SM capable of predicting a building's EPC, $Ntc$, $Nvc$, and $Nic$. This model is integrated with a MOO problem to optimize a building's retrofit considering government-funded policies, to minimize the retrofit cost, and building's $Ntc$, and maximize the retrofit's Return on Investment (ROI). The model is then deployed in a web app for users to explore their best retrofit options. Results obtained by testing the optimization problem and the interface with an existing building are documented and discussed.

This Section's workflow can be illustrated by Figure 4.1. Initially, the AOP is contextualized and described (Subsection Problem Description). Then, the existing BID is described, as well as any Model Tuning processes related to the database such as data cleaning, filtering, and feature engineering. Thus, the model tuning stage is applied simultaneously with the BID stage (Subsection Building Information Database and Feature Engineering). The third Subsection describes the SM type and compares different algorithms for SM development (Subsection Model Type). Afterward, the most suitable model is selected for the deployment process in an interface, where users can test the AOP in any building located in Portugal. The AOP results are demonstrated for a case study of a Portuguese building (Subsection Model Deployment). Finally, the obtained results and the model performance are discussed regarding their potential application and benefits (Subsection Discussion).

### 4.1.1 Problem Description

As stated in the Renovation Wave program [3], it is of critical importance to perform building retrofits as a way to reduce building energy consumption. Despite the multiple policies implemented to increase the renovation wave, there are still some barriers preventing its success, such as insufficient information on the current energy profile of buildings and policies [275], lack of trust in the actual estimation of energy savings [273], difficult decision-making processes [288] and financial obstacles [270]. In this sense, it is crucial to provide citizens and stakeholders with valuable, accurate, and comprehensible information on their energy performance at home or in a building and the benefits of specific retrofit measures.

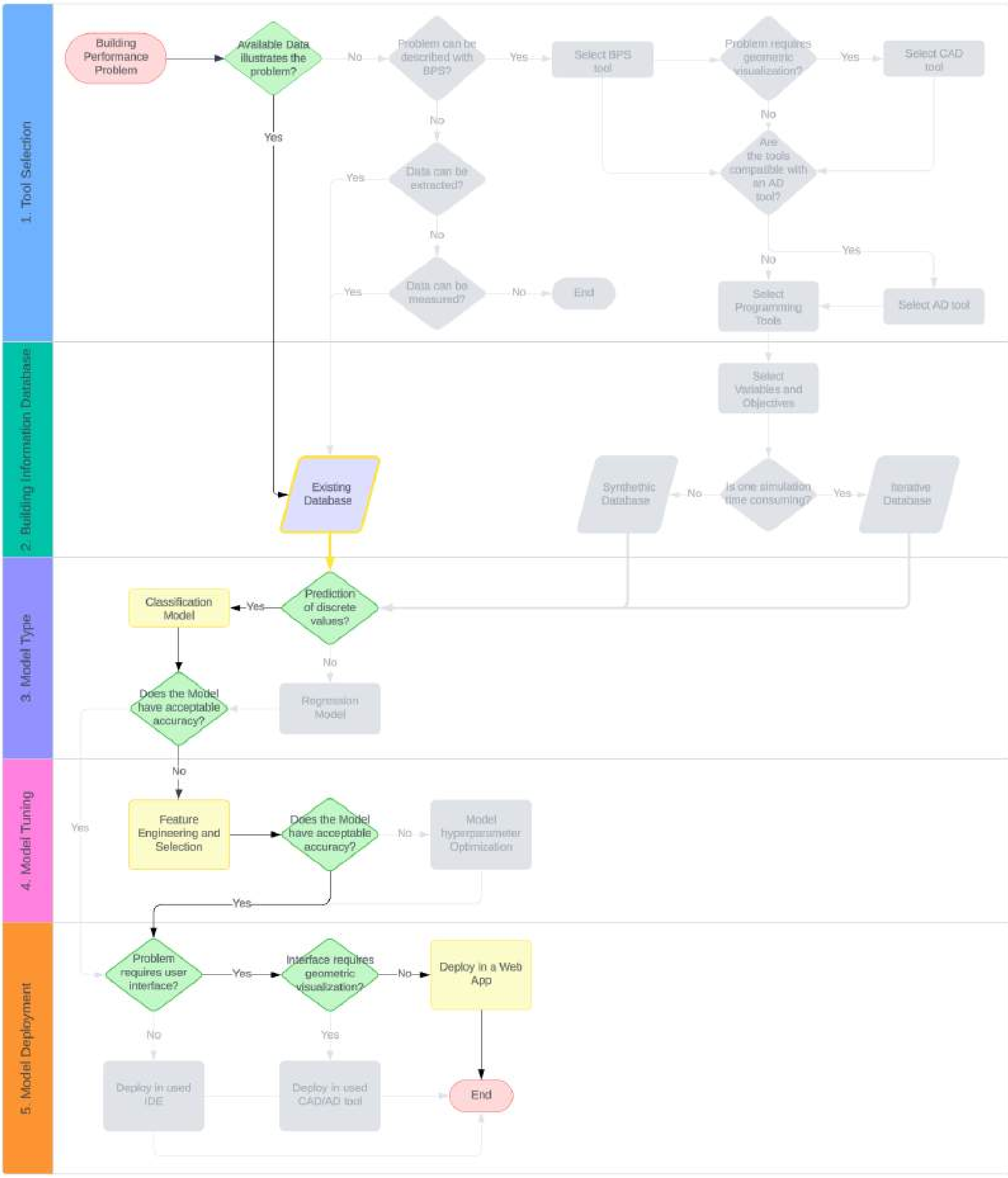



Figure 4.1: Workflow diagram for Section 4.1

The goal of this study is to strengthen the commitment to the renovation wave program [3] by developing an interface that is capable of quickly predicting and optimizing EPC indicators with easy-to-grasp inputs. Machine Learning techniques are applied to an EPC database of residential buildings in Portugal to develop regression models capable of predicting a building or household's annual energy needs, and EPC labels. Afterward, these models are integrated with a Multi-Objective Optimization (MOO) process to provide the user with a range of optimum retrofit solutions while estimating their impact on energy consumption and return on investment.

For the MOO problem, a Non-dominated Sorting Genetic Algorithm II (NSGAII) [141] was used, since it demonstrated its efficiency for similar class problems in previous research [11, 125]. The algorithm runs with a random generator object, a tournament selection method, and a population size of 25, for a maximum of 250 iterations or convergence. The MOO problem decision variables correspond to the EPC's possible retrofit solutions.

The possible retrofits for one household are organized into different intervention areas: walls, roofs, windows, heating equipment, and Domestic Hot Water (DHW) equipment. A MOO process that returns the optimum retrofit options of an EPC is programmed, considering for each variable the option of having no retrofit and only those retrofits that represent an improvement against the original features. Finally, the total retrofit cost is calculated according to standard construction solutions, equipment costs, government funds, and typology (Tables 4.3 and 4.4, adapted from POSEUR [289]).

Table 4.3: Retrofit variables costs.

| **Retrofit Variable** | **Solution** | **Cost [€/m2]** | **Cost [€/unit]** | **Funding Ratio** | **Funding limit** |
|---|---|---|---|---|---|
| Walls | Insulation | 41 | - | 65 | 4500 |
| Floors | Insulation | 13.5 | - | 65 | 4500 |
| Roof | Insulation (EPS) | 13.5 | - | 65 | 4500 |
| | Insulation (XPS) | 25 | - | 65 | 4500 |
| Glazing | PVC | 260 | - | 85 | 1500 |
| | Aluminium | 380 | - | 85 | 1500 |
| DHW | Gas boiler | - | 450 | 0 | 0 |
| | Water heater | - | 175 | 0 | 0 |
| | Boiler | - | 1750 | 0 | 0 |
| | Air-Water heat pump | - | 3750 | 85 | 2500 |
| Climatization | Gas boiler | - | 450 | 0 | 0 |
| | Water heater | - | 175 | 0 | 0 |
| | Boiler | - | 2250 | 85 | 2500 |
| | Multi-split | - | 366 | 85 | 2500 |
| Energy source | 3 solar panels | - | 6100 | 85 | 2500 |
| | 6 solar panels | - | 9400 | 85 | 2500 |

The optimization's objectives are the minimization of an EPC's *Ntc* ($o_1$ - Equation 4.1) and retrofit cost ($o_2$ - Equation 4.2), together with the maximization of the retrofit's Return on Investment over 3 years (ROI) ($o_3$ - Equation 4.3). $o_1$ is predicted by the surrogate model, $o_2$ is obtained from user inputs and retrofit costs from Table 4.3, and $o_3$ $ROI$ is obtained from Equation 4.4. The *Tax* variable represents the tax benefits obtained from the retrofit and is calculated by predicting the new EPC label and assigning the respective tax deduction. Energy Savings ($E_s$) is computed by Equation 4.5, where $p$ represents the average kWh price for electricity in Portugal (source: Eurostat [290]), $A$ the floor area of the EPC, and $t$ the number of years considered for the ROI calculation.

$$o_1 = min(Ntc_{new}) \text{ [kW h m}^{-2}\text{]} \tag{4.1}$$

Table 4.4: Multi-split equipment costs and government funds for each house typology.

| Building typology | Costs [€] | Government funds ratio [%] | Government funds limit [€] |
|---|---|---|---|
| Studio | 366 | | |
| 1-bedroom | 731 | | |
| 2-bedroom | 1096 | | |
| 3-bedroom | 1462 | 85 | 2500 |
| 4-bedroom | 1828 | | |
| 5-bedroom | 2193 | | |
| 6-bedroom | 2558 | | |
| 7-bedroom | 2924 | | |

$$o_2 = min(Cost) \text{ [€]} \tag{4.2}$$

$$o_3 = min(ROI) \ [ratio] \tag{4.3}$$

$$ROI = \frac{E_s + Tax - Costs}{Costs} \ [ratio]; \tag{4.4}$$

$$E_s = pA(Ntc_{new} - Ntc_{old})t \text{ [€]} \tag{4.5}$$

Another possible optimization problem is at an urban scale, where it is possible to analyze macro retrofit trends according to a wider sample of buildings such as neighborhoods, municipalities, and cities. As such, the developed SM is also implemented to an optimization where its objectives are to minimize the mean energy use of the building sample ($f_1$ in Eq. 4.6), the standard deviation of the energy use of the buildings sample ($f_2$ in Eq. 4.7), and the total retrofit cost of the sample ($f_3$ in Eq. 4.8), where $x$ is a binary variable representing if a specific retrofit is or is not applied, $n$ is the number of buildings in the sample and $r$ the available retrofits in Table 4.3. The considered retrofits for this problem are wall, roof, and floor insulation, window replacement, efficient HVAC units, Air-to-Water heat pumps, and solar panels for energy generation, or Domestic Hot Waters (DHW). This represents a total of eight possible retrofits per building and a total of $8n$ possible variables.

$$f_1(x_1, \ldots, x_{r\times n}) = \frac{\sum_{i=1}^{n} Ntc_i}{n} \ [\mathrm{kW\,h\,m^{-2}}] \tag{4.6}$$

$$f_2(x_1, \ldots, x_{r\times n}) = \sqrt{\frac{\sum_{i=1}^{n} (Ntc_i - f_1)^2}{n}} \ [\mathrm{kW\,h\,m^{-2}}] \tag{4.7}$$

$$f_3(x_1, \ldots, x_{r\times n}) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{r} C_{ji}}{n} \text{ [€]} \tag{4.8}$$

This approach can be adapted to other countries' databases and yield information regarding energy performance and how to manage retrofit and rehabilitation investments at a larger scale. Moreover, researchers are provided with alternative approaches for building retrofit and optimization that require less computational power and have fewer inputs than building performance simulation and optimization using complex models. Finally, these techniques can be applied to the development of tools or applications to improve the retrofit process by providing a fast but accurate analysis of retrofit alternatives.

### 4.1.2 Building Information Database and Feature Engineering

For the BID development, the first step consists of filtering unrealistic values and splitting the resulting database into a training and test set for all variables and outputs. Then, feature selection techniques are applied to retrieve the most explainable features of the model [162]. The Portuguese EPC database is used, which contains over 800000 certificates, and 88 features of household details, opaque and glazed envelopes, and systems.

From the whole database, all entries that were not households and not issued under the 2006 Decree-Law [291] were removed, since the method of calculation of the EPC changed significantly due to the EPBD [268]. Additionally, all the EPC without DHW equipment, all features with unrealistic values, and samples with over 70% of null values for all features are removed. Afterward, the most important features to calculate the EPC are retrieved. Finally, from the 800000 EPC and a total of 88 features for the whole country, 61 features and $\approx$ 740000 entries were removed, ending up with a total of 25 features and $\approx$ 60000 EPC. Both continuous and discrete features are normalized into a [-1, 1] interval. The number of features is selected according to their k-best scores [292] and for each target output with k-best = 10, 15, 20, and 25 features (Table 4.5).

The ML models prediction targets are the annual heating energy needs (*Nic*), annual cooling energy needs (*Nvc*), annual primary energy needs (*Ntc*) (Table 4.1), and *R* ratio (Table 4.2). The database with 60000 EPC, is split by 33% (1/3) to build a model validation test set and 67% (2/3) to train the models. The target prediction values are illustrated in Table 4.6 by documenting their maximum, minimum, mean, and standard deviation.

Table 4.5: EPCs' database feature list for k-best = 20.

| **General details** | **Construction elements** | **Equipment** | **Glazing** |
|---|---|---|---|
| Property | Wall type | DHW source | Window area [$m^2$] |
| Year | Wall area [$m^2$] | Heating source | Window type |
| Area [$m^2$] | Roof type | DHW type | |
| Height [m] | Roof area [$m^2$] | Heating type | |
| Typology | Floor type | Nº DHW equip | |
| Nº floors | | Nº heating equip | |
| District | | | |

Table 4.6: Test set target values distribution and indicators.

| | **Nic** [$\mathrm{kW\,h\,m^{-2}}$] | **Nvc** [$\mathrm{kW\,h\,m^{-2}}$] | **Ntc** [$\mathrm{kW\,h\,m^{-2}}$] | **R (ratio)** |
|---|---|---|---|---|
| **count** | 5914 | 5914 | 5914 | 5914 |
| **mean** | 10.46 | 1.90 | 96.10 | 0.72 |
| **std** | 7.03 | 1.70 | 86.14 | 0.51 |
| **min** | 0.00 | 0.00 | 0.00 | 0.00 |
| **max** | 48.79 | 7.00 | 660.63 | 4.20 |

### 4.1.3 Model Type

For the models' training, an Extra Trees (ET) ensemble algorithm [293], a Multi-Layer Perceptron (MLP) with 3 nodes and 25, 50, and 25 neurons [294, 295], and a Gradient Boosting (GB) for the regression problems [296] were used with default hyperparameters. These models have been extensively used in the literature for regression problems [297], particularly in building energy-related problems [298]. Root Mean Squared Error (RMSE) and $\mathrm{R}^2$ scores are recorded as performance metrics for the models' regression outputs (*Nic*, *Nvc*, *Ntc*, *R*). Performance results are compared for the three regression algorithms, and the best-performing algorithm is selected accordingly. The selected model is then subject to a k-fold cross-validation process [299] with k=6 folds. This process splits the testing subset into multiple (k) folds and yields less biased performance results.

The results obtained show that all algorithms perform similarly (Table 4.7). However, as the ET algorithm with 20 features tends to perform slightly better, it was selected for the final interface and MOO process. Specifically for the EPC prediction, the model developed with the ET algorithm has errors from -1 to +1 EPC level (Table 4.2). From the analysis of Table 4.7, it is possible to see that the $R$ model and the $Ntc$ model are the most accurate with an $\mathrm{R}^2$ of 0.84 and 0.79, respectively. While $Nic$ and $Nvc$ show $\mathrm{R}^2$ values of 0.67 and 0.41.

Table 4.7: Model training and performance indicators results.

| | | **R [ratio]** | | **Ntc** [$\mathrm{kW\,h\,m^{-2}}$] | | **Nic** [$\mathrm{kW\,h\,m^{-2}}$] | | **Nvc** [$\mathrm{kW\,h\,m^{-2}}$] | |
|---|---|---|---|---|---|---|---|---|---|
| **k-best features** | **Model** | $R^2$ | $RMSE$ | $R^2$ | $RMSE$ | $R^2$ | $RMSE$ | $R^2$ | $RMSE$ |
| 10 | ET | 0.74 | 0.26 | 0.58 | 55.70 | 0.61 | 19.11 | 0.31 | 5.85 |
| | MLP | 0.71 | 0.27 | 0.51 | 60.22 | 0.55 | 20.46 | 0.16 | 6.43 |
| | GB | 0.69 | 0.28 | 0.50 | 60.98 | 0.53 | 20.89 | 0.14 | 6.51 |
| 15 | ET | 0.82 | 0.22 | 0.72 | 45.30 | 0.65 | 18.07 | 0.36 | 5.63 |
| | MLP | 0.77 | 0.24 | 0.66 | 49.91 | 0.57 | 20.05 | 0.19 | 6.33 |
| | GB | 0.76 | 0.25 | 0.65 | 51.07 | 0.57 | 20.01 | 0.17 | 6.39 |
| 20 | ET | 0.84 | 0.21 | 0.79 | 39.77 | 0.67 | 17.70 | 0.41 | 5.41 |
| | MLP | 0.78 | 0.24 | 0.72 | 45.29 | 0.58 | 19.77 | 0.23 | 6.15 |
| | GB | 0.78 | 0.24 | 0.73 | 44.87 | 0.57 | 19.96 | 0.24 | 6.14 |
| 25 | ET | 0.85 | 0.20 | 0.80 | 38.83 | 0.73 | 15.96 | 0.61 | 4.40 |
| | MLP | 0.80 | 0.23 | 0.74 | 44.33 | 0.67 | 17.66 | 0.47 | 5.11 |
| | GB | 0.80 | 0.23 | 0.75 | 43.19 | 0.68 | 17.22 | 0.33 | 5.75 |

The selected algorithm ET with 20 features was subject to a k-fold cross-validation process in which the full database was split into six training and test sets. Results from the selected algorithm are described in Table 4.8. $R^2$ values for Fold 1 and 2 are relatively lower than the remaining folds. The mean $R^2$ shows a slight decrease to 0.80 while the RMSE shows an increase to 0.24 when compared to the performance indicators of the selected ET algorithm with k-best = 20 features (Table 4.7). Nonetheless, the validation results still show acceptable accuracy, particularly for the $R$, $Ntc$, and $Nic$ models that show $R^2$ scores $\geq$ 0.70.

Table 4.8: Cross validation results for the selected model ET with k-best = 20.

| | **R [ratio]** | | **Ntc** [$\mathrm{kW\,h\,m^{-2}}$] | | **Nic** [$\mathrm{kW\,h\,m^{-2}}$] | | **Nvc** [$\mathrm{kW\,h\,m^{-2}}$] | |
|---|---|---|---|---|---|---|---|---|
| | $R^2$ | $RMSE$ | $R^2$ | $RMSE$ | $R^2$ | $RMSE$ | $R^2$ | $RMSE$ |
| Fold 1 | 0.61 | 0.24 | 0.63 | 49.46 | 0.55 | 21.27 | 0.34 | 7.26 |
| Fold 2 | 0.79 | 0.28 | 0.76 | 50.83 | 0.67 | 21.43 | 0.50 | 6.99 |
| Fold 3 | 0.86 | 0.25 | 0.79 | 54.54 | 0.70 | 22.27 | 0.46 | 6.00 |
| Fold 4 | 0.87 | 0.24 | 0.84 | 51.21 | 0.78 | 20.87 | 0.51 | 5.86 |
| Fold 5 | 0.85 | 0.22 | 0.81 | 45.52 | 0.75 | 18.32 | 0.43 | 5.33 |
| Fold 6 | 0.83 | 0.22 | 0.80 | 44.24 | 0.75 | 19.20 | 0.45 | 5.10 |
| **Mean** | 0.80 | 0.24 | 0.77 | 49.37 | 0.70 | 20.48 | 0.45 | 6.07 |

Figure 4.2 illustrates the error (in %) distribution plot of the selected algorithm for the prediction of the test samples (x-axis) and distribution of samples in the test subset for each target feature (y-axis). For the $R$ prediction model, it is visible that most test values are distributed between 0 and 1, and respectively, the error values of the test set predictions vary between -25 and 25%. The $Ntc$, $Nic$, and $Nvc$ prediction models show most test values distributed between 0 and ≈200, ≈100, and ≈20 $\mathrm{kW\,h\,m^{-2}}$ respectively, while most error values vary between -50 to 50%. Additionally, a trend of higher error percentages for smaller target values is visible, which shows that most predictions that have an error of up to double the original value occur more frequently in smaller values. This model's behavior is seen in different scales for all models with the $R$ prediction model being the least noticeable, and the $Nic$ model the most.

### 4.1.4 Model Deployment

Initially, it is required that users fill in general information about the building/house regarding geometry, constructive solutions, heating, cooling, and DHW equipment (Figure 4.1). The features belonging to Construction elements (Table 4.5) can be estimated based on the most frequent construction solution for the respective construction period of the household. With this data input, the model predicts the total, heating, and cooling annual energy needs, and EPC label. Users may eventually compare the predicted results with their billing information and original EPC, as well as test multiple retrofits manually.

After this initial exploration, an option to optimize the described EPC is available by setting a maximum budget and filling in their household's tax amount. The MOO algorithm starts to iterate over the possible retrofit scenarios according to the EPC data previously filled in by the user. Finally, results are illustrated in interactive bar charts for each objective, and a table with the optimum retrofit solutions for

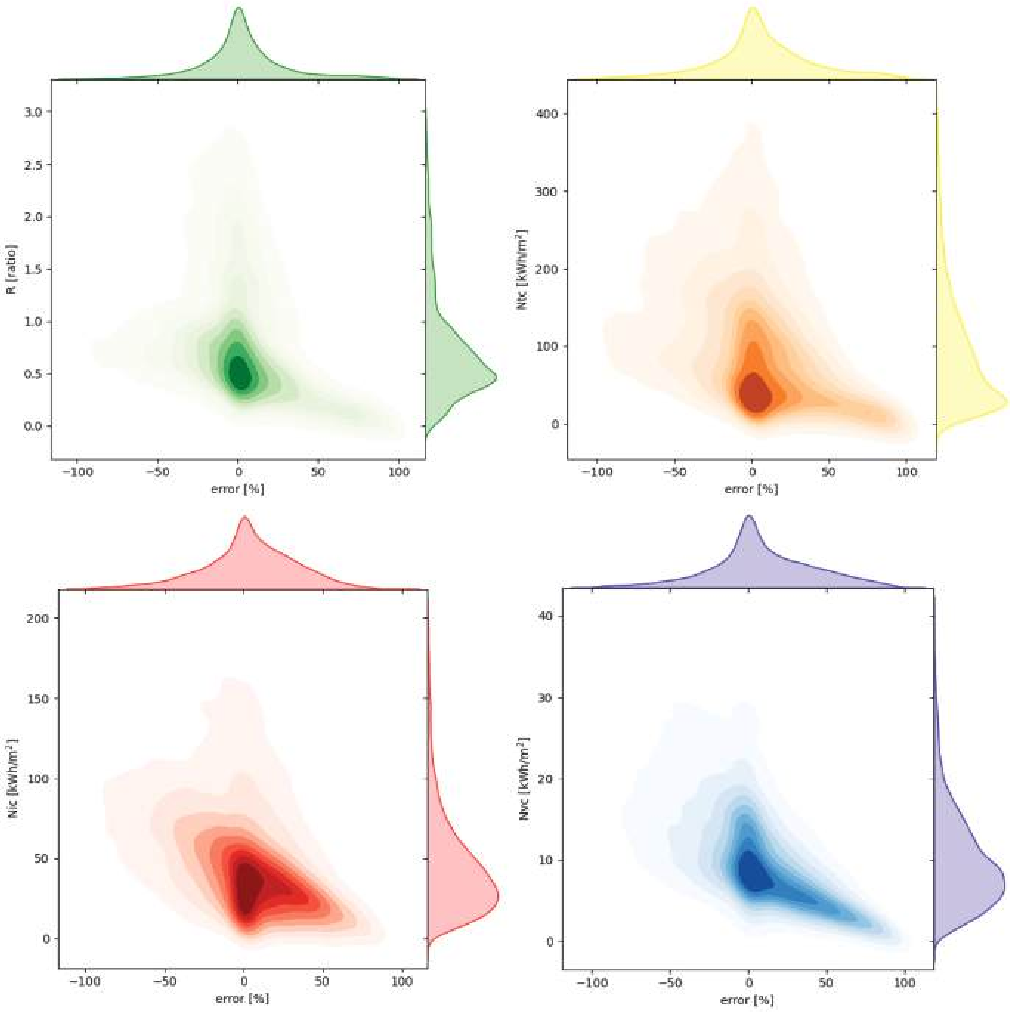



Figure 4.2: Test set values (y-axis) and error (x-axis) distribution joint plot for each regression Extra Trees model.

the maximum budget is presented. This interface was tested in a case study for a 3-bedroom household in Évora, Portugal (Figure 4.3a) with an EPC level of "F" (Figure 4.3b). This EPC was selected based on privacy concerns and because 3-bedroom households are the most common entry in the full database ($\approx$ 45%). Most features are obtained from the house's technical drawings, while the unknown features are deduced according to its construction period. Feature values for the case study are documented in Table 4.9.

Table 4.9: Feature values (Table 4.5) for the proposed case study.

| General details | | Construction elements | | Equipment | | Glazing | |
|---|---|---|---|---|---|---|---|
| **Property** | Apartment | **Wall type** | By Year | **DHW source** | Electricity | **Window area** $[m^2]$ | 9 |
| **Year** | 1996-2000 | **Wall area** $[m^2]$ | 56 | **Heating source** | Electricity | **Window type** | By Year |
| **Area** $[m^2]$ | 100 | **Roof type** | By Year | **DHW type** | Heater | | |
| **Height** $[m]$ | 3 | **Roof area** $[m^2]$ | 116 | **Heating type** | Split | | |
| **Typology** | 3 bed-room | **Floor type** | By year | **Nº DHW equip** | 1 | | |
| **Nº floors** | 2 | | | **Nº heating equip** | 2 | | |
| **District** | Évora | | | | | | |

(a) Case study 3D model. (b) Case study Energy Performance Certificate.

Figure 4.3: Case study details and geometry

### Optimization Results

The optimization process was tested with the sample house described in Figure 4.3. The respective features are seen in table 4.9 and are filled in according to the household's EPC. The results of the optimization process are described in a 3-dimensional Pareto optimality scatter chart represented in Figure 4.4. The $R$ ratio was added as a color scale of the optimal solutions to illustrate the relationships between objectives and their respective $R$. This chart illustrates the optimum solutions found by the algorithm, which are the non-dominated solutions (refer to Section 2.2, Subsection Building Performance Optimization [147]). This chart is only presented here and is not included in the interface because this type of visual communication may be considered difficult to interpret by non-expert citizens.

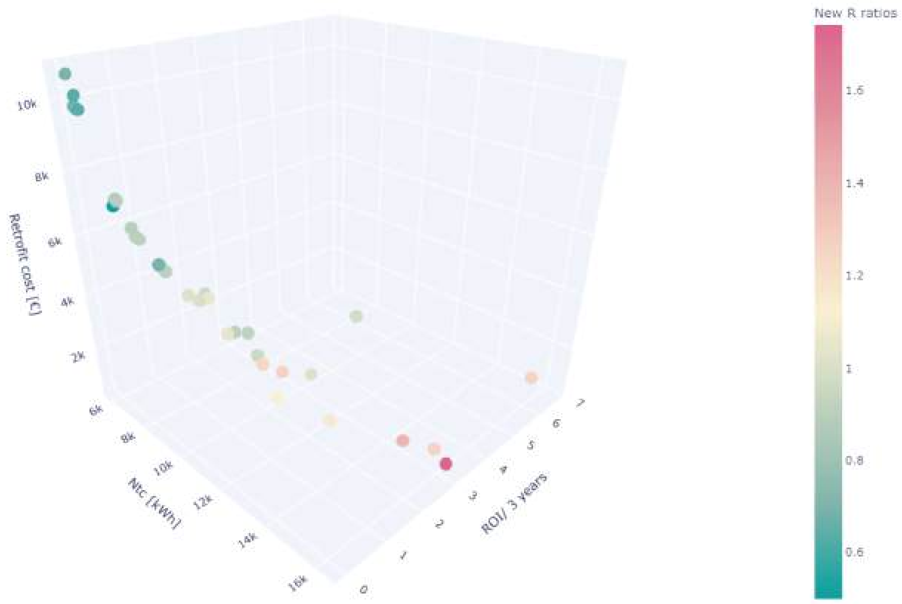



Figure 4.4: Pareto optimality chart for the single building optimization with a color scale of resulting retrofit R ratio.

Optimization results show a wide range of optimum solutions for all objectives. The objectives can vary between ≈11000 € and ≈450 € for the total retrofit costs, ≈6000 $kWh$ and ≈17000 $kWh$ for the $Ntc$, and ≈0.5 and 7 for the $ROI$ in 3 years(i.e., return in 3 years of 0.5 to 7 times the investment made). This demonstrates that the best of each objective does not necessarily represent the best solution. In the case of this EPC, it is visible in Figure 4.4 that the most expensive optimum solutions represent minimum $Ntc$ and $R$ values. However, solutions that have the highest $ROI$ in 3 years correspond to the cheapest solutions with smaller $R$ that have a more immediate impact on energy savings and tax benefits, when compared with the original features of the EPC. Finally, it is up to the user to choose the adequate $ROI$ rate, how much should be spent on the retrofit, and if it is possible to plan long-term retrofit strategies based on returns and savings obtained from cheaper retrofits.

The urban optimization problem is applied to a random sample of 20 buildings contained in the database that have an EPC of D or lower and are located in Lisbon. The results are illustrated by Figures 4.5 and 4.6. In the first figure, it is possible to see that results vary between $\approx$ 96 and $\approx$ 178 [$\mathrm{kW\,h\,m^{-2}}$] for $f_1$ (Eq. 4.6), and between $\approx$ 16 and $\approx$ 97 [$\mathrm{kW\,h\,m^{-2}}$] for $f_2$ (Eq. 4.7). Furthermore, by looking at the retrofit cost per building, it is visible that it varies between $\approx$ 1000 and $\approx$ 10000 €. With this data, it is visible a negative correlation between smaller $f_1$ and $f_2$, and the retrofit costs $f_3$. These results also show that with $\approx$ 1500 € per building, it is possible to obtain a 50% reduction in the mean energy needs of the sample.

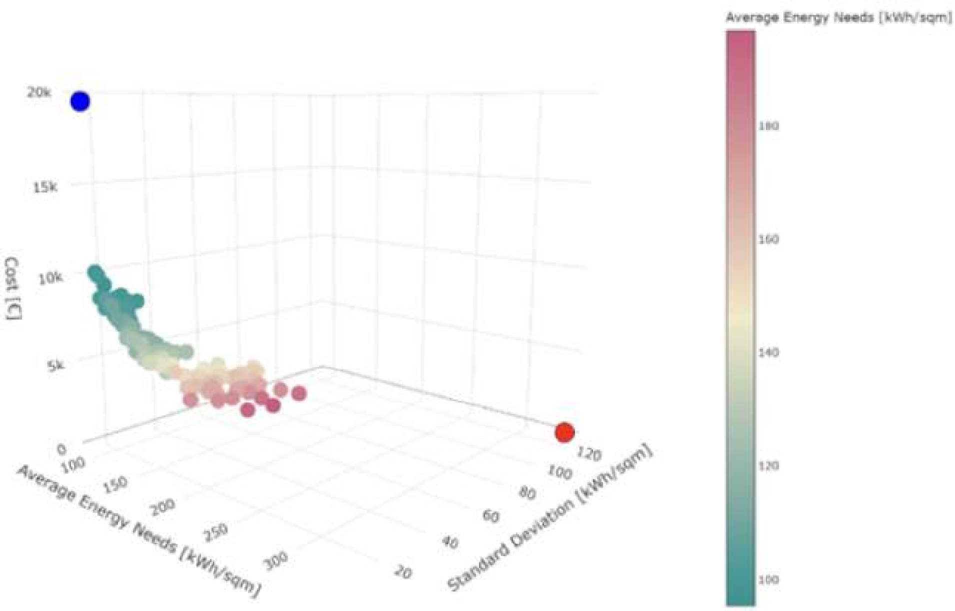



Figure 4.5: Pareto optimality chart for the urban optimization with a color scale of resulting mean average energy needs. Blue and Red dots represent the fully retrofitted and original solutions, respectively.

Since this specific urban optimization problem searches binary variables (i.e., eight different types of retrofit applied or not), it is possible to analyze the obtained results by retrofit frequency in the obtained Pareto optimal solutions. Figure 4.6 shows these results with a color scale representing the frequency of appearance of the retrofit in the building sample. It is visible in this Figure that the best solutions with minimum energy use present between $\approx$ 65% to $\approx$ 100% of the sample retrofitted with wall insulation. While flooring and roof insulation present between $\approx$ 50 to $\approx$ 65% of the sample. Retrofits like window replacement and Air-to-Water pumps seem less frequent solutions for the sample. Finally, efficient HVAC units are the most frequent retrofits present in the sample's optimum solutions. While more homogeneous results indicate that those retrofits are widely present or absent in the optimum solutions of the sample, more heterogeneous results indicate retrofits with a higher impact on the building sample.

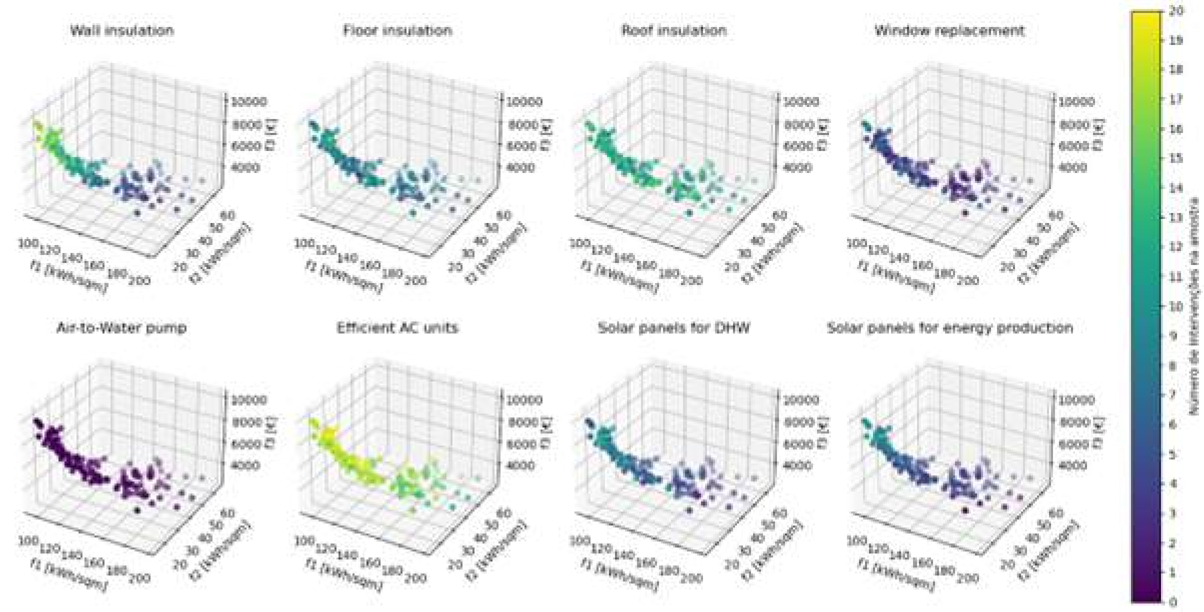



Figure 4.6: Pareto optimality chart for the urban optimization with a color scale for each retrofit frequency.

### Interface

The interface[5] initiates after loading the surrogate models. Drop-down and input boxes appear and allow the user to fill in basic general data regarding their home or building. The inputs for the proposed case study are filled in according to Table 4.9, with 16 mandatory features, and 4 optional, that can be extrapolated based on the construction period in the case of the information not being readily available (Wall, Roof, Floor, and Window types). However, if the users can obtain the required details there is a possibility to fill in and change the default extrapolated values and increase the predictions' accuracy.

After filling in the required data the user can simulate their energy needs $Nic$, $Nvc$, and $Ntc$ values, and EPC (Figure 4.7). The energy needs are presented in total $kWh$ rather than $\mathrm{kW\,h\,m^{-2}}$ for a better understanding and comparison with energy bills. Afterward, it is possible to manually change the inputs and perform analyses of retrofit impacts or keep scrolling to read the information regarding eventual government funds to support building retrofits, and tax benefits obtained from the EPC improvement.

The predicted results are compared with the original EPC for this house in Table 4.10. As seen, the predicted annual energy needs for heating ($Nic$), cooling ($Nvc$), and total ($Ntc$) have an error of $\approx$ 14 $\mathrm{kW\,h\,m^{-2}}$, $\approx$ 3 $\mathrm{kW\,h\,m^{-2}}$, and $\approx$ 23 $\mathrm{kW\,h\,m^{-2}}$ respectively. Additionally, the model fails to predict the original EPC by one level. These errors are within the documented errors for the selected surrogate model in Table 4.7.

[5]https://goncalo-araujo-epc-app-app-epc-txb2ta.streamlit.app/

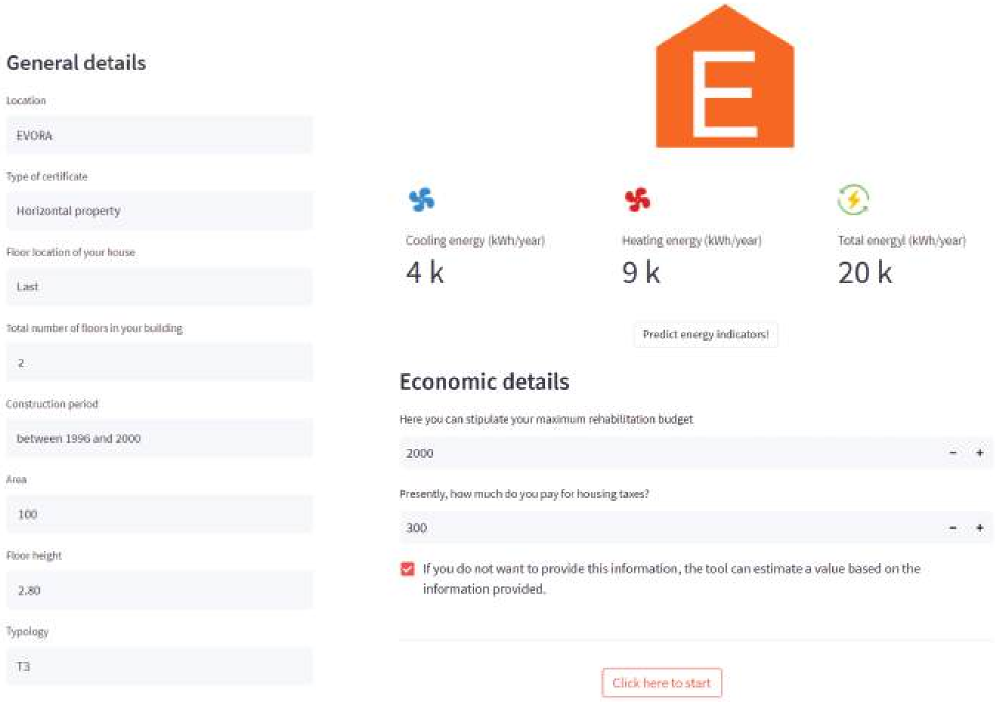



Figure 4.7: Application interface for single building Analysis and Optimization Processes (AOP).

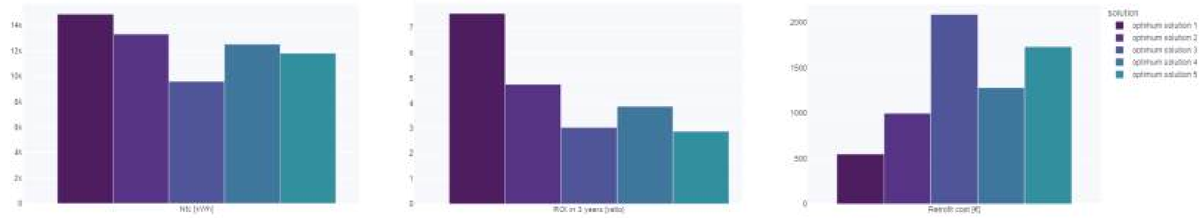
Table 4.10: Case study predicted and original results.

| | **Nic** [$\mathrm{kW\,h\,m^{-2}}$] | **Nvc** [$\mathrm{kW\,h\,m^{-2}}$] | **Ntc** [$\mathrm{kW\,h\,m^{-2}}$] | **R** **[ratio]** | **EPC class** |
|---|---|---|---|---|---|
| **Original** | 114.31 | 37.10 | 202.20 | 2.56 | F |
| **Predicted** | 102.92 | 40.99 | 225.05 | 2.10 | E |
| **error[%]** | -9.96 | 10.49 | 11.3 | -17.97 | |

After this initial analysis, the user is prompted to click a button that performs an optimization process to find optimum retrofit solutions with a specific budget to be defined. In this case study, a maximum budget of 2000€ was selected, different from the optimization process performed above, which had no limit budget. The interface runs the optimization algorithm and provides the user with a table with the obtained optimum solutions that fit the specified budget. The user can interactively explore three bar charts that illustrate these solutions' objectives' values (Figure 4.8). The bar charts were chosen as a more user-friendly option than the Pareto optimality chart to understand the results.

Results for this case study optimization show 5 non-dominated solutions that fit the specified budget (Table 4.11). Particularly, the algorithm suggested a retrofit of the roof, which currently has no insulation, followed by insulation of the exterior walls, which makes the retrofit closer to the budget limit. Cheaper variations of this solution are provided, such as changing from an electric water heater to a gas heater. The presented solutions can yield decreases in the $Ntc$ by up to $\approx$ 60%, and $ROI$ values ranging from $\approx$ 2 to $\approx$ 7.

The urban optimization problem was also implemented in a web app illustrated in Figure 4.9. The user can input a list of buildings with their input features required by the model and run an initial analysis.



(a) Total Energy needs (Ntc) (b) Return on Investment after 3 years. (c) Retrofit Cost

Figure 4.8: Interactive charts for each objective

Table 4.11: Example of the table with optimization results that can be downloaded by the user according to the specified budget limit.

| Retrofit | Walls | Roof | Glazing | DHW | DHW energy source | HVAC | HVAC energy source | Ntc [kWh] | ROI [ratio] | Retrofit cost [€] | New EPC label |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | - | EPS | - | - | - | - | - | 15473 | 7.32 | 548 | C |
| 2 | - | EPS | - | Heater | Gas | - | - | 13315 | 4.75 | 998 | C |
| 3 | ETICS | EPS | PVC | - | - | - | - | 9608 | 3.03 | 2088 | B |
| 4 | - | EPS | PVC | - | - | - | - | 12536 | 3.87 | 1284 | C |
| 5 | - | EPS | PVC | Heater | Gas | - | - | 11823 | 2.88 | 1734 | C |

Moreover, the user can also choose optimization variables of the sample (i.e., possible retrofits), as well as their cost. Finally, it is also possible to choose the optimization algorithm and its number of iterations. The obtained results can be consulted in the app with an interactive Pareto optimality chart and downloaded in a ".csv" format.

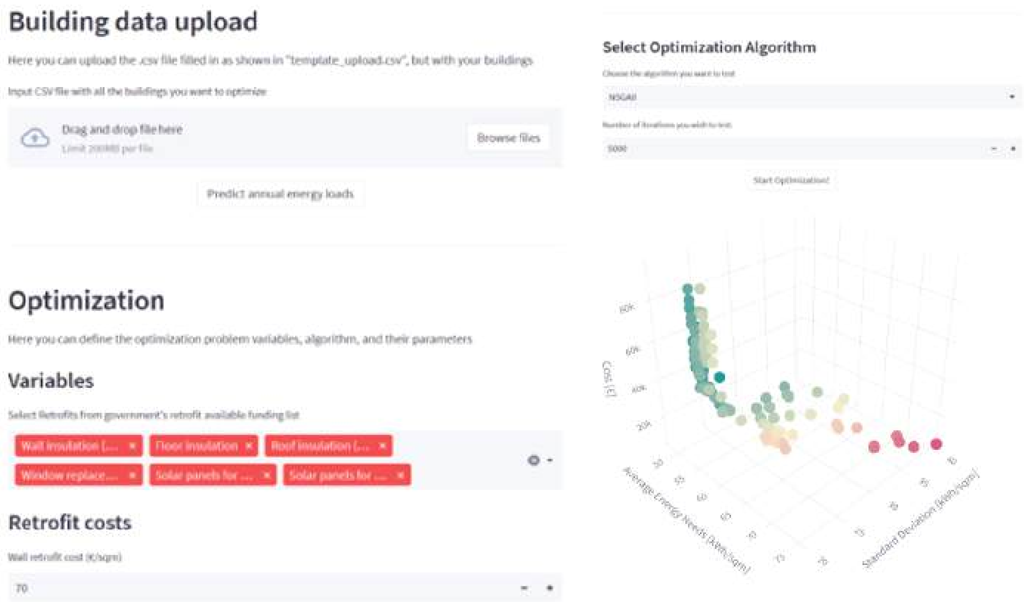



Figure 4.9: Application interface for urban Analysis and Optimization Processes (AOP).

### 4.1.5 Discussion

The results obtained in this work show $R^2$ scores of 0.84 and 0.79 and RMSE of 0.21 and 39.77 $\mathrm{kW\,h\,m^{-2}}$ for the EPC (R ratio) and energy needs (Ntc) predictions respectively. These results are adequate for integrating surrogate models and MOO into an easy-to-use web application that promotes the engagement of citizens and stakeholders in the building retrofit and EPC scheme. The validation results show slightly lower accuracy than the one seen in Burratti et al. [286] and Khayatian et al. [229] models. This may be explained by the databases used in these studies which may have different features and data. To improve the obtained accuracy in this study, new data cleaning methods can be used to improve additional features and feature engineering processes can be applied to develop additional features that might be missing from the original database.

The integration of the MOO process helped tackle conflicting objectives that have a significant weight in a citizen's decision about the rehabilitation of their household or building. For this case study EPC, the MOO process returned 5 optimum retrofit solutions capable of reducing energy savings up to 60% while minimizing the cost of the retrofit. However, optimization results should improve with the testing of multiple MOO algorithms, as well as with the tuning of each algorithm's hyperparameters. Additionally, an optimization process that explores different objectives and metrics may be helpful in the communication of results to the user.

The web app interface may improve with different communication strategies of both the input data, model exploration, and optimum retrofit solutions: (1) segmenting input and data forms into different sections; (2) assigning icons to inputs and features; (3) other interactive experiences such as interactive tables and/or charts. Furthermore, the web application would benefit from users' feedback, collaborative design, and integration of other EU countries' EPC databases. This would increase the scope and usability of the app to a larger scale. In such cases, feature engineering techniques [204, 300] (particularly undersampling [301]) can be researched to reduce the size of the database into a balanced set of buildings that is still representative of the required variables and domains. Such techniques are often used to improve a predictive model's performance when trained with an imbalanced database that presents skewed distributions of features [302, 259]. However, these techniques can also reduce the number of samples that need to be simulated, while still representing balanced and robust databases [303].

To conclude, the proposed innovative framework resulted in a SM developed with an Existing BID capable of reducing the required inputs from 88 to 20 to accurately predict an EPC class and energy use. Since no simulations are required for energy certification, this study's goal was not to speed up simulation times but to simplify and streamline the EPC process. Moreover, the model was implemented into two web apps for different AOP at the building and urban scale, showcasing its portability. Both web apps allow users to obtain valuable feedback regarding building retrofit needs at different scales with much less complexity than when using energy certification calculation. Thus, this study responds to two key factors that were identified in the research questions of this Thesis (Chapter 1, Section 1.3): AOP complexity and portability.

## 4.2 Optimization of Building retrofit and design using a Surrogate Model developed with a Synthetic Building Database

The operation of a large part of the building stock is still energy-intensive and a cause of anthropogenic carbon emissions that actively contribute to climate change [1]. This motivates sustainable building construction and retrofits in consolidated urban territories to meet the sustainable development goals set by the current political agenda [3]. Particularly, to reduce the energy consumption of our building stock, we need to assess its performance through Building Performance Simulation (BPS). However, these typically require expertise and become slow when analyzing numerous buildings. These limitations hinder the analysis of large urban areas that are critical to support the design and planning of cities, and the city policy-making sector [5]. Current practices use Urban Building Energy Models (UBEM) approaches to address this pressing issue [304]. However, these approaches still require a high level of expertise and the process is still time-consuming, error-prone, and tiresome [305]. Additionally, multiple tools are required to obtain a holistic analysis of a building's performance. Thus, it is crucial to make these approaches easier to use, faster, and portable.

To address these goals, this study proposes an approach that uses ADA [18] to create versatile surrogate models for urban analyses. Algorithmically modeled building archetypes are used as a BID, based on city-wide building properties rather than context-specific data commonly used in UBEM surrogates. This approach allows a broader use of SM for larger urban areas and mitigates the above-mentioned limitations.

With AD it is possible to generate buildings through algorithms [19], and when combined with BPS analysis one can automatically model and simulate thousands of design variations. Thus, this approach uses ADA to generate and simulate multiple instances of parametric building archetypes, compiling them into a synthetic database. With this BID, multiple Regression models are tested to develop a SM. This model promptly predicts the performance of different urban settings depending on simple inputs such as building geometry and constructions, and its accuracy and speed are worthy of comparison against conventional UBEM approaches.

This study's research methodology can be illustrated by Figure 4.10. Initially, the building AOP are described. In this study, the developed SM is used to tackle one design and one urban rehabilitation problem. The problems encompass energy performance simulated with Energy Plus [32], which is integrated with the Khepri tool [18], to automate simulations and building designs. With this approach, a Synthetic BID is developed to train a regression model capable of predicting a building's energy use. This model is then deployed in a web app for a single building design energy optimization and in a CAD tool for urban rehabilitation.

### 4.2.1 Problem Description

This study aims to develop a Synthetic BID and use it to develop a SM capable of predicting a building's Energy Use Intensity (EUI) in Lisbon, Portugal. The goals are to speed up simulation time, reduce

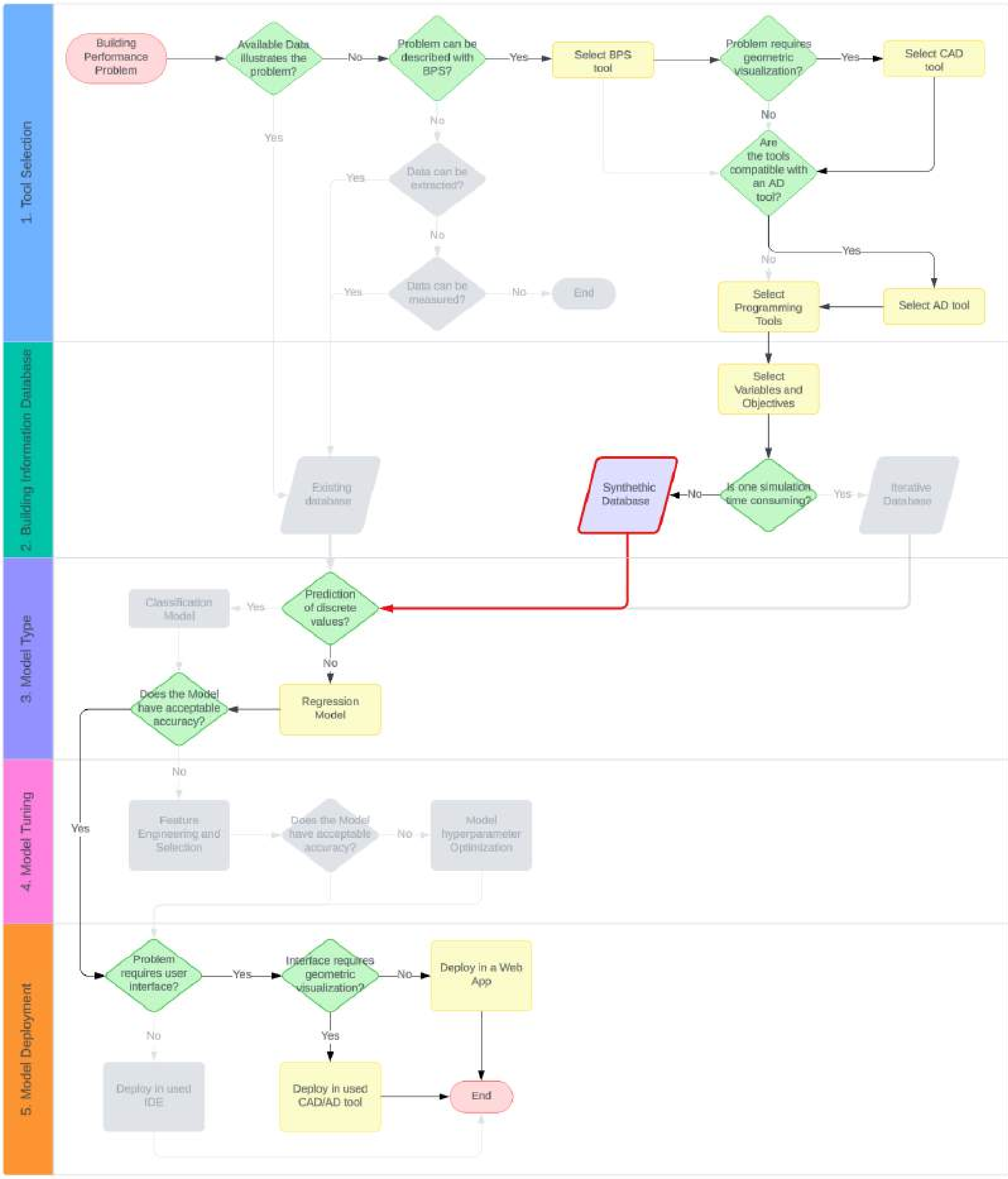



Figure 4.10: Workflow diagram for Section 4.2

the level of expertise required to obtain results and make it less dependent on BPS tools. This model was tested with two optimization problems. The first problem considers a Single-Objective Optimization (SOO) to minimize a building's EUI with the given building parameters (4.9). The second problem considers a MOO problem for the rehabilitation of a building block of 21 buildings to minimize the total EUI (4.10), total retrofit cost (4.11), and buildings' EUI standard deviation (4.12), to ensure some homogeneity of performance among all buildings [11].

$$o_1 = min(\text{EUI})\ [\text{kW h m}^{-2}] \tag{4.9}$$

$$o_2 = min(\sum_{i=1}^{21} \mathrm{EUI}) \text{ [kW h m}^{-2}\text{]} \tag{4.10}$$

$$o_3 = min(\sum_{i=1}^{21} Cost) \text{ [€]} \tag{4.11}$$

$$o_4 = min(\sigma\{\mathrm{EUI}_1, ..., \mathrm{EUI}_2 1\}) \text{ [kW h m}^{-2}\text{]} \tag{4.12}$$

For the SOO, a Genetic Algorithm was used to iterate along variables encompassing a building's geometric parameters included in the BID. For the MOO problem, the NSGA II was used to iterate along each building's retrofit variables. Thus, each building is represented as a variable with 4 options for its corresponding construction period: (1) no retrofit, (2) wall retrofit, (3) roof retrofit, and (4) wall and roof retrofit.

The MOO problem is tackled using two methods: (1) a BPS-based approach that fully simulates each building block in each iteration, and (2) using the developed SM. To test the effectiveness of using this surrogate approach, it is possible to compare it against a conventional one, regarding its execution time, expertise required, and portability.

### 4.2.2 Building Information Database

The BID developed in this study is a Synthetic database that consists of multiple building energy simulation results for combinations of different geometries and materials by construction periods. The geometric parameters considered are the number of floors $n_f$, rectangular proportion $p$, orientation $\theta$, glazing ratio $g_r$, and floor area $A$. For the construction period materials, a building's period of construction defined the materials for exterior walls, roofs, floors, windows, and the walls and roofs respective retrofit. These materials were defined in Energy Plus according to each solution's U-value [W m$^{-2}$ °C] (Table 4.12) [306]. Thus, the synthetic BID can be illustrated by Equation 4.13 and its domains, where n is the number of samples in the database.

$$f\left(\begin{bmatrix} n_f & p & \theta & g_r & A & C_p \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ n_{fn} & p_n & \theta_n & g_{rn} & A_n & C_{pn} \end{bmatrix}\right) = \begin{bmatrix} \mathrm{EUI} \\ \vdots \\ \mathrm{EUI}_m \end{bmatrix} \text{[kW h m}^{-2}\text{]} \tag{4.13}$$

$$n_f \in [1...11], \quad p \in [1...5], \quad \theta \in [0...\pi], \quad g_r \in [0.1...0.7],$$
$$A \in [50...800][\text{m}^2], \quad C_p \in \{1,2,3,4,5,6,7,8,9,10\}$$

To generate the building archetypes, and their geometric variations, ADA was used to automate the energy simulations, by describing the building domains and generating every possible design and material combination among them. In this case, all the building archetypes are modeled to uniformly divided parameter domains such as the number of floors (from 2 to 11, step size = 3), rectangular proportion (from 1 to 5, step size = 2), orientation (from 0 to $\pi$ radians, where 0 is East, step size = pi/4),

Table 4.12: U-Value [$\mathrm{W\,m^{-2}\,°C}$] for construction materials according to building construction period.

| Building Age | Age ID | Wall U-Value [$\mathrm{W\,m^{-2}\,°C}$] | Roof U-Value [$\mathrm{W\,m^{-2}\,°C}$] | Floor U-Value [$\mathrm{W\,m^{-2}\,°C}$] | Window U-Value [$\mathrm{W\,m^{-2}\,°C}$] | Wall retrofit U-value [$\mathrm{W\,m^{-2}\,°C}$] | Roof retrofit U-value [$\mathrm{W\,m^{-2}\,°C}$] |
|---|---|---|---|---|---|---|---|
| $<$1919 | 1 | 2.78 | 1.99 | 1.80 | 2.69 | 0.61 | 0.63 |
| 1919-1945 | 2 | 2.78 | 1.99 | 1.80 | 2.69 | 0.61 | 0.63 |
| 1946-1960 | 3 | 1.49 | 1.99 | 1.80 | 2.69 | 0.57 | 0.63 |
| 1961-1970 | 4 | 1.08 | 1.99 | 3.03 | 2.69 | 0.49 | 0.63 |
| 1971-1980 | 5 | 1.26 | 1.99 | 3.03 | 2.69 | 0.53 | 0.63 |
| 1981-1990 | 6 | 0.50 | 1.99 | 3.03 | 2.69 | 0.32 | 0.63 |
| 1991-1995 | 7 | 0.49 | 1.99 | 3.03 | 2.69 | 0.32 | 0.63 |
| 1996-2000 | 8 | 0.46 | 1.99 | 2.31 | 2.69 | 0.29 | 0.63 |
| 2001-2005 | 9 | 0.25 | 1.99 | 2.31 | 2.69 | 0.19 | 0.63 |
| $>$2006 | 10 | 0.25 | 1.99 | 2.31 | 2.69 | 0.19 | 0.63 |

glazing ratio (from 0 to 0.7, step size = 0.35), and floor area (from 50 to 800 m2, step size = 75 m2). This yielded a synthetic BID of $\approx$55000 simulated buildings.

The buildings are generated with an AD tool in Rhino [79] and Grasshopper [80], and simulated with Ladybug Tools [100]. Examples of the building samples varying according to their parameters can be seen in Figure 4.11. The sample is then attributed a solution material according to its construction period, and simulated.

To better visualize the results, it is possible to plot them according to the parameters domains. Figure 4.12 shows the energy simulation result (z-axis) of the BID according to different parameters present in the x- and y-axes. It is visible in the figure that most parameters have a polynomial relation with building energy use. This motivates the integration of polynomial features of degree 3 in the database when training Linear Regression (LR) models to obtain a better model fit and accuracy results for predictions outside parameter domain boundaries.

To test the developed surrogate models, instead of splitting the database into a training and test set, it was possible to obtain a building database in Lisbon containing data relative to 2193 residential buildings in Lisbon, Portugal (Figure 4.13). The available information for an urban area, including building implantation polygon, construction periods, floors, area, glazing ratio, and typology, is imported. Subsequently, automated simulations using EnergyPlus are conducted. Since this database represents the case study for the proposed optimization problems, it was considered also as a test set for the surrogate model performance evaluation in the next Subsection.

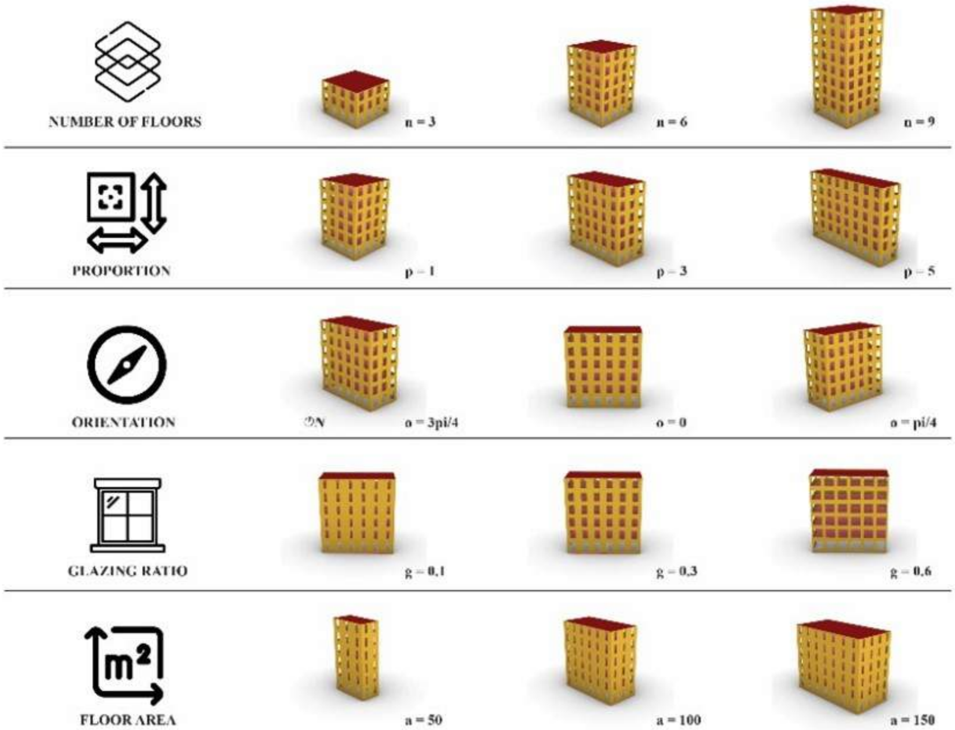



Figure 4.11: Building Information Database (BID) sample examples for chosen geometrical features.

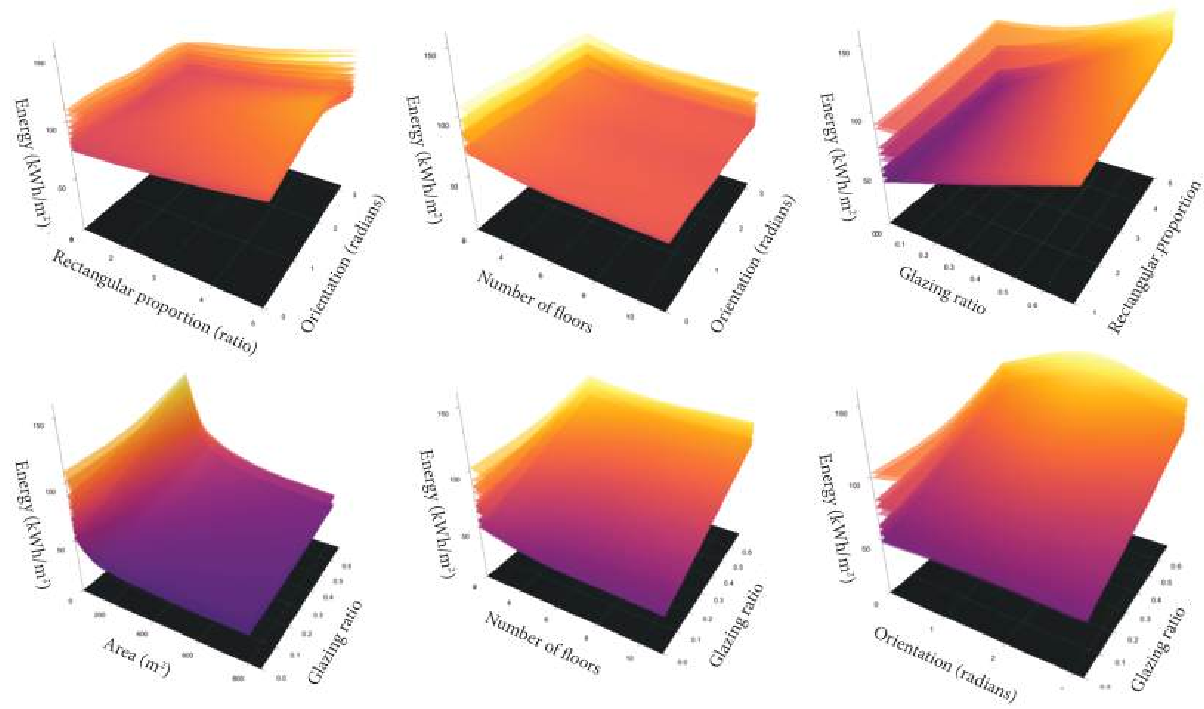



Figure 4.12: Building energy use according to the different parameters in the Building Information Database (BID). Different layers represent different building construction periods.

### 4.2.3 Model Type

The prediction of annual EUI [$\mathrm{kW\,h\,m^{-2}}$] in a building is best represented with a regression problem. As such, different supervised learning models were used from the Sci-kit learn packages [252]. Particularly,

Figure 4.13: Building test set for the Surrogate Model. Lisbon, Portugal.

in this study the models Linear Regression (LR), Ridge (a variation of Linear Regression), Random Forests (RF), and Extra Trees (ET) were trained using the developed BID.

After selecting the models, these are trained with the generated synthetic database, and their performance results were compared. Table 4.13 shows a Root Mean Squared Error (RMSE) of $\approx 15$ [$\mathrm{kW\,h\,m^{-2}}$] for the LR and Ridge models. Additionally, ensemble models such as RF and ET respectively show $\approx 10$ and $\approx 5$ [$\mathrm{kW\,h\,m^{-2}}$]. The highest Coefficient of determination ($R^2$) was obtained by the ET model with 0.95.

Table 4.13: Mean Average Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of determination ($R^2$) for the select Surrogate Models (SM).

| | **LR** | **Ridge** | **RF** | **ET** |
|---|---|---|---|---|
| **MAE** [$\mathrm{kW\,h\,m^{-2}}$] | 15.85 | 15.79 | -6.06 | -2.21 |
| **RMSE** [$\mathrm{kW\,h\,m^{-2}}$] | 15.40 | 15.40 | 9.88 | 5.44 |
| **R2 score** | 0.64 | 0.64 | 0.85 | 0.95 |

It is possible to compare the errors for each predicted building sample of the tested models by visualizing a distribution plot of the error that the developed SM presented in predicting the 2193 buildings (Figure 4.14). From the figure, it is noticeable that the best-performing model according to error indicators is the ET, which presented a smaller error distribution with a slight tendency to overestimate the EUI. This happens since the MAE is negative. The RF model also shows similar errors within smaller boundaries and with a similar overestimation of a building's energy use.

### 4.2.4 Model Deployment

With the ET model showing an acceptable accuracy, the model can now be deployed and tested with the proposed optimization problems. As seen in Subsection Problem Description, there are two proposed

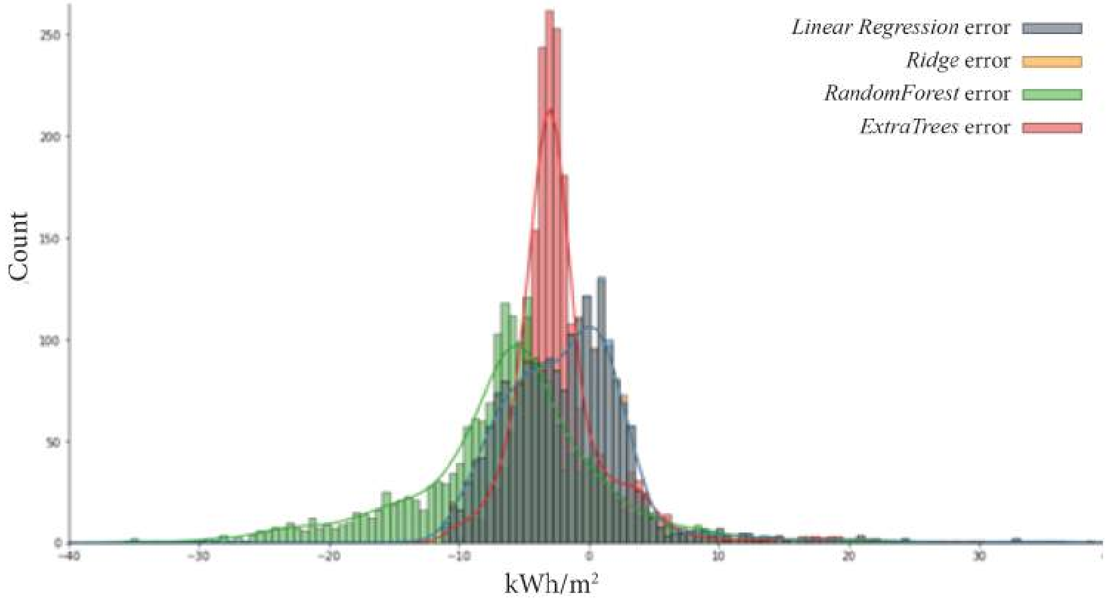



Figure 4.14: Distribution plot of prediction errors in kW h m$^{-2}$ for the tested Surrogate Models (SM).

optimization problems: A SOO to minimize the EUI of a building design (Equation 4.9); and a MOO to optimize the EUI, cost, and fairness of a retrofit (Equations 4.10, 4.11, 4.12). Thus, similar to Section 4.1, this Section is further subdivided by presenting the obtained optimization results and showcasing the resulting interface with the final SM deployed.

### Optimization Results

For the SOO problem to minimize $o_1$, 5000 iterations were tested with the NSGAII algorithm. Figure 4.15 illustrates the obtained results for all the tested solutions with minimums of ≈25 and maximums of ≈128 [kW h m$^{-2}$]. The best solution obtained was from construction period 3, but highly conditioned by lower proportion and window ratios, as well as by larger areas. However, periods 6 and 7 (Table 4.12) with insulation also provide less conditioned results of energy use. The best solutions generally show bigger areas and smaller proportions, with an optimum orientation of 0 or pi radians (which represent the same orientation) and a higher number of floors. Glazing ratios for a building in Lisbon show minimums of 0.1. Thus, Keeping in mind these values, it is possible to extrapolate design heuristics for a building in Lisbon, Portugal. Finally, by analyzing the chart, the boundaries for geometric parameters that may condition the design are easily stretched by better construction solutions.

For the MOO problem, 800 iterations spread through 40 generations were tested with the NSGAII algorithm to obtain minimum $o_2$, $o_3$, and $o_4$ (Subsection Problem Description). Minimum EUI (Equation 4.10) aims to find the combination of retrofits that yield less energy use, while the minimum cost $C$ (Equation 4.11) aims to find the cheapest combinations of retrofits. Finally, the minimum energy uses standard deviation $\sigma$ (Equation 4.12 ensures fairness among the buildings that compose the block.

This problem was solved using a conventional simulation-based approach, where each iteration is simulated, and with the developed SM. This helps compare the obtained results according to their sim-

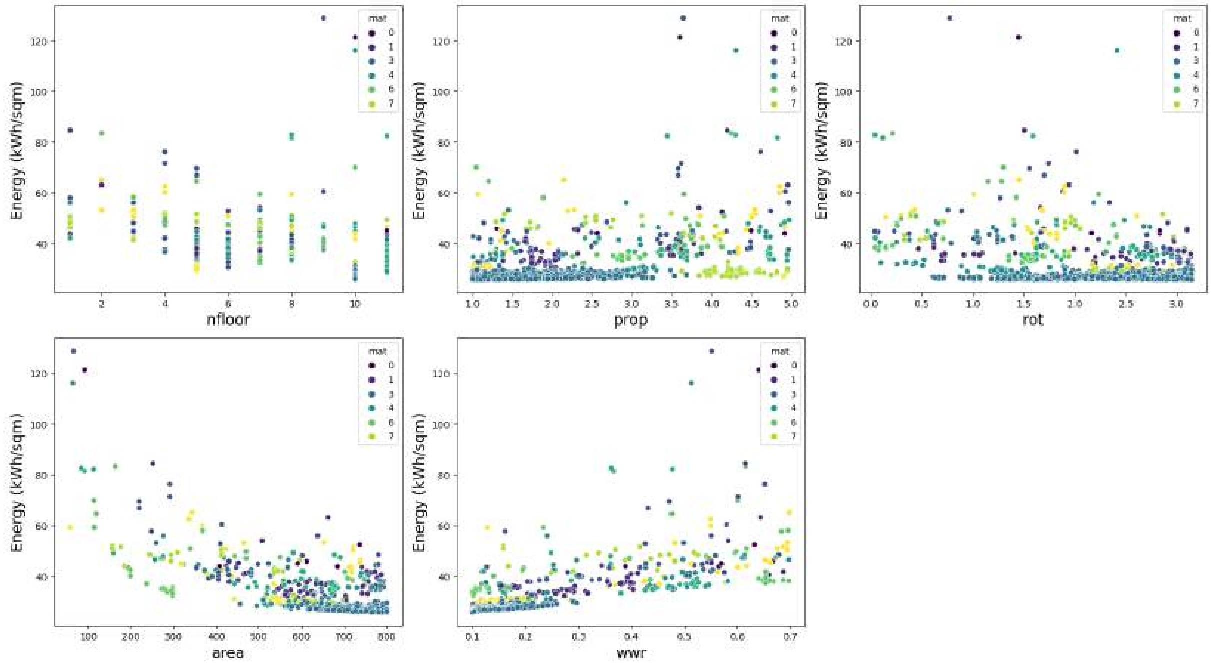
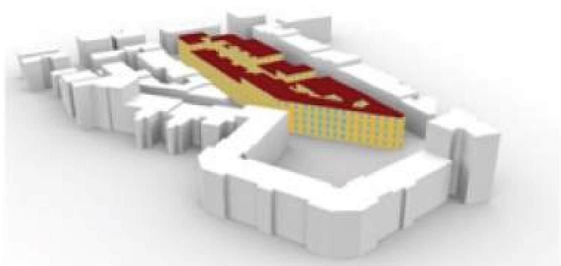
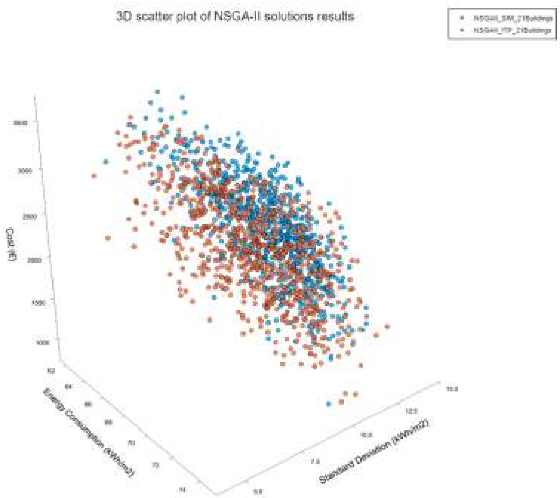



Figure 4.15: Single-Objective Optimization (SOO) results in a scatter plot for each variable by construction period for 5000 iterations with the NSGAII algorithm.

ilarity and execution time. The scatter plot in Figure 4.16 shows the existing trade-offs between the different objectives. One easily seen is that solutions that are more homogeneous and have a smaller energy consumption entail more expensive renovations. Additionally, it is visible that both solutions have a high similarity in their objective results.



(a) Sample Building block for the Multi-Objective Optimization (MOO) process.




(b) MOO results comparison between a simulation-based approach (blue) and with the developed SM (orange).

Figure 4.16: Moo problem building block (a) and obtained results (b).

To compare the execution time, Table 4.14 documents the computation time required to simulate the 2193 buildings in the test set (Figure 4.13) and to predict their energy simulation results with the developed SM. Additionally, it shows the exponential gains obtained in computation time when tackling a simple MOO problem with a simulation-based approach and with the proposed approach in this study. Results show that it is possible to speed up an optimization process up to 85x using this developed SM, which can accurately predict a simulation result for a building's energy use in Lisbon, Portugal.

The optimization algorithms explored solutions between 62 $\mathrm{kW\,h\,m^{-2}}$ and 74 $\mathrm{kW\,h\,m^{-2}}$ for the annual EUI, 5 $\mathrm{kW\,h\,m^{-2}}$ and 15 $\mathrm{kW\,h\,m^{-2}}$ for the standard deviation, and 1000 € and 3500 € for the mean cost of retrofit per building.

Table 4.14: Comparison of the execution times for both optimization and a single simulation of the full Lisbon test database (Figure 4.13).

| | **elapsed time (seconds)** | |
|---|---|---|
| | **Dataset simulation** | **Optimization** |
| **Surrogate model** | 0.08 | 791.99 |
| **Simulation** | 5820.00 | 67516.70 |

**Interface**

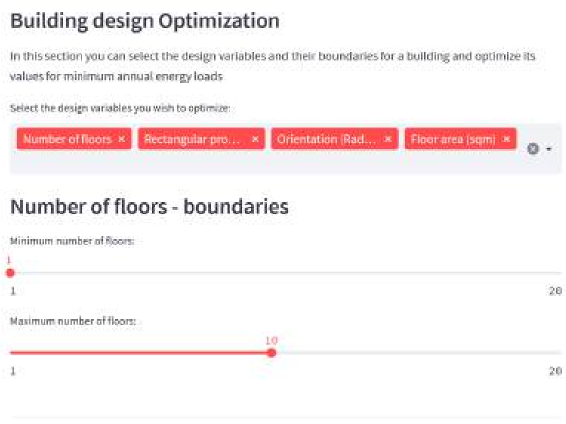
The developed SM was deployed in a web app [6] with Streamlit [245]. This interface allows users to define their SOO problem variables according to their designs and to compute faster design parameter optimization to minimize a building's energy use. To start with, the interface opens a menu that allows users to choose the design parameters that require optimization, define their boundary constraints, and run the optimization with a specified number of iterations.

A user can choose the SOO optimization variables such as the number of floors $nf$, rectangular proportion $p$, orientation $\theta$, and floor area $A$ (Figure 4.17a). Once the variables are chosen, the user can set the selected variables' respective domain, and the constant values required to run the developed SM (Figure 4.17(b)). After finishing the problem settings, the user can select any number of iterations, a desired number of optimum solutions, and a population size for the generations (Figure 4.17(c)). Finally, results can be downloaded or visualized within the web app in a scatter plot (Figure 4.17(d)). For the sample portrayed in the figure, results have shown reductions of 8% in the Energy Use Intensity (EUI) of the building.

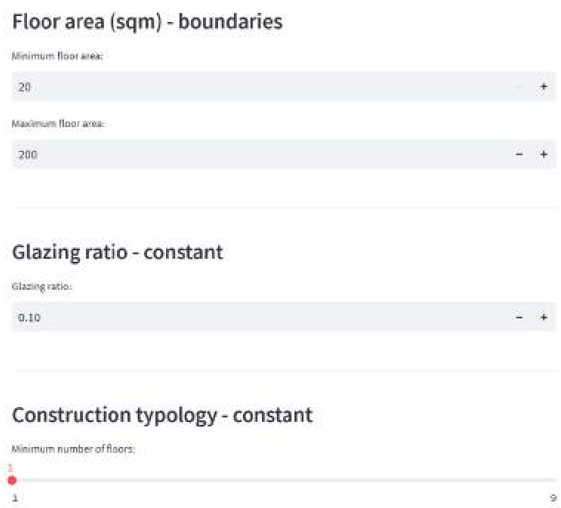
### 4.2.5 Discussion

The presented study results show high levels of accuracy for the simulation of the original case study in Lisbon of 2193 residential buildings (Table 4.13). The model has shown a Root Mean Squared Error (RMSE) of 5.44 kWh/m2, and a Coefficient of determination ($R^2$) of 0.95, which explains the target results variance. Additionally, the larger error values come from extrapolations that the prediction model made outside the parameters' domain (e.g., areas of 2000 m2 caused most of the outliers since the maximum simulated area for all archetypes was 800 m2). It is possible to assert that the prediction

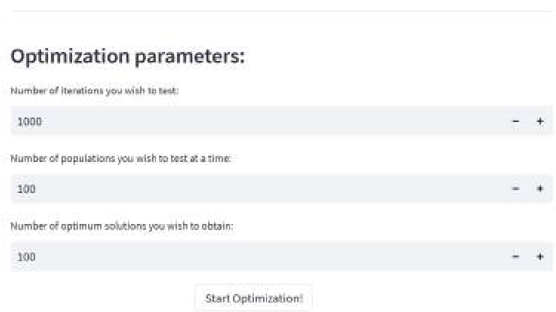
[6] https://goncalo-araujo-energy-plus-sim-app-app-energyplus-y83a4z.streamlit.app/



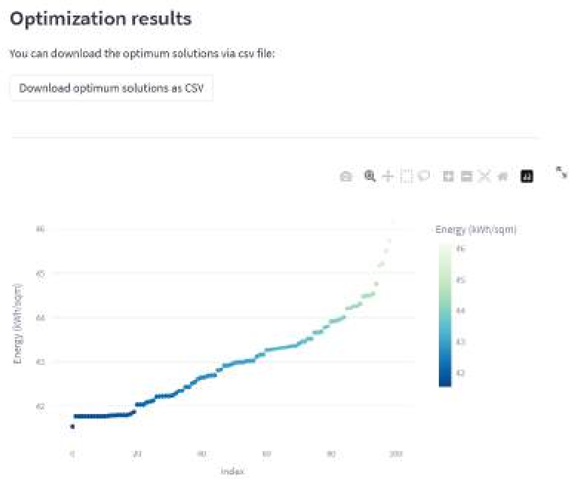

(a) Specification of optimization variables.




(b) Variable boundary selection and constant variables specification




(c) Optimization parameters and run button.




(d) Obtained results to download and visualize in a scatter plot.

Figure 4.17: Single-Objective Optimization (SOO) problem interface to find the best design and material parameters of a building in Lisbon, Portugal.

is very accurate for buildings within the domain of the continuous parameters that are required by the developed SM.

Besides evaluating the model's accuracy, a MOO and Single-Objective Optimization (SOO) problems (Equations 4.9-4.12) were tackled with both the synthetic database SM and simulation-based approaches. Figure 4.16 shows high levels of spatial similarity among the objectives of both approaches for the MOO, while Table 4.14 compares both processes by showing the elapsed time for the simulation of the original case study in Lisbon, and for the proposed MOO problem. This table shows that the SM largely outperforms the simulation approach and there are benefits obtained in increased energy prediction speed and low error rates when compared with simulations. Furthermore, the model can be easily integrated with interactive interfaces to allow even the less experienced users to perform simple Analysis and Optimization Processes (AOP) (Figure 4.17).

One consideration should be that the computation time comparisons present in table 4.14, comprise only the end-user time to perform the simulation. Thus, it does not consider the time taken with the Building Information Database (BID) to develop and train the model (Subsections Building Information Database, and Model Type). Admittedly, it takes more time to build the SM than to perform a simulation, but this time is quickly recovered when the model is used to support numerous AOP.

Nevertheless, this approach bears some limitations. The simulation of the building archetypes does not account for shading and context geometry, which can hinder simulated and predicted results. Additionally, a low number of retrofits were added as discrete parameters, with only one retrofit for each wall and roof. When one must test multiple construction solutions, the SM could take many discrete parameters, resulting in hundreds of thousands of buildings that needed to be modeled and simulated, which could be highly time-consuming. Feature Engineering (FE) techniques such as Undersampling can tackle this problem and reduce the number of samples that need to be generated by applying, for example, a normal distribution to the combination of variables. Parallel Computation techniques [126] can also be explored further to increase the number of objectives and discrete variables.

In conclusion, this study illustrates the proposed innovative framework for SM development with a Synthetic BID. The model showed high accuracy and a speed-up factor of $\approx 100$, compared with the simulation time of all the buildings. Furthermore, the model was implemented with only six inputs and for two optimization problems at different scales, showcasing its simplicity and portability. Thus, this case study illustrates the proposed framework's potential to tackle the identified research questions (Chapter 1, Section 1.3): Making AOP easier with less required inputs, faster with the model's prediction speed, and portable with its capacity to be deployed effortlessly for different problems and scales.

## 4.3 Optimization of thermochromic glazing using a Surrogate Model developed with an Iterative Building Database

To achieve Sustainable Development Goals (SDG) by reducing energy use and emissions, research has been converging towards reducing building energy consumption and achieving Net-Zero buildings [2]. Window glazing allows the transmission of natural light into the building interior, which positively affects energy consumption by lowering the electric lighting needs. However, glazing is responsible for significant heat exchanges that can harm the building's thermal performance [307]. Thus, it is no surprise that researchers have been studying methods to improve the performance and adaptation of the building glazing systems. Smart glazing technologies have shown promising results in increasing thermal comfort and reducing building energy consumption [308, 309]. Particularly, Thermochromic Glazing (TCG) is a smart passive glazing solution that incorporates a coating of a thermochromic material in the glazing system that changes its thermal and optical properties according to its surface temperature. This is achieved through a thin-layered coating of Vanadium Oxide (VO2) placed between two glass panes [309].

VO2 is generally the chosen material to produce TCG coatings because when it reaches a critical transition temperature ($T_c$), the material transitions from semi-conductor to metal according to a hysteresis process [310]. Alas, VO2 has a high $T_c$ of 68ºC, which is significantly higher than typical room temperatures. To solve this issue, manufacturers can dope the thermochromic coating with other constituents to alter the coating's $T_c$ and glass solar and visible transmittance ($\tau_{sol}$ and $\tau_{vis}$) values to obtain suitable TCG properties that provide energy consumption benefits [308, 310, 309, 311]. However, cold climates still struggle to provide the TCG chromic coating with the required temperature to transition. Therefore, TCG systems are reported to present smaller improvements against traditional glazing systems in a cold climate than a TCG system in warmer climates [312, 313, 311, 314, 315]. Because of this climatic amplitude, TCG requires different properties to yield the best performances. Other studies have focused on the effect that TCG T$_c$, $\tau_{sol}$ and $\tau_{vis}$, and hysteresis loops have on building energy needs [58, 310, 311, 316, 317, 307]. Results show that a quick transition, meaning narrow and steep switching temperature ranges, and lower $T_c$ generally represent bigger energy savings.

Most studies pointing out energy savings with TCG account for climatization and electric lighting energy needs but report a decrease in the former with an increase in the latter. This means that despite the thermal benefits provided by the TCG, its use harms daylighting performance [313, 315]. Thus, lower transition temperatures that allow the glass to switch to darker states, can eventually offset the improvements obtained in climatization energy needs with increasing electric lighting energy needs. This trade-off means that when manufacturing and applying TCG, one must consider ideal $T_c$, $\tau_{sol}$, and $\tau_{vis}$ values that improve thermal performance, without critically harming lighting performance for a specific climate. Thus, finding the best TCG properties that provide the least energy consumption for both electric lighting energy needs and climatization needs can be treated as Multi-Objective Optimization (MOO) problem.

This study integrates a SM with a MOO process of a TCG in an office room and compares it with an off-the-market TCG system in two different climates. The objectives of this MOO process are to minimize the office's heating, cooling, and electric lighting energy needs. This process helped to quantify the existing conflicts between different end-uses (electric lighting and climatization) of an office's energy needs regarding different TCG properties such as $T_c$, the switching temperature range of $\tau_{sol}$ and $\tau_{vis}$, as well as their range of values. This can help manufacturers and clients select the best-performing properties of TCG and guide its application to yield a minimum energy consumption.

The optimization of the energy use of an office can be performed with Building Performance Simulation (BPS), particularly, in Energy Plus, one can set TCG layers in glazing materials by defining their properties. This study's process can be illustrated by Figure 4.18. Since a simulation of a building with TCG is time-consuming, a Iterative Building Database is used to train the proposed SM. This database is obtained by integrating an automated simulation-based optimization process with multiple metaheuristics and a low number of iterations. This is done with the validated simulation software EnergyPlus [32] integrated into a Python environment [101]. Afterward, the model is deployed in the Python Integrated Development Environment (IDE) to solve the same MOO problem with the best-performing algorithm for 5000 iterations. The results illustrate the TCG optimum values for different climates.

This study's innovation lies in optimizing the theoretical properties of a TCG for multiple objectives and integrating a surrogate model that speeds up the whole process, removing simulations from the equation. Consequently, this work can guide TCG manufacturers into achieving the best performance for local climates, by altering the TCG's properties. This in turn, contributes to the achievement of the United Nations SDG 11 ”Sustainable Cities and Communities” and 13 ”Climate Action”, since it can help mitigate energy poverty and decarbonize the building sector, contributing to more sustainable cities and the mitigation of climate change.

### 4.3.1 Problem Description

This work is based on previous research by Teixeira et al. [315] and considers a pre-calibrated building energy simulation model of an office room facing southeast in Copenhagen, Denmark, and the same room in Lisbon, Portugal. Denmark is characterized by a cold climate, and Portugal by a warm Mediterranean climate (Figure 4.19) [318]. The office has roughly 20 m$^2$ of floor area and 10 m$^2$ of window area. The window is composed of a TCG assembly (12 mm), air gap (12 mm), and a Low-E coated glass (6 mm) (Figure 4.20). Table 4.15 describes the TCG solar-optical and thermal properties at different stages of surface temperatures. It is visible that significant changes occur mostly in the $\tau_{sol}$ and $\tau_{vis}$ values, ranging from 0.69 to 0.26 and 0.72 to 0.02 at 5ºC and 85ºC, respectively. It is visible that $\tau_{vis}$ shows a higher decrease than $\tau_{sol}$ with the thermochromic transition.

For the simulation and analysis process, EnergyPlus [32] was coupled with Eppy [101], a scripting language for Energy Plus files (IDF) in Python. With Eppy it is possible to automate specific changes in the Energy Plus file fields and objects, run IDF files, and process results. An initial simulation was performed with the standard TCG system to obtain the office's total cooling, heating, and lighting electricity rate (W). The office room was modeled with an occupation of one person during work hours and

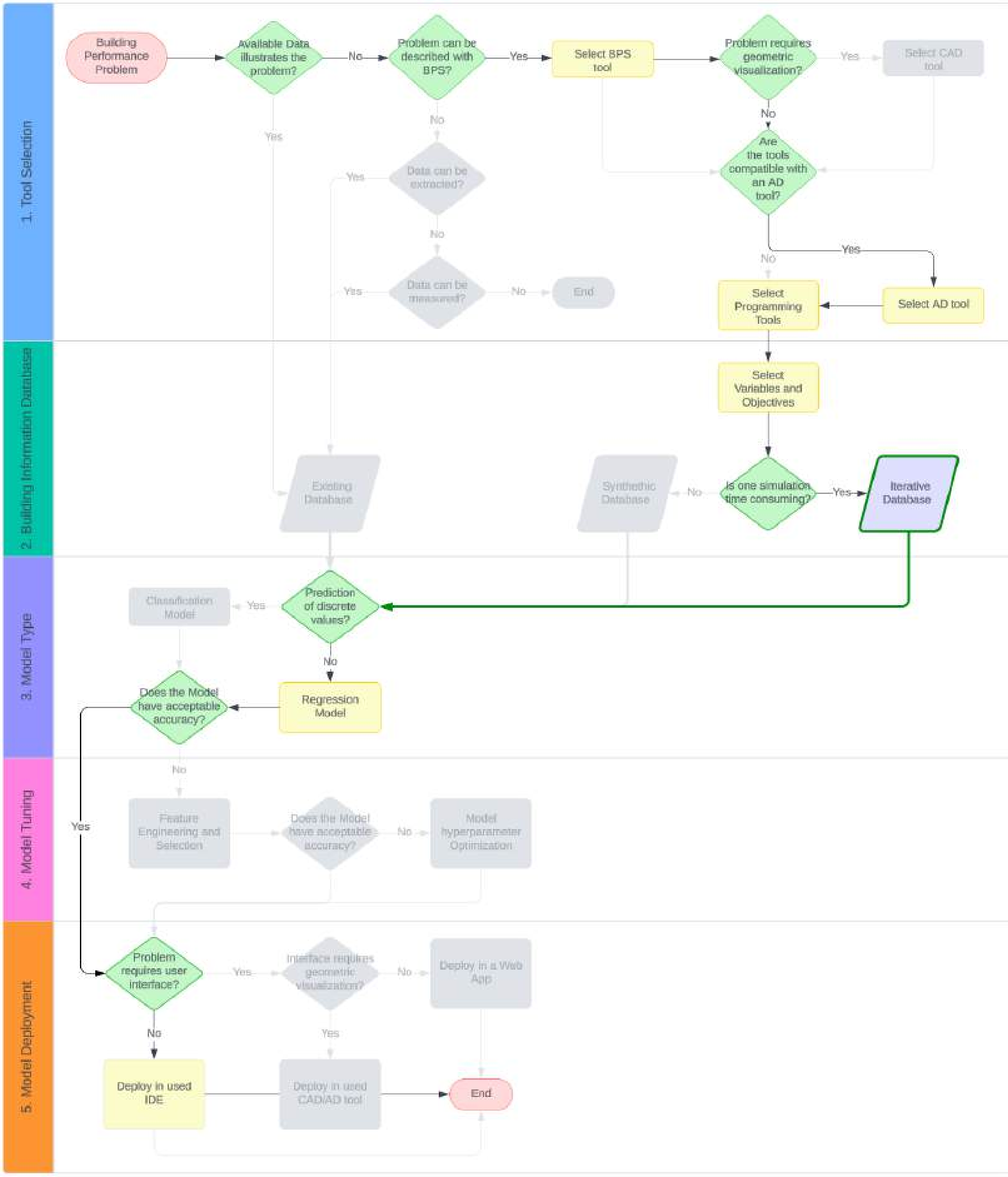



Figure 4.18: Workflow diagram for Section 4.3.

weekdays. Additionally, the electric equipment considered comprises laptop and desktop units. The zone sensors for Energy Plus were located at a desk position for one year, both for thermal and lighting outputs. The electric lighting control, Heating, Ventilation, and Air Conditioning (HVAC) parameters, and remaining simulation inputs are described accordingly in Table 4.16.

From the analysis of Table 4.15, it is possible to determine the MOO problem variables and objectives with Equations 4.14 and 4.15 to represent minimum heating, cooling (*HVAC*), and lighting ($L$) energy use. Transition temperatures are responsible for when the material gets darker, and transmittance can

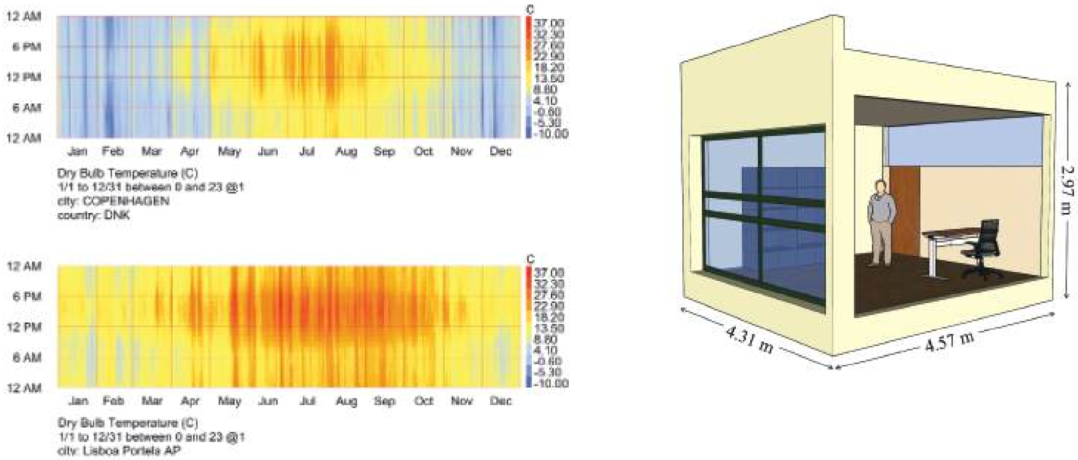



Figure 4.19: Top - Denmark (top) and Lisbon (bottom) yearly dry bulb temperature. Bottom - 3D model of the office room used as a case study.

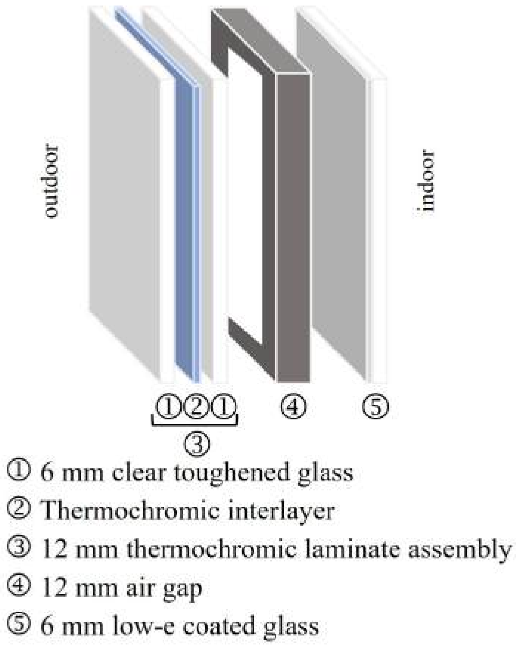



Figure 4.20: Thermochromic glazing layers.

Table 4.15: Thermochromic glazing properties simulated for each temperature state [315].

| ***T*** [℃] | **5** | **15** | **25** | **45** | **65** | **85** |
|---|---|---|---|---|---|---|
| $U[W/(m^2K)]$ | | | 1.6 | | | |
| $\tau_{vis}$ | 0.72 | 0.69 | 0.63 | 0.36 | 0.11 | 0.02 |
| $\rho_{visF}$ | 0.07 | 0.07 | 0.06 | 0.05 | 0.04 | 0.04 |
| $\rho_{visB}$ | 0.07 | 0.07 | 0.06 | 0.05 | 0.04 | 0.04 |
| $\tau_{sol}$ | 0.69 | 0.67 | 0.64 | 0.50 | 0.34 | 0.26 |
| $\rho_{solF}$ | 0.06 | 0.06 | 0.06 | 0.05 | 0.05 | 0.05 |
| $\rho_{solB}$ | 0.06 | 0.06 | 0.06 | 0.05 | 0.05 | 0.05 |

Table 4.16: Simulation inputs and outputs.

| | **Values** |
|---|---|
| **Occupation** | 1 person weekdays from 9 to 6 |
| **Air changes/hour** | 1.0 $h^{-1}$ |
| **Electric equipment** | Desktop (155 W): from 9 to 6<br>Laptop (30 W): from 9 to 6 |
| **Artificial lighting** | Available during occupation (110 W) |
| **Lighting control** | Set-point of 500 lux |
| **HVAC set-points** | During occupation: 20-24 ℃ |
| **SCOP/SEER** | 4.43/7.98 |
| **Simulation period** | 30 timesteps per hour, 1 year period |
| **Outputs** | Total Cooling Rate [W], Total Heating Rate [W], Electric Lighting Rate [W] |

affect how darker the TCG gets. Thus, the optimization variables must describe different states for the properties of the Thermochromic Glazing (TCG). One where the TCG is clear which has high transmittance $T_{solmax}$. Other where the TCG is darkening throughout the hysteresis process with a minimum temperature of transition $T_{min}$, and $T_{max}$ with different values for transmittance between a $T_{solmax}$ and a $T_{solmin}$. Finally, the final state describes the TCG fully dark to its maximum for In Table 4.15, this is visible at temperatures of 5 °C, where the TCG has a transmittance $T_{sol}$ of 0.69, while at temperatures of 85 °C this value drops to 0.26. Energy Plus then simulates this process according to the specified climate weather files' temperatures for a temperature $T$.

$$f_1 \;(T_{min},\; T_{max},\; \tau_{solmax}, \tau_{solmin}) = min(\textit{HVAC})\ \textit{kWh/m}^2 \tag{4.14}$$

$$f_2 \;(T_{min},\; T_{max},\; \tau_{solmax}, \tau_{solmin}) = min(\textit{L})\ \textit{kWh/m}^2 \tag{4.15}$$

### 4.3.2 Building Information Database

This study aims to understand and report how different TCG coating properties impact an office room's climatization and electric lighting energy needs. Thus, the main aim is to find the theoretical TCG properties that provide the case study's yearly minimum heating, cooling, and electric lighting energy use ($f_1$ in Equation 4.14 and $f_2$ in Equation 4.15). To speed up the optimization process, a SM is developed using an Iterative Building Database that leverages simulation-based optimization to compile a Building Information Database (BID) consisting of the proposed MOO problem variables and respective simulation results. This database can be described by Equation 4.16 for $i$ simulation-based iterations of the MOO.

$$f\left(\begin{bmatrix} T_{min} & T_{max} & \tau_{solmax} & \tau_{solmin} \\ \vdots & \vdots & \vdots & \vdots \\ T_{min_i} & T_{max_i} & \tau_{solmax_i} & \tau_{solmin_i} \end{bmatrix}\right) = min\left(\begin{bmatrix} \mathit{HVAC} & L \\ \vdots & \vdots \\ \mathit{HVAC}_i & L_i \end{bmatrix}\right) \mathrm{kW\,h\,m^{-2}} \tag{4.16}$$

To explore the decision variables, a function that approximates $\tau_{sol}$ and $\tau_{vis}$ values for different TCG surface temperatures is modeled (Equation 4.17). This is described according to the initial($T_{min}$) and final ($T_{max}$) temperatures of transition, and the respective minimum and maximum $\tau_{sol}$ that the TCG system can reach. This describes a linear transmittance variation during the thermochromic phasing [316]. Additionally, it comprises a static behavior when the glazing surface is below or over the temperature thresholds for each state. With this function (Equation 4.17), it is possible to approximate multiple variations of TCG systems with different properties.

$$\tau_{sol}\ (T,\ T_{min},\ T_{max},\ \tau_{solmax}, \tau_{solmin}) = \begin{cases} \tau_{solmax} & \text{if } T \leq\ T_{min} \\ \tau_{solmax} - \frac{\tau_{solmax}-\tau_{solmin}}{(T_{max}-T_{min})} \times (T - T_{min}) & \text{if } T_{min} < T < T_{max} \\ \tau_{solmin} & \text{if } T \geq\ T_{max} \end{cases} \tag{4.17}$$

$$T \in \{5, 15, 25, 45, 65, 85\},\ T_{min} \in [0, 95],\ T_{max}\ \in [0, 95],\ \tau_{max} \in [0.1,\ 0.9],\ \tau_{min} \in [0.1,\ 0.9]$$

For the simulation process, temperatures 5, 15, 25, 45, 65, and 85 ºC were considered for the TCG optical data establishment. Thus, for the MOO process, Equation 4.17 was applied to calculate the approximated $\tau_{sol}$ and $\tau_{vis}$ at the given temperatures. The decision variables are $T_{min}$, $T_{max}$, $\tau_{solmax}$, and $\tau_{solmin}$, and they can vary from 0 to 95 ºC for $T_{min}$ and $T_{max}$, and 0.1 to 0.9 for $\tau_{solmin}$ and $\tau_{solmax}$. For the $\tau_{vis}$ values, the same equation (Equation 4.17) was used by replacing $\tau_{solmin}$ and $\tau_{solmax}$ for $\tau_{vismin}$ and $\tau_{vismax}$, but with $\tau_{vismin}$ = 0 at all conditions.

Finally, the considered optimization constraints assure that $T_{min}$ is not larger than $T_{max}$, and $\tau_{solmin}$ than $\tau_{solmax}$. Figure 4.21 illustrates Equation 4.17 variables and compares the previously simulated TCG transmittance in Table 4.15 with the approximated TCG transmittance. Moreover, it describes the transition temperature ($T_c$) and transmittance variation ($\Delta\tau_{sol}$) in a TCG system. Results for this approximated TCG show errors of $\approx$ 0.05 $\mathrm{kW\,h\,m^{-2}}$and $\approx$ 0$\mathrm{kW\,h\,m^{-2}}$, for heating and cooling and electric lighting energy use in Copenhagen's office room; and $\approx$ 0.05 $\mathrm{kW\,h\,m^{-2}}$and $\approx$ 0.1 $\mathrm{kW\,h\,m^{-2}}$for Lisbon's, which shows minimum errors with the approximated values.

Since each Energy Plus TCG simulation is time-consuming (with high-resolution settings can take up to a minute), to explore a vast solution space within these variables and constraints thousands of simulations would have to be run, which would render the process highly time-consuming, or even unfeasible. As such, this BID is developed by 500 iterations performed by four metaheuristics algorithms, two from the evolutionary class of algorithms [319], and two from the particle swarm class [320]. Particularly, the NSGAII, NSGAIII [321], OMOPSO [143], and SMPSO [142] algorithms were applied to the

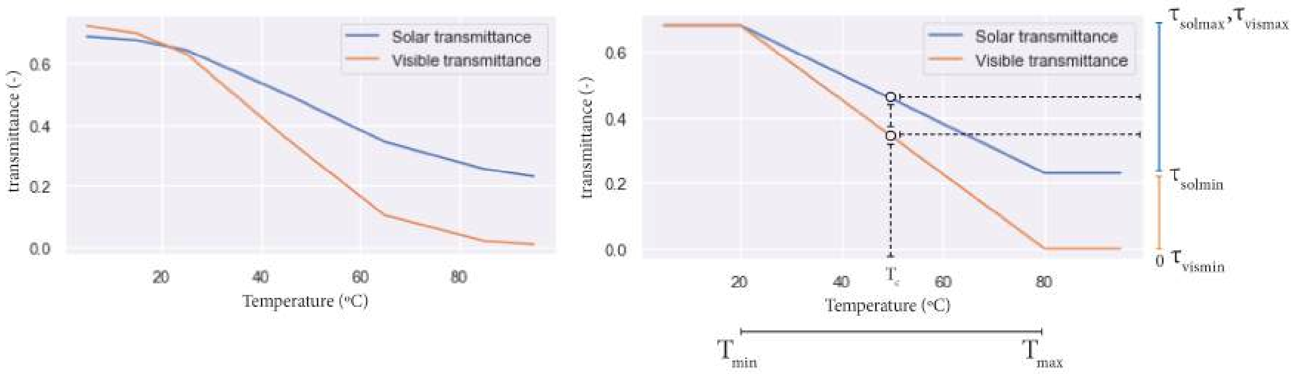



Figure 4.21: $\tau_{sol}$ and $\tau_{vis}$ values of Thermochromic glazing at different temperatures for one simulation (Left). Approximated $\tau_{sol}$ and $\tau_{vis}$ values for simulations for the optimization problem (Right).

studied optimization problem. To evaluate the optimization performance the algorithms' non-dominated solutions are calculated, which are solutions that cannot improve more in one objective, without harming the other [147].

After a total of 2000 iterations, the results, and variables are compiled into a BID to train a Supervised Learning algorithm [322] capable of building an accurate surrogate model [247, 23, 22, 24] to predict the results for the given variables significantly faster than a simulation. With the surrogate model, thousands of iterations can be performed, and optimal results are obtained at an incomparably lower computational cost than with a conventional simulation-based optimization.

### 4.3.3 Model Type

For the SM development to predict the objective functions, Machine Learning (ML) techniques were adopted to approximate $f_1$ and $f_2$ values (Equation 4.14 and Equation 4.15), which can be described as regression problems. Particularly, the supervised learning algorithm Extra Trees Regressor [322, 293] was used with SciKit-Learn package for Python [252] with an off-the-shelf computer desktop unit. A total of 2000 iterations were performed with the different metaheuristics, each with 500 iterations in which the tested solution variables described in Equation 4.17. The results are merged into a BID and split into a training and test set respectively with 70% and 30% of the 2000 total samples.

To evaluate the surrogate model accuracy the test set's coefficient of determination Coefficient of determination ($R^2$), Root Mean Squared Error (RMSE), and the elapsed time to predict the test set were documented. An error distribution between predicted and simulated results of the test sample is plotted to understand the confidence of predictions for both climates (Figure 4.22). From the error distribution plot illustrated, it is visible that the regression model for Copenhagen's climate had significantly smaller errors than Lisbon's climate, where errors were between -1 and 1 $\mathrm{kW\,h\,m^{-2}}$. However, all models have shown acceptable accuracy with at least more than 150 instances predicted with an error of $\approx 0$.

Table 4.17 shows similar results for both climates regarding the surrogate model accuracy. However, Root Mean Squared Error (RMSE) results show slight deviance for both climates. It can be observed that heating and cooling energy use has a Root Mean Squared Error (RMSE) of 0.19 and 0.05 $\mathrm{kW\,h\,m^{-2}}$ for Lisbon and Copenhagen respectively, while electric lighting energy use has a Root Mean Squared

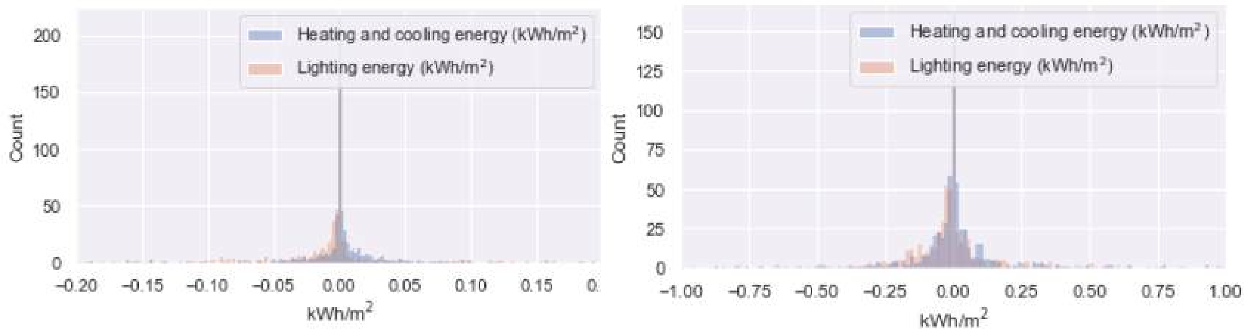



Figure 4.22: Error value distribution plot for the test sample in Copenhagen's climate (Top), and Lisbon (Bottom).

Error (RMSE) of 0.28 and 0.16 $\mathrm{kW\,h\,m^{-2}}$. This can be explained by the poorest performance of TCG systems in cold climates. Since TCG systems have less impact in cold climates, the values of $f_1$ and $f_2$ have smaller amplitudes. Thus, with similar performance levels, Root Mean Squared Error (RMSE) values will be smaller for the cold climate. Finally, it is possible to state that the regression accuracy is outstanding, and the accuracy loss with the SM predictions combined with the elapsed regression times significantly outperforms the simulation approach. It is possible now to perform thousands of iterations without the need to perform any simulation, with minimum errors.

Table 4.17: Regression model scores for both climates.

| | **Lisbon** | | **Copenhagen** | |
|---|---|---|---|---|
| **Objective** | f1 | f2 | f1 | f2 |
| **r$^2$ score** | 0.99 | 0.98 | 0.98 | 0.99 |
| **RMSE[kWh/m$^2$]** | 0.19 | 0.28 | 0.05 | 0.16 |
| **Elapsed time [s]** | 0.2 | 0.2 | 0.2 | 0.2 |

### 4.3.4 Model Deployment

BPS results for the original TCG show hourly heating and cooling rates, and the different climatization (*H*) energy use for these different climates are illustrated in Figure 4.23. This energy use is calculated by considering the office's equipment energy needs affected by the SCOP/SEER of 4.43/7.98 respectively, as described in [315]. Copenhagen's office hourly rates of *HVAC* hit maximums of 500 W, particularly during winter. Whereas in Lisbon hourly rates for *HVAC* reach maximums of 300 W during summer. Additionally, electric lighting rates for Copenhagen are superior to those for Lisbon. Finally, the office room in Copenhagen requires a yearly total of 14.45 $\mathrm{kW\,h\,m^{-2}}$for *HVAC*, 3.90 $\mathrm{kW\,h\,m^{-2}}$for $L$, and a total of 18.40$\mathrm{kW\,h\,m^{-2}}$. Whereas Lisbon respectively requires 9.70$\mathrm{kW\,h\,m^{-2}}$, 3$\mathrm{kW\,h\,m^{-2}}$, and a total of 12.7$\mathrm{kW\,h\,m^{-2}}$.

For the final optimization, the proposed SM was integrated with the MOO problem defined in Subsection Problem Description, Equations 4.14 and 4.15. The optimization objectives were to minimize both electric lighting energy use $L$ and heating and cooling energy use *HVAC*. To do that, 10000 iterations of different TCG properties were performed with the NSGAII algorithm for each climate.

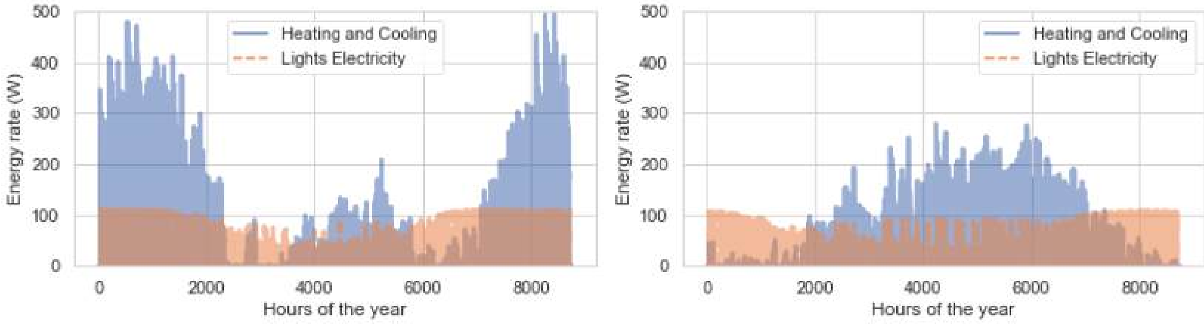



Figure 4.23: Hourly heating and cooling, and Electric lighting Rate for Copenhagen (left) and Lisbon (right).

Figure 4.24 shows the plot with all the tested solutions according to the results of each objective on the x-axis (electric lighting energy use) and y-axis (*HVAC*), and the sum of the two objectives (total energy use) on the z-axis. By looking at the graphs of both climates, a clear conflict among objectives for different TCG properties is visible. It is noticed that minimum *HVAC* both in Lisbon and Copenhagen's climate correspond to maximum $L$ values. Therefore, it is possible to conclude that modeling TCG production for a single objective, such as heating and cooling energy use, does not necessarily represent the optimal TCG solution that will save the most energy. Furthermore, it is visible that the absolute minimum value for the total energy use of the office room is represented by a balanced performance regarding *HVAC* and $L$. When comparing both climates' results, it was noticeable that there is a higher amplitude of total energy use values for the Lisbon climate than for Copenhagen's ranging between from 11 to 16 $\mathrm{kW\,h\,m^{-2}}$and 17 to 18 $\mathrm{kW\,h\,m^{-2}}$respectively.

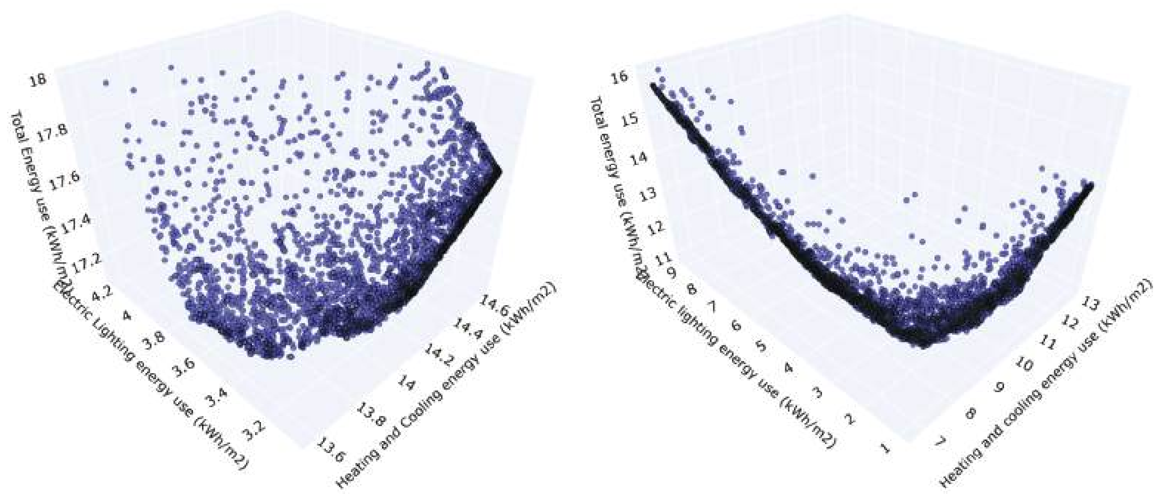



Figure 4.24: Tested solutions with NSGAII for Copenhagen (left) and Lisbon (right). Results for each objective (x- and y-axis) and the sum of both objectives (z-axis).

From these results, the TCG solutions that yielded the minimum values for $f_1$, $f_2$, and their sum were selected and compared with the original TCG system (Figure 4.25). It is visible that Copenhagen's climate TCG optimization yielded almost no significant results when compared to the original, with improvements of $\approx$ 7% of the total energy use for the TCG with minimum $f_1 + f_2$, $f_1$, and $f_2$. Results are noticeably different for the Lisbon climate since the TCG optimization shows more sensitivity to the optimization goals. For the minimum $f_1 + f_2$, the optimum TCG solution obtained improvements of $\approx$ 15% against the original system with 10$\mathrm{kW\,h\,m^{-2}}$. Moreover, the highest total energy use is represented by the TCG optimized for $f_1$ which causes a significant increase of 200% (from 3 to 9$\mathrm{kW\,h\,m^{-2}}$) in the electric lighting energy use of the office. Finally, the optimum TCG for minimum $f_2$ yields a total energy use of 13.9 $\mathrm{kW\,h\,m^{-2}}$ from which $\approx$ 90% was for *HVAC*.

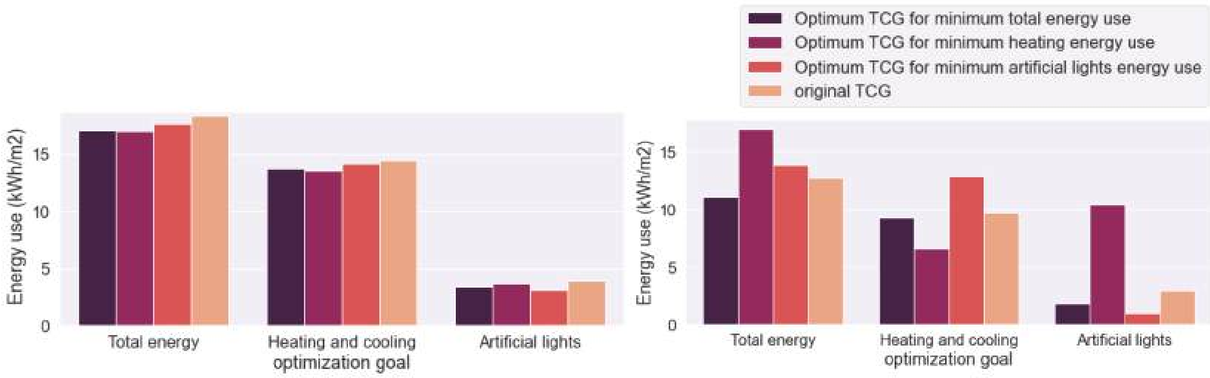



Figure 4.25: Office energy use for the Optimum thermochromic glazing of the proposed goals. Comparison with the original in Copenhagen (left) and Lisbon climate (right).

To better understand the TCG properties variation for these objectives results, the Pareto front with non-dominated solutions was illustrated according to their $T_c$ in a heatmap from 0 to 80 ºC, and according to their $\Delta\tau_{sol}$ illustrated in the size of the plot markers (Figure 4.26). This figure shows that for the cold climate of Copenhagen, lower $T_c$ values provide smaller thermal energy use, while higher $T_c$ values represent minimum lighting energy use since the TCG is switching to a darker reflective state at higher temperatures. Additionally, minimum total energy consumption occurs for TCG systems with small $T_c$ values but not the smallest. Finally, it can be noticed that all optimal TCG systems for Copenhagen have high $\Delta\tau_{sol}$.

For the climate of Lisbon, more results are worth emphasizing. Not only do the total energy consumption values have higher amplitude, but also the inherent trade-offs between thermal and lighting energy use according to different glazing systems are significantly more visible. Particularly, it is visible that higher $T_c$ and $\Delta\tau_{sol}$ values for TCG solutions provide minimum lighting energy use but maximum thermal energy use, while lower $T_c$ and $\Delta\tau_{sol}$ values provide minimum thermal energy use but maximum lighting energy use. As seen, both these heuristics do not represent minimum total energy consumption. For Lisbon's case, it is observed that minimum total energy use is represented by solutions with balanced values for both $T_c$ and $\Delta\tau_{sol}$.

To take an individual look at some TCG optimal solutions,one with the minimum *HVAC*, $L$, and total energy consumption for both climates were selected and plotted for their respective $\tau_{sol}$ and $\tau_{vis}$ according to their surface temperature (Figure 4.27). Figure 4.27A, B, and C illustrate the three optimal

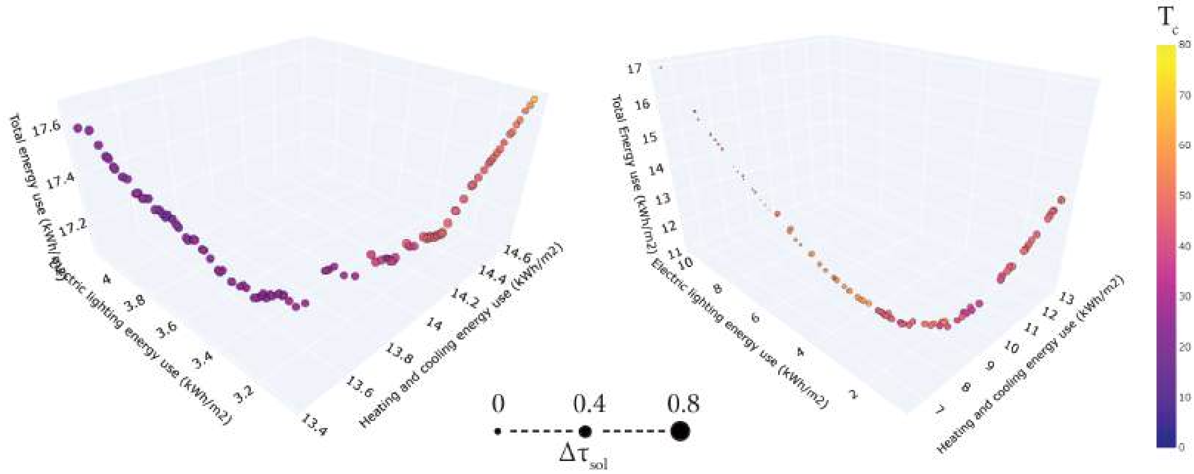



Figure 4.26: Pareto front plot of non-dominated solutions of the optimization problem for Copenhagen (left) and Lisbon climate (right). Results are mapped according to their $T_c$ and $\Delta\tau_{sol}$ values in color and size, respectively.

solutions for the climate of Copenhagen, and D, E, and F for the climate of Lisbon. Figure 4.27D shows the solution that provides the office room in Lisbon's climate with minimum heating and cooling energy use. It is visible that both the $\tau_{sol}$ and $\tau_{vis}$ have extremely low values at all temperatures. This indicates that the TCG would provide the minimum thermal energy use, but electric lighting dependence would increase for the office users to have the required 500 lux for their activities. Figure 4.27E shows the solution that provides the minimum total energy consumption. It is visible that the minimum energy use is provided by a $\tau_{solmax}$ and $\tau_{vismax}$ of $\approx$ 0.6 that transition to $\approx$ 0.3, and 0 from $\approx$ 25 to $\approx$ 65 ºC. These values grant the optimal balance between thermal and lighting energy use that provides the minimum total energy consumption. Figure 4.27F shows the solution that yields minimum lighting energy use. In this plot, it is noticed that the initial temperature of transition is $\approx$ 45 ºC, which is significantly higher than Figure 4.27D or Figure 4.27E. The final transition temperature of $\approx$ 90 ºC, and $\tau_{solmax}$ and $\tau_{vismax}$ are higher than D or E since higher $\tau_{sol}$ and $\tau_{vis}$ values increase daylighting gains.

These trade-offs are marginally less perceptible in cold climates, but they are still present. Figure 4.27A shows the optimal solution that yields minimum heating and cooling energy use for Copenhagen's climate. This solution shows initial and final transition temperatures of $\approx$ 15 ºC and $\approx$ 45ºC respectively. Additionally, $\tau_{sol}$ and $\tau_{vis}$ values range from $\approx$ 0.9 to $\approx$ 0.2, and from $\approx$ 0.9 to 0, respectively. Figure 4.27B shows the solution that returns the minimum total energy consumption. It is visible that it has the same temperature ranges as Figure 4.27A, but its $\tau_{sol}$ and $\tau_{vis}$ values range from $\approx$ 0.9 to $\approx$ 0.3, and from $\approx$ 0.9 to $\approx$0 respectively. Finally, Figure 4.27C represents the optimal solution that returns the minimum $L$. It is visible that Figure 4.27C has a slope similar to Figure 4.27F but a lower initial transition temperature of $\approx$ 15 ºC (same as A and B), and a final transition temperature of $\approx$ 82ºC. $\tau_{sol}$ and $\tau_{vis}$ ranges are similar to plots D and E except for the final $\tau_{sol}$ value of $\approx$ 0.35.

As seen, $T_c$ values and steeper $\tau_{sol}$ variations do not represent a smaller energy consumption for both climates. Particularly for cold climates, a $T_{min}$, $T_{max}$, $\tau_{solmin}$, and $\tau_{solmax}$ values of 15 ºC, 45ºC,

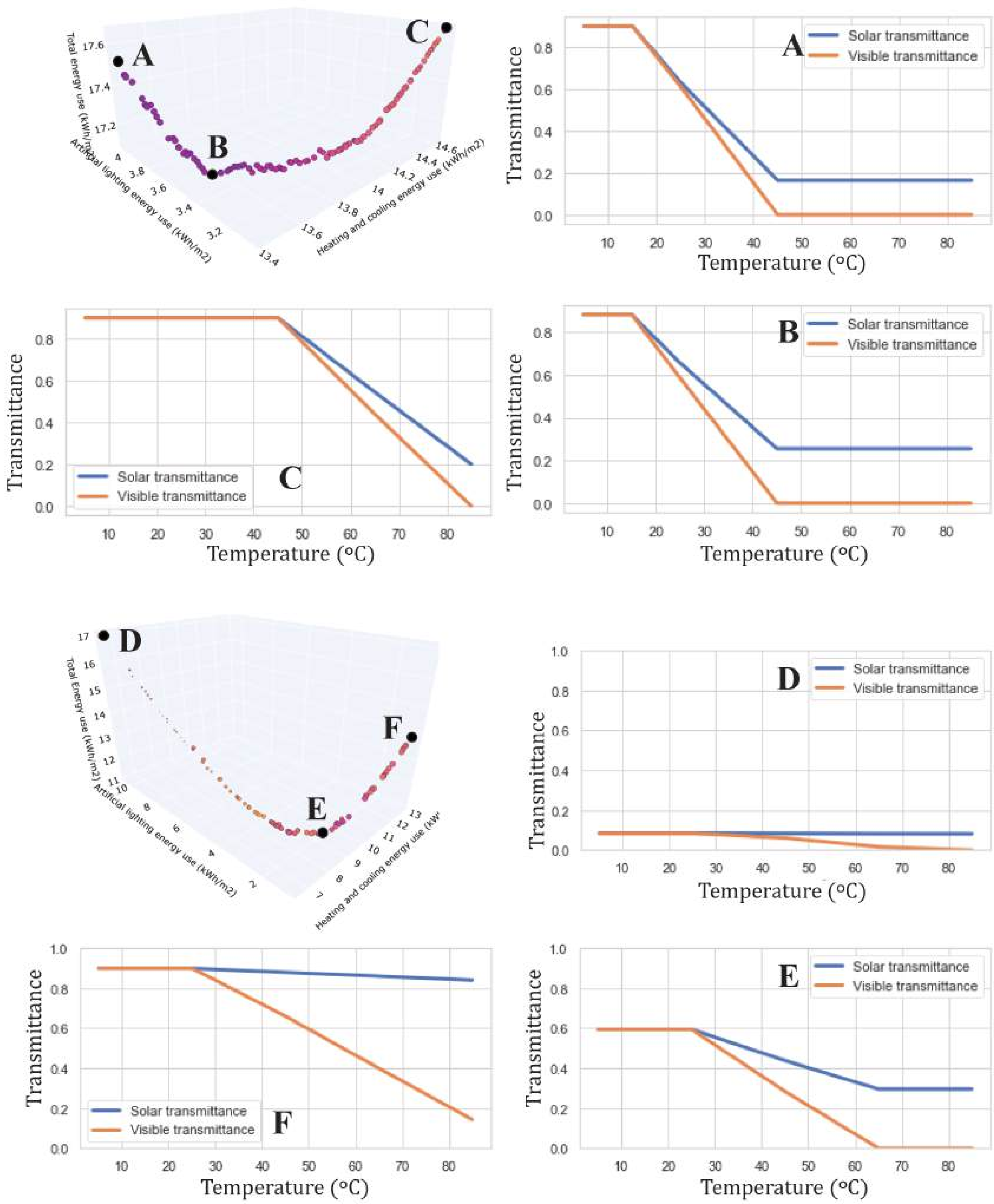



Figure 4.27: Thermochromic glazing properties for each optimal solution in both climates. Copenhagen: A) minimum heating and cooling needs, B) Minimum total energy needs, C) Minimum lighting needs; Lisbon: D, E, F, respectively.

0.9, and 0.2 respectively, yield the minimum total energy use. Whereas for warm climates, the minimum total energy use is obtained with a $T_{min}$, $T_{max}$, $\tau_{solmin}$, and $\tau_{solmax}$ values of 25ºC, 65ºC, 0.6, and 0.23

respectively.

### 4.3.5 Discussion

The added value of this study lies in (1) the MOO approach of the TCG and (2) the use of SM to speed up the simulation, making it easier, and more portable.

For (1), TCG properties studied at this extension represent advances in the literature. As an example, Saeli et al. [311] test 4 TCG solutions with different sizes and properties for 8 climates. Results show a better energy performance in warmer climates with lower $T_c$ (38.5 ºC). [317] performed a parametric sensitivity analysis to find the best TCG properties. The authors perform a parametric study of 4 different $T_c$ values and 4 different hysteresis widths in 3 different climates. Results show the best energy performance in warmer climates for a TCG with low $T_c$ (35 ºC) and a narrow switching temperature range.

The present study goes one step further since it explores thousands of possible properties for theoretical TCG with the MOO approach. In agreement with the previous findings, results show a better energy performance for warmer climates. However, not also the optimum $T_c$ (45 ºC) value was slightly higher, but also the switching temperature range was wider than previous findings. This can be explained by the vast theoretical properties explored with the MOO approach. This identified solutions with slightly lower $\tau$ that gradually switched to darker states. Thus, it did not activate the 500 lux lighting set-point as fast and contributed to savings in lighting energy without compromising thermal comfort.

For (2), the use of the SM to predict the energy performance of theoretical TCG properties made it possible to considerably speed up the simulation times. Therefore, the optimization algorithms were able to converge to optimum solutions requiring significantly less computational time. Additionally, the reduction of required features to predict the proposed objectives makes it easier to perform and report additional Analysis and Optimization Processes (AOP).

Overall, the obtained results show that TCG optimal solutions have different properties for different climates and could benefit from benchmarking these different performances to help manufacturers tailor the TCG for local climates. The observed increase of 200 % in the electric lighting energy needs could be explained by the optimum theoretical TCG for $f_1$ consisting of a significantly low $\tau_{sol}$ and $\tau_{vis}$ that is constantly darkened, thus contributing to significantly higher lighting energy needs.

The following limitations could have influenced the obtained results in Subsection Model Deployment as well as constrain the conclusions drawn: climates studied; geometry and location of the case study; and all simulation inputs included in Table 4.16. Additionally, the indoor daylighting illuminance was calculated with Energyplus [323] that uses the split-flux method [53], which is suitable to assess rooms with a cubical shape and no internal partitions such as the case of the office room in this study. This method is used during the simulation to control the electric lighting by estimating the illuminance levels at a single reference point. Other advanced simulation methods (e.g., Radiance) should be used to perform a more detailed spatial daylighting analysis.

In summary, this study illustrates the proposed innovative framework for SM development with an Iterative BID. The resulting model required only four inputs and managed to speed up the energy simulation from 30 to 0.08 seconds. The model was implemented in an optimization problem that found the inherent trade-offs between the optimization objectives and successfully found an optimum solution for two multiple climates. This case study shows how the proposed framework tackles the identified research questions (Chapter 1, Section 1.3) for more complex AOP by considerably reducing the simulation speed and complexity of the AOP.

## 4.4 Optimization of a residential block using a Surrogate Model developed with an Iterative Building Database

To obtain information regarding the building stock and define strategies to improve it, Building Performance Simulation (BPS) tools can help users to predict building design and rehabilitation impacts in multiple aspects of a building's performance [324]. BPS tools predict these results through models described by specific inputs that yield the desired outputs. With these simulations, it is possible to integrate Analysis and Optimization Processes (AOP) into building design, planning, and rehabilitation processes [122].

Unfortunately, the use of BPS tools with AOP is still out of reach for most practitioners and presents several limitations. AOP typically entail multiple objectives, which are often conflicting [122]. Additionally, most building performance indicators are outputs of BPS or extensive calculations performed by field experts [16, 24]. Because of this, optimization problems that are focused outside the practitioner's realm of knowledge are usually treated as MOO problems with derivative-free functions [122, 125, 325, 138]. Metaheuristics, a class of optimization algorithms, have been widely used in the field of optimization for the built environment with positive results.

These algorithms guide the search based on biological heuristics such as evolutionary ones and swarms of different kinds, among others [137, 326, 321, 142]. Furthermore, Wolpert and Macready [145] state with the ”No Free Lunch” theorem that no algorithm outperforms all others for all problems. This means that one has to either know from experience which algorithms yield better performance for each optimization problem or test multiple metaheuristics to find the best one [327]. Moreover, these algorithms have variables that define how they search for the optimal space, called hyperparameters. These need to be fine-tuned to provide optimum results [218]. To fine-tune optimization algorithms, one must optimize the combination of hyperparameters that yield the best indicators of their performance. The most commonly used evaluation metric for multi-objective optimization algorithms is the Hypervolume (H) [149, 150, 148].

With this in mind, this study documents the development of a Surrogate Models (SM) to solve a Multi-Objective Optimization (MOO) problem that requires time-consuming BPS for its objectives. The developed SM can not only speed up simulation times by predicting the BPS results much faster, but also reduce the number of parameters and settings required to perform a simulation. the MOO problem consists of finding the best combination of construction materials for the energy-efficient construction of a mixed-use building block with 6 buildings in Lisbon, Portugal.

This study's workflow is illustrated in Figure 4.28. Initially, an iterative Building Information Database (BID) is developed with Eppy [101] and Energy Plus [32]: This approach is used since a detailed energy simulation of 6 buildings can be time-consuming, and because the model's final deployment consists of a specific optimization problem. After the Building Information Database (BID) development, the SM parameters are fine-tuned with an Single-Objective Optimization (SOO) to maximize the model's Coefficient of determination ($R^2$). Finally, the model is deployed to optimize optimization algorithms hyperparameters, test different algorithms, and solve the MOO problem with the best performing optimization algorithm.

The BID development and MOO problem are set by describing the optimization objectives and decision variables (Subsection Problem Description). The objectives comprise the minimum total energy use of the block, minimum construction costs, and minimum standard deviation of energy needs among the buildings that compose the block. The variables are described by each building's construction materials for walls, floors, roofs, and windows. Afterward, in Subsection Building Information Database, a small simulation-based optimization is performed with different algorithms to compile the SM training database. This database is used in Subsection Model Type to train a Artifical Neural Networks (ANN) with the Keras and TensorFlow packages [210], capable of predicting the total energy use of the 6 buildings and their standard deviation.

In the Model Tuning Subsection, both the ANN layers' nodes and the optimization algorithms' hyperparameters, are optimized with a Bayesian optimization. Finally, the optimized ANN is deployed with the optimized optimization algorithm to generate thousands of iterations in the Model Deployment Subsection. Results are discussed and compared in the Discussion Subsection.

The BID development and MOO problem are set by describing the optimization objectives and decision variables (See Problem Description). The objectives comprise the minimum total energy use of the block, minimum construction costs, and minimum standard deviation of energy needs among the buildings that compose the block. The variables are described by each building's construction materials for walls, floors, roofs, and windows. Afterward, in the Building Information Database Subsection, a small simulation-based optimization is performed with different algorithms to compile the SM training database. This database is used in the Model Type Subsection to train a Artifical Neural Networks (ANN) with the Keras and TensorFlow packages [210], capable of predicting the total energy use of the 6 buildings and their standard deviation.

Finally, in the Model Tuning Subsection, both the ANN layers' nodes and the optimization algorithms' hyperparameters, are optimized with a Bayesian optimization. The optimized ANN is deployed with the improved optimization algorithm to generate thousands of iterations in the Model Deployment Subsection. Results are then discussed and compared in the Discussion Subsection.

### 4.4.1 Problem Description

The proposed MOO problem is to find the best combination of constructions for each building's surface type that minimizes the construction cost and total energy use, and maximizes fairness of performance among buildings. The case study comprises a standard midrise apartment building program from LadybugTools for building operation, usage, and schedules [100]. This program is applied to a geometric model of 6 buildings in Lisbon, Portugal, illustrated in Figure 4.29. LadybugTools integrates the Rhino3D tool with EnergyPlus by translating geometrical representations and weather files into a file folder and format that is readable by EnergyPlus. In this case study, a simulation of each building's total heating and cooling energy (J) is performed with an ideal air load system for one year with residential occupation schedules (Table 4.18). The resulting Energy Plus file can then be used with Eppy to iterate over

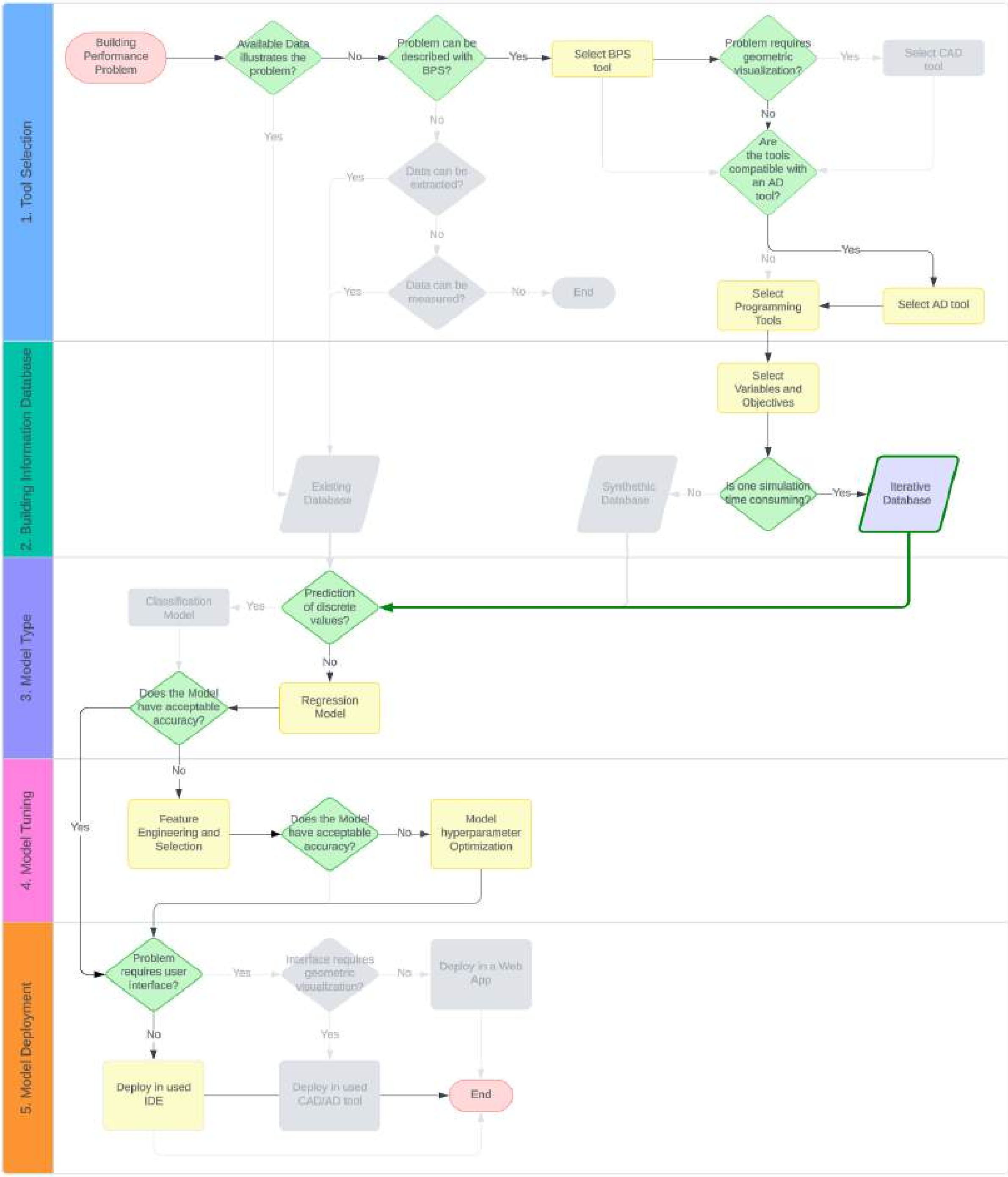



Figure 4.28: Workflow diagram for Section 4.4.

construction materials and simulate its results.

Each building can have one out of three different constructions for each surface type: exterior walls, interior floors, roofs, and windows. Thus, the optimization problem comprises the walls, roofs, floors, and windows of 6 buildings, with a total of 24 variables with 3 possible constructions each (Tables 4.19 and 4.20). The opaque materials' properties are represented from their outermost layer to their innermost, and the window materials are defined using a simple glazing system definition with the window U-value, solar ($\tau_{sol}$), and visible transmittance ($\tau_{vis}$).

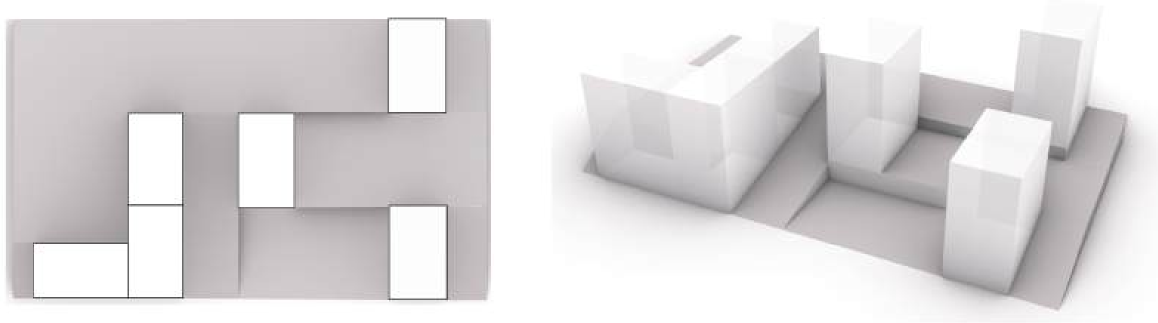


Figure 4.29: A 3D model and plan of a 6-building urban complex.

Table 4.18: Simulation setting values for the proposed case study.

| **Simulation Setting** | **Value** |
|---|---|
| Period | 1 year |
| Timestep | 1 timestep per hour |
| Program and schedules | Midrise apartment |
| Window-to-wall ratio | 0.2 |
| Outputs | Zone Ideal Load Supply Air Total Heating Energy (J)<br>Zone Ideal Load Supply Air Total Cooling Energy (J) |

The optimization variables and goals are described in Equations (4.18) to (4.21). $E_n$ (Equation (4.18)) represents the total energy use of building $n$, $f_1$ (Equation (4.19)) is the total energy use of all buildings, $f_2$ (Equation (4.20) is the standard deviation of the building sample, and $f_3$ (4.21) is the total cost of construction. The MOO problem objectives are to minimize these functions to guarantee the minimum total energy use ($f_1$), the best fairness of performance among buildings ($f_2$), and minimum costs ($f_3$).

$$E_n(x_0,\ ...,\ x_{n\times 4}) =\ Heating_n\ +\ Cooling_n\ [\mathrm{kW\,h\,m^{-2}}] \tag{4.18}$$

$$f_1\ (x_0,\ ...,\ x_{n\times 4}) = \sum E_n\ (x_0,\ ...,\ x_{n\times 4})\ [\mathrm{kW\,h\,m^{-2}}] \tag{4.19}$$

$$f_2\ (x_0,\ ...,\ x_{n\times 4}) = \sigma(E_n(x_0,\ ...,\ x_{n\times 4}))\ [\mathrm{kW\,h\,m^{-2}}] \tag{4.20}$$

$$f_3\ (x_0,\ ...,\ x_{n\times 4}) = Cost(x_0,\ ...,\ x_{n\times 4})\ [€] \tag{4.21}$$

$$x\ \in\ \{0,\ 1,\ 2\} - Number\ of\ construction\ types$$

$$n = Number\ of\ buildings$$

### 4.4.2 Building Information Database

The BID approach used in this study consists of a Iterative Building Database. As such, the SM predicts the objectives of a previously established MOO problem. Equation 4.22 illustrates the developed database. The training features are defined by each building's walls, floors, roof, and windows construction types (Table 4.19 and 4.20). As such, each set of 4 variables $x$ represents each building materials that compose the different construction types. This process repeats $n$ times for the number of buildings encompassed by this study, which is 6. Thus, the final BID has 24 training features, each with three possible categorical values.

$$f\left(\begin{bmatrix} x_0 & \dots & x_{n\times 4} \\ \vdots & \ddots & \vdots \\ x_{0_i} & \dots & x_{n\times 4_i} \end{bmatrix}\right) = min\left(\begin{bmatrix} f_1 & f_2 \\ \vdots & \vdots \\ f_{1_i} & f_{2_i} \end{bmatrix}\right) \tag{4.22}$$

The target features of this BID correspond to the objectives that require the use of BPS. Since the cost of construction can easily be calculated without any time-consuming process, the only objectives requiring a SM prediction are $f_1$ and $f_2$ (Equations 4.19 and 4.20). The optimization process is automated to simulate and compile the results into the database. The optimization algorithm chosen to explore the initial solution space was the NSGAII, which performed 1000 iterations spread through population sizes of 100. The resulting database is then split between model training and test sets at a 0.7/0.3 ratio.

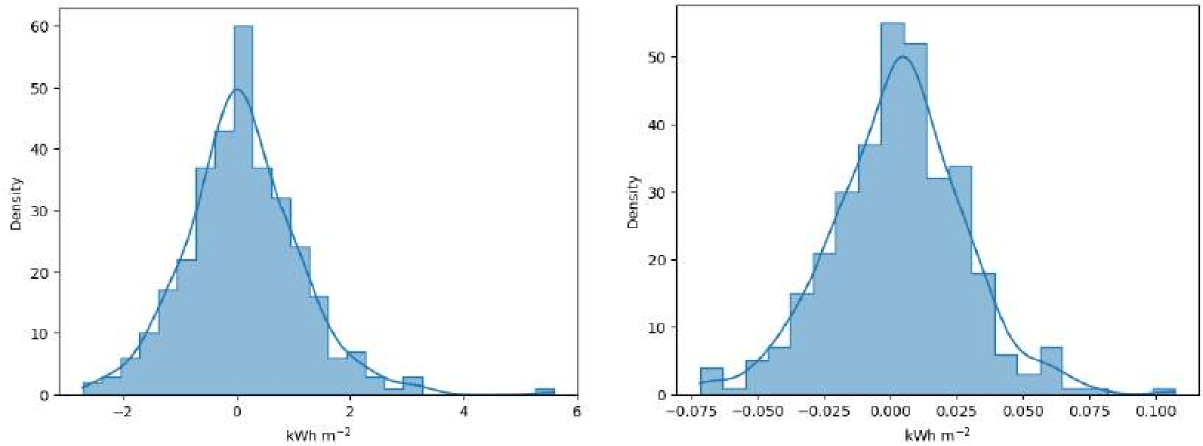
### 4.4.3 Model Type

Once the BID is compiled, it is possible to develop the SM. In this study, the chosen model was a sequential ANN with 1 dense layer, 4 convolutional layers, and 1 dense output layer. For the initial SM, arbitrary values can be set for each layer's filters (number of nodes). The first dense layer contained 25 filters, while the convolutional layers were set with 50 filters. The final layer contains only 1 filter for the desired output. The ANN filters and kernel sizes are described in Table 4.21.

The resulting SM presented an Coefficient of determination ($R^2$) of 0.84 and a Root Mean Squared Error (RMSE) of 1.11 $\mathrm{kW\,h\,m^{-2}}$ for the prediction of the total energy needs $f_1$. For the prediction of the standard deviation objective $f_2$, the model presented an $R^2$ of 0.69 and a RMSE of 0.03 [$\mathrm{kW\,h\,m^{-2}}$]. Figure 4.30 illustrates the error density in the test set for the prediction of both objectives. From these results, it is visible that the prediction accuracy of the models can be enhanced. In the Model Tuning Subsection, the developed ANN is optimized using hyperparameter optimization to find the best combination of filters per layer that yields maximum $R^2$.

### 4.4.4 Model Tuning

In this Subsection, both the SM and the optimization algorithms are optimized to yield the best prediction accuracy and results. Initially, the ANN's filters are optimized with a Bayesian Optimization to find the best combination of filters that provide the best $R^2$ [262, 220]. Afterward, the optimization algorithms and




(a) Error histogram (in kW h m$^{-2}$) of initial Artifical Neural Networks (ANN) predictions for total energy needs (Equation 4.19) among buildings.

(b) Error histogram (in kW h m$^{-2}$) of initial Artifical Neural Networks (ANN) predictions for Standard deviation of energy needs among buildings (Equation 4.20).

Figure 4.30: Initial error histograms for the developed models.

their hyperparameters are optimized to find the best H. In the next sections, the optimization results are showcased for both processes. Finally, the most suitable SM and optimization algorithm are selected for the model deployment stage.

#### Surrogate Model Hyperparameters

To optimize the layers' filters (number of nodes), the optimization process aimed to maximize the ANN R$^2$ between the test set simulation results and the model predictions (Equation (4.23)). Finally, this process was employed to train two ANN models capable of predicting $f_1$ and $f_2$ (Equations (4.19) and (4.20)), with an early stopping callback to avoid over-fitting. The optimized structure's final $R^2$ score and Root Mean Squared Error (RMSE) are documented in Tables 4.22 and 4.23. The SM obtained an $R^2$ score of 0.96 and 0.97, and an RMSE of 0.54 and 0.01 [kW h m$^{-2}$], for $f_1$ and $f_2$, respectively.

$$f_4\ (i_0, ..., \ i_n) = R^2\,(test,\ predictions) \tag{4.23}$$

$$i \in\ [6\,..\,300]\ \ - Number\ of\ filters$$
$$n = Number\ of\ layers$$

#### Optimization algorithms' hyperparameters

With the optimized ANN, it is now possible to use it to optimize multiple optimization algorithms and test their performance at the specified problem. The multiple metaheuristics selected in this study are presented along with their hyperparameters that were considered for the optimization problem. The problem is described, and the results are documented. Finally, the final optimization algorithm with optimal hyperparameters is selected for model deployment. The algorithms NSGAII and Indicator-Based Evolutionary Algorithm (IBEA) were selected from the evolutionary class, while SMPSO [142] and OMOPSO [328] were selected from the particle swarm class. All algorithms run for 500 iterations to aim for the best combination of hyperparameters that yield the best optimization performance. Table 4.24 documents each algorithm's considered hyperparameters and their specified range.

To evaluate an optimization algorithm's performance for a MOO, it is required to evaluate the non-dominated solutions found by the algorithm. These solutions are considered optimal since they cannot improve more within one objective, without harming the others. In this study, the non-dominated solutions actually represent the implicit trade-offs between the cost of construction, energy use of a set of buildings, and their fairness (Equations (4.19) to (4.21)). Thus, considering these objective domains, the most suitable optimization algorithm is one that explores the largest solution space. This is measured with the H metric [148, 149, 150]. In this case, the Hypervolume (H) is calculated using the boundary domains represented by the highest and lowest possible values for the minimum total energy use ($f_1$), standard deviation ($f_2$), and costs of construction ($f_3$).

The fine-tuning of the algorithms' hyperparameters can be described as a single objective optimization problem to find the combination of parameters that yields the maximum H in $f_{\text{NSGAII}}$, $f_{\text{IBEA}}$, $f_{\text{SMPSO}}$, and $f_{\text{OMOPSO}}$ (Equations (4.24) to (4.27)). Thus, a Bayesian optimization with 100 iterations is integrated to find the best combination of hyperparameters for each algorithm's settings running for 500 iterations.

$$f_{NSGAII}\ (P_s,\ S,\ P_m) = H(f_1, f_2, f_3) \tag{4.24}$$

$$f_{IBEA}\ (P_s,\ S,\ P_m) = H(f_1, f_2, f_3) \tag{4.25}$$

$$f_{SMPSO}\ (S_s,\ L_s\,i,\ S, P_m) = H(f_1, f_2, f_3) \tag{4.26}$$

$$f_{OMOPSO}\ (S_s,\ L_s\,i,\ S, P_m,\ \varepsilon) = H(f_1, f_2, f_3) \tag{4.27}$$

$$H - H;$$

$$Boundaries\ for\ !ht\ calculation:$$

$$f_1 \in\ [30,\ 60]\ -\ \text{kW h m}^{-2}$$

$$f_2 \in\ [0.25,\ 0.50]\ \ \text{kW h m}^{-2}$$

$$f_3 \in\ [400000,\ 900000] -\ €$$

After the Bayesian Optimization, the best hyperparameters are found, and the algorithms' non-dominated solutions are calculated and illustrated according to each objective. Afterward, the optimized algorithms' Hs are compared and the most suitable algorithm is selected for deployment.

The solution space explored by each algorithm is illustrated in Figure 4.31. It is visible that the algorithms obtained different results for the exploration of the objective space. Particle swarms explored a wider range of values, while IBEA focused more on one area of the solution space. NSGAII explored an acceptable area but failed to find solutions with higher total energy use ($f_1$, Equation (4.19)), a lower standard deviation ($f_2$, Equation (4.20)), and lower construction costs ($f_3$, Equation (4.21)), in contrast to the particle swarm algorithms. In total, the algorithms explored a solution space with values of $f_1$ between ≈45 and ≈55kW h m$^{-2}$, $f_2$ between ≈0.30 and ≈0.40kW h m$^{-2}$, and $f_3$ between ≈400,000 and

≈600,000 €.

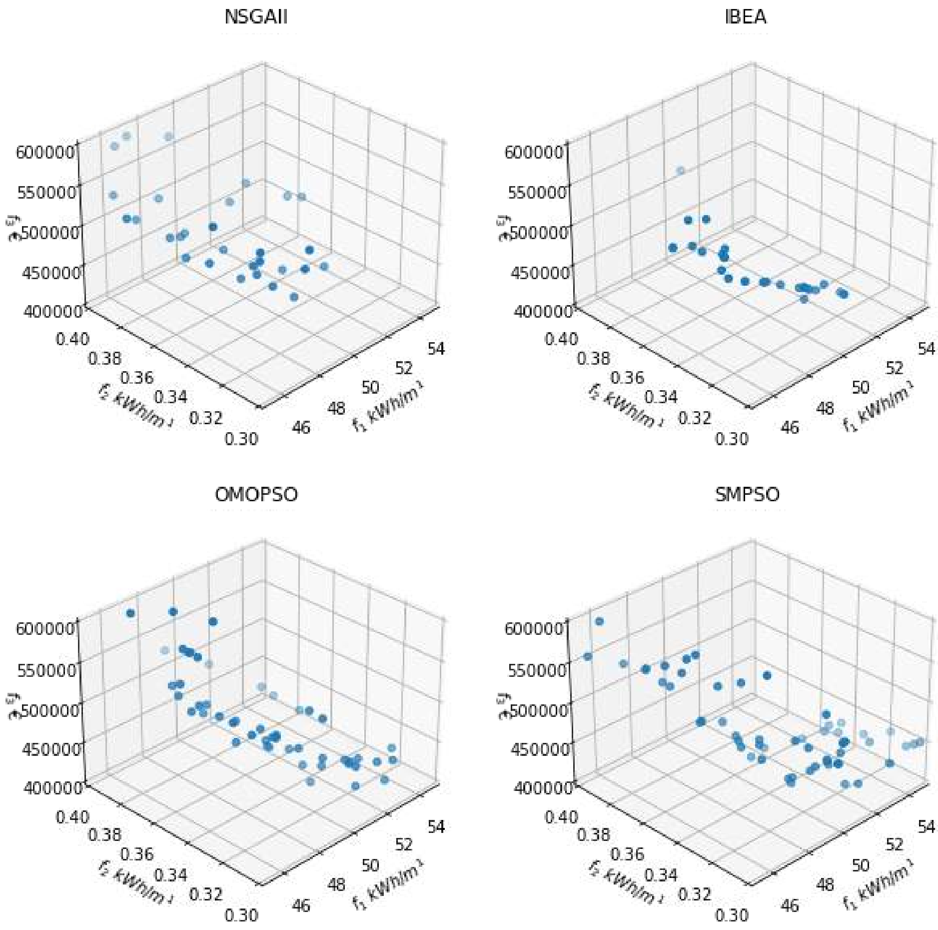



Figure 4.31: Plotted non-dominated solutions of optimized metaheuristics.

Table 4.25 documents the algorithms' Hypervolume (H) for the optimum values of their hyperparameters. The IBEA algorithm ($f_{IBEA}$, Equation (4.25)) obtained the best H with 0.54, followed by NSGAII, with 0.53 ($f_{NSGAII}$, Equation (4.24)), and SMPSO and OMOPSO, with 0.52 ($f_{SMPSO}$ and $f_{OMOPSO}$ Equations (4.26) and (4.27)). Additionally, it is visible in Figure 4.31 that the IBEA algorithm obtained the highest H because it focused on a specific area that maximized this metric, rather than exploring a wide solution space. Thus, IBEA successfully found better non-dominated solutions but ultimately failed in the exploration of the solution space of the buildings' inherent trade-offs. For these reasons, the NSGAII algorithm, which obtained the second highest H and explored a balanced solution space, was chosen to perform the final optimization.

### 4.4.5 Model Deployment

Once the model and the optimization algorithms are tuned, the Analysis and Optimization Processes (AOP) can now be deployed and solved. This is done in the Jupyter notebook Integrated Development Environment (IDE) [241]. The selected algorithm (NSGAII) performs a run of 10,000 iterations. Results show a H of 0.60, which is significantly larger when compared with the values obtained for the 500 iterations in Table 4.24. Additionally, Figure 4.32 shows that the algorithm explored values between

≈40 and 60kW h m$^{-2}$, ≈0.2 and 0.5kW h m$^{-2}$, and ≈400,000 and 900,000 €. The trade-offs between the objectives are fairly noticeable in the Figure, showing that higher costs ($f_3$) translate into better construction quality and less energy use ($f_1$). Conversely, lower construction costs show higher energy use and lower standard deviation ($f_2$), which implies that in this case, less fair solutions are obtained when the average $f_1$ value of the buildings decreases.

By analyzing the results, it is possible to find the best fitting solution with a minimum of $f_1$ and $f_2$ if a fixed budget for the construction is specified. Finally, Figure 4.32 shows how a solution with the most expensive construction for all surfaces is not necessarily the best solution, with the algorithm finding a significant number of cheaper solutions with lower $f_1$ and $f_2$. This solution enables savings of up to 22% (200,000 €) on the total cost of construction while maintaining the same performance ($f_1$) and fairness among buildings ($f_2$).

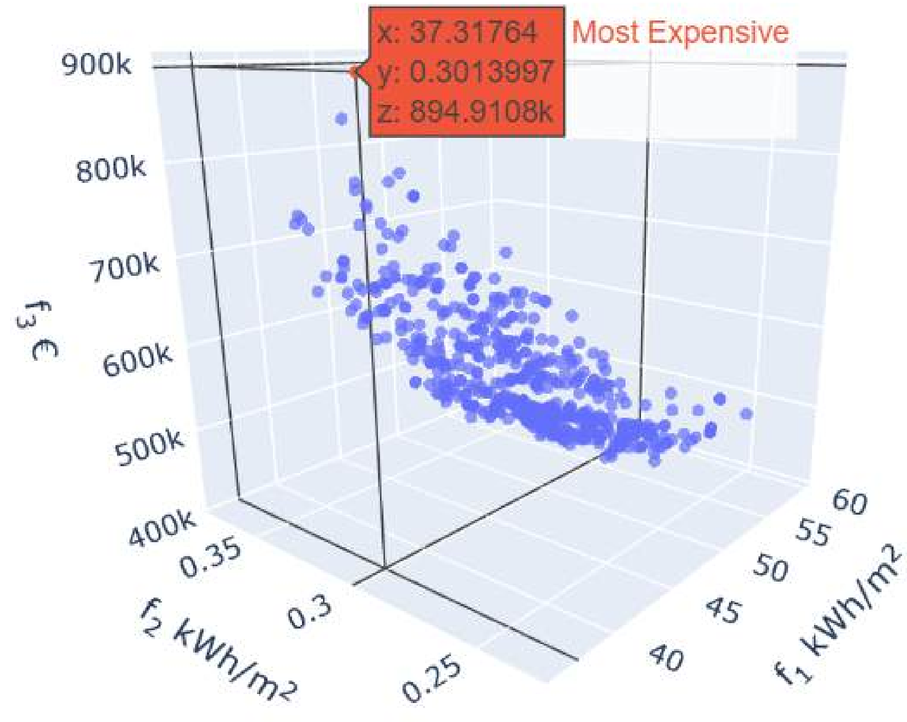



Figure 4.32: NSGAII non-dominated solutions for 10,000 iterations — the most expensive solution is represented in red.

### 4.4.6 Discussion

The use of a SM to estimate building energy use is significantly faster than using Building Performance Simulation (BPS). This increase in speed enables the application of suitable Analysis and Optimization Processes (AOP) that could otherwise take days,or months to complete with a high computational cost. Additionally, SM can be deployed in multiple platforms (e.g., web applications, programming environments, mobile applications, among others), and because they require a smaller number of inputs than simulations, they reduce the problem complexity and provide a more accessible analysis to users at any expertise level. In this case, an Iterative Building Information Database (BID) was used where the solutions tested by the optimization algorithm performed with BPS are used to train an Artifical Neural Networks (ANN). This model was used to optimize 4 different optimization algorithms, and deploy the most suitable for the specified problem.

The comparison of the optimization algorithms shows that, for this particular BPO problem, different algorithms explored different solution spaces. These can be more or less suitable for the practitioner's decision-making goals. Additionally, they show that the algorithm with the highest H obtained the best solutions, but did not explore the desired solution space. In particular, the PSO algorithms excelled in exploring a wide solution space, the IBEA algorithm obtained the highest H and best solution, and NSGAII showed an even balance between the H and solution space.

The comparison of the four MOO algorithms' H (Table 4.25) and solution spaces (Figure 4.31) provided a confident selection of the most suitable algorithm for the proposed optimization problem of minimizing the cost of construction of a 6-building block, the average energy use of the buildings, and maximizing the fairness of performance among the buildings. The obtained results show savings up to 22% of the total construction cost while maintaining a good energy performance. Additionally, they show that increasing construction costs are required for smaller average energy use, and that higher average energy use translates into more fairness of performance. This can happen due to the impact that parameters outside the scope of this analysis have on each building (e.g., glazing ratios, orientations, areas, among others). Thus, in the future, it is possible to use SM to perform extensive comparisons of optimization problems and algorithms for different aspects of building performance. This can provide a comprehensive foundation regarding common variable heuristics and the algorithms' performance.

In conclusion, this case study illustrates the proposed innovative framework for SM development with an Iterative BID. The resulting model predicts results with high accuracy, faster than a simulation (from a 20-second simulation time to a 0.02-second prediction time), and with only four input features for each building. Additionally, this study showcases the Model Tuning stage of the framework, particularly the hyperparameter optimization of the SM. Therefore, this study shows how SM can overcome the identified research challenges (Chapter 1, Section 1.3) for more complex problems by reducing AOP computational speed, and difficulty of implementation.

Table 4.19: Materials for each surface type construction solution (Part 1).

| Construction Type | Total Area $\mathrm{m}^2$ | x | Cost [€/m$^2$] | Materials |
|---|---|---|---|---|
| **Walls** | 8699.94 | 0 | 20 | Plaster 2 cm<br>Bored brick 11 cm<br>Air gap 6 cm<br>Bored brick 11 cm<br>Stucco 1.5 cm |
| | | 1 | 25 | Plaster 2 cm<br>Bored brick 15 cm<br>Air gap 6 cm<br>Bored brick 11 cm<br>Stucco 1.5 cm |
| | | 2 | 35 | Plaster 2 cm<br>Bored brick 11 cm<br>Air gap 6 cm<br>XPS 4 cm<br>Bored brick 11 cm<br>Stucco 1.5 cm |
| **Interior floors** | 9327.27 | 0 | 10 | Wood panels 12 cm<br>Stucco 1.5 cm |
| | | 1 | 25 | Ceramics 1 cm<br>Screed 8 cm<br>Lightweight slab 15 cm<br>Stucco 1.5 cm |
| | | 2 | 30 | Ceramics 1 cm<br>Screed 8 cm<br>Concrete slab 15 cm<br>Stucco 1.5 cm |
| **Roofs** | 1216.67 | 0 | 20 | Screed 8 cm<br>Waterproofing 0.2 cm<br>Screed 8 cm<br>Lightweight slab 15 cm<br>Stucco 1.5 cm |
| | | 1 | 30 | Screed 8 cm<br>Waterproofing 0.2 cm<br>XPS 4 cm<br>Screed 8 cm<br>Lightweight slab 15 cm<br>Stucco 1.5 cm |
| | | 2 | 35 | Screed 8 cm<br>Waterproofing 0.2 cm<br>XPS 4 cm<br>Screed 8 cm<br>Concrete slab 15 cm<br>Stucco 1.5 cm |

Table 4.20: Materials for each surface type construction solution (Part 2).

| Construction Type | Total Area $\mathrm{m^2}$ | x | Cost [€/m²] | $\mathrm{U[W/m^2K]}$ | $\tau_{\mathrm{sol}}$ | $\tau_{\mathrm{vis}}$ |
|---|---|---|---|---|---|---|
| | | 0 | 50 | 2.69 | 0.75 | 0.80 |
| **Windows** | 2174.98 | 1 | 80 | 1.70 | 0.38 | 0.70 |
| | | 2 | 100 | 1.25 | 0.20 | 0.70 |

Table 4.21: Initital neural network structure of Equation (4.19).

| Layer (Type) | Filters | Kernel Size |
|---|---|---|
| Dense | 25 | 1 |
| 1D-Convolutional | 50 | 2 |
| 1D-Convolutional | 50 | 2 |
| 1D-Convolutional | 50 | 2 |
| 1D-Convolutional | 50 | 1 |
| Dense | 1 | 1 |

Table 4.22: Optimum Artifical Neural Networks (ANN) structure and performance for a surrogate model of Equation (4.19).

| Layer (Type) | Filters | Kernel Size |
|---|---|---|
| Dense | 300 | 1 |
| 1D-Convolutional | 157 | 2 |
| 1D-Convolutional | 6 | 2 |
| 1D-Convolutional | 290 | 2 |
| 1D-Convolutional | 70 | 1 |
| Dense | 1 | 1 |
| **R2 score** | | 0.96 |
| **RMSE (kWh/m²)** | | 0.54 |

Table 4.23: OptimumArtifical Neural Networks (ANN) structure and performance for a surrogate model of Equation (4.20).

| Layer (Type) | Filters | Kernel Size |
|---|---|---|
| Dense | 300 | 1 |
| 1D-Convolutional | 300 | 2 |
| 1D-Convolutional | 300 | 2 |
| 1D-Convolutional | 6 | 2 |
| 1D-Convolutional | 100 | 1 |
| Dense | 1 | 1 |
| **R2 score** | | 0.97 |
| **RMSE (kWh/m²)** | | 0.01 |

Table 4.24: Considered hyperparameters and their ranges for all the optimization algorithms.

| Algorithm | Iterations | Hyperparameters | Range |
|---|---|---|---|
| | | Population size $P_s$ | [30 ... 200]. |
| NSGAII | 500 | SBX crossover $S$ | [0, 1] |
| | | Polynomial mutation $P_m$ | [0, 1] |
| | | $P_s$ | [30 ... 200] |
| IBEA | 500 | $S$ | [0, 1] |
| | | $P_m$ | [0, 1] |
| | | Swarm size $S_s$ | [30 ... 200] |
| | | Leader size $L_s$ | [30 ... 200] |
| SMPSO | 500 | Max iterations $i$ | [30 ... 200] |
| | | $S$ | [0, 1] |
| | | $P_m$ | [0, 1] |
| | | $S_s$ | [30 ... 200] |
| | | $L_s$ | [30 ... 200] |
| OMOPSO | 500 | $i$ | [30 ... 200] |
| | | $S$ | [0, 1] |
| | | $P_m$ | [0, 1] |
| | | Epsilon $\epsilon$ | [0.0001, 1] |

Table 4.25: Optimum metaheuristics hyperparameters and respective H.

| | Hyperparameter | Value | H |
|---|---|---|---|
| | $P_s$ | 30 | |
| **NSGAII** | $S$ | 1 | 0.53 |
| | $P_m$ | 0.59 | |
| | $P_s$ | 32 | |
| **IBEA** | $S$ | 0.66 | 0.54 |
| | $P_m$ | 1 | |
| | $S_s$ | 68 | |
| | $L_s$ | 59 | |
| **SMPSO** | $i$ | 67 | 0.52 |
| | $S$ | 0.43 | |
| | $P_m$ | 0.35 | |
| | $S_s$ | 30 | |
| | $L_s$ | 200 | |
| **OMOPSO** | $i$ | 200 | 0.52 |
| | $S$ | 1 | |
| | $P_m$ | 0 | |
| | $\epsilon$ | 0.001 | |

# Chapter 5

# Conclusions

## Preamble

This final part of the PhD thesis encompasses two Sections. First, a summary of the results obtained from the case studies is presented, along with a qualitative interpretation of these results. These findings are then discussed concerning the thesis's research questions and goals. Second, future research directions are proposed to guide the implementation of similar models to real-world problems.

The main findings are outlined in the first Section, considering the three approaches used for Building Information Database (BID) development: Existing BID, Synthetic BID, and Iterative BID. For each approach, the developed Surrogate Models (SM) is assessed in terms of its accuracy, the number of objectives, the number of inputs required to run the model, and its implementation. Moreover, the qualities of the developed models are discussed and compared according to their limitations, scope, development speed, and expertise required to use the model.

The main research questions are addressed for each approach, particularly those concerning SM portability, simulation time, and the expertise required for the developed models when implemented in Analysis and Optimization Processes (AOP). Afterward, this research is evaluated regarding its impact on achieving the proposed goals.

Finally, future work to overcome the limitations of each approach is suggested, alongside potential real-world applications of this research. These applications are segmented into different scopes, uses, and scales, such as building performance aspects, users and practitioners, and fields of work/industry.

## 5.1 Main Research Findings

In this Section, the main findings of each approach are compiled and outlined according to model performance, number of inputs, number of objectives, and improvements obtained with the AOP implementation. In addition, a qualitative analysis is assembled to evaluate and compare each BID approach by considering its limitations, scope, development time, and versatility. These findings are then used to discuss this research's success in achieving the established research's goals while addressing the proposed research questions.

The first approach encompassed a case study (See Section 4.1) that utilized an Existing Building Database from the Portuguese Energy Performance Certificates (EPC) to develop SM capable of predicting a building's EPC class and energy use indicators based on 20 input parameters. The models achieved an Coefficient of determination ($R^2$) of 0.84, 0.79, 0.61, and 0.47 for predicting EPC class, annual energy needs, heating energy needs, and cooling energy needs, respectively. The developed SM was subsequently integrated into web applications for two optimization problems.

The first problem's objective was to determine the optimal combination of retrofits for individual EPC upgrades, aiming to minimize annual energy consumption while considering total retrofit costs and maximizing return on investment through certificate upgrade tax benefits and government retrofit funding. The results indicated potential energy use reductions of up to 60% with more expensive solutions and reductions of up to 35% with less expensive alternatives.

The second problem focused on finding the best combination of retrofits for a group of buildings with similar EPC ratings. Here, the optimization aimed to minimize total annual energy needs, sample standard deviation, and total retrofit costs. The outcomes demonstrated potential reductions of up to 25% in annual energy needs, with associated cost reductions of up to 60% when compared with more expensive solutions.

The second approach encompasses a case study (See Section 4.2) that used a synthetic BID to develop a SM capable of predicting Building Performance Simulation (BPS) for the annual energy use of any building in Lisbon when provided with the 6 required inputs (Number of floors, proportion, orientation, glazing ratio, floor area, and period of construction). The SM showed an $R^2$ of 0.95 for predicting a simple energy simulation in Lisbon. Afterward, this model was implemented in two optimization problems, one within an Integrated Development Environment (IDE) and another within a web application.

The first optimization problem was implemented within the IDE and sought to optimize the combination of retrofits for a block of 21 buildings to yield the minimum mean annual Energy Use Intensity (EUI) $\mathrm{kW\,h\,m^{-2}}$, the standard deviation of the sample's EUI, and retrofit cost. This problem was solved with a simulation-based Multi-Objective Optimization (MOO) approach and with the developed surrogate model. The latter showed increases in speed of up to 85 times with acceptable accuracy compared to the computational time for the simulation-based optimization. Moreover, the optimization algorithms explored solutions between 62 $\mathrm{kW\,h\,m^{-2}}$ and 74 $\mathrm{kW\,h\,m^{-2}}$ for the annual EUI, 5 $\mathrm{kW\,h\,m^{-2}}$ and 15 $\mathrm{kW\,h\,m^{-2}}$ for the standard deviation, and 1000 € and 3500 € for the mean cost of retrofit per building.

The second optimization problem was implemented in a web application as a Single-Objective Optimization (SOO) to minimize BPS results for the annual EUI of a building by considering its geometric characteristics and building period. The users can select the number of iterations they wish to explore and define variables and their domains. Results for the sample tested demonstrated energy reductions of up to 8% in Energy Use Intensity (EUI).

The final approach was evaluated with two case studies (See Sections 4.3 and 4.4). Here, an Iterative BID is used to develop SM capable of predicting: (1) the BPS of the annual HVAC and artificial lighting EUI of an office with a Thermochromic Glazing (TCG) glazing in Lisbon and Copenhagen, and (2) the BPS of the annual EUI for a block of 6 buildings in Lisbon and its standard deviation. The models used 4 and 24 inputs for (1) and (2) respectively and showed mean $R^2$ scores of 0.99 in (1) and 0.97 in (2). Notably, the latter only achieved this score after a Model Tuning stage.

The first problem considered the TCG properties as variables. In particular, the maximum and minimum transition temperatures and solar transmittance. The goal was to improve the properties of the TCG to yield minimum annual HVAC and artificial lighting EUI $\mathrm{kW\,h\,m^{-2}}$. This optimization was performed with the developed SM for one weather file in Lisbon and another in Copenhagen. For Lisbon, the results have shown reductions of up to 15% in the total EUI of the office with the optimized TCG against the original. For Copenhagen, the best solution found by the optimization algorithm obtained reductions up to 7% in the total annual energy use of the office.

The second problem considered 3 different construction materials for each of the 6 building's walls, roofs, floors, and windows. The goal was to find the best combination of materials per building that yielded the minimum total annual HVAC energy use, minimum standard deviation of the sample, and minimum construction costs. Results revealed a 22% reduction in construction costs while maintaining similar performance and standard deviation compared with more expensive alternatives.

Together, all these studies fully illustrate the proposed framework in this research to improve the integration of AOP with building and urban projects. A model is developed for each BID approach except the Iterative BID. The latter was illustrated with two case studies and different models, one comprising the Model Tuning stage of the framework with the hyperparameter optimization of the model. Compared with conventional simulation-based optimizations, the obtained reduction in computational time showcases how this framework can tackle the time-consuming nature of BPS by considerably reducing the required simulation times. Moreover, the small number of feature inputs required to run the models demonstrates how this framework can address the complexity challenges that emerge from implementing AOP. Finally, the implementation of these models in optimization problems with different scales (from single building material to urban scales) illustrates how this framework overcomes portability challenges that arise from multiple building design and simulation tools and formats.

Table 5.1 documents these results summary for all the tested BID. Results show that the developed SM were highly accurate in all the tested cases. Moreover, all SM that predict a BPS result showed significant decreases in computation time compared with a simulation's. Noteworthy, Existing BID show accuracy results highly dependable on the data treatment and quality. In addition, from all the developed SM, this model was the only one not aiming to predict a simulation value.

These results can help extrapolate qualitative aspects obtained from experience from each BID approach. In particular, concerning each BID's development speed, model accuracy, number of objectives and inputs, and scope. With these qualities, each approach's qualities and limitations can be outlined.

Table 5.1: Main Findings summary regarding each Surrogate Models (SM) developed and implemented with the three different Building Information Database (BID) approaches.

| BID | Problem type | Scale | Objectives | %Improvements | Inputs | $R^2$ |
|---|---|---|---|---|---|---|
| Existing | Retrofit | Building | 3 | 60% | 20 | 0.84 0.79 |
| | Retrofit | Urban | 3 | 25% | | |
| Synthetic | Retrofit | Urban | 3 | 16% | 6 | 0.95 |
| | Design | Building | 1 | 8% | | |
| Iterative | Material | Room | 2 | 17% | 4 | 0.99 |
| | Construction | Urban | 3 | 22% | 24 | 0.97 |

### 5.1.1 Qualitative Analysis

By merging this thesis's main findings with the experience and observation during the development of the models with each BID approach, it is possible to outline some qualities and limitations concerning each one. Thus, a qualitative analysis is conducted in this Subsection to document and compare each BID approach and their respective models' qualities and limitations regarding accuracy, objective and input capacity, development speed, and scope.

The model trained with an Existing BID has shown the lowest $R^2$ scores. However, this accuracy in predicting the results heavily depends on the data quality and treatment of the BID. Notably, this approach obtained acceptable accuracy levels due to the Feature Engineering techniques applied during the BID development. Concerning the number of objectives and inputs this approach seems to be the most versatile since it does not require BPS and can contain several inputs and objectives with a small impact on the computation time. Consequently, it is also the approach that may have higher development speeds. Considering the scope of this approach, this research suggests that it is well applied to studies of building operation and forecasting. Furthermore, this approach can enhance most studies that require empirical measurements or observation data, since they do not require building performance simulation.

The model trained with a Synthetic BID has shown high $R^2$ scores, which indicates an accurate prediction of simulation results. However, this approach is significantly more limited regarding the number of inputs of the model. This happens because when the number of inputs increases, so does the number of simulations required to develop an efficient Synthetic BID. Thus, the BID's development speed can decrease significantly to the point of being unfeasible. This effect can be multiplied when more objectives that require different BPS tools are included. Concerning the scope of this approach, it is most useful in the early stages of a building project for a short number of inputs since it offers a comprehensive BID throughout all the input domains. Accordingly, this approach is versatile and can be applied for quick and easy-to-use models that can be integrated with multiple AOP.

The model trained with an Iterative BID has shown the highest $R^2$ scores. Because this BID is compiled throughout the optimization process, it can increase the capacity of inputs and objectives supported in the model. This happens because fewer simulations are required to compile the BID, always according to the number of iterations specified. However, these results do not necessarily reflect that this approach delivers the most accurate SM. Because the optimization algorithm seeks optimum solutions for the specified objectives, the resulting BID will have an unbalanced database with fewer under-performing solutions represented. Thus, it can be argued that this approach, although showing models with the highest $R^2$ scores, does not offer a comprehensive BID throughout the input domains, since it overfits the prediction of best-performing solutions. Consequently, this approach is ideal for optimization problems, since they offer a wide range of inputs and objectives, but a restricted application for different AOP.

Table 5.2 summarizes this analysis for each BID approach. Existing BID models show that their accuracy relies on the data quality and tuning. Additionally, Iterative BID models show the best $R^2$ scores but only for best-performing values. Thus, the BID that provides the most accurate models for all prediction values is arguably the Synthethic BID. However, this BID approach also has a smaller capacity for inputs and objectives, since they require more simulations to compile than Iterative BID. Consequently, a trade-off emerges between the BID detail and its development speed.

Table 5.2: Summary of findings, qualities, and limitations for the models trained with different building databases.

| | **Existing BID** | **Synthetic BID** | **Iterative BID** |
|---|---|---|---|
| **Accuracy** | High accuracy achieved through Model Tuning; Lowest $R^2$ scores. | Arguably the most accurate for all prediction values. | Most accurate for optimization problems; less accurate in predicting under-performing solutions. |
| **Number of Objectives** | Limited only by the scope of the existing databases. | Limited number of objectives for different simulations | Versatility in handling multiple objectives. |
| **Number of Inputs** | Limited only by the scope of the existing databases. | Can support fewer inputs compared to the others | Can support a larger number of inputs due to optimization process. |
| **Development Speed** | Fastest development speed because no simulations are required. | Time-consuming because a high number of simulations is required | Faster development speed because a smaller number of simulations is required. |
| **Scope** | Building Operation; measurements; when no simulation is required. | Early stage building projects; easy-to-use because of small input numbers. | Optimization problems that require simulations; final stages of building projects. |
| **Qualities** | Does not require simulations; versatile numbers of inputs and objectives; shorter development time | Offers a comprehensive database for all prediction values; low complexity due to small input numbers | Faster development when simulations are required; handles more inputs and objectives. |
| **Limitations** | Highly dependent on the quality of existing data; limited applications | Low development speed; limited detail due to fewer inputs; repeated processes for different simulation tools | Specific to the optimization problem, higher errors for under-performing solutions. |

### 5.1.2 Discussion

With these main findings and qualitative analysis of each approach, it is now possible to circle back to this Thesis's research question and goals and discuss their achievement with the studied approaches. Particularly, this Subsection will discuss each BID approach main findings from an individual and holistic perspective regarding this research's proposed goals. Finally, possible professional implementation of SM developed with each BID approach are discussed for different types of uses.

This thesis aimed to help answer the question:

*"How can the integration of Analysis and Optimization Processes (AOP) with building and urban projects be improved?"*

After documenting the literature several obstacles were identified concerning the use of Building Performance Simulation (BPS), and the implementation of AOP. For BPS in particular, it lacks portability with other building design and analysis tools; it is time-consuming when there are many simulations or the models are complex; and requires expertise to develop the simulation models and obtain valid inputs. Moreover, when integrating BPS with AOP, the iterative nature of these processes and the multiple aspects of building performance can render the project unfeasible due to high computation times.

Artificial Intelligence can be leveraged to develop SM that speed up AOP, make it easier, and more portable. However, they are usually specific to the problem and highly depend on data quality, model, and model tuning techniques. This tuning is complex and time-consuming when the model accuracy is below expectations. Overcoming these issues can help improve the integration of SM with building performance AOP. Accordingly, the proposed framework's 5 methodological advances mentioned at the beginning of this thesis were successfully showcased (Chapter 1, Section 1.4), as discussed below:

The first consisted of providing a systematic approach to develop a SM for any building performance problem. This was achieved with multiple BID approaches and with the wide case study diversity. This diversity covered every stage of the framework process considering different building performance problems, BID, model tuning techniques, and different model deployment options. This is illustrated in the flowchart that illustrates the framework that provides a comprehensive approach to all the stages of SM development for building performance.

The second methodological advance encompassed identifying different types of databases and providing suitable approaches to enhance data quality. This was achieved with the multiple databases explored throughout the case studies. In addition, some case studies encompassed model tuning techniques such as Feature Engineering (FE) and Selection, which helped improve data quality. Moreover, the framework's first stage considers data suitability according to the building performance problem at hand.

The third focused on providing workflows to select the most suitable SM. This was achieved with the multiple SM types explored for the first two case studies. In the first, the models were tested with a different number of features, while the second explored four different regression models for the prediction of an energy simulation result. These practical cases can be extended to other building performance problems.

The fourth consisted of providing workflows to enhance the model's accuracy if needed. This is achieved with the last case study, which encompassed the Hyperparameter tuning of both SM type and optimization algorithms used to increase its respective accuracy and Hypervolume. Additionally, the first case study encompasses FE and Selection techniques, which can be applied before employing a time-consuming process like Hyperparameter optimization.

Finally, the fifth was to provide implementation workflows for different building performance scopes. This was achieved with each SM deployment in the case studies. Throughout the studies, multiple methods and variations of SM deployment and use were demonstrated, ranging from web applications to 3D models. Moreover, direct implementations in the SM development Integrated Development Environment (IDE) were demonstrated, allowing more experienced practitioners to adapt it to their specific problems.

By achieving these goals, state-of-the-art AOP was advanced in different contexts and practices concerning building and urban projects. Stakeholders can assess their building or home performance and plan suitable investments and retrofits with confidence; architects can guide their designs with deployed SM developed specifically for their use; engineers can use these models to speed up otherwise unfeasible iterative analyses and optimizations; industries can optimize their materials and products for specified performance targets and different climates; and policy-makers can project the impact of their policies with more confidence and at multiple scales. Moreover, SM can be deployed to help monitor and convert data repositories to digital formats and identify possible outliers in data.

## 5.2 Future Work and Applications

The tested BID approaches in this research can improve the integration of AOP and BPS in the AEC industry with the development of SM to overcome current limitations. However, future research is needed to improve each BID development speed, input and objective capacity, and its respective SM accuracy. This Section documents and highlights possible research directions that can be taken to address these issues, and extrapolates on future research possibilities and real-world developments that can emerge.

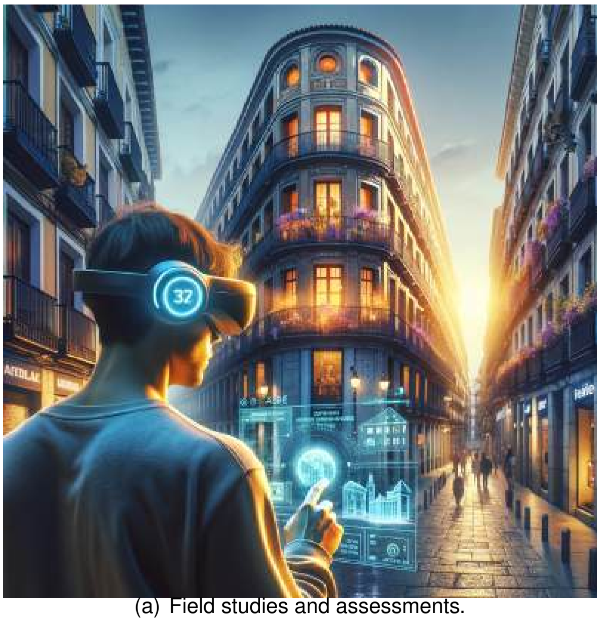
The first approach (Existing BID) shows significant improvements in AOP where no simulations are required. However, the accuracy of the resulting SM depends on the quality of existing data. To address this limitation, future research should focus on curating and streaming available BID, as well as exploring and benchmarking new model-tuning processes. This includes evaluating their effectiveness across various data types such as measurements, images/graphics, time series, and historical data.

The Synthetic BID is capable of offering a robust database that results in highly accurate SM that are easy to use with a small number of inputs. Unfortunately, when increasing the number of inputs and objectives, more simulations and different tools are required to obtain accurate predictions. This limits the BID development speed. Thus, future research direction should focus on developing new heuristics for sample generation that guarantee accurate models with a feasible number of BPS for any number of inputs and objectives. Furthermore, new types of SM can be explored to generate rendered graphic results of BPS such as Generative Adversarial Networks and Convolution Neural Networks.

The Iterative BID provides faster development speed when compared with Synthetic BID since it is capable of handling more inputs and objectives. However, results show higher errors in under-performing solutions not explored by the optimization algorithm, which narrows the SM scope to the optimization problem. In addition, to obtain high accuracy scores with the SM, multiple algorithms must be used, and their Hyperparameters should be optimized to guarantee a broad exploration of the solution space. Future research should focus on benchmarking optimization and SM algorithms for standard building optimization problems. This could allow future developers to speed up the BID and model development for future AOP.

In summary, future research suggestions encompass benchmarking algorithms, exploring different graphical representations for BPS results, experimenting with other model tuning processes, and investigating image-based BID and new SM types. This can enable real-world developments of new tools and processes to help improve the integration of AOP in building and urban projects. Moreover, it can assist in enhancing field studies and assessments, enhancing collaborative work between disciplines, curating and improving digital twin models, and developing comprehensive and fast decision-making tools to improve the evaluation of past policies while informing future ones.

Figure 5.1 illustrates future possibilities of SM deployment and use for different scopes. In (a), a field study is performed on a building to assess multiple performance indicators. Multiple building performance SM can be deployed in Augmented Reality environments. In (b), collaborative work can be improved by leveraging Large Language Models with other building performance SM to develop bot assistants for building and urban projects. Finally, (c) illustrates a fully integrated digital twin model of an urban area, and (d) envisions fast decision-making tools being used to help assess the impact of future policies. These are all promising areas for future research that can help achieve both Carbon Neutrality and Sustainable Development Goals.



(a) Field studies and assessments.

(b) Collaborative work between different fields.

(c) Digital Twin technologies.

(d) Policymaking tools to meet Sustainable Development Goals (SDG).

Figure 5.1: Examples of potential real-world applications of integrating Surrogate Models (SM) with different Analysis and Optimization Processes (AOP).

# Bibliography


[1] United Nations. The World 's Cities in 2018. Technical report, United Nations, 2018. URL https://www.un.org/en/development/desa/population/publications/pdf/urbanization/the_worlds_cities_in_2018_data_booklet.pdf.

[2] U. Nations. The Sustainable Development Goals. Technical report, United Nations, 2020.

[3] European Commission. Renovation Wave: doubling the renovation rate to cut emissions, boost recovery and reduce energy poverty. Technical Report October, European Comission, 2020. URL https://ec.europa.eu/newsroom/growth/items/690437.

[4] E. Swyngedouw, F. Moulaert, and R. Arantxa. New Geographies of Power , Exclusion and Injustice Neoliberal Urbanization in Europe : Large-Scale Urban Development Projects and the New Urban Policy. *Antipode*, 34:542–577, 2002. ISSN 0066-4812.

[5] C. F. Reinhart and C. Cerezo Davila. Urban building energy modeling - A review of a nascent field. *Building and Environment*, 97:196–202, 2016. ISSN 03601323. doi: 10.1016/j.buildenv.2015.12.001. URL http://dx.doi.org/10.1016/j.buildenv.2015.12.001.

[6] M. Batty. *Cities and complexity: understanding cities with cellular automata, agent-based models, and fractals.* The MIT press, 2007.

[7] Z. Shi, J. A. Fonseca, and A. Schlueter. A review of simulation-based urban form generation and optimization for energy-driven urban design. *Building and Environment*, 121:119–129, 2017. ISSN 03601323. doi: 10.1016/j.buildenv.2017.05.006. URL http://dx.doi.org/10.1016/j.buildenv.2017.05.006.

[8] A. Wolman. The Metabolism of Cities. *Scientific American*, 213(3):178–193, 1965. URL https://www.jstor.org/stable/24931120.

[9] S. Niza, L. Rosado, and P. Ferrao. Urban metabolism methodological advances in urban material flow accounting based on the lisbon case study. *Journal of Industrial Ecology*, 13(3):384–405, 2009. ISSN 10881980. doi: 10.1111/j.1530-9290.2009.00130.x.

[10] C. Reinhart, T. Dogan, A. Jakubiec, T. Rakha, and A. Sang. UMI - AN URBAN SIMULATION ENVIRONMENT FOR BUILDING ENERGY USE , DAYLIGHTING AND WALKABILITY Christoph F Reinhart , Timur Dogan , J Alstan Jakubiec , Tarek Rakha and Andrew Sang Massachusetts

Institute of Technology Department of Architecture. *13th Conference of International Building Performance Simulation Association, Chambéry, France, August 26-28 - 476*, pages 476–483, 2013. URL `http://web.mit.edu/SustainableDesignLab/publications/umi_introduction.pdf`.

[11] G. Araújo, I. Pereira, A. Leitão, and M. Correia Guedes. Conflicts in passive building performance: Retrofit and regulation of informal neighbourhoods. *Frontiers of Architectural Research*, 2021. ISSN 20952635. doi: 10.1016/j.foar.2021.02.007.

[12] R. Machete, A. P. Falcão, M. G. Gomes, and A. Moret Rodrigues. The use of 3D GIS to analyse the influence of urban context on buildings' solar energy potential. *Energy and Buildings*, 177: 290–302, 2018. ISSN 03787788. doi: 10.1016/j.enbuild.2018.07.064. URL `https://doi.org/10.1016/j.enbuild.2018.07.064`.

[13] J. Keirstead, M. Jennings, and A. Sivakumar. A review of urban energy system models: Approaches, challenges and opportunities. *Renewable and Sustainable Energy Reviews*, 16(6): 3847–3866, 2012. ISSN 13640321. doi: 10.1016/j.rser.2012.02.047. URL `http://dx.doi.org/10.1016/j.rser.2012.02.047`.

[14] G. R. Ruiz and C. F. Bandera. Validation of calibrated energy models: Common errors. *Energies*, 10(10), 2017. ISSN 19961073. doi: 10.3390/en10101587.

[15] C. J. Hopfe and J. L. Hensen. Uncertainty analysis in building performance simulation for design support. *Energy and Buildings*, 43(10):2798–2805, 2011. ISSN 03787788. doi: 10.1016/j.enbuild.2011.06.034. URL `http://dx.doi.org/10.1016/j.enbuild.2011.06.034`.

[16] S. Attia. Building performance simulation tools: Selection criteria and user survey. Research Report January, Université catholique de Louvain, Louvain La Neuve, Belgium, 2010.

[17] S. Attia, E. Gratia, A. D. Herde, and J. L. M. Hensen. Simulation-based decision support tool for early stages of zero-energy building design. *Energy & Buildings*, 49:2–15, 2012. ISSN 0378-7788. doi: 10.1016/j.enbuild.2012.01.028. URL `http://dx.doi.org/10.1016/j.enbuild.2012.01.028`.

[18] R. Aguiar, C. Cardoso, and A. Leitão. Algorithmic design and analysis fusing disciplines. *Disciplines and Disruption - Proceedings Catalog of the 37th Annual Conference of the Association for Computer Aided Design in Architecture, ACADIA 2017*, pages 28–37, 2017.

[19] I. Caetano, S. Garcia, Inês Pereira, and A. Leitão. Creativity inspired by analysis: An algorithmic design system for designing structurally feasible façades. In *RE: Anthropocene, Proceedings ofthe 25th International Conference ofthe Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2020, Volume 1*, pages 599–608, Bangkok, Thailand, 2020.

[20] I. Caetano, L. Santos, and A. Leitão. Computational design in architecture: Defining parametric, generative, and algorithmic design. *Frontiers of Architectural Research*, 9(2):287–300, 2020. ISSN 20952635. doi: 10.1016/j.foar.2019.12.008.

[21] Z. Qian, C. C. Seepersad, V. R. Joseph, J. K. Allen, and C. F. Wu. Building surrogate models based on detailed and approximate simulations. *Journal of Mechanical Design, Transactions of the ASME*, 128(4):668–677, 2006. ISSN 10500472. doi: 10.1115/1.2179459.

[22] E. Thrampoulidis, G. Mavromatidis, A. Lucchi, and K. Orehounig. A machine learning-based surrogate model to approximate optimal building retrofit solutions. *Applied Energy*, 281:116024, 2021. ISSN 03062619. doi: 10.1016/j.apenergy.2020.116024. URL `https://doi.org/10.1016/j.apenergy.2020.116024`.

[23] K. Bamdad, M. E. Cholette, and J. Bell. Building energy optimization using surrogate model and active sampling. *Journal of Building Performance Simulation*, 13(6):760–776, 2020. ISSN 19401507. doi: 10.1080/19401493.2020.1821094.

[24] T. Wortmann, A. Costa, G. Nannicini, and T. Schroepfer. Advantages of surrogate models for architectural design optimization. *Artiific ial Intelligence for Engineering Design , Analysis and and Manufacturing*, 29(4):471–481, 2015. doi: 10.1017/S0890060415000451.

[25] G. G. Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 4255, pages 415–426, 2006.

[26] A. Forrester, A. Sobester, and A. Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

[27] S. Seyedzadeh, F. Pour Rahimian, I. Glesk, and M. Roper. Machine learning for estimation of building energy consumption and performance: A review. *Visualization in Engineering*, 6:1–20, 2018.

[28] P. Westermann and R. Evins. Surrogate modelling for sustainable building design–a review. *Energy and Buildings*, 198:170–186, 2019.

[29] J. L. Hensen and R. Lamberts, editors. *Building performance simulation for design and operation*. Routledge, 2 edition, 2019. ISBN 9780429688539.

[30] T. Hong, J. Langevin, and K. Sun. Building simulation: Ten challenges. *Building Simulation*, 11 (5):871–898, 2018. ISSN 1996-8744. doi: 10.1007/s12273-018-0444-x. URL `https://doi.org/10.1007/s12273-018-0444-x`.

[31] Elsevier BV. Scopus api data on [topic]. `https://api.elsevier.com/content/search/scopus`, 2024. Accessed: 2024-07-19.

[32] B. Drury, O. Curtis, K. Linda, and C. Frederick. EnergyPlus : Energy simulation program. *ASHRAE Journal*, 42(April):49–56, 2000.

[33] R. Guglielmetti, D. Macumber, and N. Long. Openstudio: An open source integrated analysis platform. In *Proceedings of Building Simulation 2011: 12th Conference of International Building*

*Performance Simulation Association*, pages 442–449, Sydney, Australia, November 14–16 2011. National Renewable Energy Laboratory, International Building Performance Simulation Association.

[34] DesignBuilder Software Ltd. Designbuilder software. `https://www.designbuilder.co.uk/`, 2024. Accessed: 2024-07-19.

[35] S. E. Mattsson, H. Elmqvist, and M. Otter. Physical system modeling with modelica. *Control engineering practice*, 6(4):501–510, 1998.

[36] Autodesk. Revit insight, 2024. URL `https://www.autodesk.com/products/insight/overview`. Accessed: 2024-07-20.

[37] C. Underwood and F. Yik. *Modelling methods for energy in buildings*. John Wiley & Sons, 2008.

[38] F. C. McQuiston, J. D. Parker, J. D. Spitler, and H. Taherian. *Heating, ventilating, and air conditioning: analysis and design*. John Wiley & Sons, 1st edition, 2005.

[39] J.-H. Choi. Investigation of the correlation of building energy use intensity estimated by six building performance simulation tools. *Energy and Buildings*, 147:14–26, 2017.

[40] E. Kossecka and J. Kosny. Influence of insulation configuration on heating and cooling loads in a continuously used building. *Energy and buildings*, 34(4):321–331, 2002.

[41] D. H. Li, L. Yang, and J. C. Lam. Zero energy buildings and sustainable development implications–a review. *Energy*, 54:1–10, 2013.

[42] S. Sun, K. Kensek, D. Noble, and M. Schiler. A method of probabilistic risk assessment for energy performance and cost using building energy simulation. *Energy and Buildings*, 110:1–12, 2016.

[43] T. Olofsson and T. Mahlia. Modeling and simulation of the energy use in an occupied residential building in cold climate. *Applied Energy*, 91(1):432–438, 2012.

[44] W. Li, Y. Zhou, K. Cetin, J. Eom, Y. Wang, G. Chen, and X. Zhang. Modeling urban building energy use: A review of modeling approaches and procedures. *Energy*, 141:2445–2457, 2017.

[45] OpenStudio. Monthly electricity consumption, 2024. URL `https://nrel.github.io/OpenStudio-user-documentation/`. Accessed: 2024-07-20.

[46] Autodesk-University. Energy simulation models: Cooling loads, 2023. URL `https://www.autodesk.com/autodesk-university/`. Accessed: 2024-07-20.

[47] G. J. Ward. The radiance lighting simulation and rendering system. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 459–472, 1994.

[48] J. A. Jakubiec and C. F. Reinhart. Diva 2.0: Integrating daylight and thermal simulations using rhinoceros 3d, daysim and energyplus. In *Proceedings of building simulation*, volume 20, pages 2202–2209, 2011.

[49] S. Darula and R. Kittler. Cie general sky standard defining luminance distributions. *Proceedings eSim*, 11:13, 2002.

[50] J. Mardaljevic. Examples of climate-based daylight modelling. In *CIBSE national conference*, pages 1–11, 2006.

[51] P. R. Tregenza. Subdivision of the sky hemisphere for luminance measurements. *Lighting Research & Technology*, 19(1):13–14, 1987.

[52] P. Tregenza. The daylight factor and actual illuminance ratios. *Lighting Research & Technology*, 12(2):64–68, 1980.

[53] R. Hopkinson, J. Longmore, and P. Petherbridge. An empirical formula for the computation of the indirect component of daylight factor. *Transactions of the Illuminating Engineering Society*, 19 (7_IEStrans):201–219, 1954.

[54] A. Nabil and J. Mardaljevic. Useful daylight illuminances: A replacement for daylight factors. *Energy and buildings*, 38(7):905–913, 2006.

[55] I. Lm. Approved method: Ies spatial daylight autonomy (sda) and annual sunlight exposure (ase). *Illum Eng Soc https//www ies org/product/ies-spatial-daylight-autonomy-sda-and-annual-sunlight-exposure-ase*, 2013.

[56] J. Wienold. Dynamic Daylight Glare Evaluation. In *Elenventh International IBPSA Conference*, pages 944–951, Glasgow, Scotland, 2009.

[57] R. G. Hopkinson. Glare from daylighting in buildings. *Applied ergonomics*, 3(4):206–215, 1972.

[58] L. Giovannini, F. Favoino, A. Pellegrino, V. R. M. Lo Verso, V. Serra, and M. Zinzi. Thermochromic glazing performance: From component experimental characterisation to whole building performance evaluation. *Applied Energy*, 251(February):113335, 2019. ISSN 03062619. doi: 10.1016/j.apenergy.2019.113335. URL `https://doi.org/10.1016/j.apenergy.2019.113335`.

[59] R. Debnath and R. Bardhan. Daylight performance of a naturally ventilated building as parameter for energy management. *Energy Procedia*, 90:382–394, 2016.

[60] M. Mohsenin and J. Hu. Assessing daylight performance in atrium buildings by using climate based daylight modeling. *Solar Energy*, 119:553–560, 2015.

[61] N. Nasrollahi and E. Shokri. Daylight illuminance in urban environments for visual comfort and energy performance. *Renewable and sustainable energy reviews*, 66:861–874, 2016.

[62] I. Bournas. Daylight compliance of residential spaces: Comparison of different performance criteria and association with room geometry and urban density. *Building and Environment*, 185: 107276, 2020.

[63] K. S. Lee, K. J. Han, and J. W. Lee. The impact of shading type and azimuth orientation on the daylighting in a classroom–focusing on effectiveness of façade shading, comparing the results of da and udi. *Energies*, 10(5):635, 2017.

[64] G. Reis. What is the impact of window height and facade depth on useful daylight illuminance (udi) of an open plan commercial building?, 2021. URL https://projectstasio.com/contribution/what-is-the-impact-of-window-height-and-facade-depth-on-useful-daylight-illuminance-udi-of-an- Accessed: 2024-07-20.

[65] L. Peeters, R. D. Dear, J. Hensen, and D. William. Thermal comfort in residential buildings : Comfort values and scales for building energy simulation. *Applied Energy*, 86(5):772–780, 2009. ISSN 0306-2619. doi: 10.1016/j.apenergy.2008.07.011. URL http://dx.doi.org/10.1016/j.apenergy.2008.07.011.

[66] J. Van Hoof. Forty years of fanger's model of thermal comfort: comfort for all? *Indoor air*, 18(3), 2008.

[67] R. Dear and G. Brager. Developing an adaptive model of thermal comfort and preference, 1998. URL https://escholarship.org/uc/item/4qq2p9c6. Retrieved from https://escholarship.org/uc/item/4qq2p9c6.

[68] B. Givoni. Comfort, climate analysis and building design guidelines. *Energy and buildings*, 18(1): 11–23, 1992.

[69] B. Levitt, M. Ubbelohde, G. Loisos, and N. Brown. Thermal autonomy as metric and design process. In *CaGBC National Conference and Expo: Pushing the Boundary–Net Positive Buildings*, pages 47–58, 2013.

[70] L. Matias, S. Almeida, C. Pina Santos, M. Rebelo, and M. Correia Guedes. Adaptive thermal comfort for buildings in portugal based on occupants' thermal perception. In *Proceedings of the PLEA2009—26th Conference on Passive and Low Energy Architecture: Architecture Energy and the Occupant's Perspective, Quebec City, Canada*, pages 22–24. Citeseer, 2009.

[71] A. Pourshaghaghy and M. Omidvari. Examination of thermal comfort in a hospital using pmv–ppd model. *Applied ergonomics*, 43(6):1089–1095, 2012.

[72] F. Tartarini, S. Schiavon, T. Cheung, and T. Hoyt. Cbe thermal comfort tool: Online tool for thermal comfort calculations and visualizations. *SoftwareX*, 12:100563, 2020.

[73] S. Attia and S. Carlucci. Impact of different thermal comfort models on zero energy residential buildings in hot climate. *Energy and Buildings*, 102:117–128, 2015.

[74] V. Harish and A. Kumar. A review on modeling and simulation of building energy systems. *Renewable and sustainable energy reviews*, 56:1272–1292, 2016.

[75] L. Caldas and L. Santos. Painting with light: An interactive evolutionary system for daylighting design. *Building and Environment*, 109:154–174, 2016.

[76] W. H. Ko, S. Schiavon, G. Brager, and B. Levitt. Ventilation, thermal and luminous autonomy metrics for an integrated design process. *Building and Environment*, 145(July):153–165, 2018. ISSN 03601323. doi: 10.1016/j.buildenv.2018.08.038. URL https://doi.org/10.1016/j.buildenv.2018.08.038.

[77] T. A. Reddy. Literature review on calibration of building energy simulation programs: Uses, problems, procedures, uncertainty, and tools. *ASHRAE transactions*, 112(1), 2006.

[78] Trimble Inc. Sketchup, 2024. URL https://www.sketchup.com. Accessed: 2024-07-20.

[79] TLM, Inc. Rhino 3d. Computer software, 2020. URL https://www.rhino3d.com/. Version 7.

[80] Robert McNeel and associates. Grasshopper for rhino3d. Computer software, 2014. URL https://www.grasshopper3d.com/. Version 1.

[81] K. Terzidis. Algorithmic design: A paradigm shift in architecture. In *Architecture in the Network Society [22nd eCAADe Conference Proceedings]*, pages 201–207, Copenhagen, Denmark, September 2004. ISBN 0-9541183-2-4.

[82] B. Kolarevic. Architecture in the digital age. *Design and Manufacturing. Nueva York-Londres: Spon Press-Taylor & Francis Group*, 2003.

[83] J. Krause. Reflections: the creative process of generative design in architecture. In *Generative Arts Conference, Milan. http://citeseerx. ist. psu. edu/viewdoc/download*. Citeseer, 2003.

[84] J. FRAZER, J. FRAZER, X. Liu, M. Tang, and P. Janssen. Generative and evolutionary techniques for building envelope design in generative art 2002: 5th international generative art conference ga2002. *Italia, Milão, Dezembro*, 2002.

[85] R. Oxman. Thinking difference: Theories and models of parametric design thinking. *Design studies*, 52:4–39, 2017.

[86] M. Sammer, A. Leitão, and I. Caetano. From Visual Input to Visual Output in Textual Programming. In *Intelligent and Informed, Proceedings of the 24th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2019*, volume 1, pages 645–654, 2019.

[87] A. Leitão and L. Santos. Programming languages for generative design: Visual or textual? In *29th Conference on Education in Computer Aided Architectural Design in Europe, eCAADe 2011*, pages 549–557. Education and research in Computer Aided Architectural Design in Europe, 2011.

[88] B. Harmon. An introduction to grasshopper. https://baharmon.github.io/basics, 2024. Accessed: 2024-07-23.

[89] M. G. C. CELANI. Teaching cad programming to architecture students. *Gestão & Tecnologia de Projetos*, 3(2):1–23, 2008.

[90] P. Janssen. Visual dataflow modelling-some thoughts on complexity. In *Proceedings of the 32nd International Conference on Education and Research in Computer Aided Architectural Design in Europe (eCAADe)[Volume 2]*. eCAADe, 2014.

[91] A. Leitão, L. Santos, and J. Lopes. Programming languages for generative design: a comparative study. *International Journal of Architectural Computing*, 10(1):139–162, 2012.

[92] M. Noone and A. Mooney. Visual and textual programming languages: a systematic review of the literature. *Journal of Computers in Education*, 5:149–174, 2018.

[93] M. A. Zboinska. Hybrid cad/e platform supporting exploratory architectural design. *Computer-Aided Design*, 59:64–84, 2015.

[94] P. Janssen, R. Li, and A. Mohanty. Möbius: a parametric modeller for the web. In *Living systems and micro-utopias: Towards continuous designing. Proceedings of the 21st International Conference of the Association for Computer-Aided Architectural Design Research in Asia CAADRIA*, pages 157–166, 2016.

[95] T. Wortmann and B. Tunçer. Differentiating parametric design: Digital workflows in contemporary architecture and construction. *Design Studies*, 52:173–197, 2017.

[96] D. Nezamaldin. Parametric design with visual programming in dynamo with revit: The conversion from cad models to bim and the design of analytical applications, 2019.

[97] R. Castelo-Branco and A. Leitão. Visual meets textual: a hybrid programming environment for algorithmic design. In *RE: Anthropocene—Design in the Age of Humans, Proceedings of the 25th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), Chulalongkorn University, Bangkok, Thailand*, pages 5–6, 2020.

[98] J. Graef. Grasshopper parametric modelling. `https://jamesgraef.com/resume/grasshopper`, 2024. Accessed: 2024-07-23.

[99] Autodesk. Dynamo, 2020. URL `https://dynamobim.org/`. Version 2.17.

[100] M. S. Roudsari and M. Pak. LADYBUG : A PARAMETRIC ENVIRONMENTAL PLUGIN FOR GRASSHOPPER TO HELP DESIGNERS CREATE AN ENVIRONMENTALLY-CONSCIOUS DESIGN. In *13th Conference of International Building Performance Simulation Association*, pages 3128–3135, 2013.

[101] S. Philip. eppy Documentation, 2019. URL `https://github.com/santoshphilip/eppy`.

[102] Hydra Share. Hydra: A platform for sharing grasshopper solutions. `https://hydrashare.github.io/hydra/`, 2024. Accessed: 2024-07-23.

[103] Z. Pang, Z. O'Neill, Y. Li, and F. Niu. The role of sensitivity analysis in the building performance analysis: A critical review. *Energy and Buildings*, 209:109659, 2020.

[104] S. Alghamdi, W. Tang, S. Kanjanabootra, and D. Alterman. Effect of architectural building design parameters on thermal comfort and energy consumption in higher education buildings. *Buildings*, 12(3):329, 2022.

[105] E. Elbeltagi, H. Wefki, S. Abdrabou, M. Dawood, and A. Ramzy. Visualized strategy for predicting buildings energy consumption during early design stage using parametric analysis. *Journal of Building Engineering*, 13:127–136, 2017.

[106] M. K. Najjar, V. W. Tam, L. T. Di Gregorio, A. C. J. Evangelista, A. W. Hammad, and A. Haddad. Integrating parametric analysis with building information modeling to improve energy performance of construction projects. *Energies*, 12(8):1515, 2019.

[107] G. C. Henriques, J. P. Duarte, and V. Leal. Strategies to control daylight in a responsive skylight system. *Automation in Construction*, 28:91–105, 2012.

[108] A. Wagdy and F. Fathy. A parametric approach for achieving optimum daylighting performance through solar screens in desert climates. *Journal of Building Engineering*, 3:155–170, 2015.

[109] N. Queiroz, F. S. Westphal, and F. O. R. Pereira. A performance-based design validation study on energyplus for daylighting analysis. *Building and Environment*, 183:107088, 2020.

[110] R. Castelo Branco and A. Leitão. Integrated algorithmic design: A single-script approach for multiple design tasks. In *Design Tools: Proceedings of the 35th eCAADe conference*, pages 729–738, 2017.

[111] G. Araújo, M. C. Guedes, and A. Leitão. Integrating algorithmic processes in informal urban and architectural planning: A case study of a Maputo's neighborhood. In J. R. Á. . J. C. S. Gonçalves, editor, *Proceedings of the 35th PLEA Conference - Planning Post-Carbon Cities*, pages 1173–1178, A Coruna, 2020. University of A Coruña.

[112] A. Eltaweel and S. Yuehong. Parametric design and daylighting: A literature review. *Renewable and Sustainable Energy Reviews*, 73:1086–1103, 2017.

[113] A. A. Chowdhury, M. Rasul, and M. M. K. Khan. Thermal-comfort analysis and simulation for various low-energy cooling-technologies applied to an office building in a subtropical climate. *Applied energy*, 85(6):449–462, 2008.

[114] J. Aleixo, G. R. Araújo, and M. Correia Guedes. Comparison of passive design strategies to improve living conditions: a study in Ondjiva, Southern Angola. *Renewable Energy and Environmental Sustainability*, 6:21, 2021. doi: 10.1051/rees/2021020.

[115] S. Ramakrishnan, X. Wang, M. Alam, J. Sanjayan, and J. Wilson. Parametric analysis for performance enhancement of phase change materials in naturally ventilated buildings. *Energy and buildings*, 124:35–45, 2016.

[116] Q. J. Kwong, N. M. Adam, and B. Sahari. Thermal comfort assessment and potential for energy efficiency enhancement in modern tropical buildings: A review. *Energy and buildings*, 68:547–557, 2014.

[117] M. Liu, P. K. Heiselberg, Y. I. Antonov, and F. S. Mikkelsen. Parametric analysis on the heat transfer, daylight and thermal comfort for a sustainable roof window with triple glazing and external shutter. *Energy and Buildings*, 183:209–221, 2019.

[118] J. Khazaii. Genetic ALgorithm Optimization. In *Advanced Decision Making for HVAC Engineers*, pages 87–97. Springer, Cham, 2016. doi: doi.org/10.1007/978-3-319-33328-1_10.

[119] R. Charron and A. Athienitis. The use of genetic algorithms for a net-zero energy solar home design optimisation tool. In *Proceedings of PLEA*, volume 2006, pages 6–8, 2006.

[120] R. Evins, P. Pointer, R. Vaidyanathan, and S. Burgess. A case study exploring regulated energy use in domestic buildings using design-of-experiments and multi-objective optimisation. *Building and Environment*, 54:126–136, 2012.

[121] R. Horst and P. M. Pardalos. *Handbook of global optimization*, volume 2. Springer Science & Business Media, 2013.

[122] A.-t. Nguyen, S. Reiter, and P. Rigo. A review on simulation-based optimization methods applied to building performance analysis. *APPLIED ENERGY*, 113:1043–1058, 2014. ISSN 0306-2619. doi: 10.1016/j.apenergy.2013.08.061. URL `http://dx.doi.org/10.1016/j.apenergy.2013.08.061`.

[123] S. Attia, M. Hamdy, S. Carlucci, L. Pagliano, S. Bucking, and A. Hasan. *Building Performance Optimization of Net Zero-Energy Buildings*, chapter 5. Ernst & Sohn, 2015. doi: 10.1002/9783433604625.ch05. URL `https://doi.org/10.1002/9783433604625.ch05`.

[124] L. G. M. L. d. O. G. C. Caldas. *An evolution-based generative design system: using adaptation to shape architectural form*. PhD thesis, Massachusetts Institute of Technology, 2001.

[125] I. Pereira and A. Leitão. More is more: The no free lunch theorem in architecture. *Proceedings of the International Conference of Architectural Science Association*, 2020-Novem(November 2020): 765–774, 2020. ISSN 22093850.

[126] I. Pereira and A. Leitão. The cost of daylight: a parallelized approach to multi-objective optimization. In *Planning Post Carbon Cities: 35th PLEA Conference on Passive and Low Energy Architecture, A Coruña, 1st-3rd September 2020: Proceedings*, pages 1626–1631. Universidade da Coruña, 2020.

[127] F. Bre and V. D. Fachinotti. A computational multi-objective optimization method to improve energy efficiency and thermal comfort in dwellings. *Energy and Buildings*, 154:283–294, 2017.

[128] Q. Li, L. Zhang, L. Zhang, and X. Wu. Optimizing energy efficiency and thermal comfort in building green retrofit. *Energy*, 237:121509, 2021.

[129] M. Wetter and J. Wright. A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization. *Building and environment*, 39(8):989–999, 2004.

[130] J. H. Kämpf, M. Wetter, and D. Robinson. A comparison of global optimization algorithms with standard benchmark functions and real-world applications using energyplus. *Journal of Building Performance Simulation*, 3(2):103–120, 2010.

[131] D. Lauinger, P. Caliandro, D. Kuhn, et al. A linear programming approach to the optimization of residential energy systems. *Journal of Energy Storage*, 7:24–37, 2016.

[132] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[133] J.-M. Renders and H. Bersini. Hybridizing genetic algorithms with hill-climbing methods for global optimization: two possible ways. In *Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence*, pages 312–317. IEEE, 1994.

[134] J. Wright and V. I. Hanby. The formulation, characteristics and solution of hvac system optimized design problems. *ASHRAE Transactions*, 93, 1987.

[135] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.

[136] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM review*, 45(3):385–482, 2003.

[137] X. S. Yang. *Nature-inspired metaheuristic algorithms*. Luniver Press, 2010.

[138] C. Waibel, T. Wortmann, R. Evins, and J. Carmeliet. Building energy optimization: An extensive benchmark of global search algorithms. *Energy and Buildings*, 187:218–240, 2019. ISSN 03787788. doi: 10.1016/j.enbuild.2019.01.048. URL `https://doi.org/10.1016/j.enbuild.2019.01.048`.

[139] R. Oxman. Performance-based design: current practices and research issues. *International journal of architectural computing*, 6(1):1–17, 2008.

[140] M. DeLanda. Deleuze and the use of the genetic algorithm in architecture. *Architectural Design*, 71(7):9–12, 2002.

[141] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. ISSN 1089778X. doi: 10.1109/4235.996017.

[142] A. J. Nebro, J. J. Durillo, G. Nieto, C. A. Coello, F. Luna, and E. Alba. SMPSO: A new pso-based metaheuristic for multi-objective optimization. *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, MCDM 2009 - Proceedings*, 2:66–73, 2009. doi: 10.1109/MCDM.2009.4938830.

[143] A. C. Godínez, L. E. M. Espinosa, and E. M. Montes. An experimental comparison of multiobjective algorithms: NSGA-II and OMOPSO. *Proceedings - 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference, CERMA 2010*, pages 28–33, 2010. doi: 10.1109/CERMA.2010.13.

[144] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.

[145] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[146] I. Pereira and A. Leitão. More is more: The no free lunch theorem in architecture. *Proceedings of the International Conference of Architectural Science Association*, 2020-Novem(November 2020): 765–774, 2020. ISSN 22093850.

[147] Y. Censor. Pareto optimality in multiobjective problems. *Applied Mathematics I& Optimization*, 4 (1):41–59, 1977. ISSN 14320606. doi: 10.1007/BF01442131.

[148] A. P. Guerreiro, C. M. Fonseca, and L. Paquete. The hypervolume indicator: Computational problems and algorithms. *ACM Computing Surveys (CSUR)*, 54(6):1–42, 2021.

[149] L. Bradstreet, L. While, and L. Barone. Maximising hypervolume for selection in multi-objective evolutionary algorithms. *20076 IEEE Congress on Evolutionary Computation, CEC 2006*, pages 1744–1751, 2006. doi: 10.1109/CEC.2007.4424881.

[150] L. Bradstreet. *The Hypervolume Indicator for Multi-objective Optimisation: Calculation and Use*. PhD thesis, University of Western Australia, 2011. URL http://www.wfg.csse.uwa.edu.au/publications/PhD_thesis_Lucas_Bradstreet.pdf.

[151] P. Developers. Platypus getting started guide, 2024. URL https://platypus.readthedocs.io/en/latest/getting-started.html. Accessed: 2024-07-24.

[152] X. Sun, Y. Wang, H. Kang, Y. Shen, Q. Chen, and D. Wang. Modified multi-crossover operator nsga-iii for solving low carbon flexible job shop scheduling problem. *Processes*, 9(1):62, 2020.

[153] R. Evins. A review of computational optimisation methods applied to sustainable building design. *Renewable and Sustainable Energy Reviews*, 22:230–245, 2013. ISSN 13640321. doi: 10.1016/j.rser.2013.02.004. URL http://dx.doi.org/10.1016/j.rser.2013.02.004.

[154] S. Attia, M. Hamdy, W. O. Brien, and S. Carlucci. Assessing gaps and needs for integrating building performance optimization tools in net zero energy buildings design. *Energy and Buildings*, 60:110–124, 2013. ISSN 0378-7788. doi: 10.1016/j.enbuild.2013.01.016. URL http://dx.doi.org/10.1016/j.enbuild.2013.01.016.

[155] Z. Tian, X. Zhang, X. Jin, X. Zhou, B. Si, and X. Shi. Towards adoption of building energy simulation and optimization for passive building design: A survey and a review. *Energy and Buildings*, 158:1306–1316, 2018.

[156] R. Jin, W. Chen, and T. W. Simpson. Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and multidisciplinary optimization*, 23:1–13, 2001.

[157] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[158] M. F. Dixon, I. Halperin, and P. Bilokon. *Machine learning in finance*, volume 1170. Springer, 2020.

[159] A. Rajkomar, J. Dean, and I. Kohane. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358, 2019.

[160] S. Zhong, K. Zhang, M. Bagheri, J. G. Burken, A. Gu, B. Li, X. Ma, B. L. Marrone, Z. J. Ren, J. Schrier, et al. Machine learning: new ideas and tools in environmental science and engineering. *Environmental science & technology*, 55(19):12741–12754, 2021.

[161] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.

[162] S. García, J. Luengo, and F. Herrera. Feature selection. *ACM Computing Surveys*, 50(6):163–193, 2017. ISSN 18684408. doi: 10.1007/978-3-319-10247-4_7.

[163] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas. Data preprocessing for supervised leaning. *International journal of computer science*, 1(2):111–117, 2006.

[164] S. Patro and K. K. Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.

[165] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu. Feature selection: A data perspective. *ACM computing surveys (CSUR)*, 50(6):1–45, 2017.

[166] A. Jović, K. Brkić, and N. Bogunović. A review of feature selection methods with applications. In *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)*, pages 1200–1205. Ieee, 2015.

[167] R. M. Heiberger, E. Neuwirth, R. M. Heiberger, and E. Neuwirth. Polynomial regression. *R Through Excel: A Spreadsheet Interface for Statistics, Data Analysis, and Graphics*, pages 269–284, 2009.

[168] P. Royston and D. G. Altman. Regression using fractional polynomials of continuous covariates: parsimonious parametric modelling. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 43(3):429–453, 1994.

[169] T. Shanableh and K. Assaleh. Feature modeling using polynomial classifiers and stepwise regression. *Neurocomputing*, 73(10-12):1752–1759, 2010.

[170] S. E. Gano, J. E. Renaud, J. D. Martin, and T. W. Simpson. Update strategies for kriging models used in variable fidelity optimization. *Structural and Multidisciplinary Optimization*, 32(4):287–298, 2006. ISSN 1615-1488. doi: 10.1007/s00158-006-0025-y. URL `https://doi.org/10.1007/s00158-006-0025-y`.

[171] M. L. Stein. *Interpolation of spatial data: some theory for kriging*. Springer Science & Business Media, 1999.

[172] C. E. Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

[173] I. Bilionis and N. Zabaras. Multi-output local gaussian process regression: Applications to uncertainty quantification. *Journal of Computational Physics*, 231(17):5718–5746, 2012.

[174] J. Bernardo, J. Berger, A. Dawid, A. Smith, et al. Regression and classification using gaussian process priors. *Bayesian statistics*, 6:475, 1998.

[175] E. L. Snelson. *Flexible and efficient Gaussian process models for machine learning*. University of London, University College London (United Kingdom), 2008.

[176] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[177] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[178] I. Steinwart and A. Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[179] S. Karamizadeh, S. M. Abdullah, M. Halimi, J. Shayan, and M. javad Rajabi. Advantage and drawback of support vector machine functionality. In *2014 international conference on computer, communications, and control technology (I4CT)*, pages 63–65. IEEE, 2014.

[180] T. Maszczyk and W. Duch. Support feature machines: Support vectors are not enough. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2010.

[181] L. Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[182] Z. Zhang. Introduction to machine learning: k-nearest neighbors. *Annals of Translational Medicine*, 4(11):218, 2016.

[183] N. Bhatia. Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085*, 2010.

[184] K. Taunk, S. De, S. Verma, and A. Swetapadma. A brief review of nearest neighbor algorithm for learning and classification. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 1255–1260. IEEE, 2019.

[185] G. E. Hinton. Connectionist learning procedures. In *Machine learning*, pages 555–610. Elsevier, 1990.

[186] K. Gurney. *An introduction to neural networks*. CRC press, 2018.

[187] C. M. Bishop. Neural networks and their applications. *Review of scientific instruments*, 65(6): 1803–1832, 1994.

[188] N. M. Nawi, W. H. Atomi, and M. Z. Rehman. The effect of data pre-processing on optimized training of artificial neural networks. *Procedia Technology*, 11:32–39, 2013.

[189] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[190] F. Ghasemi, A. Mehridehnavi, A. Pérez-Garrido, and H. Pérez-Sánchez. Neural network and deep-learning algorithms used in qsar studies: merits and drawbacks. *Drug discovery today*, 23 (10):1784–1790, 2018.

[191] H. M. Gomes, J. P. Barddal, F. Enembreck, and A. Bifet. A survey on ensemble learning for data stream classification. *ACM Computing Surveys (CSUR)*, 50(2):1–36, 2017.

[192] G. Louppe and P. Geurts. Ensembles on random patches. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part I 23*, pages 346–361. Springer, 2012.

[193] L. Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[194] J. H. Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4): 367–378, 2002.

[195] C. Bentéjac, A. Csörgő, and G. Martínez-Muñoz. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54:1937–1967, 2021.

[196] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[197] G. Seni and J. Elder. *Ensemble methods in data mining: improving accuracy through combining predictions*. Morgan & Claypool Publishers, 2010.

[198] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[199] M. Hossin and M. N. Sulaiman. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process*, 5(2):1, 2015.

[200] Muthukrishnan. Understanding the classification report in sklearn, 2020. URL `https://muthu.co/understanding-the-classification-report-in-sklearn/`. Accessed: 2024-07-25.

[201] Z. Qu, X. Zhang, Y. Gao, C. Peng, Y. Wang, and P. M. Georgievitch. Detection of false data injection attack in agc system based on random forest. *Machines*, 11(1):83, 2023.

[202] D. Chicco, M. J. Warrens, and G. Jurman. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *Peerj computer science*, 7:e623, 2021.

[203] A. Masood and K. Ahmad. A model for particulate matter (pm2. 5) prediction for delhi based on machine learning approaches. *Procedia Computer Science*, 167:2101–2110, 2020.

[204] A. Zheng and A. Casari. *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc.", 2018.

[205] P. Schratz, J. Muenchow, E. Iturritxa, J. Richter, and A. Brenning. Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. *Ecological Modelling*, 406:109–120, 2019.

[206] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.

[207] M. Kuhn and K. Johnson. *Feature Engineering and Selection: A Practical Approach for Predictive Models*. Chapman and Hall/CRC, 2019.

[208] P. Juszczak, D. Tax, and R. P. Duin. Feature scaling in support vector data description. In *Proc. asci*, pages 95–102. Citeseer, 2002.

[209] C. C. Aggarwal. *Data Mining: The Textbook*, volume 1. Springer, New York, 2015.

[210] A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Inc, 2019.

[211] I. Jolliffe. Principal component analysis. In *Encyclopedia of Statistics in Behavioral Science*. Wiley, 2005.

[212] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[213] G. E. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, 2004.

[214] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24. NeurIPS, 2011.

[215] L. Yang and A. Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 2020.

[216] P. Probst, M. N. Wright, and A.-L. Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3):e1301, 2019.

[217] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(2):281–305, 2012.

[218] Y. Bengio. Gradient-Based Optimization of Hyperparameters. *Neural Computation*, 12(8):1889–1900, aug 2000. ISSN 0899-7667. doi: 10.1162/089976600300015187. URL `https://doi.org/10.1162/089976600300015187`.

[219] D. J. Toal, N. W. Bressloff, and A. J. Keane. Kriging hyperparameter tuning strategies. *AIAA journal*, 46(5):1240–1252, 2008.

[220] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng. Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019.

[221] J.-Y. Li, Z.-H. Zhan, J. Xu, S. Kwong, and J. Zhang. Surrogate-assisted hybrid-model estimation of distribution algorithm for mixed-variable hyperparameters optimization in convolutional neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(5):2338–2352, 2021.

[222] Y. Zhang, Z. O'Neill, B. Dong, and G. Augenbroe. Comparisons of inverse modeling approaches for predicting building energy performance. *Building and Environment*, 86:177–190, 2015.

[223] C. Benavente-Peces and N. Ibadah. Buildings energy efficiency analysis and classification using various machine learning technique classifiers. *Energies*, 13(13):3497, 2020.

[224] L. Magnier and F. Haghighat. Multiobjective optimization of building design using trnsys simulations, genetic algorithm, and artificial neural network. *Building and Environment*, 45(3):739–746, 2010.

[225] W.-L. Loh. On latin hypercube sampling. *The Annals of Statistics*, 24(5):2058–2080, 1996.

[226] P. Westermann, M. Welzel, and R. Evins. Using a deep temporal convolutional network as a building energy surrogate model that spans multiple climate zones. *Applied Energy*, 278:115563, 2020.

[227] T. Wortmann. Model-based optimization for architectural design: Optimizing daylight and glare in grasshopper. *Technology — Architecture + Design*, 1(2):176–185, 2017.

[228] A. Costa and G. Nannicini. Rbfopt: An open-source library for black-box optimization with costly function evaluations. *Mathematical Programming Computation*, 10:597–629, 2018.

[229] F. Khayatian, L. Sarto, and G. Dall'O'. Application of neural networks for evaluating energy performance certificates of residential buildings. *Energy and Buildings*, 125:45–54, 2016. ISSN 03787788. doi: 10.1016/j.enbuild.2016.04.067. URL `http://dx.doi.org/10.1016/j.enbuild.2016.04.067`.

[230] G. Aouad, S. Wu, A. Lee, and T. Onyenobi. *Computer Aided Design Guide for Architecture, Engineering and Construction*. Routledge, London, 2011. ISBN 9780203878750. doi: https://doi.org/10.4324/9780203878750.

[231] A. S. Denzer and K. E. Hedges. From CAD to BIM : Educational. In *AEI 2008: Building Integration Solutions*, pages 1–11, Denver, 2008. doi: https://doi.org/10.1061/41002(328)6.

[232] Graphics and CAD Blog. Graphics and CAD Blog, 2024. URL https://graphics-cad.blogspot.com/. [Accessed: December 19, 2024].

[233] MIT City Form Lab. UNA Rhino Toolbox, 2024. URL https://cityform.mit.edu/projects/una-rhino-toolbox. [Accessed: December 19, 2024].

[234] Ladybug Tools. Ladybug Tools Discourse Forum, 2024. URL https://discourse.ladybug.tools/. [Accessed: December 19, 2024].

[235] McNeel Wiki. Rhino Level 2 Training for Architecture, 2024. URL https://wiki.mcneel.com/training/rhino_for_arch/l2. [Accessed: December 19, 2024].

[236] Radiance Development Team. Radiance Lighting Simulator. Computer software, 2021. URL http://www.radiance-online.org/. version 5.4.947ea88a29.

[237] G. R. Araújo. Integrating algorithmic processes in informal urban and architectural planning. Master's thesis, Instituto Superior Técnico, 2020.

[238] H. Martinho, G. Araújo, and A. Leitão. From macro to micro. *Computer-Aided Architectural Design Research (CAADRIA)*, 2020.

[239] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017. doi: 10.1137/141000671. URL https://doi.org/10.1137/141000671.

[240] R. Castelo Branco, I. Caetano, and A. Leitão. Algorithmic representation space. *Prospectives*, 2, 2022.

[241] Project Jupyter. Jupyter Notebooks. Computer software, 2014. URL https://jupyter.org/.

[242] Wes McKinney. Pandas. Computer software, 2023. URL https://pandas.pydata.org/. Version 2.1.0.

[243] Travis Oliphant. Jupyter Notebooks. Computer software, 2023. URL https://numpy.org/. Version 1.25.1.

[244] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[245] Streamlit Inc. Streamlit. Computer software, 2023. URL https://streamlit.io/. Version 1.8.0.

[246] G. Araújo, R. Gomes, P. Ferrão, and M. G. Gomes. Optimizing building retrofit through data analytics: A study of multi-objective optimization and surrogate models derived from energy performance certificates. *Energy and Built Environment*, 2023.

[247] G. Araujo, L. Santos, A. Leitão, and R. Gomes. Ad based surrogate models for simulation and optimization of large urban areas. In *27th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA 2022)*, 2022.

[248] G. Araujo, H. Teixeira, M. Gloria Gomes, and A. M. Rodrigues. Multi-objective optimization of thermochromic glazing properties to enhance building energy performance. *Solar Energy*, 249 (October 2022):446–456, 2023. ISSN 0038092X. doi: 10.1016/j.solener.2022.11.043. URL https://doi.org/10.1016/j.solener.2022.11.043.

[249] G. R. Araújo, R. Gomes, M. G. Gomes, M. C. Guedes, and P. Ferrão. Surrogate models for efficient multi-objective optimization of building performance. *Energies*, 16(10):4030, 2023.

[250] Cam Nugent. California Housing Prices. Dataset, 2017. URL https://www.kaggle.com/datasets/camnugent/california-housing-prices.

[251] R. Castelo-Branco and A. Leitão. Comprehending algorithmic design. In D. Gerber, E. Pantazis, B. Bogosian, A. Nahmad, and C. Miltiadis, editors, *Computer-Aided Architectural Design. Design Imperatives: The Future is Now*, CAAD Futures 2021, pages 15–35. Springer, Cham, Singapore, 2022. ISBN 978-981-19-1280-1. doi: 10.1007/978-981-19-1280-1_2. URL https://link.springer.com/chapter/10.1007/978-981-19-1280-1_2.

[252] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[253] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26:159–190, 2006.

[254] N. Japkowicz. The class imbalance problem: Significance and strategies. In *Proc. of the Int'l Conf. on artificial intelligence*, volume 56, pages 111–117, 2000.

[255] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[256] X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou. On the class imbalance problem. In *2008 Fourth international conference on natural computation*, volume 4, pages 192–201. IEEE, 2008.

[257] D. Li, B. Zhang, and C. Li. A feature-scaling-based $k$-nearest neighbor algorithm for indoor positioning systems. *IEEE Internet of Things Journal*, 3(4):590–597, 2015.

[258] G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL `http://jmlr.org/papers/v18/16-365`.

[259] T. Hasanin and T. M. Khoshgoftaar. The effects of random undersampling with simulated class imbalance for big data. *Proceedings - 2018 IEEE 19th International Conference on Information Reuse and Integration for Data Science, IRI 2018*, pages 70–79, 2018. doi: 10.1109/IRI.2018.00018.

[260] I. Tomek. Two modifications of cnn. *Systems, Man, and Cybernetics*, 6:769–772, 1976.

[261] I. B. Mohamad and D. Usman. Research article standardization and its effects on k-means clustering algorithm. *Research Journal of Applied Sciences, Engineering and Technology*, 6(17):3299–3303, 2013.

[262] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[263] J. Mourão, R. Gomes, L. Matias, and S. Niza. Combining embodied and operational energy in buildings refurbishment assessment. *Energy and Buildings*, 197:34–46, aug 2019. ISSN 03787788. doi: 10.1016/j.enbuild.2019.05.033.

[264] O. Pasichnyi, J. Wallin, F. Levihn, H. Shahrokni, and O. Kordas. Energy performance certificates — New opportunities for data-enabled urban energy policy instruments? *Energy Policy*, 127 (November 2018):486–499, 2019. ISSN 03014215. doi: 10.1016/j.enpol.2018.11.051.

[265] L. Pérez-Lombard, J. Ortiz, R. González, and I. R. Maestre. A review of benchmarking, rating and labelling concepts within the framework of building energy certification schemes. *Energy and Buildings*, 41(3):272–278, 2009. ISSN 03787788. doi: 10.1016/j.enbuild.2008.10.004.

[266] European Parlament. Directiva 2010/31/UE do Parlamento Europeu e do Conselho, de 19 de Maio de 2010 , relativa ao desempenho energético dos edifícios. *Jornal Oficial nº L 153*, pages 13–35, 2010.

[267] European Parliament. Directiva 2018/844 do parlamento europeu e do conselho de 30 de maio de 2018 que altera a Diretiva 2010/31/UE relativa ao desempenho energético dos edifícios e a Diretiva 2012/27/UE sobre a eficiência energética. *Jornal Oficial da União Europeia*, L156:75–91, 2018.

[268] European Parliament. Energy Performance of Buildings Directive - EPBD (2002/91/EC), 2002.

[269] T. Nikolaou, D. Kolokotsa, and G. Stavrakakis. Review on methodologies for energy benchmarking, rating and classification of buildings. *Advances in Building Energy Research*, 5(1):53–70, 2011. ISSN 17512549. doi: 10.1080/17512549.2011.582340.

[270] M. A. Brown, A. Soni, M. V. Lapsa, K. Southworth, and M. Cox. High energy burden and low-income energy affordability: Conclusions from a literature review. *Progress in Energy*, 2(4): 042003, 2020.

[271] K. T. Gillingham, P. Huang, C. Buehler, J. Peccia, and D. R. Gentner. The climate and health benefits from intensive building energy efficiency improvements. *Science Advances*, 7(34):eabg0947, 2021.

[272] C. Reinhart, Z. Berzolla, H. D. Jacoby, J. M. Reilly, and C. Schuler. Thoughts on a federal government stimulus package for buildings. *Available at SSRN 3832381*, 2021.

[273] Y. Nidam, A. Irani, J. Bemis, and C. Reinhart. Census-based urban building energy modeling to evaluate the effectiveness of retrofit programs. *Environment and Planning B: Urban Analytics and City Science*, page 23998083231154576, 2023.

[274] A. Ferrantelli, J. Belikov, E. Petlenkov, M. Thalfeldt, and J. Kurnitski. Evaluating the energy readiness of national building stocks through benchmarking. *IEEE Access*, 10:45430–45443, 2022.

[275] C. Watts, M. F. Jentsch, and P. A. James. Evaluation of domestic Energy Performance Certificates in use. *Building Services Engineering Research and Technology*, 32(4):361–376, 2008. doi: 10.1177/0143624411404486.

[276] A. Hardy and D. Glew. An analysis of errors in the Energy Performance certificate database. *Energy Policy*, 129:1168–1178, jun 2019. ISSN 03014215. doi: 10.1016/j.enpol.2019.03.022.

[277] L. M. López-González, L. M. López-Ochoa, J. Las-Heras-Casas, and C. García-Lozano. Update of energy performance certificates in the residential sector and scenarios that consider the impact of automation, control and management systems: A case study of La Rioja. *Applied Energy*, 178 (2016):308–322, 2016. ISSN 03062619. doi: 10.1016/j.apenergy.2016.06.028.

[278] J. P. Gouveia and P. Palma. Harvesting big data from residential building energy performance certificates: Retrofitting and climate change mitigation insights at a regional scale. *Environmental Research Letters*, 14(9), 2019. ISSN 17489326. doi: 10.1088/1748-9326/ab3781.

[279] E. ISO. En iso 52000-1, energy performance of buildings — overarching epb assessment – part 1: General framework and procedures (2017). *International Organization of Standardization: Geneva, Switzerland*, 2017.

[280] E. C. for Standardization. En iso 52003-1, epb – indicators, requirements, ratings and certificates – part 1: General aspects and application to the overall energy performance (2017), 2017.

[281] E. C. for Standardization. En iso 52010-1 epb, external climatic conditions – part 1: Conversion of climatic data for energy calculations (2017)., 2017.

[282] E. C. for Standardization. En iso 52016-1 epb, energy needs for heating and cooling, internal temperatures and sensible and latent heat loads – part 1: Calculation procedures(2017)., 2017.

[283] E. C. for Standardization. En iso 52018-1, epb – indicators for partial epb requirements related to thermal energy balance and fabric features – part 1: Overview of options (2017), 2017.

[284] European Comission. DIRECTIVE 2010/31/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 19 May 2010 on the energy performance of buildings, 2010.

[285] D. Van Dijk and J. Hogeling. The new EN ISO 52000 family of standards to assess the energy performance of buildings put in practice. *E3S Web of Conferences*, 111(May):6–14, 2019. ISSN 22671242. doi: 10.1051/e3sconf/201911104047.

[286] C. Buratti, M. Barbanera, and D. Palladino. An original tool for checking energy performance and certification of buildings by means of Artificial Neural Networks. *Applied Energy*, 2014. doi: 10.1016/j.apenergy.2014.01.053. URL `http://dx.doi.org/10.1016/j.apenergy.2014.01.053`.

[287] Y. Fan and X. Xia. Building retrofit optimization models using notch test data considering energy performance certificate compliance. *Applied Energy*, 228(April):2140–2152, 2018. ISSN 03062619. doi: 10.1016/j.apenergy.2018.07.043.

[288] T. G. Reames, M. A. Reiner, and M. B. Stacey. An incandescent truth: Disparities in energy-efficient lighting availability and prices in an urban us county. *Applied energy*, 218:95–103, 2018.

[289] POSEUR. Programa Operacional de Sustentabilidade e Eficiência no Uso de Recursos. `https://poseur.portugal2020.pt/`, 2020. [Online; accessed 15-January-2020].

[290] Eurostat. Electricity price statistics. `https://ec.europa.eu/eurostat/`, 2020. [Online; accessed 15-January-2020].

[291] Portuguese Republic Diary. Decree-Law nº 80/2006, 2006.

[292] S. Begum, D. Chakraborty, and R. Sarkar. Data classification using feature selection and knn machine learning approach. In *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, pages 811–814. IEEE, 2015.

[293] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. De Sousa. Ensemble approaches for regression: A survey. *ACM Computing Surveys*, 45(1), 2012. ISSN 03600300. doi: 10.1145/2379776.2379786.

[294] J. Feng and S. Lu. Performance Analysis of Various Activation Functions in Artificial Neural Networks. *Journal of Physics: Conference Series*, 1237(2), 2019. ISSN 17426596. doi: 10.1088/1742-6596/1237/2/022030.

[295] A. Pinkus. Approximation theory of the MLP model in neural networks. In *Acta Numerica*, volume 8, pages 143–195. Cambridge University Press, 1999. doi: 10.1017/S0962492900002919.

[296] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4): 367–378, 2002. ISSN 01679473. doi: 10.1016/S0167-9473(01)00065-2.

[297] R. Alizadeh, J. K. Allen, and F. Mistree. Managing computational complexity using surrogate models: a critical review. *Research in Engineering Design*, 31(3):275–298, 2020. ISSN 14356066. doi: 10.1007/s00163-020-00336-7. URL https://doi.org/10.1007/s00163-020-00336-7.

[298] T. Østergård, R. L. Jensen, and S. E. Maagaard. A comparison of six metamodeling techniques applied to building performance simulations. *Applied Energy*, 211(October 2017):89–103, 2018. ISSN 03062619. doi: 10.1016/j.apenergy.2017.10.102.

[299] T. Fushiki. Estimation of prediction error by using k-fold cross-validation. *Statistics and Computing*, 21:137–146, 2011.

[300] B. Butcher and B. J. Smith. Feature Engineering and Selection: A Practical Approach for Predictive Models. *The American Statistician*, 74(3):308–309, 2020. ISSN 0003-1305. doi: 10.1080/00031305.2020.1790217. URL https://doi.org/10.1080/00031305.2020.1790217.

[301] J. Kang, K. R. Ryu, and H. C. Kwon. Using cluster-based sampling to select initial training set for active learning in text classification. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3056:384–388, 2004. ISSN 16113349. doi: 10.1007/978-3-540-24775-3_46.

[302] S. J. Yen and Y. S. Lee. Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, 36(3 PART 1):5718–5727, 2009. ISSN 09574174. doi: 10.1016/j.eswa.2008.06.108. URL http://dx.doi.org/10.1016/j.eswa.2008.06.108.

[303] F. Olken and D. Rotem. Random sampling from database files: A survey. *Statistics and Computing*, 5:25–42, 1995. doi: 10.1007/3-540-52342-1_23.

[304] M. Ferrando, F. Causone, T. Hong, and Y. Chen. Urban building energy modeling (UBEM) tools: A state-of-the-art review of bottom-up physics-based approaches. *Sustainable Cities and Society*, 62:102408, nov 2020. ISSN 2210-6707. doi: 10.1016/J.SCS.2020.102408.

[305] Y. Chen, T. Hong, and M. A. Piette. Automatic generation and simulation of urban building energy models based on city datasets for city-scale building retrofit analysis. *Applied Energy*, 205(April): 323–335, 2017. ISSN 03062619. doi: 10.1016/j.apenergy.2017.07.128. URL http://dx.doi.org/10.1016/j.apenergy.2017.07.128.

[306] C. Dos Santos and L. Matias. Coeficientes de transmissão térmica de elementos da envolvente dos edifícios (lnec (ed.). *Laboratório Nacional de Engenharia Civil, LNEC*, 2006.

[307] H. Ye, X. Meng, and B. Xu. Theoretical discussions of perfect window, ideal near infrared solar spectrum regulating window and current thermochromic window. *Energy and Buildings*, 49:164–172, 2012. ISSN 03787788. doi: 10.1016/j.enbuild.2012.02.011. URL http://dx.doi.org/10.1016/j.enbuild.2012.02.011.

[308] E. Cuce and S. B. Riffat. A state-of-the-art review on innovative glazing technologies. *Renewable and Sustainable Energy Reviews*, 41:695–714, 2015. ISSN 13640321. doi: 10.1016/j.rser.2014.08.084. URL http://dx.doi.org/10.1016/j.rser.2014.08.084.

[309] S. D. Rezaei, S. Shannigrahi, and S. Ramakrishna. A review of conventional, advanced, and smart glazing technologies and materials for improving indoor environment. *Solar Energy Materials and Solar Cells*, 159:26–51, 2017. ISSN 09270248. doi: 10.1016/j.solmat.2016.08.026. URL http://dx.doi.org/10.1016/j.solmat.2016.08.026.

[310] M. H. Lee. Thermochromic glazing of windows with better luminous solar transmittance. *Solar Energy Materials and Solar Cells*, 71(4):537–540, 2002. ISSN 09270248. doi: 10.1016/S0927-0248(01)00135-0.

[311] M. Saeli, C. Piccirillo, I. P. Parkin, R. Binions, and I. Ridley. Energy modelling studies of thermochromic glazing. *Energy and Buildings*, 42(10):1666–1673, 2010. ISSN 03787788. doi: 10.1016/j.enbuild.2010.04.010. URL http://dx.doi.org/10.1016/j.enbuild.2010.04.010.

[312] M. Aburas, V. Soebarto, T. Williamson, R. Liang, H. Ebendorff-Heidepriem, and Y. Wu. Thermochromic smart window technologies for building application: A review. *Applied Energy*, 255 (June):113522, 2019. ISSN 03062619. doi: 10.1016/j.apenergy.2019.113522. URL https://doi.org/10.1016/j.apenergy.2019.113522.

[313] R. Liang, Y. Wu, and R. Wilson. Thermal and Visual Comfort Analysis of an Office With Thermochromic Smart Windows Applied. *Infoscience.Epfl.Ch*, 2015. URL https://infoscience.epfl.ch/record/213310/files/2_LIANG.pdf.

[314] R. Tällberg, B. P. Jelle, R. Loonen, T. Gao, and M. Hamdy. Comparison of the energy saving potential of adaptive and controllable smart windows: A state-of-the-art review and simulation studies of thermochromic, photochromic and electrochromic technologies. *Solar Energy Materials and Solar Cells*, 200(February):109828, 2019. ISSN 09270248. doi: 10.1016/j.solmat.2019.02.041. URL https://doi.org/10.1016/j.solmat.2019.02.041.

[315] H. Teixeira, M. Glória Gomes, A. Moret Rodrigues, and D. Aelenei. Assessment of the visual, thermal and energy performance of static vs thermochromic double-glazing under different European climates. *Building and Environment*, 217(February):109115, 2022. ISSN 03601323. doi: 10.1016/j.buildenv.2022.109115.

[316] M. E. Warwick, I. Ridley, and R. Binions. The effect of transition gradient in thermochromic glazing systems. *Energy and Buildings*, 77:80–90, 2014. ISSN 03787788. doi: 10.1016/j.enbuild.2014.03.044. URL http://dx.doi.org/10.1016/j.enbuild.2014.03.044.

[317] M. E. Warwick, I. Ridley, and R. Binions. The effect of variation in the transition hysteresis width and gradient in thermochromic glazing systems. *Solar Energy Materials and Solar Cells*, 140 (June):253–265, 2015. ISSN 09270248. doi: 10.1016/j.solmat.2015.04.022.

[318] M. C. Peel, B. L. Finlayson, and T. A. McMahon. Updated world map of the Koppen-Geiger climate classificatio. *Hydrology and earth system sciences*, 11(5):1633–1644, 2007.

[319] T. Bäck and H.-P. Schwefel. An Overview of Evolutionary Algorithms for Parameter Optimization. *Evolutionary Computation*, 1(1):1–23, 1993. ISSN 1063-6560. doi: 10.1162/evco.1993.1.1.1.

[320] J. Kennedy and R. Eberhart. Particle Swarm Optimisation. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, pages 1942–1948, 1995. ISBN 0780327683. doi: 10.1007/978-3-030-61111-8_2.

[321] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014. ISSN 1089778X. doi: 10.1109/TEVC.2013.2281535.

[322] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. *ACM International Conference Proceeding Series*, 148:161–168, 2006. doi: 10.1145/1143844.1143865.

[323] USDOE. EnergyPlus Engineering Reference. Technical report, US Department of Energy, 2021.

[324] K. J. Lomas and H. Eppel. Sensitivity analysis techniques for building thermal simulation programs. *Energy and buildings*, 19(1):21–44, 1992.

[325] I. Pereira and A. Leitão. More is more: The no free lunch theorem in architecture. *Proceedings of the International Conference of Architectural Science Association*, 2020-Novem(November 2020): 765–774, 2020. ISSN 22093850.

[326] K. Deb and R. B. Agrawal. Simulated Binary Crossover for Continuous Search Space The crossover operator is believed to be the main search operator in the working of a genetic. *Complex Systems*, 9(2):115–148, 1994.

[327] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[328] M. R. Sierra and C. A. Coello Coello. Improving PSO-based Multi-Objective optimization using crowding, mutation and $\epsilon$-dominance. *Lecture Notes in Computer Science*, 3410:505–519, 2005. ISSN 03029743. doi: 10.1007/978-3-540-31880-4_35.