UNIVERSITÀ DI PARMA

UNIVERSITÀ DEGLI STUDI DI PARMA

DOTTORATO DI RICERCA IN INGEGNERIA INDUSTRIALE
CICLO XXXVI

Similitude approach for buildings energy forecasts

Coordinatore:
Chia.mo Prof. Gianni Royer CARFAGNI

Supervisore:
Chia.mo Prof. Agostino GAMBAROTTA

Tutore:
Chia.mo Prof. Mirko MORINI

Autore:
Andrea VIERI

Anni accademici 2020/2021 - 2024/2025

July 31, 2025

Declaration of Authorship

I, Andrea VIERI, declare that this thesis titled, “Similitude approach for buildings energy forecasts” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

_____

Date:

_____

"Rem tene, verba sequentur"

Cato

Acknowledgements

The conclusion of this PhD course, encapsulated in the thesis I defended and which you will read, in addition to the desire to contribute to the increase of knowledge and the advancement of mankind, also represents an important personal achievement. Beyond the satisfaction of its completion, this work has given me the happiness of completing a journey that I wanted to start almost 7 years ago when I completed my Master's degree. At that time, unfortunately or fortunately, events evolved in such a way as to convince me to hit other roads, but over the years the desire to be able to embark on such a path and hopefully complete it was still alive in me. I felt that something was missing, that the work was not yet done. And so, with this desire alive in me, I have looked over the years for the opportunity to pick up where I left off and, for these reasons, I would like to thank the people I will list below, because without each of them this would probably simply not have been possible and I would not be here today:

Prof.Gambarotta, for accepting my project proposal and giving me this opportunity. Despite knowing that it would be a long and hard journey, at every moment I felt his trust and sincere interest in what I was developing. Beyond the personal motivations and their soundness for which I embarked on this path, I am grateful for the support he gave me that I feel was important for the growth of this work.

Prof. Morini, for the opportunities of confrontation that it offered me, of which I felt enriched each time and experienced at each moment as fair, free to express my analyses and considerations, knowing that they would be evaluated for their merit. All this, gave me the confidence and sense of responsibility that allowed me to live this path to its fullest and complete it with the awareness that I had given the best and the most I could give.

Prof. Saletti, for the time she dedicated to my thesis work and articles, sharing so much much good experience to me and allowing this work to improve and grow

well day by day.

Michele R., without whose support it would not have been possible for me to be here today. Thank you for believing in me.

Siram Veolia, or as I like to call it “Mom Siram,” and all the colleagues, who supported me by providing the information I needed to complete my work and sharing with me their experience about buildings energy demands, which has been of great value for this thesis work.

My love Domitilla, for all the weekends and free time she sacrificed to stand by me. Thank you my love, for understanding me and realizing how much important it was.

My parents and sister, who most of all contributed to make me the person I am today and who accepted the sacrifices due to my being away, without ever complaining, contributing to giving me the serenity needed to complete a journey like this.

My friends Alessandro A., Alessandro B., Roberto C., you have set an example and you have been a push for me to improve myself every day. Without you, I probably would not be here today.

My friend Mirko M., with whom I spent so many hours training for our univerisity exams. Always discreet, you supported and put up with me at all times. Without you, I would not have retained the enthusiasm in the last years of graduation. Thank you for being there for me, my friend.

Matteo M., with whom I started my university journey and who in the face of my difficulties always helped me, pushing me to improve and not give up. Sometimes you made fun of me, only today I understand that you were doing it to make me overcome my limits. In many moments, you believed in me more than I did. Also

to you, I dedicate this achievement. If you had gone your way 12 years ago without telling me anything, I probably would not be here today.

Roberto B., for everything he taught me, not only about engineering, but also and especially as a person. If I have come this far, it is because, I am sure, you have watched over my path. This is the flower I promised I would bring you one day, in gratitude and memory for all you have done for me.

My grandparents Romano, Venanzio and Immacolata and my uncle Maurizio, who are no longer here with me. I am sure you too would be proud of me today.

UNIVERSITÀ DEGLI STUDI DI PARMA

Abstract

Mechanical Engineering
Department or Industrial Engineering

Doctor of Philosophy

Similitude approach for buildings energy forecasts

by Andrea VIERI

Buildings are assets characterized by environments and uses that change over time, variable occupancies, and long life cycles. In addition to this, they have high operational costs, mostly due to their energy demands, accounting for 30% to 40% of global greenhouse gas emissions. All of this makes them as complex as challenging to manage in an efficient way. Consequently, increasing efforts have been made to forecast their energy needs, with the scope of optimize their economic and environmental impact. About this, the available literature focused mainly on short-term modeling through the implementation of sets of physics-based equations (i.e., white-box), functional relationships between input and output variables (i.e., black-box), or a combination of both (i.e., grey-box). Far from reaching a state of art condition, more research is necessary on long-term forecast models, especially with the aim of reducing the energy needs than optimize the energy production. Within this context, this thesis presents an original automatic and integrated similitude approach for forecasting the energy needs of buildings from short to long-term time horizons. This is accomplished by scaling an unknown facility from a similar facility that is already known and by executing a black-box approach based on machine learning algorithms. The proposed method is implemented in real case studies in Italy to predict the energy demands (i.e., heating, cooling, and electricity) of Sant'Anna Hospital in Ferrara through the historical data of Ca' Foncello Hospital in Treviso. The results show an adjusted coefficient of correlation above 0.7 and an average error below 10% for all the energy demands, demonstrating a feasible forecast performance with a low training set-to-test set ratio.

Contents

List of Figures

List of Tables

List of Abbreviations

BMS	**B**uilding **M**anagement **S**ystem
BEM	**B**uilding **E**nergy **M**odel
RLF	**R**esidential **L**oad **F**actor
RHB	**R**esidential **H**eat **B**alance
LFs	**L**oad **F**actors
CFs	**C**ooling **F**actors
HFs	**H**eating **F**actors
l	*l*enght
O	*O*rder of complexity
SSE	**S**um **o**f Squared **E**rrors
SS	**S**um **o**f Squares
DT	**D**ecision **T**rees
ID3	**I**terative **D**ichotomiser **3**
REP	**R**educed **E**rror **P**runing
CCP	**C**ost **C**omplexity **P**runing
SVM	**S**upport **V**ector **M**achine
SVR	**S**upport **V**ector **R**egression
SVs	**S**upport **V**ectors
KKT	**K**arush- **K**uhn- **T**ucker
ANNs	**A**rtificial **N**eural **N**etworks
MLPs	**M**ulti **L**ayer **P**erceptrons
LSTM	**L**ong **S**hort- **T**erm **M**emory
PSO	**P**article **S**warm **O**ptimization
GA	**G**enetic **A**lgorithm
DE	**D**ifferential **E**volution

EA	**E**volutionary **A**lgorithms
BECMF	**B**uilding **E**nergy**C**onsumption**C**onsumption**M**odeling and**F**orecasting
MSE	**M**ean **S**squared **E**rror
RMSE	**R**oot **M**ean **S**squared **E**rror
MAE	**M**ean **A**bsolute **E**rror
MdAE	**M**edian **A**bsolute **E**rror
MAPE	**M**ean **A**bsolute **P**ercentage **E**rror
MdAPE	**M**edian **A**bsolute **P**ercentage **E**rror
RMSPE	**R**oot **M**ean **S**quared **P**ercentage **E**rror
RMdSPE	**R**oot **M**edian **S**quared **P**ercentage **E**rror
MRAE	**M**ean **R**elative **A**bsolute **E**rror
MdRAE	**M**edian **R**elative **A**bsolute **E**rror
GMRAE	**G**eometric **M**ean **R**elative **A**bsolute **E**rror
GMdRAE	**G**eometric **M**edian **R**elative **A**bsolute **E**rror
RelMAE	**Rel**ative **M**ean **A**bsolute **E**rror
PB	**P**ercent **B**etter
MBE	**M**ean**B**ias**E**rror
NMBE	**N**ormalized **M**ean**B**ias**E**rror
FSSRS	**F**ixed **S**tep**R**andom**S**earch
OSSRS	**O**ptimum **S**tep**S**ize**R**andom**S**earch
ASSRS	**A**daptive **S**tep**S**ize**R**andom**S**earch
ORSSRS	**O**ptimized **R**elative**S**tep**S**ize**R**andom**S**earch
SMBO	**S**equential **M**odel-**B**ased**O**ptimization
EI	**E**xpected **I**mprovement
PI	**P**robability of **I**mprovement
UCB	**U**pper **C**onfidence **B**ound

Physical Constants

List of Symbols

U	Thermal transmittance	$\mathrm{W\,m^{-2}\,K^{-1}}$
ΔT	Temperature difference	K
Y_i	Outcome	-
$x_{i,m}^m$	Predictors or input variables	-
α_i, β_m	Polynomial coefficients	-
ϵ_i	Random Error	-
R^2	Coefficient of determination	-
$R^2_{adjusted}$	Adjusted Coefficient of determination	-
N	Number of samples	-
p	Number of predictors	-
H(S)	Information entropy	-
IG(S,A)	Information gain	-
A	Feature column	-
p^2	Probability of success	-
q^2	Probability of failusre	-
$f_{observed,i}$	Observed frequencies	-
$f_{expected,i}$	Expected frequencies	-
σ	Variance	-
T	Decision Tree set of leaves of the tree	-
$R(T)$	Training/learning error	-
α	Decision Tree Regularization parameter	-
$r(T)$	Misclassification or prediction error	-
$F(x_i)$	Decision Function	-
b	bias	-

ϕ_i	SVR nonlinear mapping of the input space to a high-dimensional feature space	-
ξ_i, ξ_i^*	SVR positive slack variables	
C	SVR regularization constant	-
L	Lagrangian Function	-
$\eta_i, \eta_i^*, a_i, a_i^*$	Lagrangian multipliers	-
$K(x_i, x_j)$	Kernel Function	-
$\gamma(\gamma > 0), d$	Kernel parameters	-
e_t	error at time t	-
Y_t	observation value at time t	-
F_t	forecast value at time t	-
p_t	percentage error	%
HD_i, HD_j	Heating Demand	kWh
CD_i, CD_j	Cooling Demand	kWh
ED_i, ED_j	Electricity Demand	kWh
SD_i, SD_j	Steam Demand	kWh
V_i, V_j	Volumes	m^3
S_i, S_j	Surfaces	m^2
B_i, B_j	Beds	-
HDD_i, HDD_j	Heating Degree Days	°C d
CDD_i, CDD_j	Cooling Degree Days	°C d
$k_{HD_{i,j}}, k_{CD_{i,j}}, k_{ED_{i,j}}, k_{SD_{i,j}}$	Similarity constant	-
M	Number of hyperparameters	-
L	Number of trial solutions along each hyperparameter dimension	[-]

Dedicatory section

To my wife, Domitilla, who lovingly accompanied and supported me on this journey.

To my father, mother and sister, for all the sacrifices my being away from home has forced them to make.

Chapter 1

Introduction

Buildings account for approximately 40% of energy consumption and 36% of greenhouse gas emissions in the European Union, while in the world they weight for 30% and 26 % respectively[25]. This makes them the single largest energy consumer in the world. Consequently, a substantial effort has been undertaken over the time to forecast their energy demands, with the scope of optimizing their economic and environmental impact throughout all their life cycle[34]. Many efforts are carried out to understand, model, and forecast their energy consumptions, looking for new ways to improve their energy efficiency and sustainability. From an historical perspective, the early attempts to model buildings energy demands relied on engineering software packages, e.g., Energy Plus, but usually the results were not as valuable as would be expected. All of this, despite the considerable amount of time spent to setup the software and data gathered regarding the structural, geometrical and material characteristics of the target buildings. This is mainly because of anthropic loads, behaviors, and other nonlinearities, which are extremely complex and difficult to estimate, but that have significantly influence on energy demands and, therefore, consumptions. In this context, data-driven approaches represent a more efficient and effective alternative to model and forecast the energy consumption of buildings. In more detail, energy demands data are collected through smart meters and bills, then they are fed into machine learning algorithms to infer the complex relationships between energy demands and other variables, like weather, degree-days, timestamp features and building characteristics. The literature related to data-driven forecasting techniques is growing quickly, but few studies actually investigate which set of variables is more correlated with building energy demands, in order to accurately

forecast and reproduce their dynamics. This makes it a critical element for forecasting building energy demands as it represents the difference from a model able to forecast future buildings energy demands from that is not. It is worth to mention that the topic is far from having reached the state of the art condition, despite the extensive efforts and studies carried out. In fact, because of complexities inherent the variabilities and interactions that characterizes each building envelope and the vast amount of data usually needed to achieve good forecasting performance, the literature is far from having identified a definitive solution. As a matter of fact, one of the major achievement made possible by the development of artificial intelligence algorithms (e.g., machine learning and deep learning) is the opportunity to improve HVAC plant performance and load forecasting, even in face of significant uncertainty, usually hard to model through sets of equations or to investigate without an adequate set of measurement tools. The data abundance achieved though meters installations, in fact, has made possible the development of increasingly sophisticated and highly performing models and algorithms under ordinary and easily reproducible conditions. Together, this is a weakness element also, mostly for all those events hard to reproduce or predict (e.g., the COVID-19 pandemic and its resulting impact on buildings consumptions) or just when there is a lack of data (e.g., due to a breakdown or maintenance activities). In this context, it is particularly critical the usually large dataset required for training artificial intelligence-based models, especially when historical dataset is missing or is inadequate to achieve good forecasting performances. To overcome the hurdles mentioned above, in this thesis is setup a similitude based algorithm, that implements different data-driven modeling approaches deriving from artificial intelligence for building energy demands modeling and forecasting. Because each approach has its own strengths and limitations, and there is no clear winner among those available in literature, it is introduced a set of algorithms to address the complexities and uncertainties associated with building energy demands forecast. The case study focuses on Sant'Anna Hospital in Ferrara, Italy, that is assumed as target building (i.e., the building for which the energy demands are considered unknown and used for testing), while the similar building is represented by Ca' Foncello Hospital in Treviso, Italy (i.e., the building for which the energy demands are considered known and used for training). The energy demands

of these facilities are hourly-based and they include heating, cooling and electricity demands, which are analyzed through data gathered from the building management systems (BMS). In order to make the algorithm as independent as possible from the capabilities of the individual data scientist and having in mind a procedure that is as automatic as possible, the algorithm developed in this thesis incorporates automatic data pre-processing steps, like data cleaning and normalization, and hyperparameter optimization of ensemble methods. In more detail, these methods combine multiple improved models to enhance forecasting performances. The remainder of the theis is outlined as follows. Section 2 presents an overview of the available methods for forecasting building energy needs, with a particular focus on artificial intelligence based tools. Section 3 describes the methodology that enables the utilization of data from a known structure to estimate the unknown energy needs of another structure, adopting easily and economically obtainable variables. Section 4 outlines the case studies in detail, while in Section 5, the results are reported, and the final conclusions are drawn in Section 6. The primary novel contributions of this thesis work regards (i) the implementation of a similitude criteria, (ii) a data cleaning process based on plant technical specifications rather than statistical analysis, (iii) an automatic procedure for models and hyperparameters optimization selection (iii) an innovative methodology for time series analysis, and (iv) the incorporation of an automated optimization procedure for the hyperparameters tuning of the forecasting models employed.

Chapter 2

State of the art

Modeling and forecasting building energy demands is a challenging task, especially because of the many variables, their interconnections and variability over time. They are both internal and external to the target building perimeter, such as weather conditions, building envelope features, antropic factors (e.g., behavior and occupancy), lighting, electrical equipments and HVAC performance[136]. To unravel the tangle made by all of these elements concurrently affecting the problem, numerous techniques were developed, adapted and tested in recent research.

2.1 Methods for Building Energy Forecasting

The literature suggests the following unifying nomenclature in modeling processes, without any focus on building type, energy end-use distinctions, or building scale applications [20, 109]:

- White-box approaches: These models use a transparent process to solve first-principle equations describing the buildings energy behaviour.This implies that this type of approach is strongly based on the physics of the phenomena that take place within the study boundary in which the building is located. Physics-based modeling was introduced with different names, such as "engineering models"[109]. As can be easily imagined, the white-box models development of an building would require a remarkable amount of time to set the physical properties and characteristics, even for the simpler ones. In order to achieve the sufficient accuracy of building energy models (BEMs), as a starting point, it is necessary a usually time-consuming and error-prone collection of

detailed information of it. Once this step is completed, the building model is set up through its related variables and equations. Precisely because of their large number, numerous interconnections and in many cases their complexity, it is necessary to proceed numerically. This circumstance, clearly imposes the choice of an error acceptability threshold, which clearly has a direct impact on the precision, accuracy and speed of the simulations themselves. To speed up the simulation time, it is common to make use of representative buildings, called archetypes: they are usually developed after an accurate identification of the most common characteristics of different groups of analogous buildings[14]. This method analyzes portions of the buildings, producing detailed BEMs, which still preserve the ability to characterize the energy performance of the whole building portfolio with the desired level of accuracy. Moreover, it allows the accurate creation of benchmarking models at a local level through a classification process. Ultimately, another considerable benefit of this approach is the assessment of retrofit interventions by what-if scenarios[13]. However, to date, this kind of representation has required dedicated simulation engines and large amounts of data to correctly represent the buildings of a given portfolio, making the benefits listed above only potential and not real.

- Black-box approaches: These models rely on building data analysis and data mining tools to predict and forecast building energy demands and consumptions. They are mainly used for energy demand forecasting through the hierarchically important inputs selection [2, 33]. In addition, data-driven approaches based on machine learning are used to select and identify the archetypes of the main building categories[102]. Because of their interesting performance and simulations setup easiness, it is possible also to implement them at a more complex scale than single building level[109]. Focusing on the scope of this work, the most widespread black-box approaches for prediction and forecasting at a building level are as follows[107]: simple regression model (SRM), multiple linear regression (MLR), decision trees (DT), artificial neural networks (ANN), and support vector machine (SVM). All of them above rely on the

availability of prior data to forecast energy consumption. Among the downsides is that to predict a building's energy demands through such approaches, it is necessary to possess as large a historical dataset as possible. This circumstance, therefore, makes them essentially unusable in all those cases where this is not possible (such as in the case of a new building) or in the case of previously unobserved events (i.e. Covid pandemic). Nevertheless, these methods are not employed to carry out building energy demand forecasting on the short-term, espercially for plants optimization and regulation.

- Grey-box approaches: These models combine first-principle physics and data-driven approaches, resulting in a hybrid structure. As expected, they share both advantages and limitations of white and black-box models. In the recent literature, hybrid modeling presents two main orientations.In more detail, some case-studies use data-driven methods to optimize specific parameters of white-box models[103] or they combine a resistance–capacitance lumped model with a Gaussian process to predict and adjust the error of the physics-based model. The second way purely replaces parts of the physics-based models with machine learning algorithms, for instance with the aim of representing energy equipment load demand[20, 58].

  To the best of the author's knowledge, none of the articles show a clear winner among all the available algorithms. This is especially true when the building data and features are missing or unknown, making it challenging to find similarities or benchmarks. Since the main interest is the implementation in real use cases, where execution time is crucial and cost-effectiveness drives the whole process, the thesis is focused on data-driven approaches. In more detail, it proposes a method for predicting the buildings energy demands, applying similitude criteria both for verifying the results or making predictions in the absence of historical data.

2.1.1 White box approaches

White-box based approaches are physical based approaches, which require detailed information of building structure and its related typical characteristics: consequently,

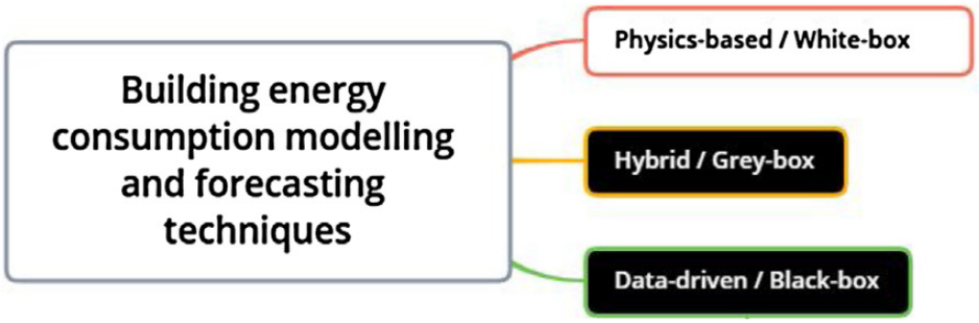

FIGURE 2.1: Methods commonly used in buildings Energy consumption performance evaluation [109].

the corresponding simulation will be usually expensive and time consuming. Recently, a series of attempts have been made by scholars in order to simplify these approaches and reduce time needed for setting up and running simulation.

As white-box approach instance, it can be mentioned Residential Load Factor (RLF) [15], an evolution of prior Residential Heat Balance method (RHB) [16], that includes both cooling and heating load procedures. The heating calculations rely on the traditional UΔT model, that has proven good results, while cooling procedure is built upon the idea of independent load components: load contributions from various sources are separately evaluated and then summed. In particular, surfaces have associated load factors (LFs) or load contributions per unit area, designated as Cooling Factors (CFs) for cooling and Heating Factors (HFs) for heating; the lasts are simply for the most part UΔT, while the firsts depend on surface construction, climate and, in some cases, surface orientation, solar absorptance or other characteristics: each LF needs to be evaluated once for a given site and construction conditions and then it is applied repeateadly to each building element of the same type.

Despite RLF formulation is conceptually transparent, unfortunately it is not possible to find any invariant model for each load component, because of interactions among them: for example, the autors reports that if a major interactions occurs between opaque surfaces and windows, the solar gain is lagged and moderated by differing amounts depending on surface construction properties; also simple convective gains, such as infiltration and ventilation, are hard to estimate, because they should be evaluated at the building-dependent peak hour [59].

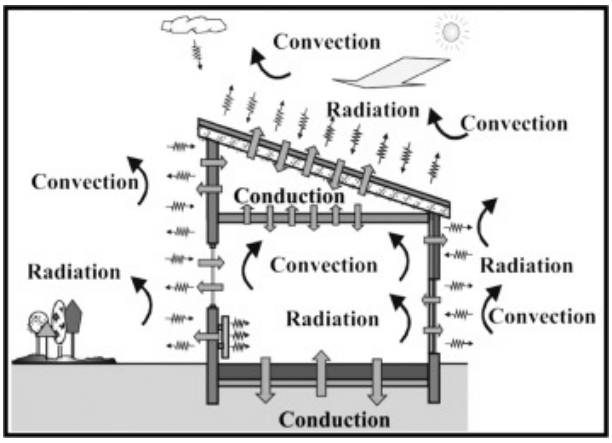

FIGURE 2.2: Simplified representation of energy flows on a generic building

As seen, development of load-component method such as RLF requires the interactions to be identified and addressed, eliminated or neglected if the effects are irrelevant compared to accuracy level the simulations want to achieve; addressing interactive effects represents the main weakness of this method, because it introduces more complexity, defeating its purposes: for this reason, RLF applicability is restricted to typical residential construction. Then, one may be tempted to consider a more general method like RHB, potentially able to handle any configuration assuming the independence of load components and developing models for each of them. As stated before, white-box approaches has three primary drawbacks and the methods listed make no exception. First, they are computationally intensive and time-consuming, that is conflicting with many applications requiring small fraction of a second per output. Second, occasional convergence failure remains troublesome. Third, it is hard to handle all the cases encountered in practice.

In fact, in order to make the model fit the case study, it is necessary to have a comprehensive set of equations able to model all the not neglible effects considered. Looking at the energy needs of a building, without taking into consideration its destination of use, one of the most common approach is to follow what stated by national or international normatives and standards, which are in Italy for HVAC and auxiliary plants the UNI TS 11300 and UNI10339, plus the so called Regional Healthcare Accreditation Laws, that rule the healthcare service requisites for hospital facilities.

Similar topic, for the internal lighting, ruled by UNI 12264-1 for italian facilities,

which set the eye-comfort level for each building and its rooms according to their destination purpose. In more detail, it allows to calculate inderectly the theoretical electrical absorption in order to satisfy the requirements.

Definetely harder to find, set and update are those very detailed informations regarding the internal of the building and linked with:

- room destination purpose: change may take place during time, making necessary to update them frequently and rising the costs. This is an occurrence that take place frequently in some kind of facilities, like hospitals, and it may impact the whole energy needs obviously;
- environmental and endogenous factors:

  thermal/cooling dispersions;

  heating/cooling free inputs;

  solar radiance;

  natural lighting free inputs;

  number of persons inside;

  free heating inputs because of electrical devices or machines.

Adding to this, detailed architecture information (i.e. internal and external wall stratigraphy) of the building are needed, both at the project and as-is level, that must be update time by time in order to take into account the performance degrade because of materials aging.

All of them, if they could be considered manageable through static simulations, it is more and more complex to update all of their statuses and values time by time in order to consider their evolution.

These complexities are emphasized by the time necessary to perform the simulations, because of the high number of equations and variables involved.

About this last point, considered that the physics of the phenomena is mainly driven by surfaces (l^2) and volumes (l^3), denoted with l the reference physical dimension, it can be stated the complexity order of the problem $O(x)$ is:

$$O(x) \approx O(l^3) + O(l^2) \approx O(n^3) + O(n^2) \tag{2.1}$$

considered that $l^3 >> l^2$, it can be also re-written as:

$$O(x) \approx O(l^3) \tag{2.2}$$

because $O(l^3) >> O(l^2)$. Hypothisizing a numerical resolution of the equation set, i.e. Newton-Raphson method, known that for each step the calculations needed are given by:

$$Calc_i = n_{equation_i} \cdot n_{variable_i} \tag{2.3}$$

So, it can be considered that:

$$n_{equation_i} \approx \prod_{i=1}^{n} n_{equation_{room_i}} \cdot n_{room_i} \tag{2.4}$$

given:

$$n_{equation_i} \approx \prod_{i=1}^{n} n_{equation_{room_i}} \cdot n_{room_i} \tag{2.5}$$

considering reasonably that for complex buidings and facilities it may be:

$$n_{equation_{room_i}} \approx 10^2 \tag{2.6}$$

$$n_{room_i} \approx 10^3 n_{variable_i} \approx 10^1 \tag{2.7}$$

it results that for each step are needed:

$$Calc_i \approx 10^6 \tag{2.8}$$

Known that the number of iterations needed to convergence are inversely proportional to the step dimensions:

$$n_{iteration} \approx \frac{1}{n_{step_i}} \tag{2.9}$$

Considering that for many variables a step of 10^-3 may be enough to achieve a good resolution, it turns out that:

$$n_{calculation} \approx Calc_i \cdot n_{iteration} \approx 10^9 \tag{2.10}$$

Then, linking the complexity of the simulation to the number of iterations needed:

$$O(n_i) \approx n_{calculation} \tag{2.11}$$

it can be easily derived that:

$$O(x) \approx O(n^3) \approx n^3_{calculation} \approx 10^{27} \tag{2.12}$$

are needed to complete a simulation with a white-box model.

Usually, a commercial computer has a CPU able to do calculations for each second in the order of GHz for each of its core. Therefore, considered a laptop with 7 core CPU, it can be turned out that:

$$\begin{matrix}\text{Computing} \\ \text{power}\end{matrix} \approx f_{CPU} \cdot n_{core} \approx 10^{10} Hz \tag{2.13}$$

then the calculation time will be approximately equal to:

$$\begin{matrix}\text{Calculation} \\ \text{time}\end{matrix} \approx \frac{n^3_{calculation}}{\begin{matrix}\text{Computing} \\ \text{power}\end{matrix}} \approx \frac{10^{27}}{10^{10} Hz} \approx 10^{16} s \approx 10^8 years \tag{2.14}$$

From this passages, then it is clear the impossibility to proceed through a white-box approach in high granularity problems, where a detailed knowledge of the building behaviour is needed, except at high cost [1]. However, it must be noticed that in white-box approaches affordable calculation costs and times can be achieved simplifying the problem structure, for example considering the building envelope as an unique environment: in that case, the variable n_{room_i} would be in the order of $\approx 10^1$; reducing also the step in the order of 10^-2, doing the demonstration again it would be find out that the amount of time needed is near to half a day. Obviously, every improvement come at a cost and, in this case, it is both precision and accuracy, that would be no longer feasible for real-case study applications in complex buildings.

[1] i.e. supercomputers and other stuffs long far away by common people;

Consequently, because of all the consideration made above and the scope of this thesis, it comes by itself that this way does not fit what it is needed in order to achieve the target.

2.1.2 Grey box approaches

Grey-box based approaches are a variant of prior quoted white-box based approaches combining statistical methods and simplified physical information to simulate building energy performance. By their nature, these approaches could be useful in every analysis that involve incomplete or uncertain data; unfortunately, in these kind of applications, one primary issue in current grey-box methods is computational inefficiency due to uncertain inputs, complex interactions and stochastic occupant behaviours [123, 84]: although this kind of model is capable of reproducing dynamics of buildings, the procedure to generate the grey-box models is usually sensitive to the typology used and the number of simulations [58, 85]. For these reasons, similarly to white-box approaches, grey-box approaches also use simplified building or the HVAC plant models and usually they are not suitable for large building simulations, for instance industrial or public facilities, making them out of any consideration in this work.

2.1.3 Black box approaches

Black-box based approaches are able to estimate building energy performance or consumption only considering historical data, without any detailed knowledge of physical information: compared to the approaches listed above, this change allows generally fast calculations and high accuracy and precision. In many data science applications, these are also usually referred as data-driven approaches, because of the statistical algorithm structures and large amount of data used. They are also part of the broader family of methods known as 'artificial intelligence', i.e. referring to that field of research that studies algorithms and models that make machines capable of reading data and, on the basis of that, make predictions and take actions to achieve a predefined goal[96]. Both conventions will be used also in this work.

Among the artificial intelligence tools available in the literature, data-driven models are simple and consist of two distinct and iteratively subsequent phases: learning and validation.

Learning process starts with a careful selection of all the parameters and modifies them through subsequent validation, namely systematical comparisions between

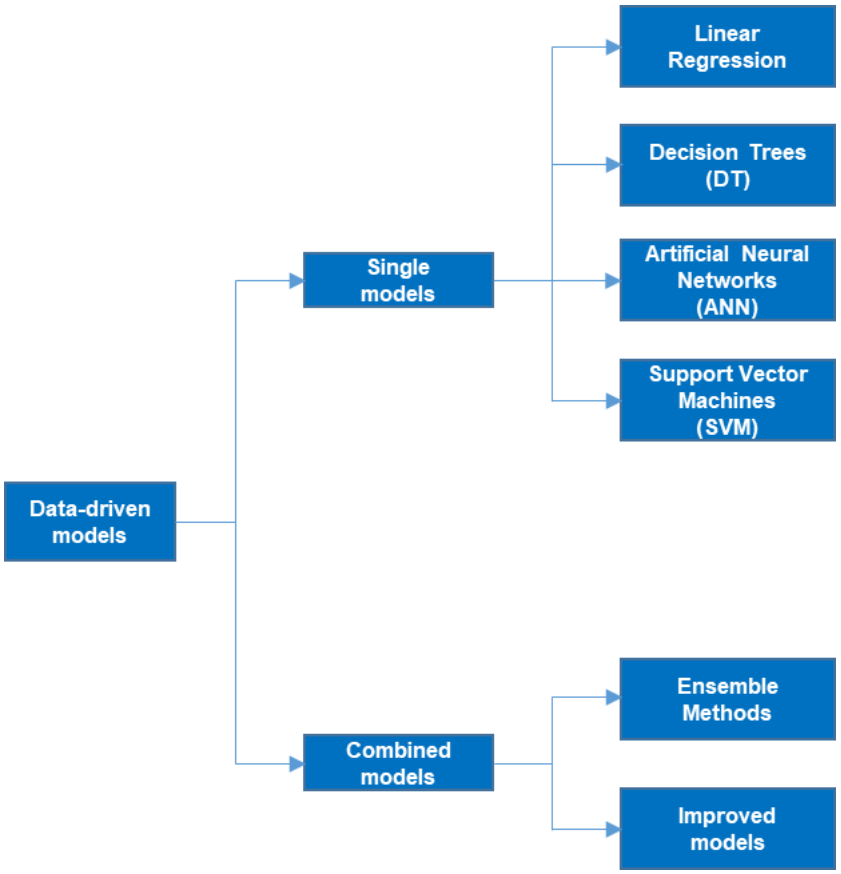

FIGURE 2.3: Data-driven models for building energy consumption [123]

the model outputs and the historical data. Once the error is lower than required threshold, the data-driven model is deemed validated and, consequently, qualified for pratical applications with new input data [123].

Because of their lower costs, speed, precision and accuracy, compared to what seen above and in all-in-one method, data-driven approaches are widely applied in several fields (e.g., medical diagnosis, political campaigns, commerce). Recently, their use has been extended to the building sector, especially to estimate building energy demands or profile energy consumption patterns. The most widespread methods are grouped in 2.3.

The present thesis will focus on data-driven prediction models, such as those pictured in 2.1, which represent the most popular models among data-driven ones.

2.1.3.1 Single models

Single models are usually the most known and more used machine learning models, both in research and application fields. Their name takes place from the fact that they are data-driven techniques implementing a single forecast algorithm for a prediction problem. Because of their simplicity, computational efficiency and easy-to-use features, they suit particularly well those problems where resources are limited. On the other hand, though, these models are prone to overfitting problems or they may struggle to generalize enough the problem in order to handle properly previously unseen events, especially in complex problem domains. Because of their effectiveness and efficiency, despite the above mentioned limitations, this models are used as valuable tools to establish baseline performance and provide insights about data correlations, serving also as blocks for more complex learning systems.

2.1.3.1.1 Statistical regression Building energy-consumption prediction relying on regression analysis link an outcome $Y_i (i = 1, 2, \ldots, n)$ to the corresponding predictors $(j = 1, 2, \ldots, m)$, investigating the relationship among different variables in a probabilistic framework. Depending on if a multiple or polynomial regression is used, usually in literature the outcome is formulated as:

$$Multiple: \quad Y_i = \alpha_i + \beta_1 x_{i,1} + \beta_2 x_{i,2} + ... + \beta_m x_{i,m} + \varepsilon_i \tag{2.15}$$

$$Polynomial: \quad Y_i = \alpha_i + \beta_1 x_{i,1} + \beta_2 x_{i,2}^2 + ... + \beta_m x_{i,m}^m + \varepsilon_i \tag{2.16}$$

Where ε_i represents the random error that is assumed to be normally distributed and α_i and $\beta_j (j = 1, 2, \ldots, m)$ are the parameters to be estimated. This is made considering the sum of squared errors (SSE), defined as:

$$Multiple: \quad SSE = \sum_{i=1}^{n} (A_i + B_1 x_{i,1} + B_2 x_{i,2} + ... + B_m x_{i,m})_i^2 \tag{2.17}$$

$$Polynomial: \quad SSE = \sum_{i=1}^{n} (A_i + B_1 x_{i,1} + B_2 x_{i,2}^2 + ... + B_m x_{i,m}^m)_i^2 \tag{2.18}$$

The estimates of all parameters is derived using least squares (LS): A_i and $B_j(j = 1,2,\ldots,m)$ in the equations 2.17 and 2.18 are the corresponding estimates of i and $j(j = 1,2,\ldots,m)$ in equations 2.15 and 2.16. SSE is then minimized , increasing to $m+1$ the number of equations; each of them is partially derivable with respect to A_i and $B_j(j = 1,2,\ldots,m)$ respectively. Finally, the prediction equation with the estimated parameters in multiple linear regression (MLR) or polynomial regression (PR) is obtained:

$$Multiple: \quad Y_i = A_i + B_1 x_{i,1} + B_2 x_{i,2} + ... + B_m x_{i,m} + \varepsilon_i \tag{2.19}$$

$$Polynomial: \quad Y_i = A_i + B_1 x_{i,1} + B_2 x_{i,2}^2 + ... + B_m x_{i,m}^m + \varepsilon_i \tag{2.20}$$

In order to quantify the variance explained by the independent variables, the coefficient of determination R^2 is introduced:

$$R^2 = 1 - \frac{SSE}{SS_{tot}} \tag{2.21}$$

Where sum of squares SS_{tot}:

$$SS_{tot} = \sum_{i=1}^{n} (Y_i - \overline{Y})^2 \tag{2.22}$$

With the mean value $\overline{Y}$:

$$\overline{Y} = \sum_{i=1}^{n} \frac{Y_i}{n} \tag{2.23}$$

As a rule of thumb, a regression equation with a R^2 near to 1 indicates it fit well original data: it must be emphasize that R^2 itself cannot determine if the coefficient estimates and predictions are biased, reason why it is necessary to assess the residual plots. In addition, R^2 coefficient has some further problems:

- Every time a predictor (i.e., an independent variable) is added to a model, R^2 increases: consequently, the more terms appear, the more it appear to have a better fit;

- If a model has too many predictors or higher order polynomials, it could model also random noise in the data. This condition is known as overfitting and produces deceptive high R^2 values and reduced predictive ability.

In order to consider each of those effect, adjusted R^2 is introduced, which compares the explanatory power of regression models that contain more than one predictor: in detail, it increases only the new term introduced improves model predictive ability more than its variance. By literature, adjusted R^2 formula is defined as:

$$R^2_{adjusted} = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1} \tag{2.24}$$

Observing the equation listed above, it's easy to guess that adjusted R^2 value is always lower than simple R^2. As seen above, statistical regression is a straightforward and simple to use approach for predicting building energy consumptions; in particular, it has been popular during early studies to predict the average consumption over long-term period of time. However, compared to other approaches such as ANN or SVM, the regression models require huge amount of historical data for training and their accuracy on short-term prediction is usually poor. It is also challenging to select a good set of predictors and an appropriate time scale to achieve a good building energy demand fitting under a wide range of environment and weather conditions. In addition to this, a big issue is that in some cases the selected variables are unexpectedly dependent: the unforeseen correlations among them in such cases result in an unpredictable level of accuracy in the regression outputs [136]. In the last years, scholars have studied some modifications for statistical regression aimed to solve inaccuracy problem [61] [8] [118] [48], claiming in most cases better results that ANN's models [123]. Anyway, in most cases statistical regression are adopted to estimate the important parameters needed for characterizing building energy performance, designing, tracing and analyzing building thermal behavior [69] or, also, drafting heating control strategy for energy saving [121]. The main applications concern their use for administrative buildings [7] and the design of new buildings [87].

2.1.3.1.2 Decision Tree Decision Tree (DT) is a flowchart-like structure in which each internal node represents a test on an attribute, each branch represents the outcome of the test and each leaf node presents a class label or predicted value, each assigned after computing all attributes. The topmost node of DT is known as the root node, where the input data are splitted into different groups based on some predictor variables predefined as splitting criteria. These split data form sub-nodes as branches emanating from the root node; the data on sub-nodes will undergo either further or no splits: the former, succeeding data split is conducted to form new subgroups as child-branches emanated graphically at the next level, whereas the latter are called leaf nodes and the corresponding data group at those levels are their final outputs. Each path from root to leaf represents a classification rule. In DT analysis, most known algorithms used to achieve error optimized splits are [53]:

- ID3;
- Gini Index;
- Chi-Square;
- Reduction in variance.

ID3 is the core algorithm for building decision trees and was developed by J.R.Quinlan thirty years ago [88]. This algorithm employs a top down, greedy search with no backtracking: it uses Entropy index and Information Gain to build a decision tree, defined as follows:

- Information entropy:

$$H(S) = \sum_{x \in X} -p(x) \log_2 p(x) \tag{2.25}$$

- Information gain:

$$IG(S, A) = H(S) - \sum_{t \in T} -p(t)H(t) = H(S) - H(S|A) \tag{2.26}$$

Once calculated information entropy and gain for every attribute, the algorithm choose that with the largest information gain as the decision node, dividing the

dataset by its branches and repeating the same process on every branch. Then, ID3 algorithm runs recursively on non-leaf branches, until all data are classified. Gini index algorithm, named so in honor of the Italian statistician and sociologist Corrado Gini that developed the Gini coefficient [46, 47], is conceptually simpler than the previous: it assumes that if two items are selected from a population at random, if the population is pure the probability to choose an item owning to that is equal to 1; like the previous algorithm, it works with categorical target variable "success" of "failure" and performs only binary splits; Gini index is calculated differently depending on if it is referred to sub-nodes or split:

- Gini index for sub-nodes is calculated using formula (p^2+q^2), namely sum of square of probability for success and failure;
- Gini index for split is calculated using weighted Gini score for each node of that split;

As it can imagine, the node split will take place on the attribute with the higher Gini index. The third listed algorithm, so called Chi-Square, is completely different and it find out the statistical significance between the differences among sub-nodes and parental nodes, measured by sum of square of standardized differences between observed and expected frequencies for each node:

$$\chi_i^2 = \sum \frac{(f_{observed,i} - f_{expected,i})^2}{f_{expected,i}} \tag{2.27}$$

Similarly to the previous algorithm, it works with categorical target variable "success" or "failure" and the steps to calculate chi-square for a split are two:

- for individual node, chi-square is calculated by deviation for both success and failure;
- chi-square of split is calculated using sum of all chi-square of each node of the split.

The three algorithms listed above are only for categorical target variable, consequently not suitable for the purposes of this work. Because of that, we shall introduce a fourth algorithm: reduction in variance. Conceptually, it uses the standard formula of variance to choose the best split:

$$\sigma = \sum_{i=1}^{n} \sqrt{\frac{(\sigma_n - \overline{\sigma})^2}{n}} \tag{2.28}$$

Also in this case, the steps needed are two:

- calculate variance for each node;
- calculate variance for each split as weighted average of each node variance.

the split with lower variance is selected as the criteria for splitting population; even if it has not been written explicitly until know, reduction in variance is a typical greedy algorithm, since it follows the problem heuristic of making the locally optimal choice at each stage, with the idea of finding a local optimum: in many problems, a greedy strategy does not produce a global optimal solution, but it is able to found local optimal solutions that well approximate a global optimal solution in a reasonable amount of time and with an acceptable error. Until now, focus has been put only on main algorithms commonly used to yield a decision tree: another important issue is about the optimal dimension of it; a tree that is too large risks to overfit training data and poorly predict new samples, while a small tree might not detect structural informations. In addition to this, reading between the lines no one of the algorithm listed above gave any stopping criteria on decision trees growth: this is because of so called "horizon effect", i.e. that is impossible to tell in advance how much the addition of a single extra node will decrease error. For this reason, growing the tree until each node contains a small number of instances, then pruning to remove nodes that do not provide any additional information, it is a wide used strategy. Obviously, pruning aims to reduce the size of a learning tree without affecting predictive accuracy as measured by a cross-validation set. Literature is flourishing of pruning methods for decision tree algorithm; among the most famous ones there are:

- reduced error pruning (REP);

- cost complexity pruning (CCP);

The REP algorithm is one of the simplest and commonly used pruning techniques; it evaluates the cost at each decision tree node in order to establish whether to convert or not the node into a leaf, prune the left or the right child [89]. The algorithm prunes the nodes of a brunch until sub-trees of an internal node are pruned and stops immediately even if one sub-tree is kept [6]. Cost complexity pruning relies on a different philosophy: it generates a series of trees T_0,...,T_m, where T_0 is the original tree and T_m is the root node alone. At step i , a tree is generated removing a sub-tree from i-1 tree, which is chosen minimizing the following cost function[24]:

$$R_\alpha(T) = R(T) + \alpha|f(T)| \tag{2.29}$$

Where:

- f(T) a function that returns the set of leaves of tree T;
- α is a regularization parameter;
- R(T) is the training/learning error;

The latter is defined as:

$$R(T) = \sum_{t \in f(T)} r(t) = \sum_{t \in f(T)}^{l} R(T) \tag{2.30}$$

With:

- $\sum_{t \in f(T)} r(T)$, is the sum of misclassification or prediction errors at each leaf;
- $r(t)$, is misclassification rate or prediction error;
- $R(T)$, is the training/learning error.

Once a sub-tree has been removed, then it is replaced with a leaf node whom value is chosen as in the tree building algorithm. In the end, when the series of trees has been created, the best tree is chosen by generalized accuracy as measured by a training set or cross-validation. As for the limitations of this approach, apart from being mostly applied in non-complex applications, such as buildings in residential areas, it is usually unstable in the presence of noisy or non-linear data [112].

2.1.3.1.3 Support Vector Machine The Support Vector method, originally designed to solve pattern recognition problems, has evolved during years becoming a tool for solving multidimensional function estimation problems. Nowadays, in fact, this method has been widely applied to solve regression problems and estimate an underlying relationship among nonlinear inputs to the continuous real-valued target; in particular, the Support Vector Machine (SVM) used for regression is called as Support Vector Regression (SVR) [123]. Its core task is to build a decision function $F(x_i)$ with a good generalization ability, whose elements are nonlinear functions parameterized by small subset of training data, called the Support Vectors (SVs): this type of function representation is useful for high-dimensional input space, because the number of free parameters in this representation is equal to the number of SVs, but it does not depend on the dimensionality of the space [114]. Given an input x_i, the decision function is usually formulated as:

$$F(x_i) = \langle w, \phi(x_i) \rangle + b \tag{2.31}$$

Where the bias b $\in \mathbb{R}$, $\langle \bullet, \bullet \rangle$ and w represent the dot product and weight defined in $\mathbb{R}^n$ and ϕ_i is a nonlinear mapping of the input space to a high-dimensional feature space[33]; b and w are unknown in the equation 2.31 and are estimated minimizing the regularized risk function, which is formulated as follows:

$$\frac{1}{2}||w||^2 + C\frac{1}{l}\sum_{i=1}^{l} L_\varepsilon(y_i, F(x_i)) \tag{2.32}$$

The first term $||w||^2$ is the regularized term, which minimization make function as flat as possible and contribute to control the function capacity; the second term, where C is the regularization constant, is the empirical error measured by the insensitive loss function, which is defined as below [115]:

$$L_\varepsilon(y_i, F(x_i)) = |y_i - F(x_i)| - \varepsilon = 0, |y_i - F(x_i)| \geq \varepsilon \tag{2.33}$$

This equation set defines a region of thickness where if the predicted value fall inside the loss is zero, otherwise the loss magnitude is equal to the difference between the predicted value and the radius of the tube [33]. Both C and ε are determined by

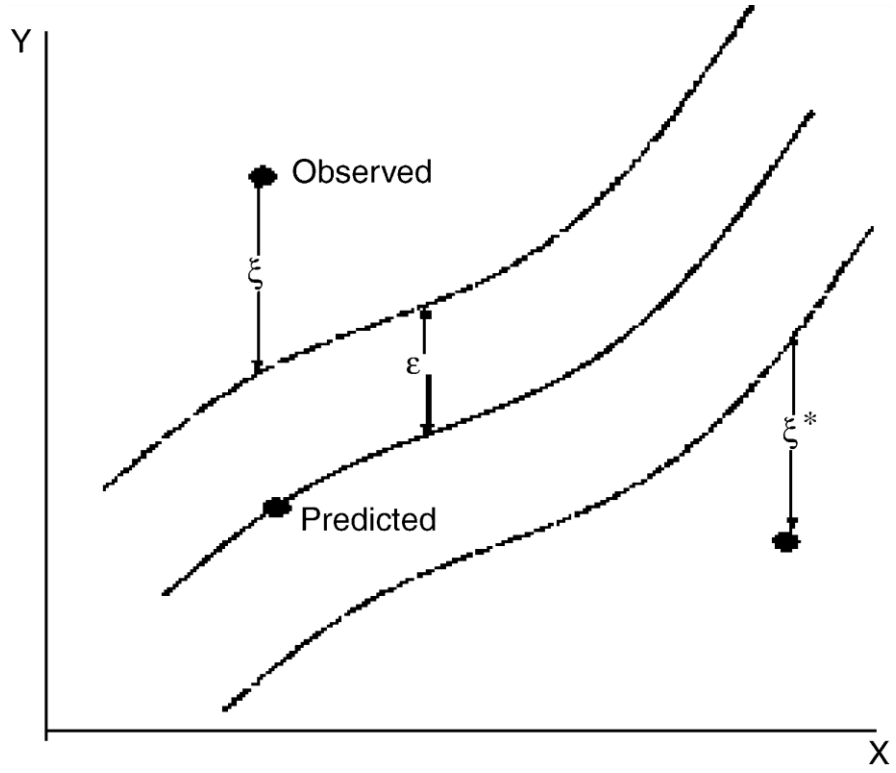

FIGURE 2.4: The parameters for the support vector regression [115]

users.

Then, to estimate b and w, the equation 2.32 is transformed to the primal objective function by introducing the positive slack variable ξ_i, ξ_i^* [33]:

$$Minimize \frac{1}{2}||w||^2 + C\frac{1}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*) \tag{2.34}$$

Subject to the following constraints:

$$\begin{aligned} y_i - w \cdot \phi(x_i) - b &\leq \varepsilon + \xi_i \\ w \cdot \phi(x_i) + b &= 0 \\ \xi_i^* &\geq 0 \end{aligned} \tag{2.35}$$

The optimization problem can be solved more easily in most cases in its dual formulation by an introduction of a Lagrangian function L:

$$\begin{aligned} L = \frac{1}{2}||w||^2 + C\frac{1}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}(\eta_i\xi_i + \eta_i^*\xi_i^*) + \\ - \sum_{i=1}^{l} a_i(\varepsilon + \xi_i - y_i + w \cdot \phi(x_i) + b) + \\ - \sum_{i=1}^{l} a_i^*(\varepsilon + \xi_i^* - y_i + w \cdot \phi(x_i) + b) \\ \eta_i^*, a_i^* \geq 0 \end{aligned} \tag{2.36}$$

Where η_i, η_i^*, a_i, a_i^* are Lagrange multipliers. Hence, the partial derivatives of L with respect to the primal variables equal to zero gives the saddle point condition:

$$\delta_b L = \sum_{i=1}^{l}(a_i + a_i^*) \tag{2.37}$$

$$\delta_w L = \sum_{i=1}^{l}(a_i + a_i^*)\phi(x_i) \tag{2.38}$$

$$\delta_{\xi_i} L = C - a_i^* - \eta^* \tag{2.39}$$

Finally, substituting the latter 2.39 in the 2.36 enable the twofold optimization problem:

$$\begin{aligned} Maximize W(a_i, a_i^*) = \sum_{i=1}^{l} y_i(a_i - a_i^*) \\ -\varepsilon\sum_{i=1}^{l} y_i(a_i + a_i^*) \\ -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(a_i + a_i^*)(a_j + a_j^*) \\ (\phi(x_i) \cdot \phi(x_j)) \end{aligned} \tag{2.40}$$

Subject to the following constraints:

$$\begin{aligned} \sum_{i=1}^{l} y_i(a_i - a_i^*) \\ a_i, a_i^* \in [0, C] \end{aligned} \tag{2.41}$$

Through condition 2.38:

$$w - \sum_{i=1}^{l} y_i(a_i - a_i^*)x_i = 0 \tag{2.42}$$

Therefore:

$$F(x_i) = \sum_{i=1}^{l}(a_i - a_i^*)\phi(x_i)\phi(x) + b \tag{2.43}$$

In order to achieve a more compact and exploitable equation, the kernel function $K(x_i, x_j)$ is introduced, which value is equal to the inner product of two vectors x_i and x_j in the feature space $\phi(x_i)$ and $\phi(x_j)$, that is:

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \tag{2.44}$$

and the previous equation can be rewritten as:

$$F(x_i) = \sum_{i=1}^{l}(a_i - a_i^*)K(x_i, x) + b \tag{2.45}$$

Introducing a kernel function is not just about simplifying the equation form, rather thanks to it is possible to perform all necessary computation directly in input space, without having to compute the map $\phi(x)$. Among the most popular kernel functions are the:

$$Linear\ kernel: \quad K(x_i, x_j) = x_i \cdot x_j \tag{2.46}$$

$$Polynomial\ kernel: \quad K(x_i, x_j) = (x_i \cdot x_j + 1)^d \tag{2.47}$$

$$Radial\ basis\ function\ kernel: \quad K(x_i, x_j) = (x_i \cdot x_j + 1)^d \tag{2.48}$$

Where γ ($\gamma > 0$) and d are the kernel parameters. By the use of different kernel functions is possible to build different learning machine, each with different types of decision surfaces. About the first term of the equation 2.46, accordingly to the Karush-Kuhn-Tucker (KKT) conditions, only a certain number of coefficients $(a_i - a_i^*)$ will be non-zero values and, consequently, the data point associated with them will have an approximation error equal to or larger than ε and these are respectevely the data points laying on or outside the bound decision function, referred

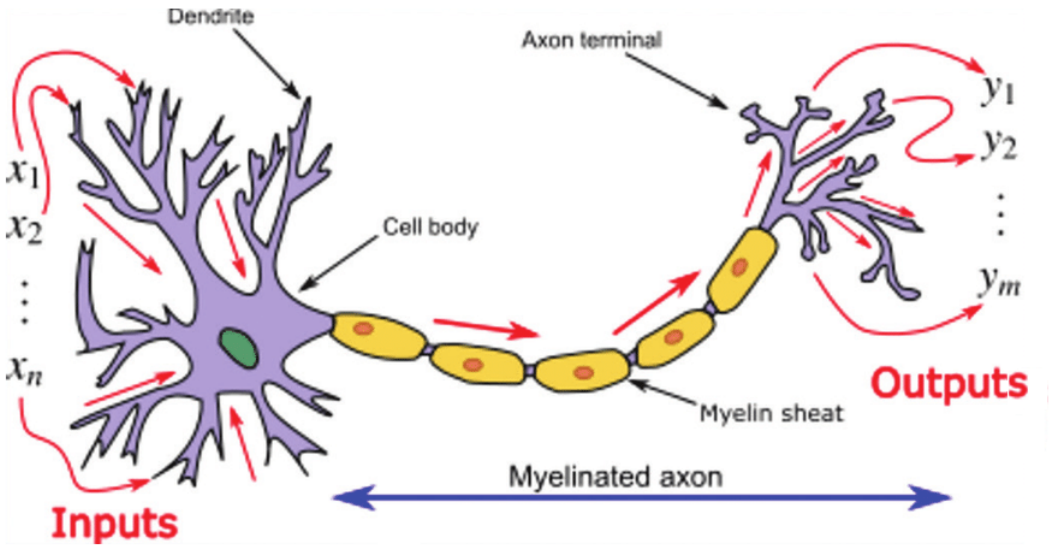

FIGURE 2.5: Axon-synapse-dendrite connection [93]

to as support vectors [33]. In this regard, choosing ε hide a trade-off between the sparseness and the closeness to data: as rule of thumb, respectively, the larger the , the fewer is the number of support vectors and sparser the representation of the solution and viceversa [26]. The main applications of the model aimed to forecast the combined cooling and heating energy demand in residential low energy buildings [83], the electrical load of an office building at the university [66] and the day-ahead electrical consumption of an office building [43]. In more detail, the last two case studies performed a comparison of SVR with ANNs, and the former results outperformed the latter in all the metrics.

2.1.3.1.4 Artificial Neural Network Artificial neural networks (ANNs) are biologically inspired computational networks, that attempts to mimic how the human brain processes information [95]. It consists of a set of neurons, each of which represents a node, that are connected to other nodes through links These correspond to biological axon-synapse-dendrite connections, like that represented in figure 2.5. Finally, each link has a weight, determining the strength of one node's influence on another [127]. The networks constitutes a directed, weighted graph.

As in the figure above 2.6, neurons receive inputs, combine the input with their internal state or activation function and produce outputs using an output function: the most valuable characteristic of the activation function is that it provides a smooth

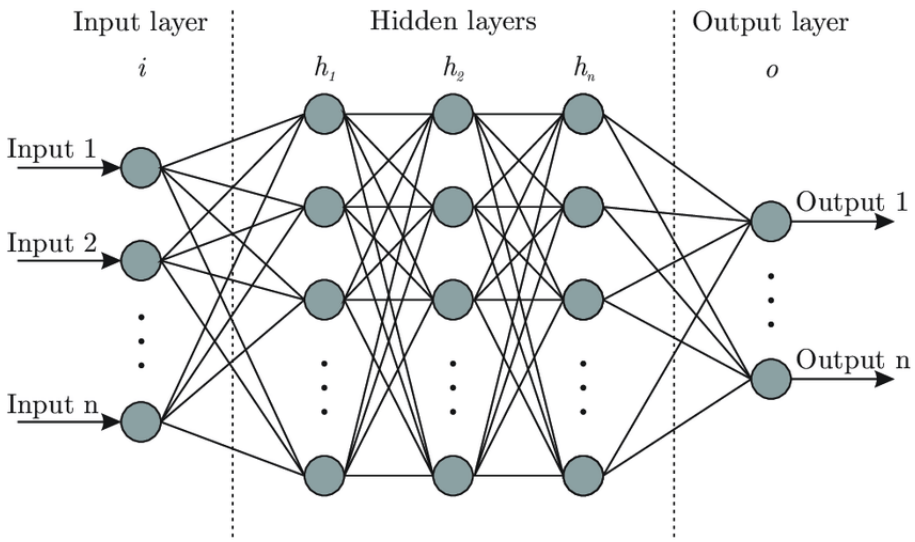

FIGURE 2.6: ANN architecture [22]

transition as input values change [126]. Then, each connection, to which is assigned a weight that represents its relative importance, transfer the output of one neuron as input to the linked one: a neuron can have multiple input and output connections, but each connection can link just two neuron among them. Therefore, the output of ANNs is computed through a propagation function, which consists of a weighted sum of input neurons and their relatives connections. Among the various types of ANNs, in this paragraph, it is dedicated a focus on multilayer perceptrons (MLPs) with backpropagation learning algorithms. MLPs are the ANNs most commonly used for a wide variety of problems. They are based on a supervised procedure and comprise three layers, as shown in the figure 2.6: input, hidden layers and output. Neurons are organized into multiple layers, where neurons of a layer connect only to neurons of the previous or next layers; the layer that receives external data is named input layer, while that produce the ultimate result is the output layer. In between them could be one or more hidden layers.

The figure 2.6 is a representation of ANNs as commonly perceived, but different kind of architecture are possible: for example, single layer and unlayered networks are also used. It is important to highlight that between two layers multiple connection patterns can also exist. In more detail, they can be fully connected, if every neuron in one layer connect every neuron in the next layer, or pooling, where a group

of neurons in one layer connect to just one neuron in the next layer [28]. When neurons present only this kind of connections, they form a directed acyclic graph and they are also known as feedforward networks; networks that allow connections between neurons in the same or previous layers are called recurrent networks [73] and particularly welcome to deal with time series datasets without random data and, consequently, sequence events [22]. After choosing the ANNs architecture, a learning process take place to specify all needed connection weights and biases, usually through historical data used as benchmark to address the response of the ANNs model for each given input. In more detail, learning process consists in the adjustment of the weights and optimal threshold of the neural network, in order to better adapt the network to handle a task by considering sample observations and improve the result accuracy, i.e. minimizing observed error. The learning process is complete when examining additional observations error rate does not reduce futher. Downstream of learning process, if error rate is unacceptable, the network must be redesigned in the majority of cases through a cost function periodically evaluated during learning process, whose cost is defined as a statistic value: as long as cost function output continues to decrease, learning process keeps going. Cost function is choosen considering desirable properties, such as convexity, or other arising from the model; however, nothing prevents it can be set ad hoc. Another useful method to increase ANNs accuracy is backpropagation: as the cost function, this method takes action on connection weights to reduce for each error found during learning, but unlike that the error amount is divided among connections. Technically, backpropagation starts calculating the partial derivative with respect to the weights of the cost function. Then, neural network weights are updated with one of the several available methods. Because of their ability to model and reproduce non linear phenomena and processes, fault tolerance, robustness and noise immunity, ANNs have been successfully applied in many disciplines and case studies. In fact, it provides a flexible way to handle regression and classification problems without explicitly specifying any relationships between the input and output variables. Specific attention has been dedicated to forecasting building energy demands, in particular, in tertiary [3], residential [65] and educational buildings [38].In fact, they provide a

flexible way to handle regression and classification problems without explicitly specifying any relationships between the input and output variables. This is a positive feature, considering that buildings exhibit predominantly non-linear dynamics and highly complex input-output relationships. On the other hand, it must be pointed out that the architecture choice and learning rate optimization process still rely on ad hoc methods: this implies that ANNs are case-dependent and have to be designed and validated each time for every different application. Thus, in building contexts, ANNs cannot effectively handle tasks requiring the prediction of unforeseen events. It is also worth mentioning another category of data-driven methods, namely Long Short-Term Memory (LSTM) models, which has been employed for instance to forecast heat loads in multi-floor buildings[117] or to reproduce general electricity consumption data for an operating year [105]. LSTM models are mainly based on neural networks and allow long-term prediction. Nonetheless, no study has used them in combination with similitude methods.

2.1.3.2 Combined Models

In the realm of machine learning, combined models have emerged as a powerful approach to enhance predictive performance and robustness. These models leverage the strengths of multiple individual algorithms or architectures to create a more comprehensive and accurate system. By integrating diverse learning paradigms, combined models can capture complex patterns and relationships in data that might be overlooked by single models. This approach is particularly valuable in scenarios where the underlying data structure is multifaceted or when dealing with high-dimensional datasets. The synergy achieved through model combination often leads to improved generalization capabilities and reduced overfitting, making it an attractive option for tackling the challenging real-world problems across various domains.

In particular, with respect to the literature and, as shown in figure 2.3, two types of combined models can be distinguished: improved models and ensemble models.

Improved models typically refer to enhancements made to a single base model or algorithm. These improvements can involve fine-tuning hyperparameters, modifying the model architecture, or incorporating additional features or preprocessing

steps. The goal is to optimize the performance of the original model without fundamentally changing its core structure or combining it with other distinct models.

Ensemble models, on the other hand, involve combining multiple diverse models to create a more powerful predictive system. These models can be of the same type (e.g., multiple decision trees in a random forest) or different types (e.g., combining a neural network with a support vector machine). Ensemble methods, such as bagging, boosting, and stacking, aim to leverage the collective wisdom of multiple models to improve overall performance and reduce errors. The key distinction is that ensemble models maintain the individuality of each constituent model while combining their outputs, whereas improved models focus on enhancing a single model's performance through modifications to its internal structure or parameters.

2.1.3.2.1 Improved Models Improved models, also known as "hybrid methods" [67], are defined as the combination of single models and optimization techniques. Accordingly to the literature for building energy needs forecast, they include Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and Differential Evolution (DE) algorithms.

In the context of optimization algorithms, Particle Swarm Optimization (PSO) emerges as a significant paradigm within the broader landscape of computational intelligence. Conceptualized by Kennedy and Eberhart in 1995 [56], PSO draws inspiration from the collective behavior observed in natural systems, particularly the synchronized movements of bird flocks and fish schools. This bio-inspired, population-based stochastic optimization technique shares certain fundamental characteristics with evolutionary computation methodologies, such as Genetic Algorithms (GA), yet distinguishes itself through its unique operational mechanisms. The PSO algorithm initializes by generating a population of candidate solutions, referred to as particles, randomly distributed within the problem space. These particles traverse the multidimensional search space in pursuit of optimal solutions. Unlike evolutionary algorithms that rely on genetic operators such as crossover and mutation, PSO's optimization process is driven by the social interaction and information exchange among particles. Each particle's trajectory is influenced by both its individual best-known position and the global best position discovered by the swarm collective,

facilitating a balance between exploitation of known good solutions and exploration of the search space. The iterative nature of PSO involves updating the particle positions and velocities across successive generations, guided by a fitness function that evaluates the quality of each solution. This process continues until a termination criterion is met, which may be a predetermined number of iterations, achievement of a desired fitness level, or convergence of the swarm. The absence of complex evolutionary operators in PSO contributes to its computational efficiency and ease of implementation, making it an attractive option for a wide range of optimization problems in various domains.

Like any tool, PSO has its obvious strengths, but also its weaknesses, with the latter often determining whether or not its application is successful. In this regard, literature shows that the main weaknesses of such an optimisation algorithm lie in [91][137]:

- low diversity or quality solutions;
- early convergence;
- particle stacking in local optima;
- computational inefficient to update parameters.

The Genetic Algorithm (GA) is a metaheuristic search methodology inspired by the biological natural selection and genetic evolution, beloging to the larger class of Evolutionary Algorithms (EA)[74]. In a genetic algorithm, a population of candidates (usually called individuals, creatures, organisms, or phenotypes) is characterized by a set of properties (its chromosomes or genotype), which can be mutated and altered and evolved toward better solutions accordingly to the metrics of the optimization problem. Each candidate solutions, traditionally, are represented in binary as strings of 0s and 1s, but other encodings are also possible[124]. The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called "generation". In each generation, the fitness of every individual in the population is evaluated: the fitness is the value of the objective function in the optimization problem being solved. The more

fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. A genetic algorithm requires typically:

- a representation of the solution domain;
- a fitness function to evaluate the solution domain.

Once the genetic representation and the fitness function are defined, the GA proceeds initializing the population of solutions and improving it through iterative mutation, crossover, inversion and selection operators.In addition to the main operators above, other heuristic techniques are usually employed to make the algorithm faster or more robust. In detail, the speciation heuristic penalizes crossover between candidate solutions too similar among themselves, encouraging therefore population diversity and helping preventing premature convergence to less optimal solutions[32].Then, the generational process is repeated until a termination condition has been reached. Common stopping conditions are:

- A solution is found that satisfies minimum criteria;
- Fixed number of generations (or iterations) reached;
- Allocated budget (computation time/money) reached;
- A plateau is reached, i.e. successive iterations no longer produce better results;
- successive iterations no longer produce better results than the highest ranking solution's.

or a combination of them. As every algorithm or method, also genetic algorithm has its owns limitations, especially as compared to alternative optimization algorithms:

- iterative fitness function evaluation for complex problem is often the most prohibitive and limiting segment of this approach [31]. In this case, it may be necessary to use an approximated fitness that is computationally efficient than an exact evaluation;

- Genetic algorithms do not scale well with complexity [131]: in detail, if the number of elements exposed to mutation is large, there is often an exponential increase in search space size. It makes extremely difficult to use this optimization technique on problems such as designing an engine, a house or plane. In order to make such problems tractable through evolutionary search, each one must be broken down into the simplest representation possible. Hence it is typically seen evolutionary algorithms encoding designs for fan blades instead of engines, building shapes instead of detailed construction plans, and airfoils instead of whole aircraft designs. The second problem of complexity is the issue of how to protect parts that have evolved to represent good solutions from further destructive mutation, particularly when their fitness assessment requires them to combine well with other parts[72].

- Like others evolutionary algorithms, a solution is "good" only in comparison to other previously discovered solutions[42];

- GAs may have a tendency to converge towards local optima or even arbitrary points rather than the global optimum of the problem. This means that it does not “know-how” to sacrifice short-term fitness to gain longer-term fitness[17];

- Operating on dynamic data sets may genomes start to converge early on towards solutions which can be no longer valid for later data. To remedy this issue, several methods have been proposed by increasing genetic diversity somehow and preventing early convergence, either by increasing the probability of mutation when the solution quality drops (called triggered hypermutation), or by occasionally introducing entirely new, randomly generated elements into the gene pool, called "random immigrants".[113];

- GAs cannot effectively solve problems in which the only fitness measure is a single right/wrong measure (like decision problems), as there is no way to

converge on the solution (so called "no hill to climb"). In these cases, a random search may find a solution as quickly as a GA [64];

- For specific optimization problems and problem instances, other optimization algorithms may be more efficient than genetic algorithms in terms of speed of convergence. Alternative and complementary algorithms include evolution strategies, evolutionary programming, simulated annealing, Gaussian adaptation, hill climbing, and swarm intelligence (e.g: ant colony optimization, particle swarm optimization) and methods based on integer linear programming. [64].

The suitability of genetic algorithms is dependent on the amount of knowledge of the problem: well-known problems often have better, more specialized approaches.

Such algorithms have been used in the literature for the optimisation of ANN hyper-parameters. In particular, an intensive study was carried out on a tertiary building in China aimed to compare a PSO-ANN to simple ANN and GA-ANN (GA was used for the same purpose as PSO)[27], in order to better understand which was most suitable to building energy forecast. Two databases were used to predict hourly electricity consumption for:

1. a research building from ASHRAE dataset located in USA with four-month data collection including WS, SR, RH and OAT;

2. a campus library located in East China with 100-day data collection including estimated occupancy and daily OAT.

For the former, 70% of the data were used for training and 30% for testing. For the latter, 93% of the data were used for training and the remaining 7% for testing. PSO-ANN gave better forecasting results than GA-ANN and ANN, with MAPE of 1.6%, 1.9% and 2.2%, respectively, for the ASHRAE database. PSO-ANN also outperformed GA-ANN and ANN when applied on the Chinese library building with a MAPE of 5.9%, 7.1% and 8.0%. A downside, on the other hand, lies in the fact that such optimisation algorithms often require a large number of generations and, consequently, calculation time to reach convergence.

Furthermore, one of the main limit of those optimization methods and machine learning models relied on the fact that a model optimized for short-term forecasts was not suitable for the long-term one [135], both because of machine learning models and hyperparameters tuning. In order to overcome this hurdles, a different approach was tried implementing Differential Evolution for timeseries forecast in tertiary buildings by the authors in [135], in order to make either short or medium term forecast possible, while also long-term one is still out of the range. In detail, DE is a population-based algorithm, which contains mutation, crossover and selection steps through repeated generations till the termination criteria is achieved. The DE algorithm utilizes randomly sampled pairs of object vectors to guide the mutation operation instead of relying on probability distribution functions as in the case with other Evolutionary Algorithms (EAs). The distribution of the differences between randomly sampled object vectors is determined by the distribution of these object vectors. The distribution of the object vectors is mainly determined by the corresponding objective function's topography, and so the biases where DE tries to optimize the problem match those of the function to be optimized. This enables DE to function robustly and more as a generic global optimizer than other EAs [129]. DE commences the search process by randomly generating NP number of D-dimensional parameter vectors $X_{i,g}$ (where i=1,2, ..., NP and g represents the current generation): it must be noted that the initial population (g=0) is generated randomly to cover the entire search space and NP does not change during the optimization process. Once the initialization process is performed, the algorithm evolves through the mutation, crossover and selection operators, as a classic evolutionary one. The main difference with them relies in the way those steps are carried out [60]:

- Mutation: in DE is carried out to facilitate random perturbations on the population: for each population member, a mutant vector is generated by randomly selecting (usually three in the basic version of DE) different individuals in the population and then adding a scaled difference of any two of the three vectors to the third one. In EA, this step is performed thanks to random perturbations on previous generations of population members or using predefined probability distributions;

- Crossover: DE algorithms implement a binomial crossover, where each parameter is exchanged with a probability distribution [122], while EAs use many crossover methods, like one-point, two-point or also uniform crossover [68];

- Selection: compares a trial vector and its parent solution in the current population to decide the winner to survive into the next generation, i.e. replacing the parent only if the offspring is better. In EAs roulette wheel or tournament for the next generation [125].

The characteristic mutation method, combined with easier parameter tuning and faster convergence speed, make the DE particularly effective in all those continuous optimisation problems, such as those that characterise, for example, the frequent updating of hyperparameters in machine learning models in the prediction of building energy requirements over time horizons ranging from short to medium term.

By virtue of these characteristics, this optimisation algorithm was tested and compared with PSO and GA mentioned above for the hyperparameters tuning of Support Vector Machine regressor in the prediction of time series referring to the energy consumption of buildings[135]. As an evolutionary step of hybrid models, in order to fill the gap among short to long term forecast which usually require different models for each time horizon, the authors proposed a model based on a weighted average of models, in particular ϵ-SVR and ν-SVR. Concerning building type and energy requirements, the case study addressed an institutional building in an university campus in Singapore, mostly consisting of offices and laboratories, having a ground floor surface of 3894 m^2. For the scope, it was utilized the energy data collected by meters for a period of one year with a thirty minutes sampling interval frequency, splitting it into two datasets for daily and half-hourly energy consumption forecasts, with 384 and 209 training points and 96 and 52 test points respectively, with a training/test points ratio of 4:1. Once the model parameters corresponding to the least MAPE are optimized, the model have been tested on the testing datasets: in detail, the performance of the proposed DE optimized model is compared with a single SVM optimized with Genetic Algorithm and Particle Swarm Optimization, where each model in the experiment is run 20 times and models and those with the lowest MAPE among the 20 trials are selected for the paper. Accordingly to authors'

findings, the proposed model takes a training time of about three hours (not mentioning hardware performance) and it shows a daily dataset average MAPE of 5.843, while for half-hourly dataset is 3.768, much lower than ϵ-SVR and ν-SVR alone. This case study and its results, although limited to a single building, one type of energy carrier and a training time that make it unfeasible for real-time applications, nevertheless it demonstrates the need for specialised approaches for each type of problem addressed, which means relying on a specific model for each task.

2.1.3.2.2 Ensemble Methods In statistics and machine learning, ensemble methods combines multiple learning algorithm to achieve a better predictive performance than it could be got with any singular method [86]: the main hypothesis of them is that combining multiple models together can often produce a much more powerful and flexible model [94]. In fact, most of the times, basics models perform poorly because of high bias or too high variance. As Latins used to say:"In medio stat virtus": on the one side, it is desirable a good degree of freedom to resolve the underlying complexity of data and contemporary, on the other side, not too much, in order to avoid high variance and achieve robustness [94]: this is the well know bias-variance tradeoff.

In addition to this, ensemble learning methods presents also an interesting set of underlying benefits and reasons why they may be preferred [86]:

- Statistical reason – as previously written, neural networks or other automated classifiers/regressors good performance on training data does not imply at least equal generalization/prediction performance: a set of classifiers or regressors with similar training performance may have different generalization/prediction performances, in particular if the test dataset used is not representative of the future field data. In such cases, combining the outputs of many classifiers/regression function by averaging might reduce the risk of selecting a poorly performing classifier/regression function and, consequently, having poor performance: it must be pointed out that the averaging may not perform better than the best classifier/regression function in the ensemble, but it certainly reduces a lot the overall risk of making a particularly poor selection;

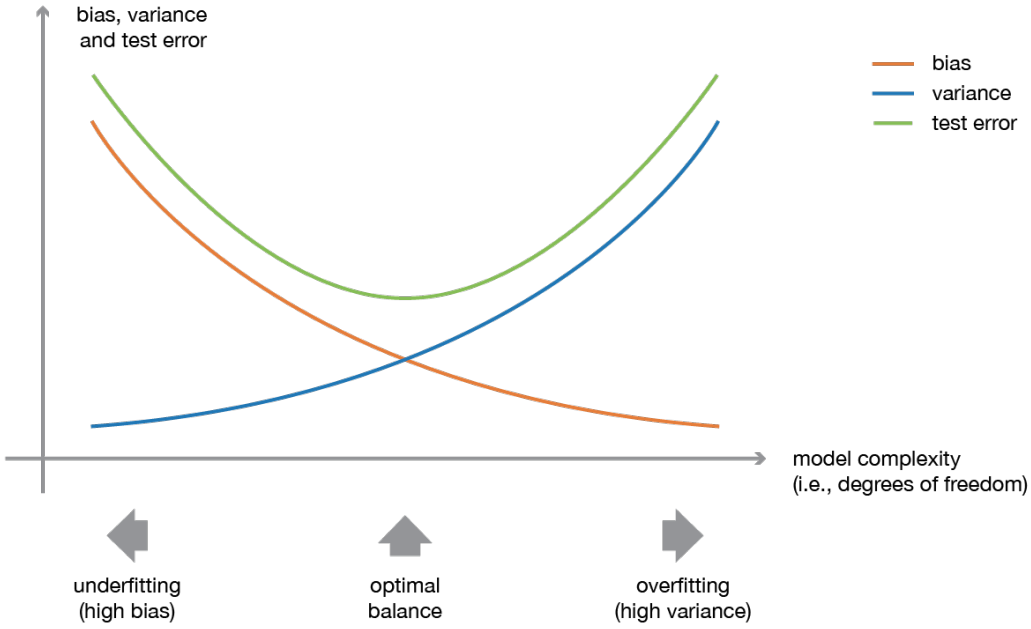

FIGURE 2.7: Bias-variance tradeoff[94]

- Large/small volumes of data – some applications involve volumes of data that are too large to be handled by just one classifier/regression function. In such cases, partitioning data into smaller subsets, training different classifiers/regression functions with different subsets of data and combining their outputs using combination rules widely has proved to be a more efficient and effective approach. On the other side, ensemble methods have been used also to address problems involving too little volumes of data. Having a representative and adequate training dataset is of primary importance for the classification/regression algorithm to successfully learn the underlying complexity of data and making predictions. In absence of this, resampling techniques can drawn overlapping random subsets from the available data, that can be successfully used to train different classifiers/regression functions and, then, creating the ensemble;

- Complex decision boundaries – some problems are just too difficult for a given classifier or regression function to solve, because the decision boundary that separates data of different classes may be too complex or lie outside the space

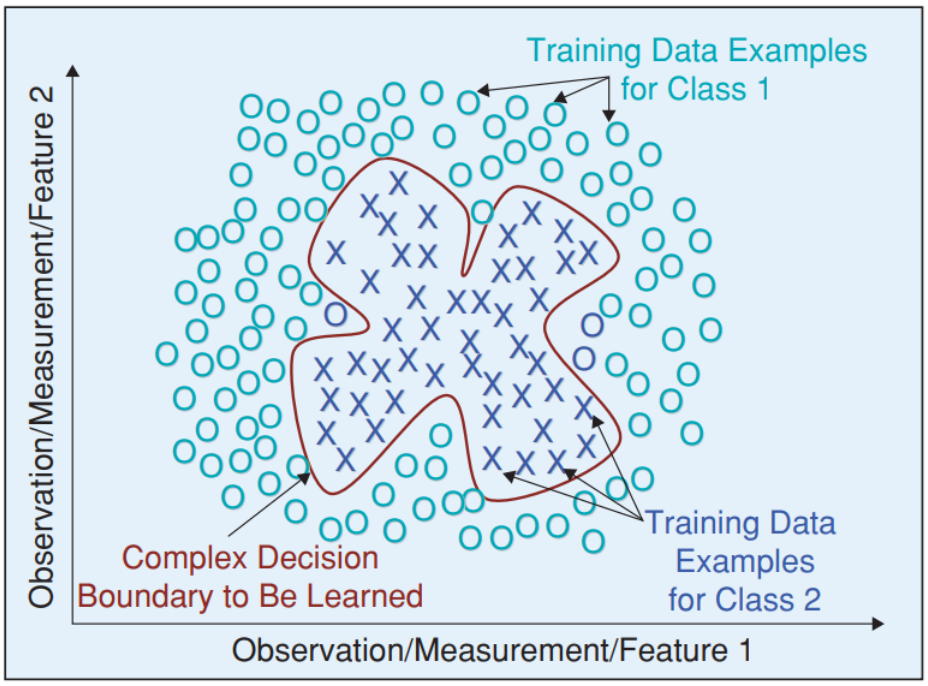

FIGURE 2.8: complex decision boundary that cannot be learned by linear or circular classifiers[94]

of functions implemented by the chosen classifier/regression model. For instance, a linear classifier cannot successfully deal with a complex decision boundary as the one depicted in 2.8.

However, an appropriate combination of an ensemble of linear classifiers can learn that nonlinear boundary; for instance, as in 2.9, a collection of circular or elliptic decision boundary generated by an ensemble of such classifiers can deal with the underlying complexity of the decision boundary dividing the data space into smaller and easier to learn partitions, where each classifier learns only one partition. Despite the example above referred to a classification problem, obviously the same is valid for a regression problem, where a complex distribution of numerical data can be learned and predicted an ensemble of decision tree, for example.

- Data fusion – when various data set from different sources and with heterogeneous features are available, a single classifier/regression model cannot handle effectively the information within all the data. In such cases, ensemble

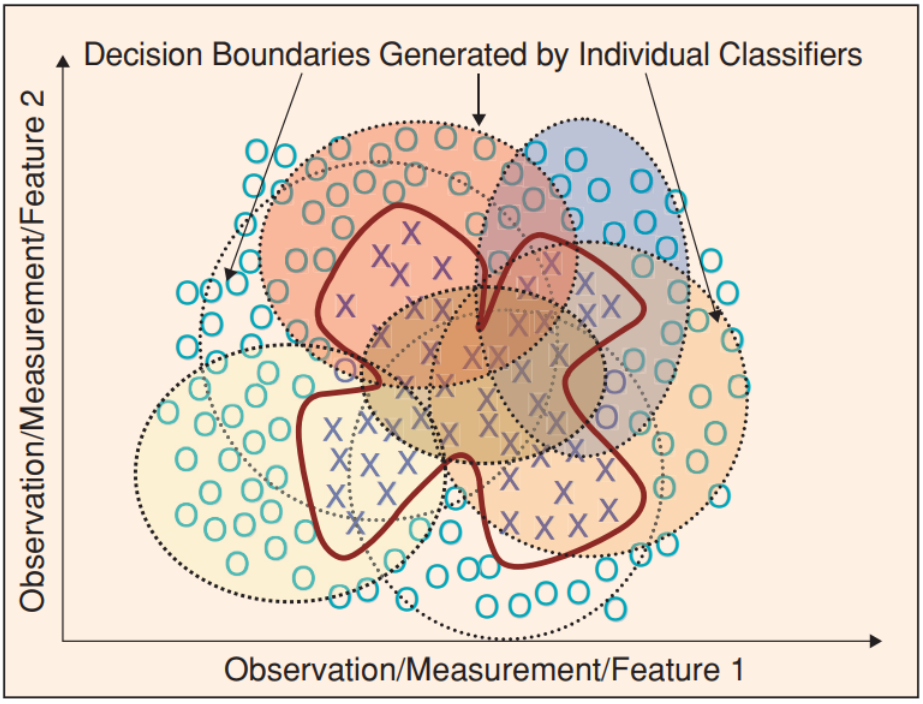

FIGURE 2.9: Ensemble of classifiers spanning the decision space [94]

based methods can perform a data fusion process, namely integrating multiple data sources to produce a more consistent, accurate and useful information than provided by any individual data source. In detail, test from data with different number and type of feature, which cannot be used jointly to train a single classifier or regression mode, are used to train different classifier or regression models, whose outputs are combined into a single one.

In order to set up an ensemble learning method, the first step is to select the base models that have to be aggregated; by literature, an ensemble model is said homogeneous when it takes a single base learning algorithm and weak learners are trained in different ways. Viceversa, other ensemble learning methods use different kind of base learning algorithms: so, heterogeneous weak learners are combined into a so called heterogeneous ensemble model.

Common types of ensembles are [77]:

- Bayes optimal classifiers – is an ensemble of all the hypotheses in the hypothesis space;

- Bayesian parameter averaging – approximate the Bayes Optimal Classifier by sampling hypotheses from the hypothesis space and combining them using Bayes' rule;
- Bootstrap aggregating (bagging) – builds multiple models, typically of the same type, from many subsamples of the training dataset;
- Boosting – it consists in building multiple models, usually of the same kind, and each of them learns to fix the prediction errors of a prior model in the chain;
- Stacking – building multiple models, in most cases of different each other, and having a supervisor model that learns how to best combine the predictions of the primary models;
- Bucket of models – it is a model selection algorithm based on cross-validation that is used to choose the best model for each problem;

The implementation of ensemble modelling mainly involves the two following steps[37]:

1. several sub-models are implemented;
2. their forecasting results are compared, weighted accordingly to their accuracy metric and combined to generate the optimal output of the ensemble model.

Regarding the general ensemble modeling process, accordingly to the literature [119], there are two main strategies that can be performed:

- homogeneus modeling;
- heterogeneous modeling.

In homogeneous modeling, the original dataset sub-samples are created and then processed through a single machine learning model. Their results, so called base-learner, are weighted based on their prediction performances, which are evaluated accordingly to choosen metrics. Then, they are combined into the ensemble model. Accordingly to the available literature, two additional paths can be followed [5]:

- sequential learning;
- parallel learning;

The sequential learning, also known as in series learning, generates base learners sequentially in order to exploit their interdependence: typical example of this technique are the boosting algorithms [98]. The parallel learning refers, instead, to bootstrap aggregation method [23] and it generates base learners in parallel to exploit conversely their indipendence. From this different way of working, it comes that the former aims to reduce the variance of each base learners estimates, while the latter targets the bias reduction.

Moving on to the second strategy for ensemble modeling, the heterogeneus modeling, it is referred to stacking techniques[128] and develops in two steps:

1. training on the same dataset several different single forecasting models;
2. give the ensemble model weighting the forecasting results from each base model.

Obviously, the choice of weak learners must be as much as possible coherent with the model aggregating process: if base models are characterized by a low bias and high variance, the chosen aggregating method must tend to reduce variance, and viceversa [94].

Accordingly to the most recent literature findings, this approach have been found to outperform both single and homogeneous machine learning models because of their inherent ability to leverage the point of strenghts of the different implemented models [57], ability to handle complex patterns in time-series data [133] or forecasting scenarios [54], incorporating LSTM architecture in the model and endowing it with the capacity to catch hard to find non-linear relationships among input and output variables: these features give unique versatility, adaptability and performance robustness in many applications [106].

Because of their extreme flexibility, enseble methods have gained interest for building energy consumption modeling and forecasting (BECMF), providing generally better prediction accuracy compared to single models. Obviously, every gain comes at a cost, that for ensemble methods is the complexity increase: in fact, their

implementation can be some how challenging and it usually requires a good machine learning expertise to select the right approach. Furthermore, it is considered as a fully black-box process and their case studies until now have been performed on short-term time horizon forecasting [20].

2.2 Forecasting horizon

Forecasting horizons are usually split in short, medium and long-term [75, 130], each of them aiming at different energy management and saving scopes, that can be listed as follows in descending order of citations:

- short-term: ranging from the next minute to the next week, it is an essential tool for building energy systems management such as HVAC systems [38] or to manage local energy production, storage and supply [21]. It represents the majority of the case studies.
- medium-term: with a range from the next week to several months ahead, it is implemented mainly for energy storage systems and building equipment maintenance planning [90].
- long-term: it provides information from the next year and over and it is widely used for designing and planning tasks [90].

Although essential for long-term sustainability strategies, only a minor proportion of the case studies seem to cover this time horizon[20].

2.3 Forecasting scope

Concerning the purpose of forecasts, the literature shows that a large proportion of publications refer to overall energy forecasts and forecasts of thermal and cooling loads (separate or combined), while a very small proportion of studies deal with other loads forecasts. In this regard, the literature shows that there is a gap with regard to the forecasting of other energy loads, such as lighting and motive power, which in many cases can account for a large share of the overall demand in certain

building categories, especially the less complex ones such as residential buildings [35, 45].

2.4 Accuracy metrics

The choice of the right metrics is one of the key steps for the successful training of a machine learning algorithm and the optimisation of its hyperparameters. In general, it can be stated that each metrics provide unique informations about model forecasting performances and behaviour, whose value lies in the choice of the most representative metric during model training and optimization of its hyperparameters, as well as in the comparison with other metrics in order to obtain further information useful for understanding the phenomenon and how to improve the forecasting model. In fact, they are able to prioritise the robustness of a model or its optimisation for specific tasks, depending on what the objective of the work being done is. In general, every metric start from the definition of error e_t, that is intended as the difference between the observation Y_t at time t and its corresponding forecast F_t, in formula:

$$e_t = Y_t - F_t \tag{2.49}$$

According to the literature, there are four types of metrics that can be used to measure time-series forecast performances[52]:

- scale-dependent measures;
- measures based on percentage errors;
- measures based on relative errors;
- relative measures.

Scale-dependent measures are among those commonly used in literature and their took their name from the fact that their scale depends on the scale of data itself. This type of metrics are useful when used on the same dataset or set of data having same scales, while it should be avoided across datasets that have different scales (unless a scaling process, obviously). Among the most widely used scale-dependent metrics there are those based on absolute errors and squared errors:

$$Mean\ Squared\ Error\ (MSE) = mean(e_t^2) \tag{2.50}$$

$$Root\ Mean\ Squared\ Error\ (RMSE) = \sqrt{MSE} \tag{2.51}$$

$$Mean\ Absolute\ Error\ (MAE) = mean(|e_t|) \tag{2.52}$$

$$Median\ Absolute\ Error\ (MdAE) = median(|e_t|) \tag{2.53}$$

$$Mean\ Bias\ Error\ (MBE) = mean(\frac{|e_t|}{n}) \tag{2.54}$$

From an historical point of view, MSE and RMSE are popular among authors because of their theoretical relevance in statistical modeling. The second one is more widely used because of being in the same scale of the data, although their sensitivity to outliers caused their use to be not reccommended in the evaluation of forecast accuracy [10].

The second family of metrics, that are based on percentage errors, has the advantage of being independent of the scale of the data. This makes them particularly useful and therefore widely used to make possible performance comparisons of models operating on different datasets. Defined the percentage error as follows:

$$p_t = 100 \cdot \frac{e_t}{Y_t} \tag{2.55}$$

the widest used percentage error-based metrics are:

$$Mean\ Absolute\ Percentage\ Error\ (MAPE) = mean(|p_t|) \tag{2.56}$$

$$Median\ Absolute\ Percentage\ Error\ (MdAPE) = median(|p_t|) \tag{2.57}$$

$$Root\ Mean\ Squared\ Percentage\ Error\ (RMSPE) = \sqrt{mean(p_t^2)} \tag{2.58}$$

$$Root\ Median\ Squared\ Percentage\ Error\ (RMdSPE) = \sqrt{median(p_t^2)} \tag{2.59}$$

The main flaw, or disadvantage, of this type of metric is the fact that it gives an infinite or indefinite result when $Y_t = 0$ for any t in the period of interest or has an extremely skew distribution for values close to zero. The most immediate consequences are that generally MAPE may be often larger than MdAPE and it is

impossible to rely on these metrics if data involves small counts, because Y_t equal to zero may occur frequenty.Further related to this is the fact that these metrics, in order to be used effectively, presuppose as an assumption that the measured quantity is characterised by a significant zero, which for example makes them unusable in the case of temperatures in degrees Fahrenheit or Celsius. A further element of caution in the use of these metrics is that they generally place much greater weight on 'positive' errors than on 'negative' ones. By virtue of this, in order to mitigate the negative aspects while retaining the positive ones of a metric independent of the scale of the data, alternative metrics to MAPE and MdAPE[63], known as 'symmetrical' metrics, or transformations, e.g. on a logarithmic basis, have been evaluated over time to make them more stable. [29]

The third family of metrics, based on relative errors, is also independent of the scale of the data and is characterised by a different scaling of the data. In particular, these metrics refer to the comparison between the error e_t and the error obtained through another benchmark forecasting method e_t^*, formulated as follows:

$$r_t = \frac{e_t}{e_t^*} \tag{2.60}$$

Usually the benchmark method mentioned is the random walk, from which derives that $F(t)$ is equal to the last observation. From the $r(t)$ formulation is possible to derive the following metrics:

$$Mean\ Relative\ Absolute\ Error\ (MRAE) = mean(|r_t|) \tag{2.61}$$

$$Median\ Relative\ Absolute\ Error\ (MdRAE) = median(|r_t|) \tag{2.62}$$

$$Geometric\ Mean\ Relative\ Absolute\ Error\ (GMRAE) = gmean(|r_t|) \tag{2.63}$$

and so forth. From literature, however, it results that among the most used relative errors based metrics there are the MdRAE and GMdRAE [11, 39]. Moving to their use and characteristics, an element that makes the use of this type of metric critical lies in the fact that the benchmark error e_t^* has a positive density probability for zero values,resulting in a particularly high variance (ideally infinite) of $r(t)$

when e_t^* has small values. In order to fix this inconvenient, by literature is suggested the use of "winsorizing" to trim the extreme values[11]: this process help to avoid the above mentioned hurdles related to e_t^* but conversely it translates in additional complexity to calculations and, morover, a certain grade of arbitrariness in specifing the amount of trimming.

As a final category of metrics for measuring performance in time series forecasting there are relative measures, which are represented for example by the relative mean absolute error (RelMAE), expressed as follows:

$$Relative\ Mean\ Absolute\ Error\ (RelMAE) = \frac{MAE}{MAE_b} \tag{2.64}$$

$$Normalized\ Mean\ Bias\ Error\ (NMBE) = \frac{MBE}{F_t} \tag{2.65}$$

Similar measures can be defined also for RMSEs, MdAEs, MAPEs, etc., according to the specific situation and need. In more detail, for such calculations, the random walk or "naïve" method (where F_t is equal to the last observation) is the most commonly used benchmark method for this type of calculations, together with the frequently used mean method (that is, F_t equal to the mean of all the observations). The main advantage of this methodology is their intepretability, by making it immediately clear if the proposed forecast method is bringing an improvement compared to the benchmark forecast method: for instance, RelMAE < 1 means that the proposed forecast method is performing better than the benchmark and conversely for RelMAE > 1. On the other hand, among the negative aspects it is worth to mention that this metrics require several forecast on the same series to make it possible the MAE (or MSE and so on) calculation. A common situation where it is not possible to implement such metrics are those when accuracy forecast is measured out-of-sample across multiple series. Despite their horizon can be only one, it make no sense to measure for example the MAE across multiple series, because of the different scales they may have. Among the relative measures, another metric used is the "Percent Better" that is formulated as:

$$PB\ (MAE) = 100\% \cdot mean(IMAE < MAE_b) \tag{2.66}$$

$$PB\ (MSE) = 100\% \cdot mean(IMSE < MSE_b) \tag{2.67}$$

where I is an operator that yields the values of zero or one, in accordance with the expression:

$$I(e_t) = \begin{cases} 1, if MAE < MAE_b \\ 0, if MAE > MAE_b \end{cases} \tag{2.68}$$

$$I(e_t) = \begin{cases} 1, if MSE < MSE_b \\ 0, if MSE > MSE_b \end{cases} \tag{2.69}$$

for MAE and MSE respectively and similarly for any other metrics used. This kind of metric has the merit to highlight the percentage of cases when a forecast method is better than the benchmark method, but it does not return any information about the possible improvement that can be achieved. For example, the forecast method could have a MAE slightly better than MAE_b in 99 cases and one single case with a MAE far worse that could vanish all the 99 slight improvements, without the metric returning this information.

Sorted in descending order of use in the literature, among those most frequently used are[20]:

- Mean Absolute Percentage Error (MAPE);
- Root Mean Square Error (RMSE);
- Coefficient of Variation of RMSE (CV-RMSE);
- Mean Average Error (MAE);
- Coefficient of Determination (R^2);
- Mean Square Error (MSE);
- Mean Relative Error (MRE);
- Mean Bias Error (MBE);
- Normalized Mean Bias Error (NMBE).

In some rare cases, specific metrics are also available in the literature, such as relative error[62], average error[80] and accuracy rate [116], but will not be addressed in this thesis, as the aim is to maintain a general, replicable and robust approach to individual applications.

In more detail, according to literature sources above mentioned, MAPE, RMSE, CV-RMSE and MAE have the largest share and are respectively assessed in 53%, 47%, 38% and 36% of the reviewed studies, while the other remaining R^2, MSE, MRE, MBE and NMBE are far less used, in detail they are implemented in 27%, 16%, 9%, 2% and 4% of the cases.

As it can be easily imagined, each metric provides specific information on forecast performance and model behaviour for each dataset [52]. One of the first choices, for example, is whether to employ unit-based (such as MSE or RMSE) or error percentage metrics, where the firsts should be preferred if the study is focused mainly on a single model, dataset or building typology, while contrariwise the seconds bring more information in comparing a moltitude of them. A further element influencing the choice of metrics lies in elements surrounding the problem addressed, but no less important for that, such as data availability. It should come as no surprise that the most frequently used metrics correspond precisely to those that require less data in order to be considered sufficiently representative, since, according to the literature consulted, in the majority of case studies (namely 59%) the simulations are run with less than a year's worth of available data[20], an amount often not sufficient to evaluate the use of other metrics than the firsts four, with significant and meaningful results.

2.5 Data characteristics

Input data are the driver of all the approaches and techniques, directly affecting modeling and forecasting performances. Data characteristics can be classified in the following types [20]:

- origin of data;
- features;

- timeseries granularity;
- amount of data used for training;

Looking at the origin of the data, in their turn are listed as real, simulated or benchmark, depending if they are extracted from devices or documents (like energy meters/environment sensors or billings respectively), physics-based models related to existing or unexisting buildings (using softwares like Energy Plus or Trynsis, for example) or publicly-available dataset for research scopes and comparing forecasting algorithm performances respectively. Moving to the features, according to the literature above mentioned, it can be divided in six groups:

- weather data, grouping all the weather data related to outdoor conditions;
- indoor environment, used to characterize the indoor building conditions;
- occupancy and occupants behaviour;
- time indicators, that deliver informations about how the building is used and its energy behaviour;
- past time-steps, accounting the potential impact of past events on the present and predicted building energy needs;
- building characteristics, that is information about the building passive and active systems, maily related to the envelop and plants/utilities respectively.

Other parameters or types of information, such as barometric pressure, cloud cover or evaporation, are usually given little consideration due to their little influence on building energy consumption and requirements. Regarding all of the informations above mentioned, of course, involve an effort to be acquired and, consequently, can be said to represent a cost, be it purely technical or even economic. The first two types are considered to be among the most influential in affecting the energy consumption and needs of buildings: in more detail, outdoor air temperature, outdoor relative humidity and solar radiation are included in several studies in the literature[20] and, because related to the internal comfort of the buildings, especially the tertiary ones, they are strongly related to energy consumptions and needs

[4]. Furthermore, these variables are also among those usually easy and cheap to acquire, as they are normally available from sensors, simulations or possibly public databases. In this regard, variables such as building occupancy, on the other hand, are among the least used precisely because they are difficult and expensive to acquire and, above all, to update over time, despite the fact that the energy needs of the building and its use are recognized among those with the greatest impact. Precisely, because one of the main challenges lies in the acquisition of the occupancy measurements of the rooms and the building[132], in some cases reference has been made to estimates occupancy based on the opening hours of the buildings: in more detail, time related variables, for example type of the day, day of the week and time of the day are considered in order to replace other time-dependent measurements such as occupancy, weather and equipment usages. Similar considerations can be done for past load energy demands, which are considered because of the information they can bring related to buildings operating conditions and similar events to the future states of the building energy demand. Despite their widely recognized importance and a good number of publications reling on this data characteristics, in many cases availability issues arise, with only a minority share of publications that can lean on one or more years of data.

Thus moving on to the third category of data characteristics, timeseries granularity, it is usually linked to the need of the studies being carried out and does not represent a technical limit, obviously except in cases where there are no monitoring systems adequate for the purpose. In this regard, small timestamps or high frequency samplings, for example 1 minute, are usually utilized in all those situations where one is interested in being able to identify and reproduce specific events on small time scales that influence the energy demand patterns of a building. On the other hand, they can induce high variances in the temporal forecasts of energy demands, introducing complexity and, consequently, important challenges in obtaining accurate forecasts[70]. On the contrary, larger granularities, such as data ranging from weekly to monthly, are usually useful for acquiring informations about the characteristics of the envelope[111] or because they are linked to socio-economic aspects[134], but in no way are they applicable and useful in forecasting the energy needs of a building for its daily management. The fourth and last category of data characteristics is the

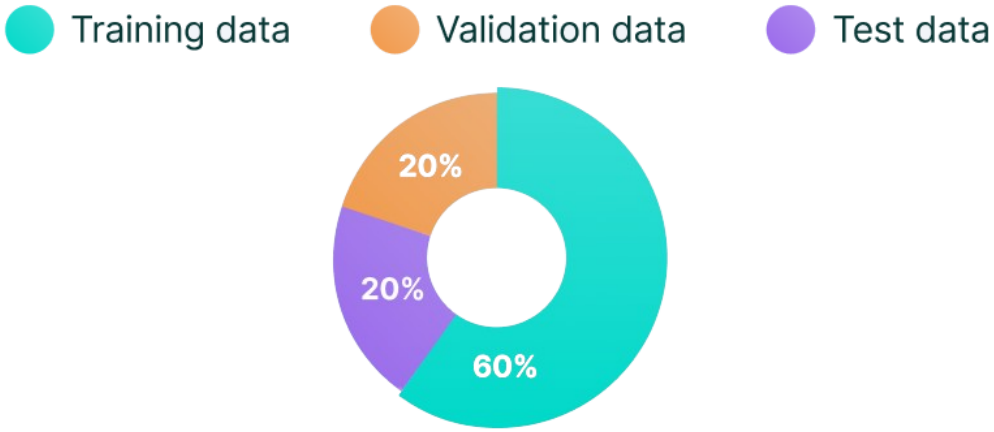

FIGURE 2.10: Typical training-validation-test data split ratio[12]

amount of data used to train, test and validate the models. This characteristic is, obviously, particularly important in all those so-called data-driven techniques and, for which, it generally turns out to be one of the most painful points, since such techniques usually need significant amounts of data from the first training phase for the performance and the related metrics to be satisfactory. With reference to the latter, in particular, the literature review shows that almost half of the presented case studies rely on a quantity of data between one and six months and a slightly lower share characterised instead by datasets with more than a year of data[20]. In addition to the amount of data itself, it is also very important how the data is divided between training, testing and the possible validation set: in the literature, there are different data splitting ratios, but it can be considered that a good ratio is around 80:20 training/test set [55], or 70:20:10, if the validation set is also present[12].

2.6 Data preprocessing

Data preprocessing, also known as data cleaning or data preparation, is the process of detecting and correcting corrupt or inaccurate records from a dataset, identifying incorrect, incomplete, irrelevant parts of the (raw) data and then modifying, replacing or deleting the dirty or coarse data[97]. It is a crucial step in every modelization project, especially those involving machine learning ones, but at the same time it is important to highlight that no all the data have to undergo it, according to literature.

Typical application of data preprocessing techniques are noise removal, features extraction, outliers minimization or removal, dealing with missing attributes/values, fractured data reparation, determining incompatibility among other used parameters, mapping relationships (multiple parameters) and other purposes [36].

2.7 Research gap

The literature review has highlighted that a wide range of solutions are available for forecasting the energy needs of a building. Indeed, the choice between single models and combined models highly depends on the type of forecast and on the specific requirements of the case study. Looking for the best model or algorithm for each application is far from being a real artificial intelligence approach since it always relies on human setup and skills. Furthermore, it must be pointed out that, usually, a good approach today might not be as good in the future, so there is no sense in looking for the best global model and best algorithm for a case-based situation. In addition to this, many self-claimed artificial intelligence models may be weak in terms of how they adapt to unfavorable contexts, such as lack of data. Every mentioned algorithm relies only on monitoring data, which makes them useless if the monitoring system is unavailable. Moreover, they are not robust enough to manage unforeseen circumstances because of too intrusive statistical-based data-cleaning approaches, case-based models, and time-consuming hyperparameter optimization algorithms. For all these reasons, it is believed that the most appropriate approach is to equip the algorithm with a set of models (e.g., combined models) and optimization tools that can deal with different events that it might face, predicting the building energy demand with adequate accuracy and precision. Furthermore, this approach can prove useful when there is a lack of data.

In addition, to the best of the author's knowledge, the available literature does not address the problem of data scarsity through similarity methods, resorting rather in some cases to data generated by simulations (so-called synthetic data). Furthermore, the models usually are difficult to adapt and reuse in other contexts or in the presence of events that are not clearly defined within the source dataset. This thesis fills this research gap by introducing a new method having the following fully

automatic features:

- A similitude criteria, which allows to overcome the data lack issue through the historical data from similar buildings, in order to effectively train and test the model;
- A data preprocessing and hyperparameter optimization procedure, which aims at the same time to further improves the combined model performance and making its development human skills free;
- A methodology to compare different data-driven models and the select of the most suitable onem based on performance evaluation metrics that can be modified according to specific needs, making it adaptable to every context.

Method	Pros	Cons	Reference	Use case
LR	Easy and fast	Potentially inaccurate in short-term forecasting	[7]	University, electricity
			[87]	Office, heating and cooling
			[61]	Office, cooling
			[69]	School, heating and cooling
DT	Flexible and interpretable	Instable with noise and outliers, not good for nonlinear data	[134]	Residential, heating, cooling, electricity
			[111]	Residential, heating, and cooling
			[112]	Residential, electricity
ANN	Highly flexible and adaptable	Case dependency, subject to overfitting	[3]	Tertiary, electricity
			[65]	Residential, electricity
			[38]	Educational, cooling
SVM	Manage large and complex datasets, effective with high dimensional features, robust	Complex set up, resource intensive, sensitive to hyper-parameters	[83]	Residential, heating and cooling
			[66]	University and offices, electricity
			[43]	Office, electrical
Improved models	High accuracy and automation, able to handle complex tasks	Resource intensive, data-hungry, complex	[27]	Residential, heating and cooling
			[135]	University and offices, electricity
Ensemble models	Robust versatile, easy to implement	Opaque, resource intensive, complex	[104]	Residential, heating and cooling
			[120]	Institutional, electricity

TABLE 2.1: Data-driven approaches for forecasting building energy demand.

Chapter 3

Materials and Methods

3.1 Introduction

The literature review has highlighted that a wide range of solutions are available for forecasting energy needs of a building, but no one of them is a real winner in this task. Looking for the best model or algorithm for each application is far from being a real artificial intelligence approach, relying every time on human setup and skills. Furthermore, it must be pointed out that usually a good solution today might be a bad one for the future, so looking for the best model and best algorithm for a case-based solution sounds like a no-sense. In addition to this, those data-driven approach or self-claimed artificial intelligence models are weak on a main aspect that differs on what is intelligent from what it is not: adaptability to unfavorable context, i.e. lack of data; every algorithm mentioned relies only on monitoring data, that makes them useless if the monitoring system is not available, and are not robust enough to manage unforeseen circumstances, because of too intrusive statistical-based data-cleaning approaches, case-based models and time-consuming hyperparameters optimization algorithms. Accordingly, the first one is to identify a structure that is similar to the one under investigation and for which it is possible to derive energy demand profiles. To do this, criteria have been established to trace similar structures, even when a substantial database of information is not available. This work aims to provide a codified and usable form of criteria and ways of operating usually shared by professionals, of which little or nothing can be found in literature and which are often learned experientially. In more detail, the similarity approach is based on a multi-level validation, where the first level is based on the

physical/structural characteristics of the buildings, while the second level uses artificial intelligence techniques to analyze energy demand profiles and patterns. This double validation clearly makes the results more robust, as the similarity approach is based both on objective characteristics of the buildings and on the analysis of their energy demand behaviour. This makes the similarity criterion overall easier to understand and adapt to the individual context, without giving up the power of more structured, but less transparent, methods such as those in the AI field. The two levels of similarity validation allow for an initial "static" assessment based on the overall "structural" characteristics of the buildings (defined as geometry and intended use), while the artificial intelligence models enable a "dynamic" assessment of their behavior, capturing complex and nonlinear relationships in energy demand patterns. This final step allows for the validation or, if necessary, correction of the preliminary conclusions from the first one, providing a more complete and sophisticated assessment of the actual similarity between the buildings.

In this sense, the "first level" based on characteristic ratios provides an initial quantitative assessment of similarity. This allows to exclude buildings that are too different from each other, thus reducing the risk of comparing intrinsically inconsistent buildings and improving the algorithm's efficiency, focusing computational resources and more complex analyses on potentially similar ones.

Once a similar structure or energy demand profile has been identified based on available consumption information, the first consideration when making predictions of any kind is that the historical series on which they are based and referred to must be representative of the phenomenon under analysis and, for this reason, free of outliers that negatively influence the results.

After the data cleaning step, input variables of the dataset are normalized, in order to balance all the input features and mitigate the effect of variables with large numerical values or extensive ranges, that could exert a disproportionate influence on the model's performance. This normalization procedure, indeed, aims to avoid that single variable unduly skews the model's predictions due to its scale or variability.

At the end of this phase, as the final preprocessing step, for each of the energy needs to be forecasted, the dataset will be divided into a training set, a test set, and

a validation set. In more detail, the training set and the test set will be used to train and test the model used to validate the similarity between the target building and the building selected as similar, while the training set and the validation set will be used for the subsequent predicting phase, in which a futher model is used to forecast the energy needs of the target building.

After the "first level" and the preprocessing phase, the "second level" of the similarity criterion takes place, aimed at identifying through a "cooperative learning" approach the best AI model to validate and represent the similarity between buildings. In more detail, the cooperative learning is an artificial intelligence approach that focuses on the collaborative analysis of different data sources or models, with the goal of identifying and validating common patterns and similarities between them. This is not a true knowledge transfer, but rather an exploratory process that aims to understand whether and to what extent different domains share significant characteristics: in the specific case of buildings energy needs, this phase allows to verify whether two buildings exhibit similar energy behaviors, statistically validating these similarities through machine learning models.

At an operational level, the algorithm selects the most significant variables for each model and the target variables, ranking them by importance based on their contribution to maximize the chosen metric. It employs a progressive scoring system, utilizing only the highly correlated variables or incorporating additional variables if they enhance the metric, and it aims to optimize the model's performance while mitigating overfitting issues. At this stage, the model with the highest $R^2_{adjusted}$ is considered more reliable, while the other metrics have more a control scope.

In this regard, using a specific model, especially the one with the best metrics, to compare and validate the similarity between buildings ensures several advantages:

- a fair comparison, as using the same model for all buildings to be compared with the target building ensures a uniform basis for comparison: any differences in the results, therefore, will be due to the characteristics of the buildings themselves and not to the type of model used;

- a standard procedure, as the use of a single model allows for a single, consistent comparison method across all buildings, simplifying the analysis process and thus allowing for clearer interpretation of the results. Furthermore, if a model has proven to be more effective at identifying common energy demand patterns between the target building and potentially similar buildings, it clearly makes sense to use it for all the next comparisons (given its greater ability than other models to detect similarities and differences, as well as providing more reliable and consistent results);

Comparing the results between multiple models, in fact, would have introduced greater complexity, due to the need to manage multiple models simultaneously, but above all greater difficulties in interpreting and validating the results, especially in the case in which conflicting results were to emerge, making it difficult to justify which one to consider more reliable and with the risk of ending up making a completely arbitrary average of the results; in general, a single well-chosen model certainly provides clearer and more direct results, thus simplifying the subsequent presentation and discussion phases, as well as making the methodology itself more linear and understandable.

Once the most performing model and the set of variables most correlated to the phenomenon have been identified at the end of the cooperative learning phase, the predictive (i.e. transfer learning) phase of the model begins. In more detail, transfer learning represents a more advanced technique that allows to transfer the knowledge acquired from a source domain to a target one. This approach is particularly effective when limited data is available in the target domain, as it allows the use of learning performed on a related task. In the context of building energy needs forecasting, once similitude has been validated through cooperative learning, transfer learning is implemented by training a new model on data from the source building (that is similar) to forecast the energy needs of the target building. This knowledge transfer is possible thanks to the preliminary validation of energy similarity and the selection of relevant variables identified during the cooperative learning phase. The new model then leverages the acquired understanding of the source building's energy patterns to generate accurate forecasts for the target building.

The use of two separate models, one for similarity and one for forecasting the energy needs of buildings, is believed to have the following advantages:

- better performances, as the similarity model is optimized to identify common patterns between the target building and buildings selected as similar, while the predictive model is specialized to forecast time series and, therefore, energy needs: this also ensures that each model can be optimized for its specific task;
- method transparency, as the separation of tasks between the two models makes their use and mutual purpose clearer: the first model has the task of validating the similarity hypothesis coming from the "first level", while the second model applies this knowledge for forecasts;
- control, as the use of two distinct models makes it easier for each phase to understand the results of each phase and better identify and resolve any problems that may arise, as well as optimise each step separately.

Once these steps are completed, both the model used to verify the similarity between the buildings and the one used to predict the needs of the target building, based on knowledge of those of the similar building, are selected.

3.2 First-level similitude

The algorithm developed requires following the steps outlined in Figure 3.1, which aims to replicate the steps typically taken in investigating the energy performance of a building (as described, for example, in the UNI CEI EN 16247-1 standards regarding energy audits). In particular, to establish similarity between two buildings, it is necessary to investigate first the following variables:

- Intended use;
- Climate area;
- Year of construction;
- Geometric factors (volumes, surfaces, number of floors above ground);
- Energy needs ratios;

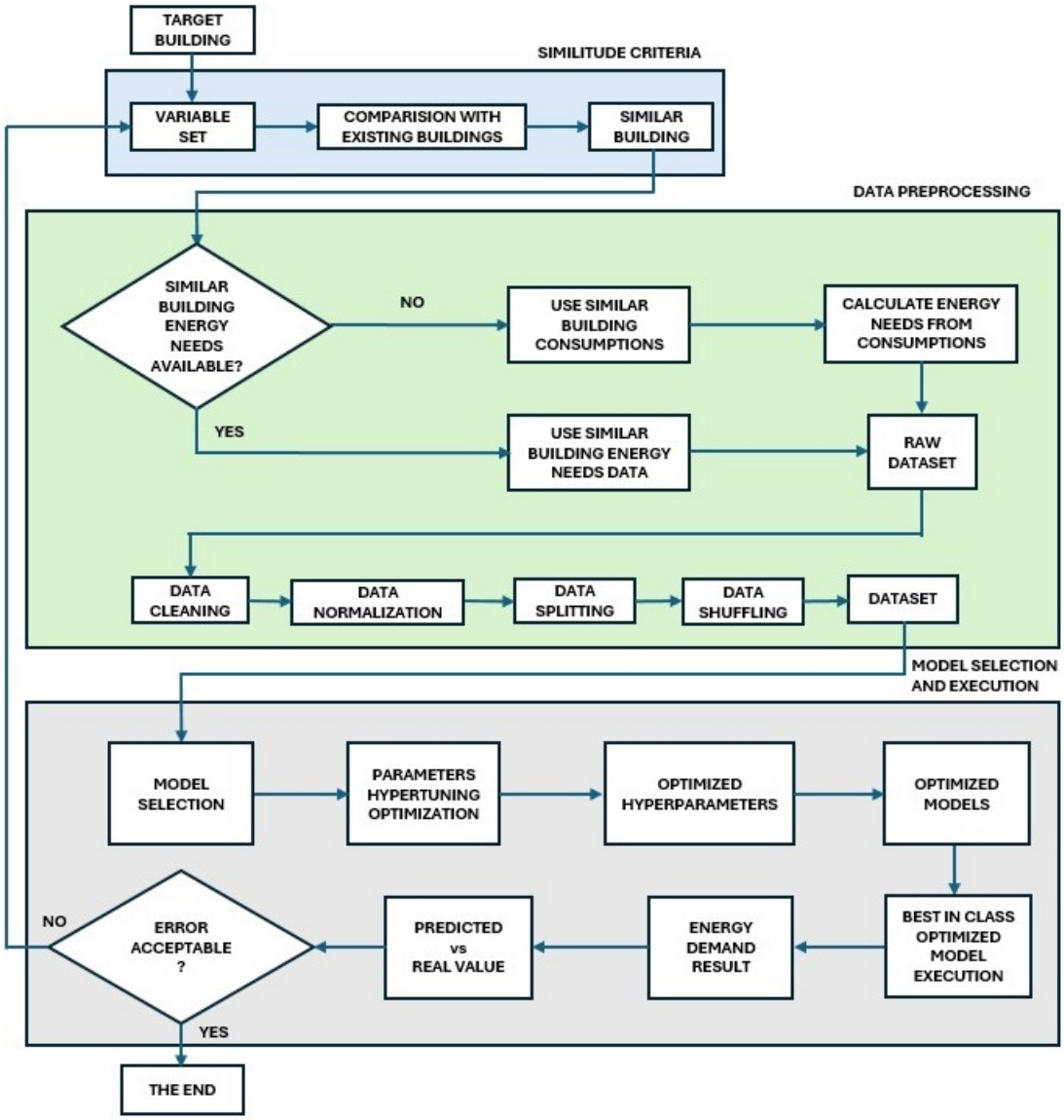

FIGURE 3.1: Steps constituting the integrated artificial intelligence method for forecasting buildings energy needs

- Beds (in the case of hospital facilities).

The variables above listed are considered having in mind the criterion that similar buildings have similar energy demands. In the list order, the intended use typically drives the energy demand profiles shape on different time scales, whether short, medium, or long term. The climate zone typically impacts the magnitude of the energy demands, especially those correlated to the weather factors like thermal and cooling needs. The year of construction, instead, is generally linked to the construction criteria adopted and, consequently, with the energy class and, therefore, energy losses of the building (recently constructed buildings, in particular, must meet specific requirements, as stated in the European Directive 2010/31/EU). Geometric factors, such as surfaces and volumes or their ratio (known as building form factor), affect both the definition of comfort-related needs and energy losses: in other words, the building envelope efficiency level. Finally, in the case of hospital structures, the number of beds is typically an indicator of the level of service and the technological intensity compared to other buildings with the same intended use: in particular, low number of beds with the same surface area indicates a high level of technological intensity, due, for example, to a high incidence of intensive care units, resuscitation or high-dependency units. Once a set of buildings having similar characteristics is identified, it must be checked whether they have or not characteristics that could be considered in similitude with each other. In this regard, the various energy demands are analyzed and the quantities influencing their behavior and, consequently, their profiles are determined. In more detail, it is well-known that heating is influenced by volume and heating degree days during winter, while cooling by surfaces area and cooling degree days during summer. Because they are also impacted by the building level of service and its technological intensity, a third element of similarity is added to these two elements: for hospital facilities, because the more they are advanced and the more space they offer to their residents, it is decided to combine it with the total surface area and the number of beds ratio. The same applies to electricity and steam consumption, which are usually related to the intensity of care provided and the number of hospitalizations that the facility can accommodate. In general, each requirement of the i-th facility is assumed to be a function of

given combinations of parameters, as reported in the following equations:

$$HD_i = f(V_i, HDD_i, \frac{S_i}{B_i}) \tag{3.1}$$

$$CD_i = f(S_i, CDD_i, \frac{S_i}{B_i}) \tag{3.2}$$

$$ED_i = f(\frac{S_i}{B_i}) \tag{3.3}$$

$$SD_i = f(\frac{S_i}{B_i}) \tag{3.4}$$

Hence, the similarity criterion between i-th and j-th facilities provides that:

$$HD_{i,j} = k_{HD_{i,j}} \cdot HD_i = HD_j \tag{3.5}$$

$$CD_{i,j} = k_{CD_{i,j}} \cdot CD_i = CD_j \tag{3.6}$$

$$ED_{i,j} = k_{ED_{i,j}} \cdot ED_i = ED_j \tag{3.7}$$

$$SD_{i,j} = k_{SD_{i,j}} \cdot SD_i = SD_j \tag{3.8}$$

where:

$$k_{HD_{i,j}} = f(\frac{V_i}{V_j}, \frac{HDD_i}{HDD_j}, \frac{\frac{S_i}{S_j}}{\frac{B_i}{B_j}}) \tag{3.9}$$

$$k_{CD_{i,j}} = f(\frac{S_i}{S_j}, \frac{CDD_i}{CDD_j}, \frac{\frac{S_i}{S_j}}{\frac{B_i}{B_j}}) \tag{3.10}$$

$$k_{ED_{i,j}} = f(\frac{\frac{S_i}{S_j}}{\frac{B_i}{B_j}}) \tag{3.11}$$

$$k_{SD_{i,j}} = f(\frac{\frac{S_i}{S_j}}{\frac{B_i}{B_j}}) \tag{3.12}$$

Thus, the i-th structure can be held in similarity with the j-th structure if:

$$\frac{HD_i}{HD_j} = 1 \tag{3.13}$$

$$\frac{CDi}{CD_j} = 1 \tag{3.14}$$

$$\frac{ED_i}{ED_j} = 1 \tag{3.15}$$

$$\frac{SD_i}{SD_j} = 1 \tag{3.16}$$

within a range of +/- 10 %. Once this step has been completed, the j-th building can be assumed similar to the i-th building. The creation of the raw dataset requires, at this point, that the data of the j-th structure (whose consumption is unknown) are simply linked to those of the i-th structure (the consumption of which is known).

3.3 Second-level similitude

As mentioned in the introductory chapter, the second level of similarity aims to verify and validate its existence at the "dynamic" level, identifying common patterns and complex relationships in the profiles of different energy needs. To this end, it is necessary to define dedicated metrics that provide a measure of the validity of this similarity, select the most influential variables, and optimize and train a model that serves as a common basis for further comparisons. The following paragraphs provide more information on the individual steps mentioned above.

3.3.1 Preprocessing

Data preprocessing stands as a critical cornerstone in the thesis pipeline, playing a pivotal role in the efficacy and reliability of predictive models. This essential step involves a series of automatic techniques aimed at transforming raw data into a format that is both suitable and optimal for subsequent analysis and modeling. In this context, preprocessing not only enhances the quality of input data itself but also significantly influences the performance, interpretability, and generalizability of machine learning algorithms implemented. Techniques such as normalization, feature scaling, handling missing values and dimensionality reduction are instrumental in mitigating biases, reducing noise, and uncovering latent patterns within the dataset. Moreover, effective preprocessing can address common challenges dealing with real-world data, fostering more robust and accurate models. The judicious application of preprocessing methods lead to improved convergence rates during training, reduced computational complexity, and enhanced model stability across

diverse datasets and events. As such, a thorough exploration and implementation of data preprocessing techniques is not merely a preliminary step but a fundamental aspect that can significantly shape the trajectory and outcomes of machine learning models, warranting in-depth investigation and analysis within the scope of this work.

3.3.1.1 Data structure and data type

Once the structure in similarity is identified, the algorithm proceed with "preprocessing" of the data. As it can be observed by figure 3.1. The first step of preprocessing involves establishing the structure and type of the datasource, which must then be processed by the algorithm to define the dataset used for training the building energy demands prediction model. To this end, the first step of preprocessing involves determining whether or not the energy demands of the building in similarity are available. In more detail, depending on the situation, two data structures are identified, which through appropriate steps will allow defining the rawdataset that will be preprocessed. If energy demand data of the structure in similarity are available, these will be joined to input variables mentioned in paragraph 3.2, constituting the raw dataset. If, on the other hand, the energy needs of the similar building are for some reason unavailable, such as because the building is very small and the benefits of more detailed monitoring would not outweigh the costs of installing one or more additional submeters, the algorithm can calculate them through the consumptions or absorptions of the primary energy vectors measured to the respective building meters and the efficiency chains between them and the building. Specifically, for heating, cooling or steam energy demands, the latter case will correspond to the respective substations, where present, or the main pipe system detachments from which distribution to the individual users usually takes place through specific circuits. For electrical energy demands, it will correspond to the low voltage delivery point, which can be the secondary substations or the meter itself, depending on the size and the building absorptions. Finally, if there are self-production systems, such as photovoltaic, solar thermal or cogeneration/trigeneration systems, their contributions to the building's energy needs must be appropriately taken into account. For thermal, cooling and steam vectors, if building energy demand meters

are installed at the branch off points or in the substations usually no special precautions are necessary. This is because their production typically flow into the main distribution network, making their contribution indistinguishable. Otherwise, once the energy demands of the building have been calculated from its consumptions, self-productions must be added to the latters, in order to obtain the actual building energy demands. A small side note must be made, however, for dissipation systems, such as drycoolers or similar for the protection of internal combustion engines for cogeneration/trigeneration systems, which must be subtracted from the respective overall demand. A separate discussion must be made, however, regarding self-production of electricity. In the case of electricity self-production systems, such as photovoltaic or cogeneration/trigeneration ones, the amount of self-produced electricity must be added to the quantities measured at the general meter or at the distribution cabin of the building, subtracting the amount sold to the grid.

3.3.1.2 Data cleaning

Once the raw dataset is established, data cleaning represents a necessary process for accurate energy demands forecast. This is true in any strategy, model, and algorithm for many (if not all) real-world applications, where the data quality is largely impacted by the numerous situations and circumstances can affect the continuity and performance of the data collection systems. This can happen because of, for example, scheduled routine maintenance operations that need to be performed in a planned or unplanned way, making necessary to power off the interested portion of the system for safety reasons. For all these reasons, data preprocessing is a fundamental step to obtain the data quality required for training machine learning models to produce satisfactory results in terms of correlation, precision and accuracy. Indeed, the primary target is about preparing a dataset that allows the model to learn significant correlations between input and output variables, so that it has sufficient generalization capabilities to accurately predict unforeseen events. In order for the training process to lead to a robust model, the pre-processing phase must address the following key aspects:

- Outlier removal: anomalous or extreme values negatively affect the learning process, leading to inaccurate or poor results. For these reasons, it is key to

identify and handle outliers through techniques such as removal or replacement;

- Missing value imputation: datasets frequently present missing or incomplete data points, which need to be addressed through appropriate methods like statistical imputation or learning-based estimation;

- Handling extreme values: excessively high or low values of the same variable or compared to others have a negative impact model performance, making input variables range more important than its actual correlation with the output ones. To mitigate such extreme input values over the outcome, normalization and clipping techniques are frequently implemented;

- Bias correction: datasets may present systematic biases or distortions, that can even invalidate the model's learning process and its consequent results. Data sampling or weighting techniques are critical in identify and correct biases;

- Temporal dependency removal: data may be characterized by temporal dependency or even correlations among instances, which are obviously undesired while looking for input-output variables correlations [1].In this situation, randomization or shuffling techniques may be helpful to enable model learns generalizable patterns.

Unlike to what is available in literature, the data-cleaning approach proposed in this thesis is based on technical and engineering considerations. This choice is made in order to both isolate outliers according to objective criteria and at the same time still monitoring any anomalous phenomena occurrence. More specifically, the data cleaning process is based on the building's energy production plant technical constraints. This is made through the identification of its components imposing the slowest load ramp for each energy demand category, since all the others will necessarily have to adapt to them and regulations systems will prevent too ramp ups or downs that could impact the other items or machineries. Identifying the "weak" elements in the energy vectors supply chain clearly requires studying how and by what

[1] It is different in reducing the forecasting maximum error, a situation when it may be even a desired data feature

	Power rating [MW]	Operation range [%]	Ramp rate [%power\minute]
Gas boilers	Up to 300	16-100	4-6
CHPs	0.01-20	30-100	20-50
Electrical chillers \Heatpumps	0.08-6	50-100	10-20
Absorption chiller	0.1-7.0	20-100	10-20
Electric grid	—	—	1-10

TABLE 3.1: Ramp limits of different types of power plant.

each of them are supplied. Specifically, for each of the energy demands, it is identified the supplying item with the lowest ascent or descent ramp, as it is technically more vulnerable to stresses resulting from sudden load variations. Its correnspondding limits values are set as a threshold for identifying potential outliers for each energy demand to be forecasted. The refinement of the data cleaning thresholds then occurs thorugh successive iterations, analyzing the impact that the different values have on the forecast of the building's energy demand. Regarding the choice of limit values, the literature mentions those concerning fire tube boilers, cogeneration engines [7], absorption [1, 30] and electrically powered refrigeration units and heat pumps [49] and electrical distribution [108], as reported in Table 3.1.

It is observed that those to be cautiously managed are the thermal ones, mainly because of their usually high temperature differences, that can vary from 50°C degrees or more during the startup and shutdown phase to 10°C degrees while operating at nominal loads. In fact, other energy demands and their plants are characterized by less stressing temperature deltas. For example, cooling plants operate usually with no more than 10°C during startup and shutdown phase to achieve a temperature difference of 5°C degrees among supply and return flow at nominal load. Aside, the electrical plants typically present very steep ascent or descent rates, because the utilities can be turned-on or turned-off simply by opening/closing a switch: to this effect, the electric chiller units are the slowest items, reason why the electrical utilities acceptable ascent and descent ramps thresholds it is decided to be considered coincident with those of the cooling units. About the thresholds setup, it is worth mentioning that for a plant the typical most stressful and critical moment is represented by the startup phase. This is because the imposed forcings may be excessive compared to the structural and mechanical capacities of it, leading to materials

fatigue or even failures. On the contrary, the shutdown phase is less critical, because as a matter of fact it is physically impossible to impose a descent ramp on the plant. Even in the case where everything is turned off instantaneously, the decrease in its operating temperatures would be however linked to the dissipation rates, usually around 2%/hour for plants of the sizes being studied. The only exception is represented by electrical plants, for which the previously mentioned reasoning applies. In light of these considerations, for all the energy demands analyzed, it is decided to impose as acceptable limit thresholds equal values for all the energy demands according to the logic above mentioned.

3.3.1.3 Data normalization

Following the data cleaning step, the acquired dataset undergoes min-max normalization. This process is essential to mitigate the disproportionate statistical influence of variables characterized by large magnitudes or variation ranges over the smaller ones. Normalizing them, namely making them vary among 0 and 1, allows to prioritize instead their actual correlation with the output variables. Once normalization is completed, a further step is needed before splitting the dataset in a traning and a test set and then proceeding with training the machine learning model.

3.3.1.4 Data splitting

Once the dataset has been normalized, it must be divided into a training set, a test set, and a validation set. Data splitting is a fundamental preprocessing step that can significantly influence the training of AI models and, therefore, have a significant impact on their performance and generalization ability: a correct split helps to avoid overfitting problems and ensures an objective evaluation of the model's performance.

In more detail, the first phase involves dividing the data into training sets, test sets, and validation sets, where the data from the similar building constitutes the training set, while the data from the target building constitutes the test set and the validation set in a proportion that must be defined (usually 80/20).

At this stage, the method sets a different use of datasets depending on the phase in which the algorithm is operating. Specifically, the similarity verification phase

involves a cooperative learning approach, in which data from the training set and the test set are shuffled before proceeding with their split, model training and hyperparameter tuning. The goal is to ensure that the model is trained on a dataset representative of both buildings and then verifies its similarity on an equally representative dataset. To ensure that the similarity results are not given by chance or a particularly performing data mix, once the model is trained, cross-validation is performed both for hyperparameter tuning and for verifying model performance. Conversely, in the transfer learning phase, the training set consists solely of data from the similar building and the validation set from the target one. This is to verify the effective transfer of knowledge from the energy demand model of one building to the other.

This strategy differentiation reflects the different objectives of the two approaches: cooperative learning aims to identify common and complex patterns across buildings, while transfer learning aims to effectively transfer knowledge from one to another. Careful selection of the splitting strategy is therefore crucial to the methodological validity of the study and the reliability of the results obtained.

3.3.1.5 Data shuffling

Before proceeding with the model training, another crucial preprocessing step is the "data shuffling". In the context of this thesis, data shuffling plays a cornerstone role in the model training process of the cooperative learning phase. In more detail, it provides the dataset splitting into smaller batches or mini-batches for computational efficiency reasons, ensuring that each one containes a random and diverse subset of the entire dataset itself. Then, this process involves the randomly reorder of the data points sequence within the dataset. It is worth to mention that this is obviously carried out mixing training data with test ones. As described, this step allows to effectively break any inherent patterns or correlation that exists out of the original data collection sequence. The randomization mitigates also the impact of potential biases or patterns that may lie in the original data order. The immediate consequence is that the model can be trained on a more generalized and representative portion of the dataset, learning more robust and generalizable representations. A second aspect of no less importance is that this process of data mixing in fact introduces a

degree of independence between consecutive samples. This is an interesting property, especially for statistical and machine learning algorithms employing stochastic optimization techniques. Furthermore, data shuffling impacts also in the hyperparameters tuning: specifically, it ensures that the training batches are truly random and independent, preventing any specific subset of data from overly influencing the models. However, despite its benefits, it is interesting to observe that data shuffling is not universally applied. Time series data or online learning scenarios are among the cases where data shuffling might not be implemented, because the interest in preserving timestamps temporal dependencies. For the scope of the cooperative learning phase, because there is no interest in reproducing specific time series, but rather to find strong correlations between input variables and output variables, data shuffling is an excellent tool to break the interference of negative phenomena on the model training and increase its performance and robustness.

In this thesis work, the data shuffling process is implemented and performed through the built-in functions provided by the respective machine learning libraries, in order to ensure a consistent and reproducible approach across all the experiments.

To conclude this section, it is important to note that applying data shuffling in the transfer learning phase, on the other hand, can compromise the model's ability to recognize and reproduce specific time series, especially seasonal or periodic ones. For this reason, in the subsequent forecasting phase, it was decided not to adopt this technique in model training.

Completed the preprocessing stage, the dataset is ready to be implemented in the machine learning model training.In more detail, it must be split then into a training set and a test set. Due to the case study of this thesis, it is make sense to split the dataset taking as training set the known building dataset, while as test set unknown's one, setting consequently the size of the training set, the test set and their ratio.

3.3.2 Metrics selection

The choice of metrics is a step that as mentioned in section 2.4 is able to impact both the cooperative learning as well as the transfer learning phase, influencing both the models hyperparameters tuning and algorithm chosen for these scopes.

For this purpose, two types of metrics must therefore be distinguished: one as a criterion by which the similitude among buildings will be validated and another according to which the prediction performances will be considered acceptable.

More articulate, however, it is the process regarding the metrics on which the accuracy of the algorithm and its model for comparing energy needs is assessed.

In particular, while all the considerations made above remain valid, and considering what is reported in the literature at paragraph 2.4, in the present work it is decided to proceed by employing more than one performance evaluation metric: in more detail, a primary metric is established, according to which performance is deemed satisfactory or not, and secondary metrics are useful to address algorithm and models improvements.

Secondary metrics are introduced in order to have additional information about the performance of the model and the algorithm, so as to understand its actual behaviour depending on the structure under investigation and the expected energy requirements.

Indeed, while satisfactory degrees of correlation may emerge, it is also important that the model's accuracy and precision are known, so that the relative uncertainty can be effectively managed.

Regarding this, it is important to highlight once more that the aim of cooperative learning phase is to investigate if the similitude among similar and target buildings is consistent, training the model in such a way that it is actually able to establish significant correlations between their input and output variables. This may come even at the expense of a maximum error that is not so low, but to the advantage of the model's forecasting capabilities even when facing events or circumstances that are not contained in the starting dataset, but which can nevertheless be modelled: one thinks, for example, of a particularly extreme season or a particularly high number of hospital admissions. These are all situations that are in themselves anomalous if one considers that they are not contained in the starting dataset, but which can be equally adequately represented, if a particularly significant correlation is established between these input and output variables.

In particular, bearing in mind that the aim of the cooperative learning is first of all to identify a model capable of predicting if two or more buildings can be considered

"dynamically" similar or not, the main metric (also known as the 'north pole star' metric) will certainly be R^2. For these reasons, it is considered of primary importance to optimize hyperparameters accordingly. As a logical step, it was decided to adopt R^2 as optimization metric, useful to validate the buildings similitude though a metric representative of the correlation among them. So as to favour robustness and flexibility of the model when facing events not contained in the starting dataset and even considering the purpose of the thesis about predicting the energy demands of a building for which they are not known, starting from the data of a known structure, following this, further secondary metrics for comparison are MAPE, RMSE and maximum error.

In the next transfer learning phase, it was decided to use the same metrics mentioned above, albeit in a different order to fit the predictive scope than the similitude one. In particular, given the similarity between the two or more identified buildings, the aim of this phase is to train a model capable of predicting the energy needs of the target building starting from the similar one. In this sense, the north pole star becomes the MAPE, as it is more indicative of the models' actual ability to reproduce the time sequence and validate its performance, while the other metrics (including the R^2, which was the "north pole star" in cooperative learning) take on the role of secondary and, therefore, aimed at improving the model's performance. In particular, the R^2 is useful to verify whether the predictive capabilities are obtained for example at the cost of generalization ability (i.e. overfitting), while the max error and the RMSE give useful information on how representative the mean percentage error (MAPE) is.

3.3.3 Machine Learning Model and Variables Selection

Once the preprocessing phase is completed, the algorithm proceeds to select models to be subsequently optimized and compared, in order to identify the one that performs best for each energy demand. To this end, in the cooperative learning phase the algorithm first proceeds by identifying the variables most impacting the good performance of each selected model, assigning to each input variable a feature importance score based on their ability to increase model performance relatively to the chosen accuracy metric. By selecting only those variables that are most important

to model performance and eliminating those that are not, this step in the algorithm allows each model's performance to be optimized while reducing the risk of overfitting.Having completed this step, the algorithm then proceeds by selecting each of the models that are part of the set at hand in order to first optimize their hyperparameters and then make a comparison based on the chosen accuracy metric. At this scope, as mentioned in the previous chapters and paragraphs, one model is probably not enough to handle the complexity of the problem addressed. Based on this, it was decided to equip the algorithm with a set of machine learning models that could be chosen, optimized and compared. Precisely, these models are selected from the current literature, referred to in Chapter 2, and they are:

- Multi-Layer Perceptrons (MLPs);
- Support Vector Regression Machine;
- Random Forest;
- Extremely-Randomized Trees.

As mentioned, in order to identify the best performing model among them, the algorithm then proceeds with hyperparameter optimization for each one. Regarding the optimization methods and algorithms implemented, a more in-depth discussion is given in the following paragraphs.

It is important to underline that the variable selection takes place exclusively during the cooperative learning phase: the underlying hypothesis is that the variables, if they are useful for representing the similarity between the target building and the one identified as similar, will then also be valid for predicting its energy needs.

3.3.4 Hyperparameters optimization

The Machine Learning Model hyperparameters optimization is one of the key steps, both of the present thesis work and in increasing its performance. The hyperparameter optimization cannot take place during the training process itself, but rather it must be defined before training and then verified once the training has ended, looking at the improvement their optimization has brought on the performance metrics. In fact, their optimization control the general behavior of each machine learning

model, influencing its learning ability and, consequently, its performance. Taking into account the potential infinite amount of optimal or suboptimal solutions for the hyperparameters optimization problem and the number of models to be optimized, it is decided to introduce an automatic algorithm at the scope. This is among the major steps of this work thesis, considering that it frees the machine learning model hyperparameters optimization from the data scientist skills.

In more detail, the algorithm optimizes the hyperparameters of the various models in both the cooperative learning and transfer learning phases, using the objective metrics indicated in the paragraph 3.3.2. The following details the techniques used in this thesis for hyperparameter optimization.

3.3.4.1 Grid search

The first and simplest strategy that comes in mind dealing with non-linear problems and multi-variable optimization is to consider every possible solution, trialing every one taken from an imaginary grid drawn in the search space and picking the one that performs better accordingly to the optimization metric.In detail, Grid search is an intuitive, heuristic optimization method in which the design space is discretized into a finite number of mutually disjoint partitions of equal volumes [79]. The process of dividing the search space in many slices and searching the solution into it gives the name to the search algorithm itself: gridsearch.

This type of search and, therefore, optimization algorithm is considered a suitable candidate when[71]:

- the overall number of parameters to optimize (M) is small. Given a grid M-dimensional, the number of testable solutions amount to L^M , where L is the number of trial solutions along each M dimension of the search space;
- the solution lies within a specific range of values, that can be used to limit the the search space and the calculation needed;
- the problem can be computed quickly enough that the time needed to test the L^M solutions is feasible;

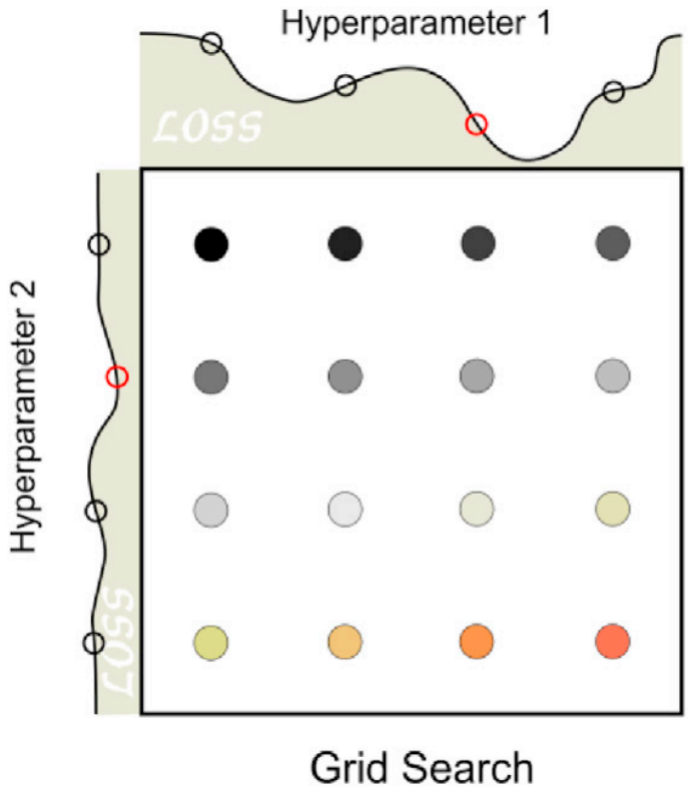

FIGURE 3.2: Representation of Grid Search hyperparameters search scheme

- the optimization function is smooth enough over the grid space scale Δm, ensuring that the maximum or minimum is not missed or the optimized value is not too far from the real maximum or minimum.

As a matter of fact, it is difficult to have an a priori idea of the behaviour of the black-box model as its hyperparameters are changed, which should lead one to discard this type of approach regardless. In the present thesis, however, it was decided to include it as it is equally true that a more brute-force approach may in any case be useful if more efficient optimisation algorithms in general encounter difficulties in identifying the optimal set of hyperparameters.

Despite decades of research in global optimization and the publishing of several optimization algorithms, there are many reasons why gridsearch is still on the edge of hyperparameters optimization state of art, that is:

- it gives some degree of insight about the search space;
- no technical barrier or overhead to the implementation;

- it is easy to implement and parallelize;
- it typically finds better trial point than manual search;
- it is reliable, especially on low-dimensional spaces (e.g. 1-d, 2-d).

On the other hand, grid search alone can lead to poor results in practice: this can be observed quite well through the figure 3.2 [82], in which it is evident how a fair distribution of the search space can lead to sub-optimal solutions, especially where the latter is characterised by high noise, reason why in the present work it was decided to explore further optimisation techniques.

3.3.4.2 Random search

Among the family of derivative-free or, so called, black-box optimization methods, Random Search is historically fully part of it. Its first appearance in literature is dated 1953 by R.L. Anderson [9], although it did not appear under the name by which it is known today until 11 years later by L.A. Rastrigin[92]. The main difference with the gridsearch method lies on replacing the exhaustive enumeration of all combinations by selecting them randomly, both on discrete and continuous search spaces. A counterintuitive benefit of this approach is the opportunity to explore much more values than a gridsearch could do, considering the same amount of iterations, outperforming it especially when in low intrinsic dimensionality problems, i.e. when only few hyperparameters affect the machine learning algorithm performances. Another interesting property of this approach is its embarassing parallel, that is the minimal or no-dependency upon communication among tasks or their results, making effortlessly the split of the components (i.e. hyperparameters optimization) in a number of tasks that can be executed concurrently.

Assuming a search space of n dimensions, the random search algorithm is operationally composed of the following steps:

1. it is given a cost or fit funciont f: $\mathbb{R}^n \rightarrow \mathbb{R}$ that must be maximized;
2. initialize the variable x with a random position in the search space;
3. until a stopping criteria is met, repeat the following steps:

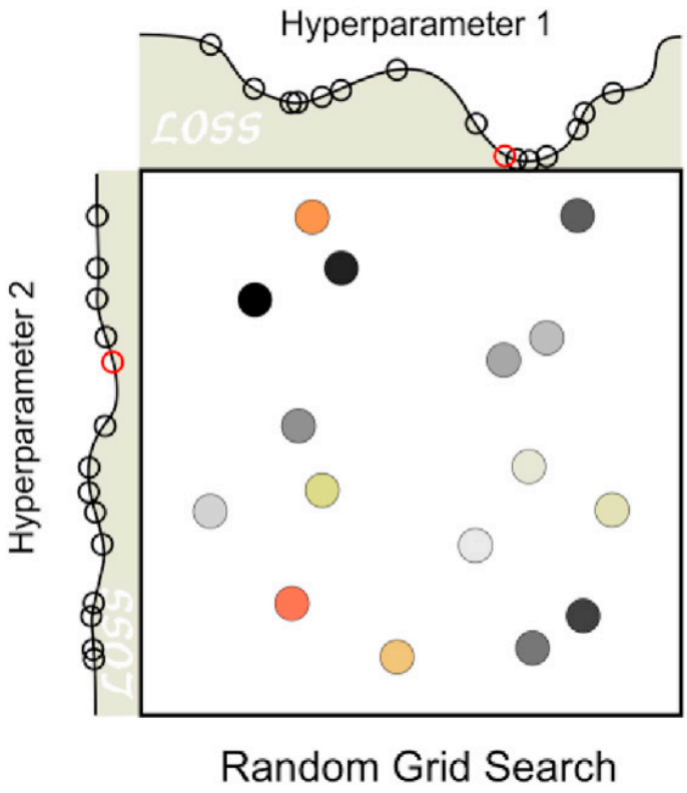

FIGURE 3.3: Representation of Random Search hyperparameters search scheme

(a) sample a new position y in the hypersphere of a given radius surrounding the position x;

(b) if the $f(y) > f(x)$ then $x = y$.

A variety of Randomized Search variants are available in literature with more or less structured search:

- Fixed Step Random Search (FSSRS): it is the basic algorithm proposed by Rastrigin [92];
- Friedman-Savage procedure[41]: it searches each parameter sequentially with a set of guesses having a space pattern among themselves and the search space boundaries;
- Optimum Step Size Random Search (OSSRS): theorised by Schumer and Steiglitz [100], it is an evolution of the FSSRS, where the sampling hypersphere radius is optimally adjusted, speeding convergence to the optimum: unfortunately, because the need to repeatedly sampling in order to approximate the

optimal radius make it too expensive by a computational point of view to be implemented.

- Adaptive Step Size Random Search (ASSRS): it is an evolution of the previous algorithm, theorised by the same authors, where hypersphere radius is heuristically adapted generating two candidates at each step, one with the current step size and the second with a larger step size. The latter one is taken as new nominal step size only if it leads to improvements, otherwise the nominal step size is reduced if after several iterations no improvement is achieved.

- Optimized Relative Step Size Random Search (ORSSRS)[99]: conceived by Schrack and Choit, it approximate the optimal step size through an exponential decrease

Since, in general, there may be no information known a priori that would make one random search algorithm preferred towards another, in the present study it was decided to implement a pure random search optimization algorithm, similar to FSSRS proposed by Rastrigin, coupled with Cross Validation, as proposed by Bergstra and Bengio [18]. Like Rastrigin method, this approach use random sampling as core principle to explore the search space, neither relies on gradient information, making it suitable for non-differentiable or discontinuous functions, and it is designed to global optimum search than focusing on local minima. Despite these similarities, the above mentioned approach distinguish itself from the Rastrigin's work for the following features:

- theoretical framework: Bergstra and Bengio in their Randomized Search CV specifically targeted machine learning hyperparameters optimization, while Rastrigin's work was primarily focused on general function optimization.

- hyperparameters importance: this concept introduced by Bergstra and Bengio dimonstrated that only few hyperparameters significantly affect machine learning model performances, reducing the computational effort that would required randomly exploring the search space for each hyperparameter.

- parallelization: taking into account modern advancements both in hardware and software, Bergstra and Bengio introduced parallelization in order to optimize the execution time;

- cross-validation: in order to increase robustness, avoiding for example overfitting issues, each hyperparameter combination is tested across multiple data-splits.

All these characteristics make it an excellent candidate for including this approach as a further complement to the optimiser, although it is not in itself sufficient to complete the spectrum of possible algorithms, as more targeted ones may be required in certain situations to further reduce the computational effort required.

If one compares figure 3.2 with figure 3.3, from literature [82] it can be seen how a random search approach can lead to better results than a grid search. It is worth mentioning, that this may be true for search spaces characterised by high noise, while it may not be so evident otherwise.

3.3.4.3 Bayesian optimization

Generally attributed to Jonas Mockus, Bayesian optimization is a global optimization method for black-box functions based on sequential design strategy [76], usually implemented to expensive-to-evaluate functions, like for example noisy ones. In detail, the Bayesian optimization is used for problems concerning the maximization or minimization of a function a "black-box" (i.e. having an unknown structure) continuos function f, that mathematically takes the form $max_{x \in A}\ f(x)$ or $min_{x \in A}\ f(x)$, where A is the set of points x. As partly mentioned above, this type of algorithm is particularly advantegeous where in general $f(x)$ is hard to evaluate because of the computational costs: it is important to highlight that only $f(x)$ is considered in its evaluation and not its derivatives [40]. In order to deal with the black-box structure of the $f(x)$ function, the first step of the algorithm is to assign a prior probability distribution at it (also known as "prior"), in order to evaluate its behaviour that is usefull to form a posterior distribution from prior values update over the objective function: in turn, the posterior distribution is used to build up the acquisition function, (infill sampling criterion) aimed to determine the next query point. Among

the methods to assign prior and posterior distribution over the objective function, the most common ones are kriging and Parzen-Tree Estimator [19], which respectively rely on one Gaussian distribution process or two separate distributions for maximums and minimums.

One of the main limitations, not only of this optimisation algorithm, but also of the others seen above, lies in the handling of problems characterised by high dimensionality, that leads to an exponential increase of the iterations needed before a solution is met: a commonly accepted rule of thumb is to consider a problem high-dimensional if the number of dimensions is over 20 ($d > 20$)[101]. Recently, an approach has been introduced to overcome this drawback by subdividing the high-dimensional space into subspaces with smaller dimensions[78]: in itself, this would not represent a novelty, as this is at least theoretically also possible for other optimisation algorithms, such as grid-search, random-search and so on. The real novelty lies in the adaptive capabilities of the Bayesian optimisation algorithm: Bayesian optimization's ability to build and update a model of the objective function allows it to focus on promising subspaces dynamically, making it much more suitable for dealing with high-dimensional optimization challenges, than it could be possible with a static, by-hand approach that could be pursued in the previous mentioned approaches.

Taking some step more and specializing the topic for the hyperparameters tuning of machine learning models, as far as for the previous mentioned optimization algorithm, also in this case some differences are here among theory and their applications. In detail, in the context of the machine learning hyperparameters optimisation, the reference algorithm for Bayesian Optimisation is the so-called Sequential Model-Based Optimization (SMBO)[51, 50, 81]. Considering an application characterized by a costly to evaluate fitness function $f{:}X{\to}\mathbb{R}$, essentially it approximate $f(x)$ with a cheaper to evaluate surrogate function: in general, the SMBO algorithm inner loop optimize the surrogate function or its transformation. The point x^* that maximize it is the new proposal for the evaluation of the real function f. Regarding the optimization criteria, the following are the main one:

- Expected Improvement (EI);

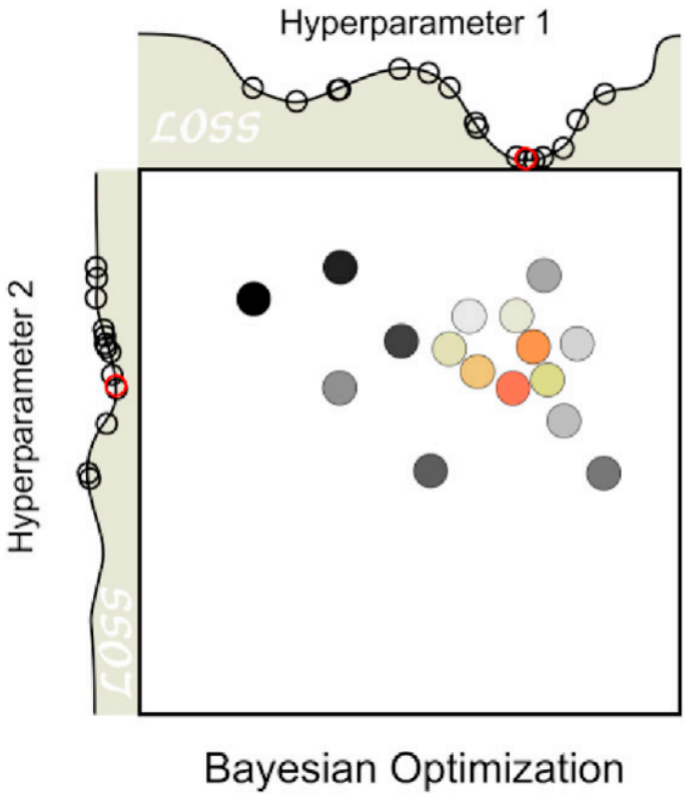

FIGURE 3.4: Representation of Bayesian Optimization hyperparameters search scheme

- Probability of Improvement(PI);
- Upper Confidence Bound (UCB);
- minimization of the Conditional Entropy;
- bandit-based criterion;

As can be seen from the figure 3.4 [82], the prerogative of the Bayesian optimisation algorithm compared to the previous two is to concentrate the calculations where it is most likely to obtain the minimisation (or maximisation, as the case may be) of the optimisation function, thus considerably reducing the amount and intensity of the computational effort, while at the same time obtaining significantly better results.

Because its flexibility that make it work well in many different settings and its intuitive process, in the present work Expected Improvement optimization criteria for the Bayesian Optimization algorithm of machine learning model hyperparameters tuning is implemented.

3.3.5 Optimized model selection and training

In addition to hyperparameters optimization, to ensure an unbiased evaluation of model performance and generalization ability, k-fold cross-validation in the model training is implemented: this resampling technique partitions the training set and the test set into k equal-sized folds, where one test fold is used for validation, and the remaining k-1 training folds are used for training. Cross-validation is then repeated k times, with each fold serving as the validation set once: this process mitigates overfitting by providing a more reliable estimate of a model's performance on unseen data. By training and evaluating on different subsets, it accounts for potential idiosyncrasies in the training data, ensuring a robust assessment of generalization ability. In these experiments, k = 10 is adopted, as it is a common value that balances computational efficiency and reliable performance estimation. The dataset is therefore randomly partitioned into 10 folds, and the models are trained and evaluated 10 times, with each test fold serving as the validation set once. Then, performance metrics are averaged across all folds, providing an unbiased estimate of the models' generalization capability. The cross-validation procedure is implemented using the scikit-learn library in Python, ensuring a consistent and reproducible approach. This rigorous evaluation strategy allows to mitigate overfitting risks and obtain a realistic assessment of the proposed models' performance, strengthening the validity and generalizability of the findings.

The selection of the Machine Learning model is performed autonomously by the algorithm on the basis of the assigned metric. In particular, the metric adopted for the choice of the best model for the cooperative learning phase is $R^2_{adjusted}$, while for transfer learning phase MAPE was the chosen one. Regarding, cooperative learning, this choice was made because the interest is to train a model able to reliably investigate if two or more buildings have similar energy needs "dynamics", establishing a robust degree of correlation between input and output variables. This is useful as the model will also be able to effectively predict events not contained in the source dataset. In more detail, therefore, this property will also be reflected in the ability to predict the energy demands of the target building. Instead, the transfer learning model is mainly a prediction tool capable of minimizing forecasts errors, because

correlation problems are considered as already managed in the previous steps.

3.4 Results acceptance

Once the machine learning model has been chosen, optimized and trained, it is executed. The results obtained based on, it is worth remembering, the data of the similar building are compared with those of the target building. At this point, if the accuracy metrics acceptability threshold are met, the results are considered acceptable. Then, the selected, optimized and trained models are used from then on to forecast the building's energy needs. Conversely, if the results are not acceptable based on the set threshold, the algorithm then proceeds to select another similar building, on which to restart the previously exposed algorithm. It is worth mentioning that this circumstance, of the result unacceptability, can occur both during the first execution cycle and during the subsequent ones. In fact, the algorithm presented is executed cyclically and, at each execution, such verification is carried out to ensure the goodness of the results, changing or not the similar building and, consequently, all that follows, in order to ensure the best possible performance based on the chosen accuracy metric .

Chapter 4

Case study

This section provides a description of the application of the developed method and the related case study. In order to prove the effectiveness of the proposed methodology and, at the same time, test its limits. Choosing the case study, it is felt that it should focus on buildings characterised by multiple energy needs, randomness of use of the utilities and modelling complexity.

At the same time, in order to ensure the replicability of what was studied, it is considered that these characteristics should be attainable in standard buildings, therefore comparable with each other and replicable.

In this sense, hospital structures present all these characteristics put together: in fact, they are often characterized by multiple simultaneous energy needs (typically, heat, cold, electricity and, in some cases, also steam), whose profiles can take on in some cases difficult to predict, due to the urgency that access to the hospital structure may have, and by the complex modeling, by virtue of the different comfort, healthiness and instrumental requirements that characterize the various rooms.

4.1 Target building

The Sant'Anna Hospital of Cona, located near the city of Ferrara in the Emilia-Romagna region of northern Italy in figure 4.1, serves as the target building for this work: with approximately 900 beds, the hospital handles over 27,000 hospitalizations annually. [44]. For the climatic area, it is classified as "E" with 2'313 heating degree days referred to standard conditions set by Italian law DPR 412/93.

The hospital buildings of the site (4.1) are supplied by heating and cooling districts. These are fed by an integrated energy system (4.2), whose items are listed

FIGURE 4.1: Sant'Anna Hospital of Cona (Ferrara, Italy)

below and that takes place in the power center adjacent to the hospital, composed as follows:

- Four identical natural gas-fed boilers (HBW);
- Three identical natural gas-fed steam generators (SB);
- A cogeneration engine (CHP), for contemporary production of electricity and heat at different temperature levels[1];
- An absorption chiller (AC), fed by a portion of the heat recovered from the cogeneration engine;
- Four identical electricity-fed chillers (ECs);

The energy center is grid-connected, so that the surplus electricity from the CHP can be injected into the power grid, while electricity is purchased when the demand (including the power to the electric chillers) exceeds the production.

The monitoring network and its equipment is designed and installed to collect all the information needed to run the plant efficiently, gaining information and insights about the status of the technical equipment and the energy needs of the buildings. In detail, the following energy variables are monitored:

[1] in detail, heat is recovered from intercooler, oil circuit, cooling water and flue gases: these are used to heat the water flow sent to the hospital's heating circuit or the absorption chiller

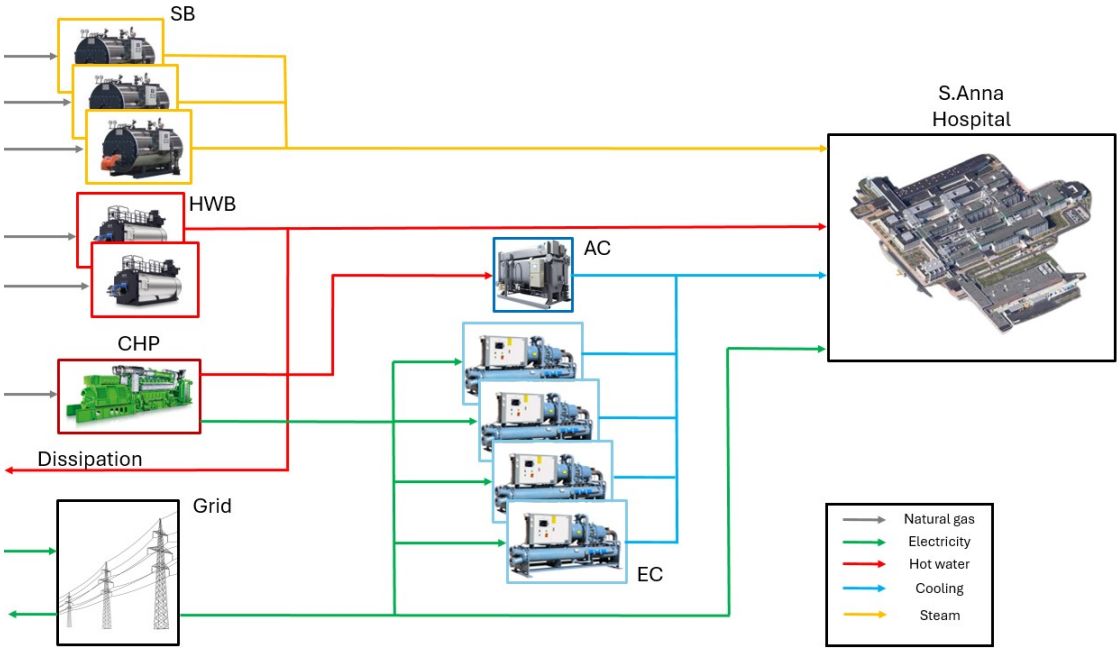

FIGURE 4.2: Sant'Anna Hospital of Cona (Ferrara, Italy)

- natural gas consumption;
- boilers heat production;
- CHP heat production;
- CHP heat dissipation;
- CHP electricity production;
- ABS heat consumption;
- ABS cooling production;
- electric chillers cooling production;
- hospital heat demand;
- hospital cooling demand;
- electricity withdrawn from the grid;
- electricity injected into the grid.

In detail, the Building Management System (BMS) operates via a Bacnet/IP communication protocol, which is typical for the management of modern complex building/plant assemblies. The system architecture is standard and, according to a bottom-up representation as in figure 4.3 presents:

- field components: energy meters (EMs), sensors, actuators, etc., i.e. everything that sends measurement signals or receives and executes regulation commands;
- PLCs (programmable logic controllers): these are the components that collect data and signals and give commands to the individual components in the field, acting on them according to their technical constraints (load up or down ramps, opening or closing commands, e.g. valves, etc.);
- Controllers/accentrators: they receive signals from several PLCs and are responsible for giving commands that are consistent with the management logic of the system, so as to prevent different parts from conflicting or the system from operating under sub-optimal conditions;
- server/workstation: is responsible for collecting all signals from the various controllers and centralisers. Inside it is contained the overall management logic of the system.

Between the field components and the PLCs, there are also interface modules, which are in charge of converting the protocol of the signals coming from the field components (typically ModBUS) to BACnet/IP; the use of the former is favoured above all for the control of the equipment and machines that make up the system by virtue of its greater speed and robustness. The ModBUS typically presents a direct connection between field components and PLC (that is, "many wires" as components to connect), while the BACnet presents a single "wire" that enters and exits in each individual field component.

If on one hand this reduces installation costs and simplifies the system architecture and the addition of new components, it follows that the system is intrinsically slower (having to call each individual component) and intrinsically less resilient (the

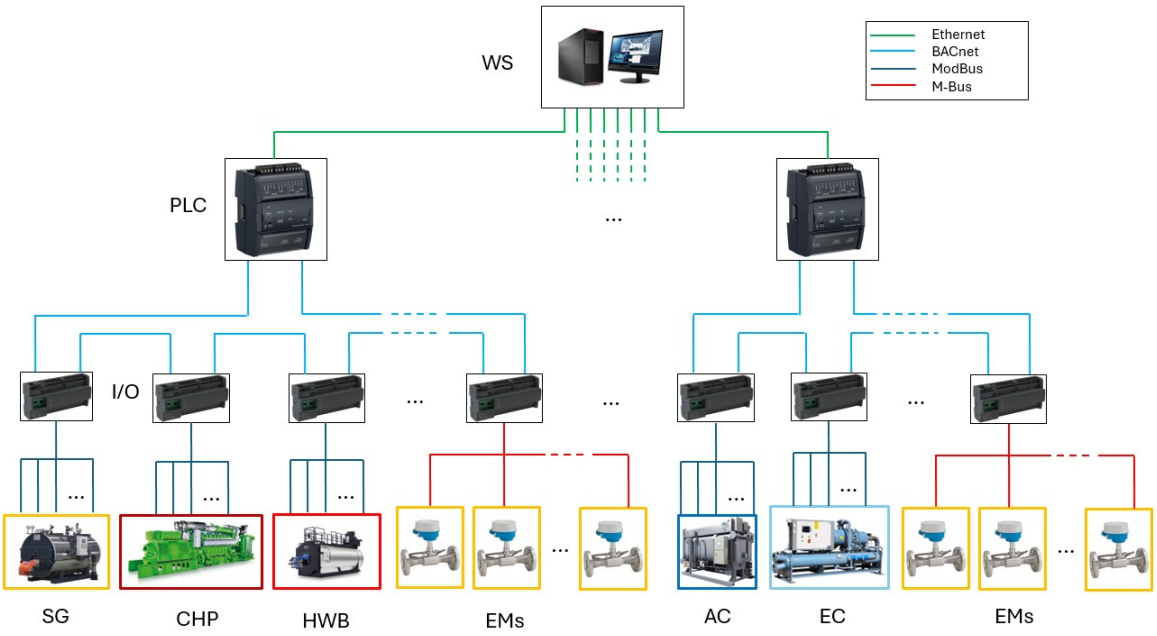

FIGURE 4.3: BMS scheme

interruption of the communication cable, at any point, would determine the impossibility of communicating with all the components connected to it).

The data collected from the server, finally, are stored in a storage system in csv format, in order to be able to query the collected data for further analysis as needed. In the case of the site S.Anna hospital, the data collected by the server are also addressed to an on-top optimizer for the system through csv or json files, which the former is in charge of periodically optimizing the operation of the site based on the objective functions set [44]. In addition to the above, the files are gathered and sent to the industrial partner's on-cloud data lake, which achieves two important objectives: making the system more resilient through cloud backups and enabling comparative site analysis through BI tools, for example. Specifically, regarding this last point, collecting data in the data lake from different sites managed by the industrial partner allows comparing different buildings energy needs and identifying any similarities between them or further analyzing their energy behaviour through benchmarks, KPIs or demand profile analysis.

Regarding the real-time consumption and energy flows through the main distribution hospital circuits, two meters are installed in order to monitor the real-time

Paramater	Boiler	CHP plant	Absorption chiller	Electric chiller	Steam generators
Number of units	4	1	1	4	3
Input	Natural gas	Natural gas	Heat	Electricity	Natural gas
Output	Heat	Electricity | Heat	Cold	Cold	Steam
Nominal output [-]	2.6	1 | 0.93	0.52	1.6	1.3
Nominal efficiency [-]	0.92	0.447 | 0.422	0.77	2	0.87
Supply temperature [°C]	93	93	6	6	—
Return temperature [°C]	73	73	12	12	—
Supply pressure [bar]	-	-	-	-	3

TABLE 4.1: Technical parameters of the conversion units in the Cona's hospital power station. The nominal output power is normalized with respect to the CHP electrical output for confidentiality reasons.

data. The new sensors installed in the site to monitor its heating, cooling and electrical load are listed in Tables 4.2. It is worth mentioning that the steam generators, placed in a dedicated area, have little interaction with the rest of the infrastructure. Indeed, they provide heat for services that cannot be delayed and, therefore, are not available for other tasks. Dedicated sensors and actuators are not installed. The natural gas supplied to the steam generators is counted in the total natural gas consumption, but the management of these units is standalone. Real-time local data are transferred through the standard industrial protocol Modbus TCP, considering the technical specifications of the BMS software.

4.2 Reference building

Following the matching procedure described in the previous section, a reference building was identified within the building stock that exhibits the closest similarity to the target building. This section presents a detailed analysis of the reference building's physical and construction characteristics, highlighting the key features that led to its selection. The comprehensive examination encompasses geometric parameters, envelope properties, and technical systems, providing a foundation for understanding the subsequent energy performance comparison.

4.2.1 Physical and construction characteristics of the reference building

According to the criteria presented in chapter 3, the facility identified as "similar" is the Ca' Foncello Hospital in Treviso in figure 4.4, with its 862 beds, an annual number of hospitalizations of 37,000, similar categories of energy needs, and a total

Energy flow	Monitored variables	Sensors installed
Heating	Supply/return temperature and volumetric flow rate	Ultrasonic flow meter (E+H Proline Prosonic Flow 91W), ×2 Pt100 thermometer (E+H RTD omnigrad TST310)
Cooling	Supply/return temperature and volumetric flow rate	Ultrasonic flow meter (E+H Proline Prosonic Flow 91W), ×2 Pt100 thermometer (E+H RTD omnigrad TST310)
Electrical energy sold to and purchased from the grid	Voltage, current, power factor	Enel distribuzione Actaris High Precision Multifunction Electronic meter (Telematica Sistemi MT860-MID)

TABLE 4.2: Sensors installed for monitoring the energy demand of Cona Hospital.

FIGURE 4.4: Ca'Foncello Hospital in Treviso

area of 167,000 square meters [110]. With regard to the climatic area, it is classified as "E" with 2'378 heating degree days referred to standard conditions set by Italian law DPR 412/93.

The transformation of energy vectors takes place inside a power center adjacent to the hospital, which houses a trigeneration plant with an architecture similar to that shown in figure 4.2 for the Cona hospital, dedicated to the production of hot and chilled water, steam and electricity. Also the plant characteristics are similar to the target building, as it can be observed in table 4.3.

The distribution of energy vectors also presents characteristics similar to those observed for the target building. In particular, as regards the thermal vector, this is distributed to the hospital building through a "ring" type network, from which the branches are created that serve the individual distribution columns on the floors of the building. The monitoring of the thermal requirement is carried out through a dedicated meter corresponding to the building distribution ring.

In the case of the cooling energy vector, also for the reference building there are several distribution lines, three in particular, each one has a dedicated energy meter. As regards the electrical energy demands, there are bidirectional meters both on the main electrical cabin and the emergency cabin, metering both absorbed from and

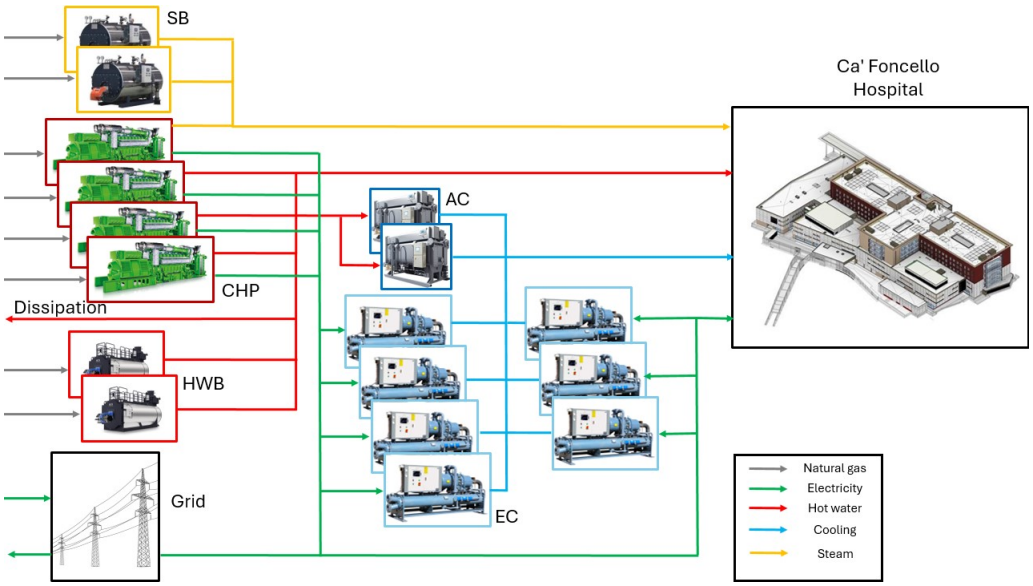

FIGURE 4.5: Ca'Foncello Hospital in Treviso

sold to the grid electricity, and on the individual self-production systems, i.e. the CHP engines.

Similarly to the target building, the reference building is equipped with a data collection and storage system within the BMS, which, on the basis of the available information, is in charge of the automatic and efficient management of the plant, while providing useful information about the operating conditions of the plant and the utilities on the building side. Among the monitored quantities are listed:

- natural gas consumption;
- boilers heat production;
- CHP heat production;
- CHP heat dissipation;
- CHP electricity production;
- ABS heat consumption;
- ABS cooling production;
- electric chillers cooling production;
- hospital heat demand;

Paramater	Boiler	CHP plant	Absorption chiller	Electric chiller	Steam generators
Number of units	2	1	3	7	2
Input	Natural gas	Natural gas	Heat	Electricity	Natural gas
Output	Heat	Electricity | Heat | Steam	Cold	Cold	Steam
Nominal output [-]	2	1.13 | 0.93	0.51	2.67	1.5
Nominal efficiency [-]	0.92	0.460 | 0.408	0.7	5.7	0.90
Supply temperature [°C]	90	90	7	7	—
Return temperature [°C]	70	70	12	12	—
Supply pressure [bar]	-	-	-	-	12 | 6

TABLE 4.3: Technical parameters of the conversion units in the Ca Foncello's hospital thermal power station. The nominal output power is normalized with respect to the CHP electrical output for confidentiality reasons.

- hospital cooling demand;
- electricity withdrawn from the grid;
- electricity injected into the grid.

Both hospitals require heating, cooling, and electricity, as well as high-temperature energy for steam production, which is utilized for various special utilities such as the sterilization department and laundry. The steam demand is excluded from the current thesis because no real-time data are available.

4.3 Comparative analysis

Upon the physical characteristics analysis, this section presents a detailed comparison of the energy consumption patterns between the target and reference buildings. The comparative analysis focuses on temporal energy demand profiles, examining both daily and seasonal variations. Through this investigation, the degree of similarity in terms of energy behaviour is evaluated, validating the effectiveness of the matching methodology and establishing the basis for energy performance prediction. Multiple metrics and visualization techniques are employed to quantify and illustrate the alignment between the two buildings' energy profiles.

The yearly needs for these energy sources in the year 2023, normalized with respect to electricity for confidentiality purposes, are presented in Table 4.5. Moreover, the table reports the ratio among Ca'Foncello and Cona energy needs calculated accordingly to the similitude approach shown in Section 3, ensuring the "static" similarity between the buildings.

Energy flow	Monitored variables	Sensors installed
Heating	Supply/return temperature and volumetric flow rate	Ultrasonic flow meter (E+H Promag 50P50/1H), ×2 Pt100 thermometer (E+H RTD TST90)
Cooling	Supply/return temperature and volumetric flow rate	Ultrasonic flow meter (E+H Promag 50P1Z), ×2 Pt100 thermometer (E+H RTD TST90)
Electrical energy sold to and purchased from the grid	Voltage, current, power factor	Enel distribuzione

TABLE 4.4: Sensors installed for monitoring the energy demand in Ca'Foncello hospital.

	Heat	Cold	Electricity	Steam
Cona Hospital	0.761	0.226	1	0.176
Ca' Foncello Hospital	0.863	0.873	1	0.353
Ca'Foncello (similitude scaling)/Cona	0.966	1.01	1	0.999

TABLE 4.5: Yearly energy demands of the Ca'Foncello hospital (for the year 2023) normalized to Cona hospital for each energy need.

4.3.1 Energy needs profile characterization

This section provides an analysis of the energy needs profiles of the target building and the similar one. The aim is to demonstrate the dynamics and relationships between the various energy needs, thus highlighting both the similarities and differences between the two buildings. This will also allow for a better understanding of the potential strengths and weaknesses of the models presented in the following chapters and sections, also in light of its actual predictive capabilities.

4.3.1.1 Yearly energy needs

The annual analysis of energy demand profiles (4.6) for heating, cooling, and electricity highlights the seasonal pattern for each of these energy needs. Specifically, heating demand exhibits a peak in the winter months and a minimum in the summer ones. An interesting detail for both buildings is the increase in heating needs near the end of August and the beginning of September, coinciding with a decrease in the facility's cooling needs. This is essentially consistent with the use of the facilities, where the provision of services related to environmental comfort is not strictly limited by law constraints.

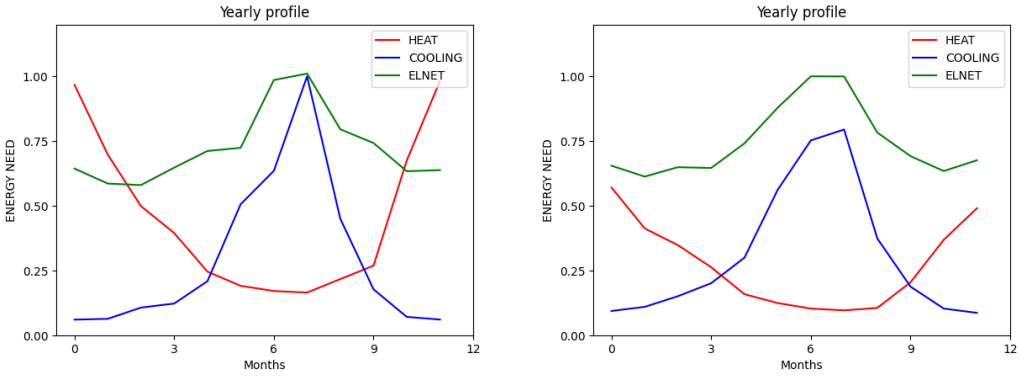

FIGURE 4.6: Yearly normalized energy needs of Cona (left) and Ca' Foncello (right) hospitals

Finally, an anomalous element is undoubtedly represented by the electricity need, which in both buildings mirrors the cooling energy need trend: this is likely due to the installation of small air conditioning devices (portable and similar ones) probably located inside the building and therefore difficult to separate from the rest of the consumption. This hypothesis is further confirmed by the fact that in the winter

months, there is a similar increase in electricity consumption, likely linked to devices such as heaters or heat pumps (note, in particular, how heating energy need decreases between January and February and increases between November and December).

4.3.1.2 Winter energy needs

During the winter season, there are no significant differences between the different days of the week in terms of energy needs, as it can be observed in figure 4.7. Regarding heating needs, both facilities adopt "night reduction" regimes: in particular, in the case of the Cona hospital, the most significant reductions are conducted between the early hours of the morning and 6:00 a.m., then restarting in the early afternoon and, finally, shortly before midnight.

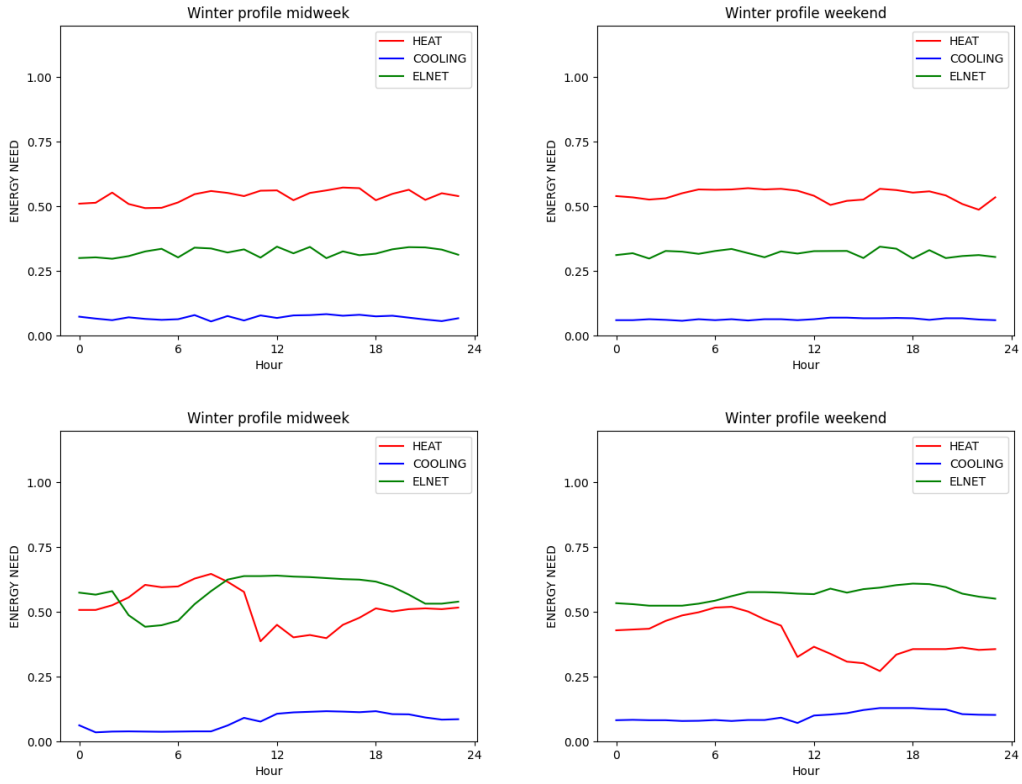

FIGURE 4.7: Winter daily normalized energy needs of Cona (upper) and Ca' Foncello (down) hospitals during weekdays (left) and weekends (right)

The main difference between weekdays and weekends can be observed in the less jagged profiles on weekends, presumably also due to the typically lower access

to healthcare facilities and the building's thermal inertia. In the case of the Ca' Foncello hospital, however, the "reduction" regimes are fairly uniform between weekdays and weekends, with the latter being concentrated in two bands instead of three (as in the previous case), specifically between midnight and 6:00 a.m. and between 11:00 a.m. and 4:00 p.m. Cooling demand, as one might expect, is virtually minimal during the winter season, also due to the fact that both buildings are located in a relatively cold climate zone (E). No notable variations are observed between weekdays and weekends. Finally, regarding electricity demand, it can be observed that the trend is also fairly constant across the different hours of the day for both facilities. Only in the case of Ca'Foncello hospital during the week a decrease was observed in the late evening and early morning hours, likely due to a specific lower number of patients accessing the hospital and, therefore, to be considered an isolated incident.

4.3.1.3 Summer energy needs

For summer energy needs, from figure 4.8, heating and cooling ones show fairly similar trends between the two buildings.

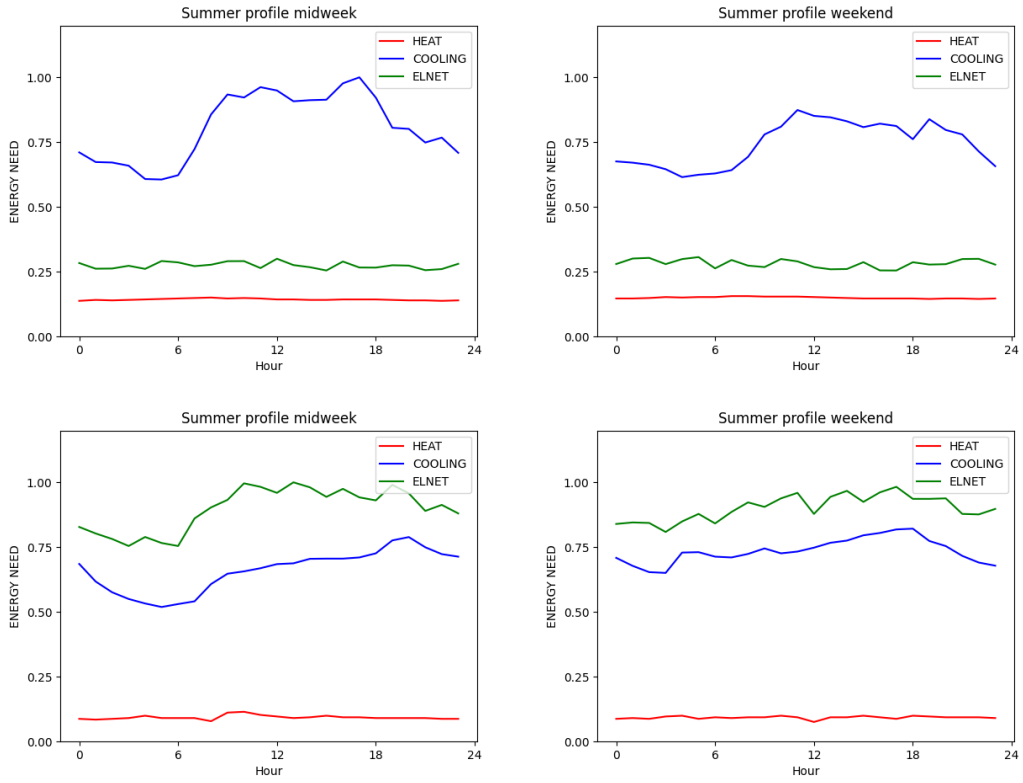

FIGURE 4.8: Summer daily normalized energy needs of Cona (upper) and Ca' Foncello (down) hospitals during weekdays (left) and weekends (right)

Specifically, thermal needs are essentially constant throughout the day, aligned with the baseload, while cooling energy need increase during the early morning hours and then decrease towards evening. What can be observed is a much steeper ramp in the case of the Cona hospital compared to the Ca'Foncello hospital, and a shift in maximum and minimum temperatures: the Cona building, in fact, has a more concentrated cooling need during the central hours of the day, while the Ca'Foncello hospital is more distributed. The real difference, however, lies in the trend in electricity needs, which can be observed to be fairly constant in the case of the Cona hospital, while the Ca'Foncello hospital during weekdays shows an increasing trend in the morning, then a slight decrease during the rest of the day; this dynamic is weaker and dilated during the weekends.

4.3.1.4 Mid-season energy needs

Moving to the mid seasons, specifically from April 16th to June 20th and from September 21st to October 14th, from figure 4.9 can be observed that all energy needs are midway between the winter and summer seasons, in terms of intensity, while the dynamics remain very similar to those of summer needs, with very low and constant heating needs throughout the day and cooling and electricity needs on the days increasing in the early hours of the day and decreasing from the late afternoon/evening.

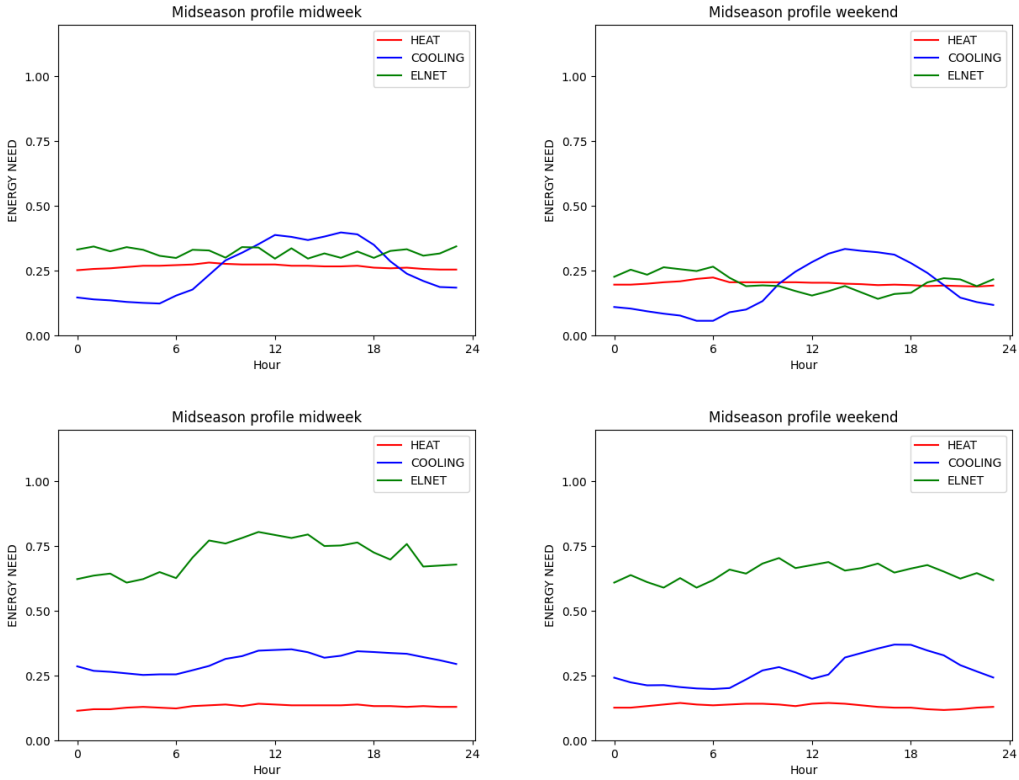

FIGURE 4.9: Mid-season daily normalized energy needs of Cona (upper) and Ca' Foncello (down) hospitals during weekdays (left) and weekends (right)

4.4 Simulations setup

In this section, information regarding the setup of the simulations is provided. In more detail, with reference to the steps outlined in chapter 3, in order to make the study presented replicable, the parameters useful for the setup of the simulations are given. Since in this case the similar building energy demands are available, the simulation setup will refer directly to it regarding the algorithm steps. Regarding the case when energy demands data are not available, this is presented in the case study faced in [44].

4.4.1 Dataset

For both hospital facilities, post-Covid yearly data are considered. Because of the significant impact on energy consumption due to the high number of hospitalizations during that period, this allows to test the algorithm's performance in the presence of unforeseen events such as the Covid pandemic. The energy demands are hourly collected thorugh meters installed in the substation facilities and gathered on the Building Management System (BMS). Despite higher frequencies and data granularity are available, it is decided to focus on hourly data in order to make this work comparable with others in literature, which standard datasets are usually hourly-based. Both hospitals exhibit a similar distribution of energy demands, particularly in terms of heating, while cooling and electrical needs appear at first glance to be slightly higher in the structure of the Ca' Foncello hospital. However, what should draw attention is how both are roughly balanced in their respective structures, indicating an incidence of surfaces (and therefore volumes) with intensive care unit, high-dependency unit and surgery rooms that is roughly equivalent between the two hospitals.

To this end, the first step involves selecting input variables that possess a significant degree of correlation with output variables. The overall available variables have been reported in previous sections 4.1 and 4.2. In more detail, a starting selection of the variables involved only those referring to the needs of the structures under investigation: for the heating and cooling needs, reference was made to the quantities

measured at sub-central level, so as to take the exclusion of all those quantities referring to production parameters. It is a different matter, however, with regard to the electrical demand: since there are plants equipped with trigeneration, this is given by the sum of the electrical production of the cogenerators and the electrical absorption from the grid, excluding the electrical absorption of the chiller units and electricity sold to the grid.

As regards steam, however, the prediction of its needs is not the subject of this work, as its use is not monitored in real time and is mostly limited to the laundry, kitchen and sterilization services of the hospital facilities mentioned above.

In addition to the variables related to the energy demands of the facility, which thus represent the output variables, i.e. the object of forecasting, there is also the choice of input variables, i.e. the variables used to predict the energy requirements by means of the algorithm and then the chosen model.

Among these, both those obtainable from the decomposition of the information content of the timestamp are considered, i.e. the days, weekdays, months, years and hours, and the information coming from weather services, such as external temperature and relative humidity, as well as those relating to the structure hospital and its dimensions, i.e. volumes, surfaces and number of beds.

4.4.2 Preprocessing

The first step involves setting the optimal thresholds for data cleaning, with the aim of having sufficiently clean data for the simulations. For the ascent and descent ramps of the needs, the values set in Table 4.6 are adopted, based on the plants serving each of the forecasted needs, as seen in section 3.3.1.

	Heat	**Cold**	**Electricity**
Ramp up threshold [(%\min)]	8.5	8.5	8.5
Ramp down threshold [(%\min)]	2	2	2

TABLE 4.6: Data cleaning pre-processing thresholds.

The latter constraints are set equal to the former to reduce the outliers impact, as they are more restrictive. The values are set according to technical specifications of the power centre items and the regulations allowed by the BMS, looking significantly steeper than what is asserted in the literature. This approach is taken to

avoid excessively cleaning the data, which could eliminate valid ones, and to ensure that the algorithm is trained on situations that are not always perfectly "linear". Instead, it is designed to recognize and account for potentially anomalous situations or those representative of the actual management of the plant and its real limits. After the data are cleaned, they are then normalized, splitted and shuffled. In detail, normalization is carried out for each variable accordingly to its minimum and maximum value, while splitting is performed considering as traning set the known building data and as test set the unknown ones. Regarding shuffling, in the cooperative learning phase it is performed randomly inside each batch of data, mixing training data with test ones, while it is avoided among training and validation one in transfer learning.

In more detail, regarding data splitting and shuffling, as mentioned in Chapter 3, this is conducted differently depending on whether the process is in the cooperative learning phase (i.e., verifying and validating the similarity between two buildings) or the transfer learning phase (i.e., forecasting energy needs). In more detail, the starting dataset is split into a training set, a test set, and a validation set. The training set consists exclusively of data from the similar building, while the test set and validation set consist of data from the target building (in an 80/20 ratio, respectively).

At this point, in the cooperative learning phase, the data from the training set and the test set are mixed together and then shuffled. This allows the model to identify any pattern representative of both structures in the next model training and hyperparameter optimization phase. Conversely, in the transfer learning phase the training set consists only of the data from the similar building: the model is trained and its hyperparameters are optimized to then verify its predictive capabilities on the validation set consisting only of the data from the target building.

4.4.3 Hyperparameters optimization search space

Once the final dataset is ready, it is used to complete the phase of model selection and hyperparameters tuning. In particular, for the optimization algorithms of the demand forecasting models, these are compared to the same number of simulations to have a representative measure of their actual performances. The R^2 metric is chosen for the optimization of the hyperparameters of each of the models, as the aim

is to identify an optimal set of parameters capable of maintaining a sufficient level of problem generalization. Based on these considerations, the searches are carried out through the solution space shown in Table 4.7.

	Hyperparameter	**Range**
Extremely Randomized Trees	Estimators	(10, 5000)
	Max depth	(1, 110)
	Min samples split	(2, 10)
	Min samples leaf	(1, 4)
	Max features	(0.1, 0.99)
Random Forest	Estimators	(10, 5000)
	Max depth	(1, 110)
	Min samples split	(2, 10)
	Min samples leaf	(1, 4)
	Max features	(0.1, 0.99)
Support Vector Regressor	Regularization parameter	(0.1, 10)
	Tolerance error	(0.01, 0.1)
	Gamma	(0.01, 0.1)
	Degrees	(2, 10)
Artificial Neural Network	Layers	(1, 10)
	Neurons for each layer	(2, 10'000)
	Activation	Relu
		Logistic
		Identity
		Tanh
	Learning rate	Constant
		Inverse scaling
		Adaptive
	Initial learning rate	(0.001, 0.01)
	Alpha	(0.1, 10)

TABLE 4.7: Hyperparameters search space for different models.

To conclude this section, it's worth mentioning that the hyperparameter optimization phase requires the search space to be equally large for both cooperative learning and transfer learning phases. This is because the model's hyperparameter optimization is influenced solely by the data it is trained on, not by the assigned task. As it will be seen in the next section, the choice of metrics by which the hyperparameters are optimized needs a different approach.

4.4.4 Hyperparameters optimization metrics

Regarding the choice of the hyperparameter optimization algorithm for machine learning models, some thresholds are chosen in order to address correctly the optimization process, both in the cooperative learning and transfer learning phase.

In cooperative learning, as mentioned in Chapter 3, for the model the goal is to be able to determine whether there is a similarity between the target building and the similar building, and to what extent. In this case, it is particularly important that there are no overfitting or underfitting conditions, and therefore that the model is sufficiently capable of generalizing the observed phenomenon. For this reason, hyperparameter optimization must reflect this requirement and this is certainly represented by the $R^2_{adjusted}$, a metric typically used to obtain information on the level of goodness of the model and whether it is sufficiently balanced (i.e., whether its generalization capabilities are satisfactory, i.e., whether it does not fall into underfitting or overfitting).

In detail, a minimum and a maximum value of the $R^2_{adjusted}$ is considered in order to avoid underfitting and overfitting respectively, while maximum iterations and execution time are set in order to avoid any kind of unefficient use of computational resources. The exact values considered for each one of them are reported in Table 4.8.

Cooperative learning	
Metric	**Threshold**
$R^2_{adjusted}$ minimum	0.70
$R^2_{adjusted}$ maximum	0.95
Maximum iterations	100
Time of execution [s]	900
Transfer learning	
MAPE maximum	10.0%
MAPE minimum	2.00%
Maximum iterations	1'000
Time of execution [s]	900

TABLE 4.8: Metrics thresholds.

Among these, the metrics given the most consideration are the $R^2_{adjusted}$ and the

execution time: the former, as is well known, represents the goodness-of-fit index of the model and its related variables to the observed phenomenon, while the latter is useful for evaluating the feasibility of implementation in real-world applications, where there are time constraints for updating the models and their respective results.

On the contrary, in transfer learning, the requirement (at least the most pressing one) is to have accurate forecasts of the target building energy needs, which is why MAPE was chosen as the hyperparameter optimisation metric. This choice was also dictated by the fact that the risk of overfitting is less pronounced at this stage: this is because the most of the "knowledge" has already been learned in the previous phases. In particular, it can be assumed that the preliminary similitude check acts as a guarantee that any overfitting will only affect specific details of the target building, not the fundamental relationships between the variables and the fundamental energy dynamics learned through the similar building. In particular, verification of the similarity consistency (or at least that the criteria are met) allows to affirm that transfer learning is applicable and, by virtue of this, any overfitting would indicate that the model is particularly specialised in reproducing the behaviour of the target building based on that of the similar building, which corresponds to the objective of this phase.

Chapter 5

Results

5.1 Introduction

In this section, the results achieved for the different parts of the presented algorithm are reported. First, the data cleaning output, with the pre and post-cleaning situation, highlights how the data is improved and less affected by outliers. Subsequently, the comparison between the various optimization algorithms applied to the demand forecasting models is presented, showing the optimal hyperparameter sets, the execution times required by each one to reach the most performing solution, and to which value of the optimization metric they correspond. Finally the results of the simulations for each of the energy demands under investigation are reported, with reference to the results of the target structure corresponding to the Cona Hospital.

5.2 Preprocessing

The datacleaning thresholds imposed according to table 4.6 results in a decrease of the quantities of available data. From table 5.1 it is clear the importance of datacleaning itself in real-world application, because the presence of 'dirty' data represent as a fact a not negligible factor. Moreover, the energy demands most affected by this circumstance are precisely those related to thermal and cooling needs. This is because their respective equipments, plant components, and distribution systems are subject to the most frequent and lengthy maintenance operations, than electrical systems, for example.

These inconveniences, due to routine situations on the plants, clearly have a negative impact on the amount of data available for training machine learning models,

	Heat	Cold	Electricity
Raw dataset	31'823	31'823	31'823
Preprocessed dataset	8'538	11'274	22'119
Training/Test data ratio	6:1	12:1	3:1

TABLE 5.1: Number of timestamps before and after the preprocessing phase.

reducing the available amount. In spite of this, as can be seen from Table 5.1, the ratio among training data to test data is still satisfactory and, although lower than in the literature consulted where ratios of 250:1 are also reached, in line with good practice in statistics (where a ratio of 2/3 and 1/3 between training set and test set respectively is recommended). In particular, the presence of gaps or 'NaN' values is the most encountered issue. This takes place because of the interruption of power supply to the meters during maintenance of the affected plant components. In addition to that, since the meters record cumulative values, they generate 'spikes' upon resumption of operation, resulting from the difference between the last value transmitted before the power interruption and the first value upon resumption of operation. For the cooperative learning phase the next step, finally, is the shuffling of data: shuffling training and test datasets is a standard practice in machine learning with significant implications for model training and evaluation. By randomizing the order of data points, shuffling prevents bias, improves convergence in iterative algorithms, and mitigates the effects of non-IID data. In the evaluation phase, shuffling enhances the robustness of performance estimates, defects overfitting, and ensures fair comparison between different models or hyperparameter configurations. Models trained and evaluated on shuffled data tend to exhibit greater accuracy, reliability, and reduced variance in predictions. Once the similitude is validated also at a "dynamic" level, the same operation is repeated in the transfer learning phase, where, however, data shuffling is not carried out, as the aim is to predict the energy needs of the target building based on the knowledge gathered through the data of the similar building.

5.3 Cooperative learning

5.3.1 Variables importance

The first result investigated is the feature importance, that is the score of each input variable accordingly to its importance in improving model performance according to the chosen performance metric, namely their correlation with the output variables. From the results are listed in Table 5.2, it is evident that heating and cooling demands are mainly influenced by the external temperature and the month in which they are supplied to the building.

Feature	**Score**		
	Heat	**Cold**	**Electricity**
External temperature	0.51924	0.64182	0.13274
Month	0.16965	0.20344	0.19385
Relative humidity	0.02506	0.03586	0.04428
Hour	0.01915	0.02317	0.06441
Day	0.01744	0.01633	0.11193
Year	0.04194	0.06486	0.25697
Weekday	0.00647	0.00520	0.06299
Volume	0.06669	0.00336	0.04033
Surface	0.0706	0.00300	0.05095
Number of beds	0.06375	0.00296	0.04155

TABLE 5.2: Feature importance for cooperative learning

The same applies to electricity demand, which may subtend portable or split system type equipment presence. The other variables weights are aligned with each other respect to the three energy needs, assuming greater or lesser importance depending on the forecasted demand. About this, for example, relative humidity shows an importance score between 1.5 and 2 times greater in the cooling and electricity demands forecast than heating one, which could be related to the incidence of latent heat in the dehumidification process. In similar manner, the time at which the electrical demand arises is of high importance, as it can be observed by time, day, year and day of the week feature importance scores. On the other hand, the building characteristics, such as geometric factors, its intended use and the level of service offered (e.g. beds) have a greater influence on the heating and electricity

demands. This fact may be because they are linked to the actual use of the building and its spaces, more than it is for cooling, which is usually supplied to environments characterised by continuous use and constant temperatures during the whole year, i.e. surgery rooms and intensive care units.

5.3.2 Machine learning model selection

Based on the criteria given in the table 4.8, machine learning models suitable for predicting the energy needs of the target building are derived. In more detail, as can be gleaned from the literature review in the chapter 2 and table 5.3, more than one model was found to fit the scope.

Energy need	Machine Learning model	Hyperparameters optimizer	Convergence steps
Heat	Random Forest	Bayesian	2
Cooling	Random Forest	Bayesian	2
Electricity	Extremely Randomized Trees	Bayesian	2

TABLE 5.3: Results of the machine learning model selection for cooperative learning.

5.3.3 Hyperparameters optimization

The hyperparameter optimization process yielded the results listed in Table 5.4. Comparing the hyperparameters, the thermal and cooling energy demands require much less computational effort than the electrical energy demand, although the larger volume of data available for the latter in the training.

5.3.4 Accuracy metrics

With regard to the predictions accuracy, the main metrics are reported in Table 5.5, and the "data predicted-data test set" diagram is provided, making possible to compare the results for each energy demand.

Comparing Table 5.4 with Table 5.5 results, it is observed that electricity needs much more features to be above the acceptable thresholds, compared to heating and

Energy need	Machine Learning model	Hyperparameter	Value
Heat	Random Forest	Estimators	3052
		Max depth	43
		Min samples split	9
		Min samples leaf	2
		Max features	0.72
Cold	Random Forest	Estimators	3052
		Max depth	43
		Min samples split	9
		Min samples leaf	2
		Max features	0.72
Electricity	Extremely Randomized Trees	Estimators	613
		Max depth	387
		Min samples split	9
		Min samples leaf	2
		Max features	0.7

TABLE 5.4: Results of the hyperparameters optimization for cooperative learning.

Energy need	RMAE	RMSE	MAPE [%]	$R^2_{adjusted}$	Execution time [s]
Heat	0.1785	0.2291	9.390	0.9422	1.253
Cold	0.1049	0.1893	9.6787	0.9413	56.89
Electricity	0.2931	0.4700	9.051	0.7481	35.72

TABLE 5.5: Metrics results for cooperative learning.

cooling demands. About this, heating shows fast time calculations and less computational effort, compared to the other ones. Among the all three mentioned, cooling shows off the lowest accuracy and highest execution time.

5.3.5 Energy needs profiles

In this section, the profiles of each of the energy demands are reported. In detail, energy needs profile for the winter, summer and intermediate seasons (understood as autumn or spring), compared with what is predicted by the presented model are available. In general, although a good fit is observed for all three of them, the most relevant information will be highlighted from time to time.

The Figure 5.1 regarding heating demands shows a good prediction and correlation long all over the range, although a slight tendency to overestimate can be

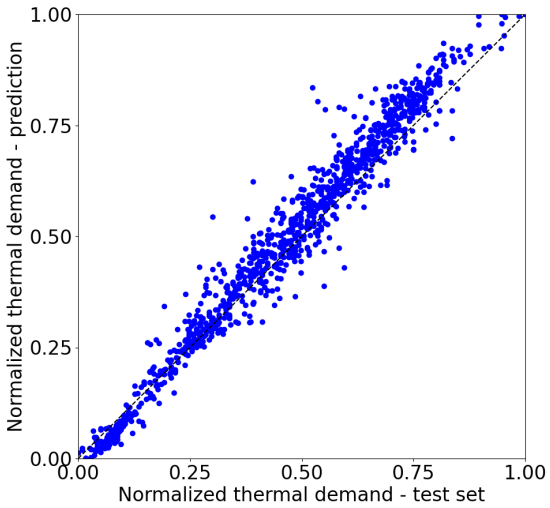

FIGURE 5.1: Predicted vs. test set results for thermal demand.

observed at the highest requirements, roughly from 50% to above. A similar argument can be made for the scattering of data, which is higher at higher thermal loads: this could be attributed to the influence of variables not considered in the modeling, such as solar radiation, wind or crowding of hospital facilities.

In any case, it is important to point out that these "imperfections" are definitely contained and do not significantly affect the goodness of what the metrics highlight and observe in general about the work done so far. Similar observations for the cooling demand, which differs from the thermal demand in that it shows no clear tendency to overestimate or underestimate in any of the requirement ranges, as shown from figure 5.2 showing itself in fact as the most balanced of all three.

In more detail, it can be observed a slightly tendency to understimate the real energy need near to the nominal load. Residuals dispersion is more or less constant in the whole range of loads, with a light increase at high cooling loads.

Separate considerations should be made for the electricity need, which first of all shows a greater dispersion of points than the other two energy demands predicted by the model, as it can be observed in figure 5.3. This can certainly also be attributed to the lower degree of correlation of the model in predicting electricity needs, compared to what is shown, on the other hand, for thermal and refrigeration needs. There is a trend in overestimating the real electrical need in the whole range

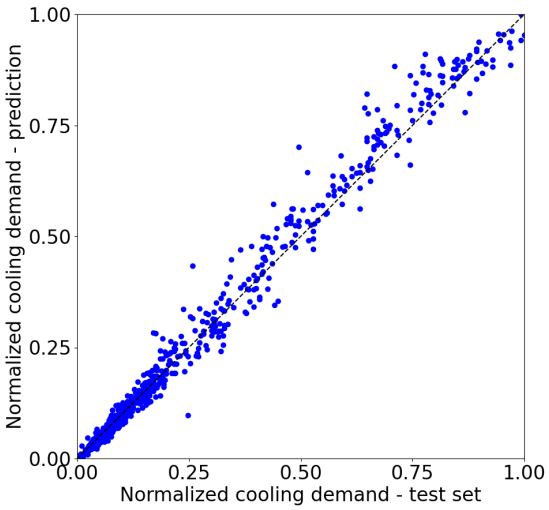

FIGURE 5.2: Predicted vs. test set results for cooling demand.

of loads, with a light improvement near to the nominal loads. However, all these considerations result in a good prediction of load profiles for all three energy needs, as can be observed from the figures 5.4 to 5.12. In detail, for all three of them, the actual profiles are compared with those predicted by the algorithm for three weeks, each one representative of a period of the year: winter, autumn/spring (as intermediate seasons) and summer are considered.

In more detail, comparing heat demand during the winter, summer and intermediate (spring/autumn) seasons, represented by figures 5.4, 5.5 and 5.6 respectively, shows a slight positive offset precisely when thermal needs are higher, i.e. precisely during the winter season.The genesis of this offset can be attributed to the constraints imposed at the stage of data cleaning, of which table 4.6. In fact, comparing the upward and downward profiles of the energy demands of the above figures it shows that the predicted values of the former are more closely matched to the actuals of the latter, in terms of the slope of the line.

Looking in more detail at each season period for each energy demand, as can be seen from figure 5.4, the thermal energy demand profile is reproduced rather faithfully, if one sets aside the tendency to overestimate or underestimate actual values at

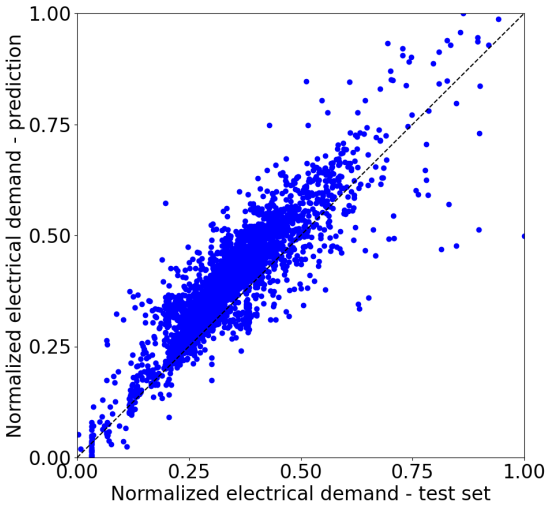

FIGURE 5.3: Predicted vs. test set results for electricity demand.

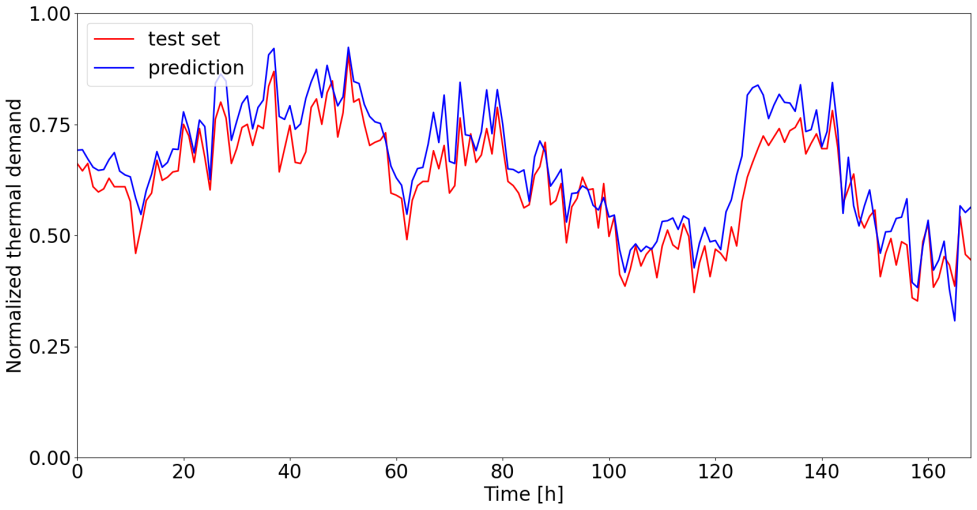

FIGURE 5.4: Thermal energy needs predicted (blue) vs. real (red) – winter week.

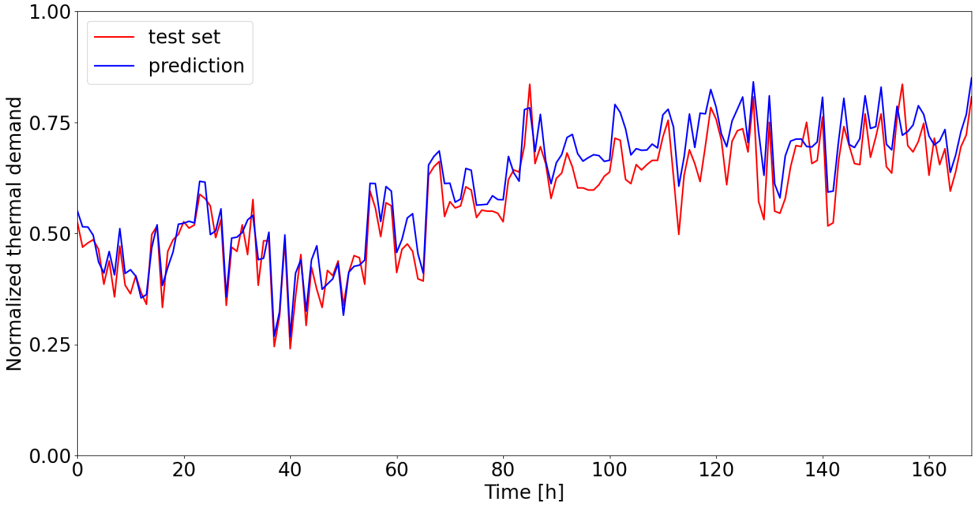

FIGURE 5.5: Thermal energy needs predicted (blue) vs. real (red) – intermediate season week.

certain times of the year. Once more, this is probably linked to the data cleaning criteria set in the preprocessing part. In more detail, the more restrictive constraint imposed on the downward profile of requirements than on the upward profile means that the model is not trained and, therefore, unable to recognize and correctly reproduce precisely those changes that are more abrupt than those contained in the clean dataset.

Different discussion are made when the heating demand is low. In fact, on the other hand it can be observed that some of the peaks are not faithfully reproduced, as it is the case at the higher ones, or in some cases these are completely invented or missing. This circumstance can be attributed, in contrast to the above, to the limitations imposed on the hyperparameters optimization search space: this fact, evidently, prevented the Random Forest model from creating additional decision nodes along each of the trees. This meant that the model is unable to recognize and model additional case histories, which would allow the peaks to be reproduced with adequate precision and accuracy.

As for the profiles of cooling demand during the same periods, these can be observed in the figures 5.7, 5.9 and 5.8. There is a marked improvement in the adherence of predicted profiles to actuals compared to what was previously found for heating needs. This indicates that the constraints on data cleanliness and search

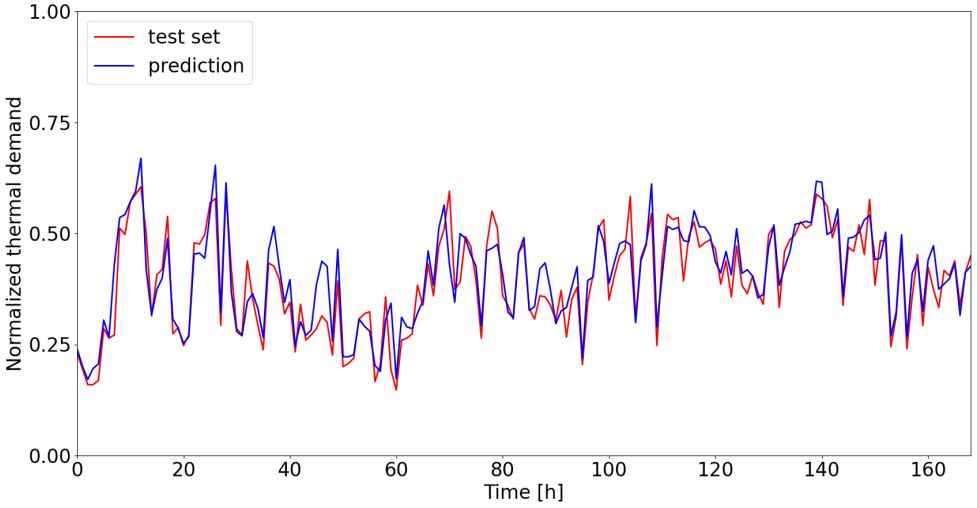

FIGURE 5.6: Thermal energy needs predicted (blue) vs. real (red) – summer week.

space are probably better calibrated than in the previous case.

This circumstance is related to the fact that the excursions of cooling need are significantly lower for a refrigeration hydronic circuit than for a thermal one, both in terms of flow rates and temperatures. Regarding the first one, it must be observed that not all spaces and, therefore, utilities subject to heating are necessarily cooled, causing the processed flow rates and, consequently, their variations to be lower. In particular, in a hospital, the areas having higher cooling requirements are those with higher intensity of care, e.g. intensive care units, high-dependency units and surgery areas, which are characterized by continuous and stable operations over time. Second point, a hydronic cooling circuit operates with ΔT temperature ranges between supply and return flow of about 5°C (typically, 7°C in supply and 12°C in return).

Conversely, a hydronic heating circuit can operate with much larger temperature differences: typically, for areas served by AHUs or fancoils, there are ΔT temperature ranges of 10°C between supply and return (usually, 50°C as supply temperatures and 40°C as return temperatures), but there is nothing to prevent such temperature differences under certain circumstances from even increasing (for example, due to sudden demands from utilities or different emission items, such as radiators). In fact, unlike what was highlighted for cooling, heating is substantially extended

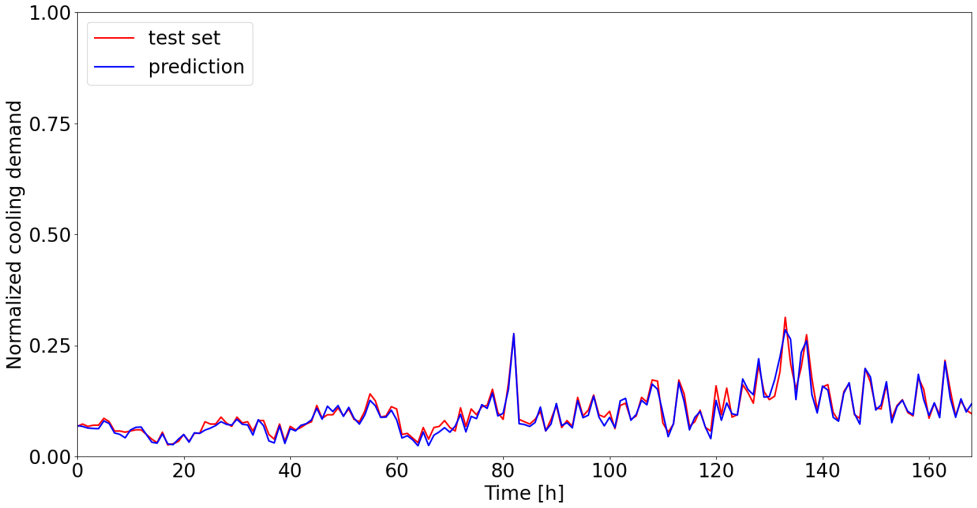

FIGURE 5.7: Cooling energy needs predicted (blue) vs. real (red) – winter week.

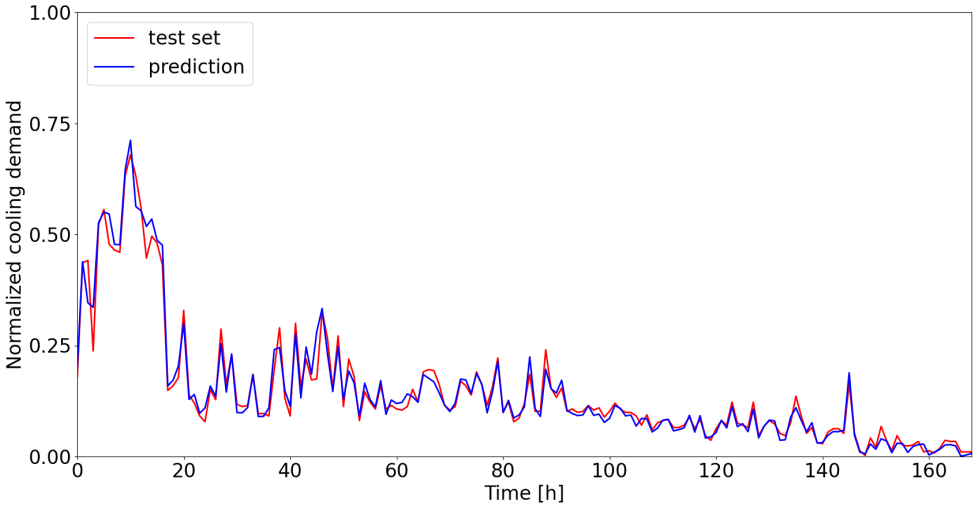

FIGURE 5.8: Cooling energy needs predicted (blue) vs. real (red) – intermediate season week.

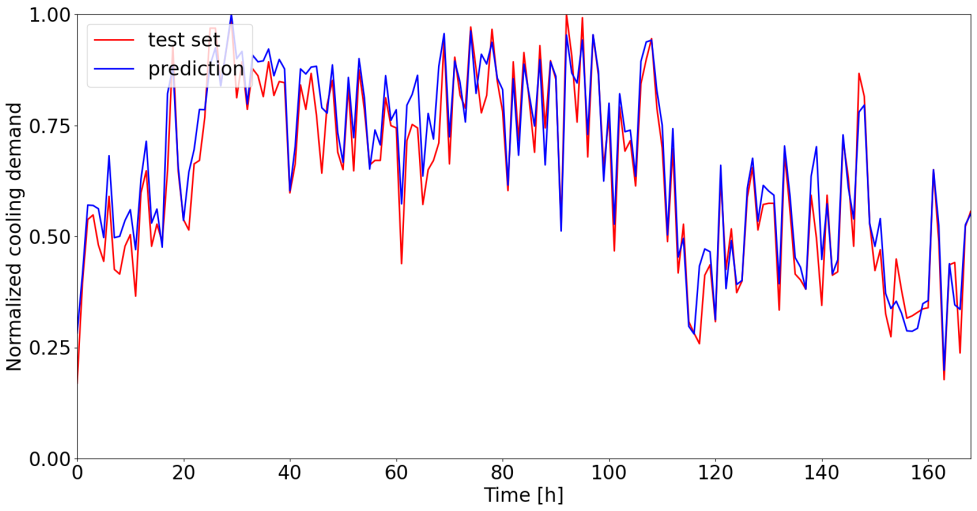

FIGURE 5.9: Cooling energy needs predicted (blue) vs. real (red) – summer week.

to all spaces of the hospital structure, each of which is subject to different thermo-hygrometric, ventilation and service system requirements, elements that drastically increase the variability of behavior. This implies that the variations in cooling need are much smaller than those in heating ones, making the constraints on data cleanliness and search space properly calibrated so that the model accurately and precisely reproduces the demands profile.

The electricity demand forecast, in contrast to the previous cases, presents the least satisfactory results among the three investigated, as also mentioned in the section 5.3.4. The greater computational complexity is found, in fact, in demand profiles that are decidedly less regular than those for heating and cooling.

The essentially zero inertia of the system, of the users and of its energy vector means that the variations are much more sudden and frequent than those that can be recorded in the case of heating or cooling demands. Furthermore, this circumstance finds its coherence in the fact that the algorithm itself leans towards an ET type model, rather than an RF one, as anticipated in the previous paragraph 5.3.4.

The comparison between the electricity demand profiles in the winter, summer and intermediate seasons can be observed in the figures 5.10, 5.12 and 5.11. These show a good reproduction of the profiles when the needs are lower, i.e. close to the base load. This circumstance, which occurs mostly in the winter period, also

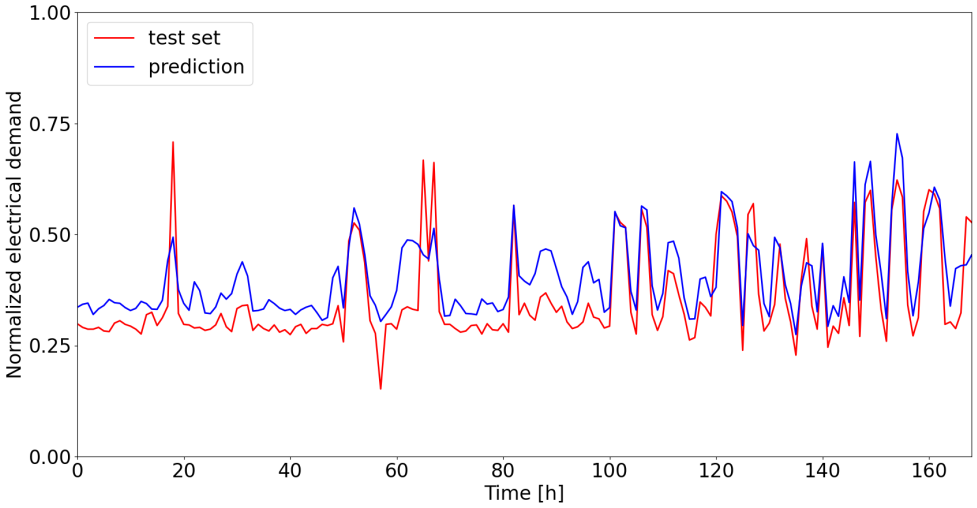

FIGURE 5.10: Electricity energy needs predicted (blue) vs. real (red) – winter week.

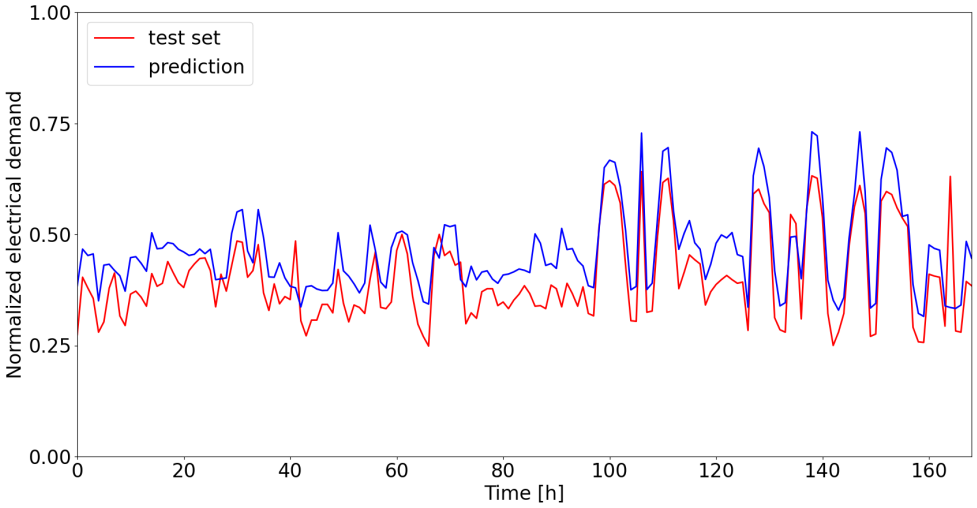

FIGURE 5.11: Electricity energy needs predicted (blue) vs. real (red) – intermediate season week.

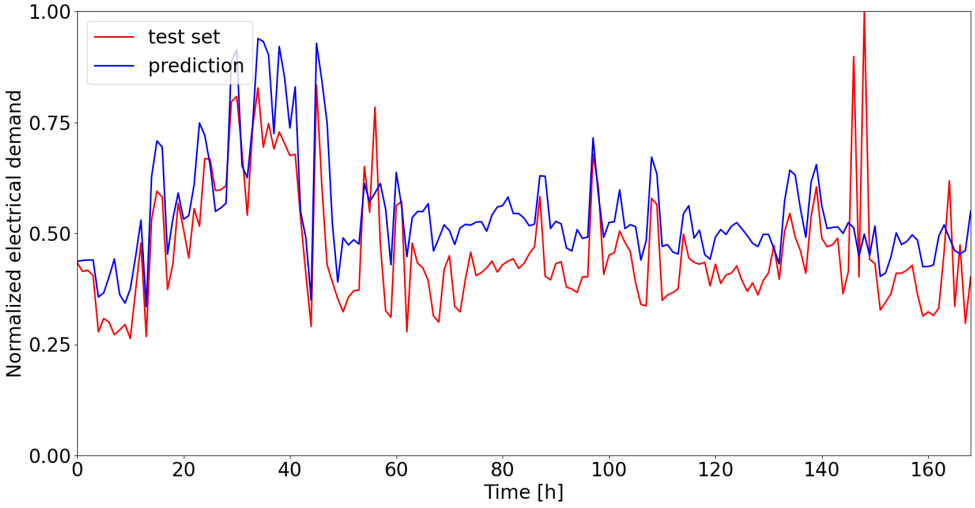

FIGURE 5.12: Electricity energy needs predicted (blue) vs. real (red) – summer week.

corresponds to the situation in which there is less variability in the electrical needs of the facilities. For the opposite reasons, however, a slight worsening is recorded when the loads are higher.

In greater detail, in the summer season, a positive offset is noted between the expected and actual demand profile. As in the cases previously observed and commented, this deviation is attributable to the inability of the model to follow and reproduce with adequate precision and accuracy the ramps of ascent and, above all, of descent of the demand. This circumstance, as in the other cases, is most likely to be found in an imperfect tuning of the data cleaning constraints by the algorithm. In fact, a non-perfect reproduction of some of the peaks, both ascending and descending, is observed in the figure 5.12.

5.4 Transfer learning

This paragraph concludes the Chapter 5 by presenting the results obtained from transfer learning by training a model to predict each of the energy needs, once the similitude is validated.

5.4.1 Variables importance

It should be noted that variable importance at this stage has a more theoretical than practical scope, as they have been validated in the previous cooperative learning phase. This is further confirmed by the fact that the gap between the weights of importance is much narrower in the transfer learning phase than in the cooperative learning phase.

Feature	Score		
	Heat	Cold	Electricity
Surface	0.00856	0.01483	0.01238
Volume	0.00183	0.00188	0.01729
Beds	0.00181	0.00827	0.00418
External temperature	0.00171	0.00010	0.00012
Month	0.00035	0.00012	0.00014
Year	0.00020	0.00068	0.00030
Hour	0.00004	0.00003	0.00005
Relative humidity	0.00002	0.00006	0.00008
Day	0.00002	0.00002	0.00002
Weekday	0.00002	0.00001	0.00002

TABLE 5.6: Feature importance for transfer learning

Compared to cooperative learning, it can be seen that the variables present a substantially reversed order of importance, with those related to geometry being more influential in predicting energy needs than ones linked to weather.

5.4.2 Machine learning model selection

Not surprisingly, compared to the previous case, neural networks, in particular MLPRegressor, are the preferred model for each of the energy needs at this stage, as it can be observed by table 5.7. This is in line with the literature, in which neural networks usually perform better when the target is to reproduce time series.

5.4.3 Hyperparameters optimization

As regards the model's hyperparameters, it can be seen that the "most complicated" energy need to predict (i.e. requiring the greatest number of neurons, hidden layers

Energy need	Machine Learning model	Hyperparameters optimizer	Convergence steps
Heat	MLP	Bayesian	2
Cooling	MLP	Bayesian	2
Electricity	MLP	Bayesian	2

TABLE 5.7: Results of the machine learning model selection for transfer learning.

and iterations) is the thermal one, followed closely by the electrical, while the cooling is still the easier to predict. In particular, it should be noted that the structure of neural networks is rather simple, although it requires thousands of neurons, although distributed over a single layer: this characteristic allows us to deduce that the relationships underlying the prediction of energy needs are essentially linear or, in any case, fairly simple.

In fact, the simulations were concluded with 14 iterations for the heating and cooling energy needs forecasts and 13 for the electricity one, which further confirms the simplicity of the models and their execution.

5.4.4 Accuracy metrics

Moving on to the metrics, in table 5.9 it can be observed that, with the exception of execution time, in the transfer learning phase they are improved compared to those of table 5.5 in the cooperative learning phase. This element represents an encouraging confirm that what was learned during the cooperative learning is used with benefit during the transfer learning phase, which can therefore claim to have successfully transferred the knowledge acquired from the similar building to the target building.

In this regard, it can be useful to compare the results of the transfer learning phase with those of the cooperative learning: it must be highlighted that the two phases involved training the models and optimizing their hyperparameters with different reference metrics, in particular the $R^2_{adjusted}$ for the cooperative learning phase and the MAPE for the transfer learning phase, making this stage to be cautiously handled. In this regard, it may be beneficial to keep in mind that, rather than a direct comparison between the two phases (which would make little sense), what it

is investigated is whether there have been improvements in the metrics from one phase to the next, as this would confirm that the knowledge has been acquired and correctly transferred from one phase to the other.

In more detail, there are improvements in the RMSE of the cooling and electricity demand and in all the RMAE, which allows to state that the model has certainly improved its ability to correlate the input variables with the output ones.

As regards the MAPE, it can be observed that for all three energy needs the values are within the acceptability range presented in the table 4.8. The improvement observed compared to the results in the table could be due to the use of this metric as an optimization metric for the model hyperparameters and, therefore, not directly linked to an actual performance improvement.

The results regarding the $R^2_{adjusted}$ metric are also interesting, as at this stage it shows better values than those found in cooperative learning. In more detail, it can be observed that the values are higher, although in the case of electricity energy need, they even exceed the acceptability threshold. This indicates potential overfitting issues, i.e. excessive specialization of the model and a risk of problem generalization ability loss. This circumstance should be mitigated by the fact that training set is made by the similar building data only. In any case, it is certainly an element worth of further investigations and studies, especially considering further building type.

5.4.5 Energy needs profiles

At the end of this chapter, a comparison is made between the real energy needs profiles of the target building represented by the validation set and those predicted by the model based on the training set made by the similar building data for the three energy needs investigated.

In detail, starting from the thermal energy need profile, figure 5.13 visually highlights that real and predicted are almost aligned with each other, with even the peaks or sudden load changes (more or less pronounced) corretly reproduced.

Given the metrics and what has just been presented, it can be assumed that the model has substantially captured the complexity of the phenomenon and the buildings similitude, without losing its ability to generalise.

However, to better understand the actual performance of the model in the case of thermal needs, also in this case it is necessary to use the diagram shown in the lower side of figure 5.13, the same of those presented for cooperative learning in the paragraph 5.3.5.

In particular, from the diagram it can be observed that at low loads the model tends to slightly underestimate energy needs, while this tendency tends to correct itself at higher ones. Furthermore, compared to the cooperative learning phase shown in the figure 5.1, the forecast dispersion is reduced. Despite a lower $R^2_{adjusted}$ in the case of transfer learning compared to cooperative learning, this is directly related to the improvement of the RMSE and RMAE metrics, mentioned in the paragraph 5.4.4 and that can be found comparing table 5.5 with table 5.9.

Moving on to the cooling need, it is observed how the forecast is equally good, or even better considering the much more complex profile compared to the previous case, made up of numerous and sudden load variations. These load variations are those that put the model in the most difficulty, where from figure 5.14 it can be observed that it slightly overestimates the load reductions and slightly underestimates the increases. This is even more evident analyzing the diagram with the actual and predicted normalized values as x-axis and y-axis respectively in the same figure. In this case, a comparison with figure 5.2 shows that the dispersion is certainly smaller than during the cooperative learning phase, which is reflected in the improvement in accuracy metrics also in this case. However, the columnar arrangement shown in the aforementioned scatter plot is striking and, indeed, represents a clear evidence that the model may have poor predictive capabilities. In more detail, the arrangement of points on this type of plot is typical of situations in which the model fails to discriminate well between different input situations, producing the same prediction for cases that actually have different values. This is a sign that the model may have limited predictive capability or be too simplified to capture the complexity of the real data.

In this regard, it should be noted that attempts have been made to force more complex models than that used by the algorithm, but the result has only been to reduce its dispersion, without ever affecting the columnar look. This circumstance therefore suggests that a larger dataset or one with more varied data could help

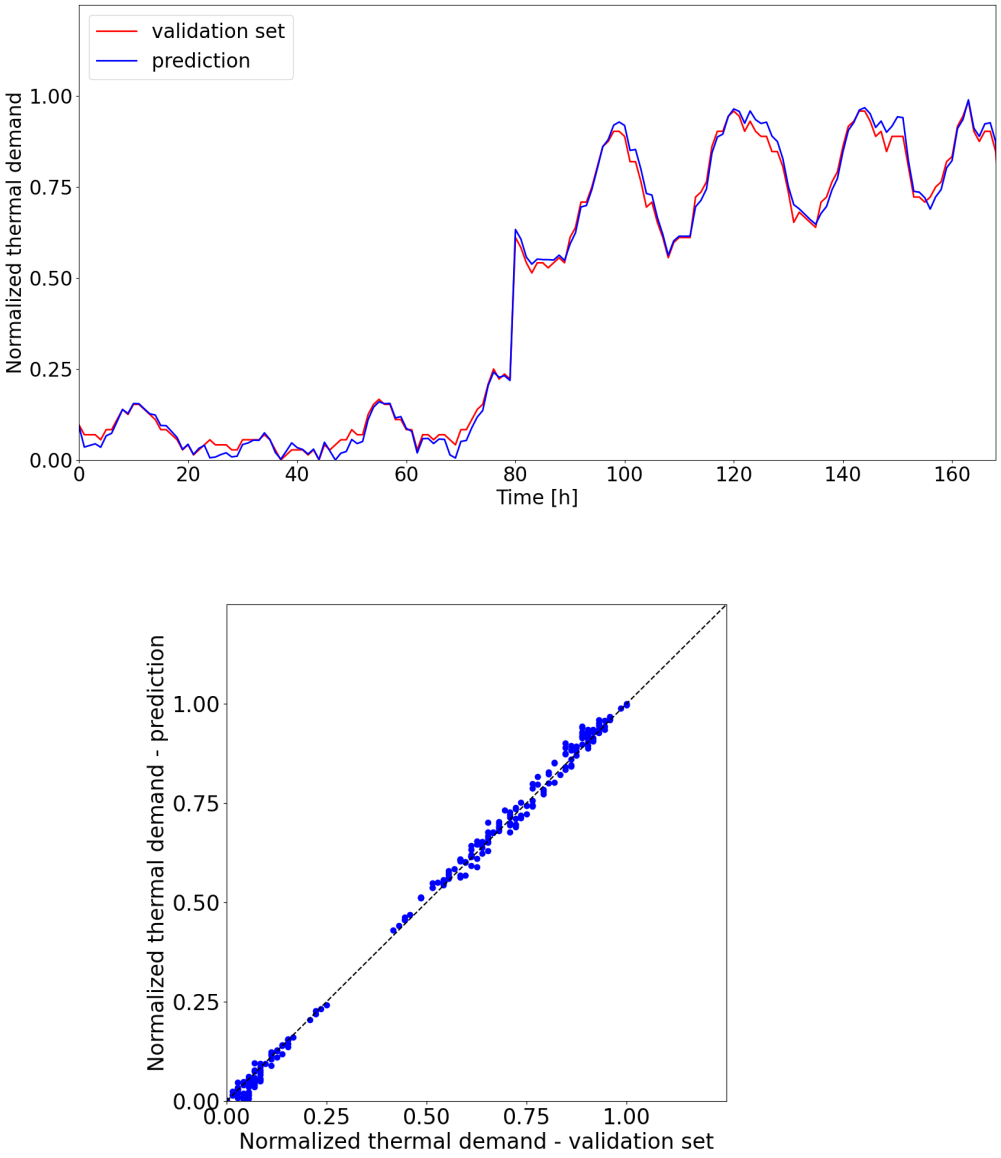

FIGURE 5.13: Thermal need profile predicted and real(up) and diagram to assess model performance (down)

improve the model's training and, therefore, its performance.

Finally, moving on to the electrical need, this is particularly challenging and interesting: even if at first glance it seems quite simple, with a relatively flat profile, the sudden, isolated and, apparently, unjustified peaks (i.e. load variations) certainly represent an interesting validation for the model, which is clearly called upon to generalise the phenomenon and correctly identify the energy need and reproduce it.

In particular, observing the diagram at the top of the figure, it can be seen that

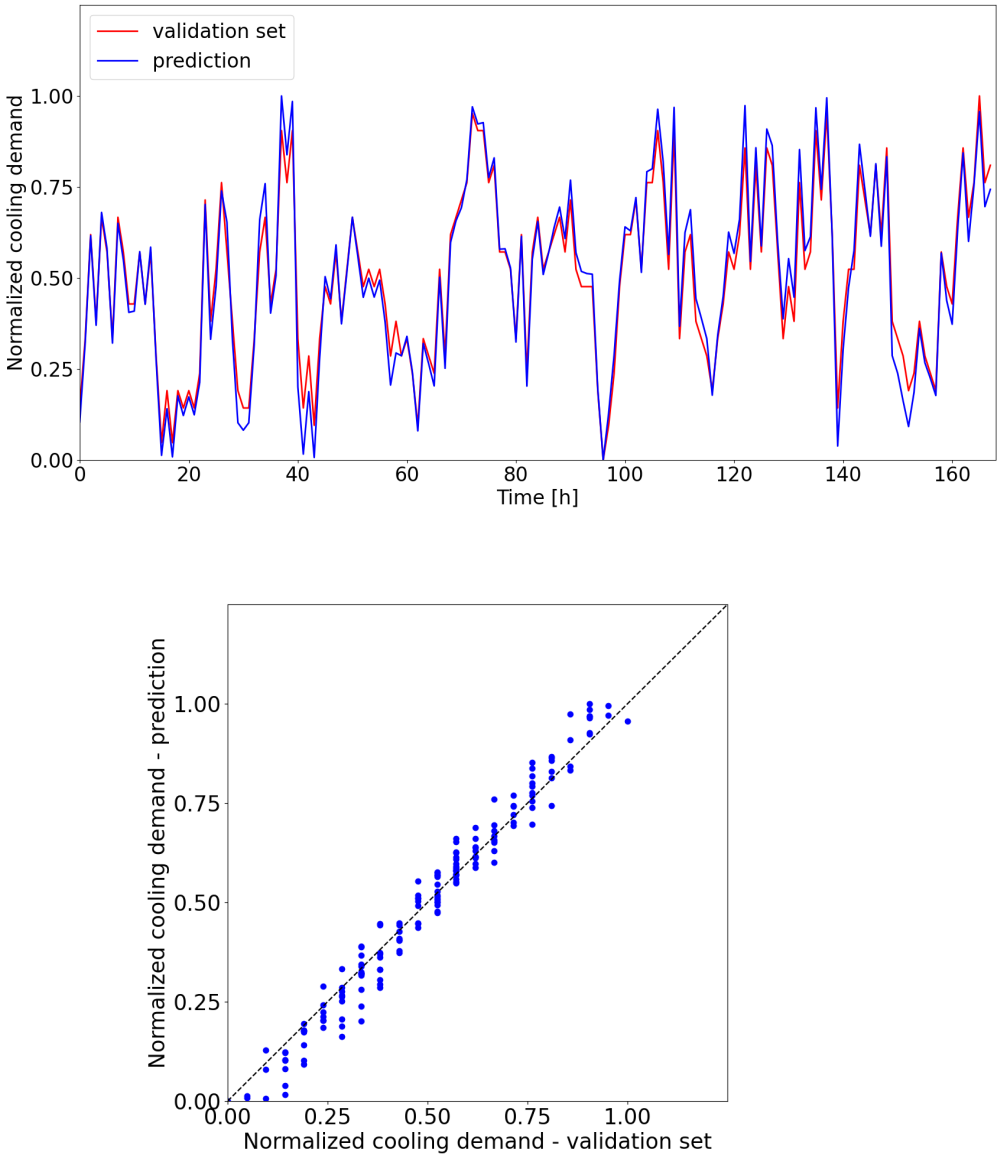

FIGURE 5.14: Cooling need profile predicted and real(up) and diagram to assess model performance (down)

the peaks are reproduced correctly, both in terms of time and intensity. The quality of the forecasts, however, is lower for the rest of the demand profile, where there is a substantial underestimation, albeit slight, of the actual electricity need. It should be noted, however, that the real profile is still predicted fairly well, with the model able to track even small load variations. Eventually, in an operational phase this deviation could be managed by introducing a constant capable of aligning the predicted and real values.

Even in this case, however, any further improvement of the models requires more

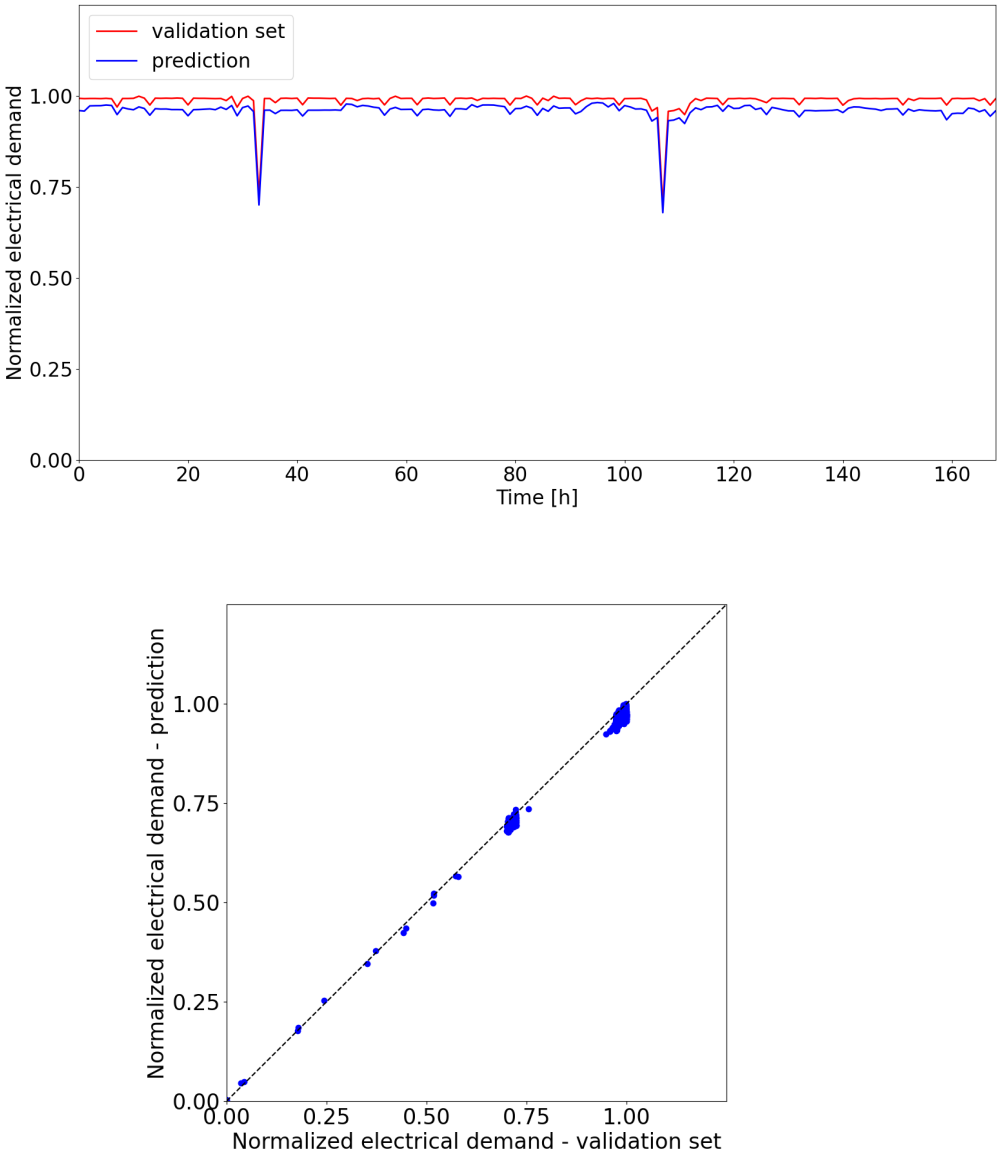

FIGURE 5.15: Electrical need profile predicted and real(up) and diagram to assess model performance (down)

in-depth studies and, above all, a more varied and broader dataset. Similarly to the cooling need, in fact, in this case too forcing more complex models did not produce any significant improvements, which is why it is believed that the dataset is the place where further improvements can be achieved.

Energy need	Machine Learning model	Hyperparameter	Value
Heat	MLP	Layers	1
		Neurons for each layer	5'000
		Activation	Relu
		Learning rate	Constant
		Initial learning rate	0.001
		Alpha	0.0001
Cold	MLP	Layers	1
		Neurons for each layer	1000
		Activation	Relu
		Learning rate	Constant
		Initial learning rate	0.001
		Alpha	0.0001
Electricity	MLP	Layers	1
		Neurons for each layer	1'500
		Activation	Relu
		Learning rate	Constant
		Initial learning rate	0.001
		Alpha	0.0001

TABLE 5.8: Results of the hyperparameters optimization for transfer learning.

Energy need	RMAE	RMSE	MAPE [%]	$R^2_{adjusted}$	Execution time [s]
Heat	0.1363	0.0225	8.219	0.8371	42.48
Cold	0.0487	0.0032	4.449	0.9411	13.05
Electricity	0.0468	0.0029	2.595	0.9974	35.55

TABLE 5.9: Metrics results for transfer learning.

Chapter 6

Conclusions

The thesis presents a method for the energy needs forecast of a building for which they are not known ahead. In particular, the proposed methodology focuses on complex buildings characterised by high and complex profile energy demands. As a first step, the proposed method identificates a scale model of the building under investigation, of which as much informations as possible are available and from which the starting dataset can be obtained: for their study and in order to validate the before mentioned methodology, it is decided to use two hospital buildings, specifically S.Anna hospital of Cona and Ca'Foncello hospital in Treviso, respectively as target and similar buildings. In more detail, an automated algorithm capable of replicating the manual steps typically taken while investigating similitude among buildings and predicts their energy demand is implemented.

To this end, first of all, a check of the similitude consistency between the buildings at a "static" level is carried out, through dimensionless ratios presented in the chapter 3.This can effectively streamline the whole process, while reducing errors associated with building selection.

After that, the data cleaning process takes place, which coupled with technical constraints has proven highly effective in enhancing the algorithm's robustness by selectively utilizing data deemed "realistic", even if they deviates from the median values. This approach further minimizes the amount of data required to accomplish the task successfully, furthermore enabling a similitude approach.

Then, after a suitably cleaned dataset is obtained on the basis of technical plant constraints, the cooperative learning phase investigates if the similitude is consistent

at a "dynamic" level, with the set of models and the related hyperparameter optimisation algorithms chosen and trained accordingly. In this stage, for all energy needs investigated in the two facilities, the method's performance evaluation metrics are met, with $R^2_{adjusted} > 0.7$, prediction errors of less than 10% and run times of less than a minute, all using training set/test set ratios of 3:1 to 12:1, far lower than the literature consulted. Moreover, including an optimizer, coupled with hyperparameter tuning, data shuffling, and cross-validation techniques, make the entire process more resilient to changes or anomalies that may arise overtime. The implementation of ensemble methods enables rapid execution times, even with large datasets, with satisfactory accuracy in testing similitude among buildings.

Once the similitude is validated at the end of the cooperative learning phase, the transfer learning one takes place. At this stage, the same steps are replicated training the AI model on similitude building training set and assessing its performances on the target building validation set. To perform this task, hyperparameters are optimized according to different accuracy metrics than that of cooperative learning, i.e. MAPE instead of $R^2_{adjusted}$, because of the focus on forecast performances than generalization abilities. The transfer learning phase has highlighted also some limitations that had not emerged during the cooperative learning phase: in more detail, if the data cleaning carried out in the preprocessing stage has been particularly useful in reliably establishing whether there is a similarity between the target building and the one taken as a model, on the other hand the considerable amount of discarded data meant that in the transfer learning phase the performances of the models are not so satisfactory, in particular with regard to the cooling and electrical needs. In detail, in the first one, the scatter plot shows that the cooling model may have limited generalization capabilities despite metrics that are still acceptable compared to the imposed thresholds, while in the second, a constant underestimation of the building's actual needs is highlighted together with overfitting issues that may be a redflag for a lack of generalization ability.

In conclusion, by developing an enhanced understanding of the mechanisms that drive energy demands in a complex system, the key variables influencing these needs and the processes governing energy usage similarities, this work enables more rapid forecasting of a building's energy needs from the outset. This allows for the

timely implementation of energy conservation measures aimed at evaluating and improving overall efficiency, as well as conducting insightful "what-if" scenario analyses to deepen the understanding of the building's energy profile without relying on time-consuming simulations. Furthermore, the implementation of a similitude approach makes the lack of data no-longer a problem.

Collectively, these findings enable more efficient and effective energy management strategies, potentially paving the way to significant economic and environmental benefits across the energy sector.

For what concerns future developments, they include test and validation of the methodology on a larger pool of buildings, with different or the same uses as in this thesis work, as well as identifying improvements to further reduce the training/test ratio.

Bibliography

[1] Hussein Abushamah, Ondřej Burian, and Radek Škoda. "Design and Operation Optimization of a Nuclear Heat-Driven District Cooling System". In: *International Journal of Energy Research* 2023 (Aug. 2023), pp. 1–18. DOI: 10.1155/2023/7880842.

[2] A.S. Ahmad et al. "A review on applications of ANN and SVM for building electrical energy consumption forecasting". In: *Renewable and Sustainable Energy Reviews* 33 (2014), pp. 102–109. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2014.01.069. URL: https://www.sciencedirect.com/science/article/pii/S1364032114000914.

[3] Muhammad Waseem Ahmad, Monjur Mourshed, and Yacine Rezgui. "Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption". In: *Energy and Buildings* 147 (2017), pp. 77–89. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2017.04.038. URL: https://www.sciencedirect.com/science/article/pii/S0378778816313937.

[4] Yacine Allab et al. "Energy and comfort assessment in educational building: Case study in a French university campus". In: *Energy and Buildings* 143 (2017), pp. 202–219. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2016.11.028. URL: https://www.sciencedirect.com/science/article/pii/S0378778816315717.

[5] Mohammad H. Alobaidi, Fateh Chebana, and Mohamed A. Meguid. "Robust ensemble learning framework for day-ahead forecasting of household based energy consumption". In: *Applied Energy* 212 (2018), pp. 997–1012. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2017.12.054. URL: https://www.sciencedirect.com/science/article/pii/S0306261917317695.

[6] Rajanikanth Aluvalu and Rinkal Patel. "A Reduced Error Pruning Technique for Improving Accuracy of Decision Tree Learning". In: *International Journal of Engineering and Advanced Technology (IJEAT)* 3 (June 2014).

[7] Khuram Pervez Amber et al. "Energy Consumption Forecasting for University Sector Buildings". In: *Energies* 10.10 (2017). ISSN: 1996-1073. DOI: 10.3390/en10101579. URL: https://www.mdpi.com/1996-1073/10/10/1579.

[8] Nima Amjady. "Short-term hourly load forecasting using time-series modeling with peak load estimation capability". In: *IEEE Transactions on Power Systems - IEEE TRANS POWER SYST* 16 (Aug. 2001), pp. 498–505. DOI: 10.1109/59.932287.

[9] R. L. Anderson. "Recent Advances in Finding Best Operating Conditions". In: *Journal of the American Statistical Association* 48.264 (1953), pp. 789–798. DOI: 10.1080/01621459.1953.10501200. eprint: https://www.tandfonline.com/doi/pdf/10.1080/01621459.1953.10501200. URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1953.10501200.

[10] J. Scott Armstrong. "Evaluating Forecasting Methods". In: *Principles of Forecasting: A Handbook for Researchers and Practitioners*. Ed. by J. Scott Armstrong. Boston, MA: Springer US, 2001, pp. 443–472. ISBN: 978-0-306-47630-3. DOI: 10.1007/978-0-306-47630-3_20. URL: https://doi.org/10.1007/978-0-306-47630-3_20.

[11] J.Scott Armstrong and Fred Collopy. "Error measures for generalizing about forecasting methods: Empirical comparisons". In: *International Journal of Forecasting* 8.1 (1992), pp. 69–80. ISSN: 0169-2070. DOI: https://doi.org/10.1016/0169-2070(92)90008-W. URL: https://www.sciencedirect.com/science/article/pii/016920709290008W.

[12] Pragati Baheti. "Train Test Validation Split: How To and Best Practices". In: (2024). Accessed 01-09-2024. URL: https://www.v7labs.com/blog/train-validation-test-set.

[13] Ilaria Ballarini, Stefano Paolo Corgnati, and Vincenzo Corrado. "Use of reference buildings to assess the energy saving potentials of the residential building stock: The experience of TABULA project". In: *Energy Policy* 68 (2014),

pp. 273–284. ISSN: 0301-4215. DOI: https://doi.org/10.1016/j.enpol.2014.01.027. URL: https://www.sciencedirect.com/science/article/pii/S0301421514000329.

[14] Ilaria Ballarini et al. "Improving energy modeling of large building stock through the development of archetype buildings". In: *Building Simulation* (2011), pp. 2874–2881.

[15] C.S. Barnaby and Jeffrey Spitler. "Development of the Residential Load Factor Method for Heating and Cooling Load Calculations". In: *ASHRAE Transactions* 111 (Jan. 2005), pp. 291–307.

[16] C.S. Barnaby, Jeffrey Spitler, and D. Xiao. "The residential heat balance method for heating and cooling load calculations". In: *ASHRAE Transactions* 111 (Jan. 2005), pp. 308–319.

[17] Alireza Behroozsarand and Sirous Shafiei. "Optimal control of distillation column using Non-Dominated Sorting Genetic Algorithm-II". In: *Journal of Loss Prevention in the Process Industries* 24.1 (2011), pp. 25–33. ISSN: 0950-4230. DOI: https://doi.org/10.1016/j.jlp.2010.08.009. URL: https://www.sciencedirect.com/science/article/pii/S0950423010001075.

[18] James Bergstra and Yoshua Bengio. "Random search for hyper-parameter optimization". In: *J. Mach. Learn. Res.* 13.null (2012), 281–305. ISSN: 1532-4435.

[19] James Bergstra et al. "Algorithms for hyper-parameter optimization". In: *Proceedings of the 24th International Conference on Neural Information Processing Systems*. NIPS'11. Granada, Spain: Curran Associates Inc., 2011, 2546–2554. ISBN: 9781618395993.

[20] Mathieu Bourdeau et al. "Modeling and forecasting building energy consumption: A review of data-driven techniques". In: *Sustainable Cities and Society* 48 (2019), p. 101533. ISSN: 2210-6707. DOI: https://doi.org/10.1016/j.scs.2019.101533. URL: https://www.sciencedirect.com/science/article/pii/S2210670718323862.

[21] M. Bouzerdoum, A. Mellit, and A. Massi Pavan. "A hybrid model (SARIMA–SVM) for short-term power forecasting of a small-scale grid-connected photovoltaic

plant". In: *Solar Energy* 98 (2013), pp. 226–235. ISSN: 0038-092X. DOI: https://doi.org/10.1016/j.solener.2013.10.002. URL: https://www.sciencedirect.com/science/article/pii/S0038092X13004039.

[22] Facundo Bre, Juan Gimenez, and Víctor Fachinotti. "Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks". In: *Energy and Buildings* 158 (Nov. 2017). DOI: 10.1016/j.enbuild.2017.11.045.

[23] Leo Breiman. "Bagging Predictors". In: *Machine Learning* 24.2 (1996), pp. 123–140. ISSN: 1573-0565. DOI: 10.1023/A:1018054314350. URL: https://doi.org/10.1023/A:1018054314350.

[24] Stone Breiman Olshen. *Decision Tree — link.springer.com*. https://link.springer.com/referenceworkentry/10.1007/978-0-387-30164-8_204. Accessed 26-10-2024.

[25] *Buildings Energy System - IEA — iea.org*. https://www.iea.org/energy-system/buildings. [Accessed 25-10-2024]. 7-2023.

[26] L.J. Cao and Francis Tay. "Support vector machine with adaptive parameters in financial time series forecasting". In: *Neural Networks, IEEE Transactions on* 14 (Dec. 2003), pp. 1506 –1518. DOI: 10.1109/TNN.2003.820556.

[27] Mauro Castelli et al. "Prediction of energy performance of residential buildings: A genetic programming approach". In: *Energy and Buildings* 102 (2015), pp. 67–74. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2015.05.013. URL: https://www.sciencedirect.com/science/article/pii/S0378778815003849.

[28] Dan Ciresan et al. "Flexible, High Performance Convolutional Neural Networks for Image Classification." In: July 2011, pp. 1237–1242. DOI: 10.5591/978-1-57735-516-8/IJCAI11-210.

[29] Charles Coleman and David Swanson. "On MAPE-R as a measure of cross-sectional estimation and forecast accuracy". In: *Journal of economic and social measurement* 32 (Dec. 2007), pp. 219–233. DOI: 10.3233/JEM-2007-0290.

[30] Johnson Controls. *YHAU-CL/CH Single Effect Hot Water Absorption Chiller (Mod A)*. https://docs.johnsoncontrols.com/chillers/api/khub/documents/v0s5_zvbFvXiMABui_Vp9w/content. [Accessed 14-10-2024]. 2024.

[31] M. Davarynejad et al. "Evolutionary hidden information detection by granulation-based fitness approximation". In: *Applied Soft Computing* 10.3 (2010), pp. 719–729. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2009.09.001. URL: https://www.sciencedirect.com/science/article/pii/S1568494609001616.

[32] Kalyanmoy Deb and William M. Spears. "Population Structures C 6 . 2 Speciation methods". In: 1997. URL: https://api.semanticscholar.org/CorpusID:3547258.

[33] Bing Dong, Cheng Cao, and Siew Eang Lee. "Applying support vector machines to predict building energy consumption in tropical region". In: *Energy and Buildings* 37.5 (2005), pp. 545–553. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2004.09.009. URL: https://www.sciencedirect.com/science/article/pii/S0378778804002981.

[34] M. Economidou et al. "Review of 50 years of EU energy efficiency policies for buildings". In: *Energy and Buildings* 225 (2020), p. 110322. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2020.110322. URL: https://www.sciencedirect.com/science/article/pii/S0378778820317229.

[35] *Energy consumption in households*. URL: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Energy_consumption_in_households.

[36] A. Famili et al. "Data preprocessing and intelligent data analysis". In: *Intelligent Data Analysis* 1.1 (1997), pp. 3–23. ISSN: 1088-467X. DOI: https://doi.org/10.1016/S1088-467X(98)00007-9. URL: https://www.sciencedirect.com/science/article/pii/S1088467X98000079.

[37] Cheng Fan, Fu Xiao, and Shengwei Wang. "Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques". In: *Applied Energy* 127 (2014), pp. 1–10. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2014.04.016. URL: https://www.sciencedirect.com/science/article/pii/S0306261914003596.

[38] Cheng Fan et al. "Assessment of deep recurrent neural network-based strategies for short-term building energy predictions". In: *Applied Energy* 236 (2019), pp. 700–710. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2018.12.004. URL: https://www.sciencedirect.com/science/article/pii/S0306261918318221.

[39] Robert Fildes. "The evaluation of extrapolative forecasting methods". In: *International Journal of Forecasting* 8.1 (1992), pp. 81–98. ISSN: 0169-2070. DOI: https://doi.org/10.1016/0169-2070(92)90009-X. URL: https://www.sciencedirect.com/science/article/pii/016920709290009X.

[40] P. Frazier. "A Tutorial on Bayesian Optimization". In: *ArXiv* abs/1807.02811 (2018). URL: https://api.semanticscholar.org/CorpusID:49656213.

[41] Milton Friedman and L. J. Savage. *Planning experiments seeking maxima*. New York [u.a.], 1947.

[42] Dan Fylstra Frontline Solvers. *Genetic and Evolutionary Algorithms Versus Classical Optimization*. 2024. URL: https://www.solver.com/press/backgrounder-genetic-and-evolutionary-algorithms-versus-classical-optimization.

[43] Yangyang Fu et al. "Using Support Vector Machine to Predict Next Day Electricity Load of Public Buildings with Sub-metering Devices". In: *Procedia Engineering* 121 (Dec. 2015), pp. 1016–1022. DOI: 10.1016/j.proeng.2015.09.097.

[44] Agostino Gambarotta et al. "Demonstrating a smart controller in a hospital integrated energy system". In: *Smart Energy* 12 (2023), p. 100120. ISSN: 2666-9552. DOI: https://doi.org/10.1016/j.segy.2023.100120. URL: https://www.sciencedirect.com/science/article/pii/S2666955223000278.

[45] Yazeed Yasin Ghadi, M.G. Rasul, and M.M.K. Khan. "Energy Savings by Fuzzy Base Control of Occupancy Concentration in Institutional Buildings". In: *Energy Procedia* 105 (2017). 8th International Conference on Applied Energy, ICAE2016, 8-11 October 2016, Beijing, China, pp. 2850–2858. ISSN: 1876-6102. DOI: https://doi.org/10.1016/j.egypro.2017.03.628. URL: https://www.sciencedirect.com/science/article/pii/S1876610217306823.

[46] Corrado Gini. "Measurement of Inequality of Incomes". In: *The Economic Journal* 31.121 (1921), pp. 124–126. ISSN: 00130133, 14680297. URL: http://www.jstor.org/stable/2223319 (visited on 05/28/2022).

[47] Corrado Gini. "Variabilità e mutabilità". In: *Reprinted in Memorie di metodologica statistica (Ed. Pizetti E* (1912).

[48] Yin Guo et al. "Hourly cooling load forecasting using time-indexed ARX models with two-stage weighted least squares regression". In: *Energy Conversion and Management* 80 (2014), pp. 46–53. ISSN: 0196-8904. DOI: https://doi.org/10.1016/j.enconman.2013.12.060. URL: https://www.sciencedirect.com/science/article/pii/S0196890414000259.

[49] N. Hovenga. *Dynamic Characteristics of Industrial Heat Pump Operation—A Modeling Study*. https://resolver.tudelft.nl/uuid:25adc273-6d05-4a98-9319-be64d8e30bb8.

[50] Frank Hutter. "Automated configuration of algorithms for solving hard computational problems". PhD thesis. University of British Columbia, 2009. DOI: http://dx.doi.org/10.14288/1.0051652. URL: https://open.library.ubc.ca/collections/ubctheses/24/items/1.0051652.

[51] Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. "Sequential Model-Based Optimization for General Algorithm Configuration". In: *Learning and Intelligent Optimization*. 2011. URL: https://api.semanticscholar.org/CorpusID:6944647.

[52] Rob J. Hyndman and Anne B. Koehler. "Another look at measures of forecast accuracy". In: *International Journal of Forecasting* 22.4 (2006), pp. 679–688. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2006.03.001. URL: https://www.sciencedirect.com/science/article/pii/S0169207006000239.

[53] Rishabh Jain. *Decision Tree. It begins here*. 2017. URL: https://www.coursehero.com/file/63589772/decision-tree-gini-indexpdf/ (visited on 05/28/2022).

[54] DM Jose, AM Vincent, and GS Dwarakish. *Improving multiple model ensemble predictions of daily precipitation and temperature through machine learning techniques. Sci Rep 12: 4678*. 2022.

[55] V. Roshan Joseph. "Optimal ratio for data splitting". In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 15 (Apr. 2022). DOI: 10.1002/sam.11583.

[56] J. Kennedy and R. Eberhart. "Particle swarm optimization". In: *Proceedings of ICNN'95 - International Conference on Neural Networks*. Vol. 4. 1995, 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968.

[57] Mergani A. Khairalla et al. "Short-Term Forecasting for Energy Consumption through Stacking Heterogeneous Ensemble Learning Model". In: *Energies* 11.6 (2018). ISSN: 1996-1073. DOI: 10.3390/en11061605. URL: https://www.mdpi.com/1996-1073/11/6/1605.

[58] Eui-Jong Kim et al. "Urban energy simulation: Simplification and reduction of building envelope models". In: *Energy and Buildings* 84 (2014), pp. 193–202. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2014.07.066. URL: https://www.sciencedirect.com/science/article/pii/S037877881400615X.

[59] Frédéric Kuznik et al. "A review on phase change materials integrated in building walls". In: *Renewable and Sustainable Energy Reviews* 15.1 (2011), pp. 379–391. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2010.08.019. URL: https://www.sciencedirect.com/science/article/pii/S1364032110002716.

[60] Miguel Leon and N. Xiong. "Investigation of Mutation Strategies in Differential Evolution for Solving Global Optimization Problems". In: vol. 8467. Feb. 2014. ISBN: 978-3-319-07172-5. DOI: 10.1007/978-3-319-07173-2_32.

[61] Zhengwei Li and Gongsheng Huang. "Re-evaluation of building cooling load prediction models for use in humid subtropical area". In: *Energy and Buildings* 62 (2013), pp. 442–449. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2013.03.035. URL: https://www.sciencedirect.com/science/article/pii/S037877881300203X.

[62] Dandan Liu, Qijun Chen, and Kazuyuki Mori. "Time series forecasting method of building energy consumption using support vector regression". In: *2015*

IEEE International Conference on Information and Automation. 2015, pp. 1628–1632. DOI: 10.1109/ICInfA.2015.7279546.

[63] Spyros Makridakis. "Accuracy measures: theoretical and practical concerns". In: *International Journal of Forecasting* 9.4 (1993), pp. 527–529. ISSN: 0169-2070. DOI: https://doi.org/10.1016/0169-2070(93)90079-3. URL: https://www.sciencedirect.com/science/article/pii/0169207093900793.

[64] Jyoti S. Kale Manjusha P . Dhoke and Pavan K. Dhoke. "Investigation on adaptive genetic algorithm and metaheuristic methods within stochastic optimisation". In: 6 (2017), pp. 835–840. ISSN: 2394-5869.

[65] Daniel L. Marino, Kasun Amarasinghe, and Milos Manic. "Building energy load forecasting using Deep Neural Networks". In: *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*. 2016, pp. 7046–7051. DOI: 10.1109/IECON.2016.7793413.

[66] Joaquim Massana et al. "Short-term load forecasting in a non-residential building contrasting models and attributes". In: *Energy and Buildings* 92 (Apr. 2015). DOI: 10.1016/j.enbuild.2015.02.007.

[67] Mohammad Azhar Mat Daut et al. "Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods: A review". In: *Renewable and Sustainable Energy Reviews* 70 (2017), pp. 1108–1118. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2016.12.015. URL: https://www.sciencedirect.com/science/article/pii/S1364032116310619.

[68] Usama Mehboob et al. "Genetic Algorithms in Wireless Networking: Techniques, Applications, and Issues". In: *Soft Computing* 20 (June 2016). DOI: 10.1007/s00500-016-2070-9.

[69] Olfa Mejri, Elena Palomo Del Barrio, and Nadia Ghrab-Morcos. "Energy performance assessment of occupied buildings using model identification techniques". In: *Energy and Buildings* 43.2 (2011), pp. 285–299. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2010.09.010. URL: https://www.sciencedirect.com/science/article/pii/S0378778810003208.

[70] R. Mena et al. "A prediction model based on neural networks for the energy consumption of a bioclimatic building". In: *Energy and Buildings* 82 (2014), pp. 142–155. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2014.06.052. URL: https://www.sciencedirect.com/science/article/pii/S0378778814005349.

[71] William Menke. "Chapter 9 - Nonlinear Inverse Problems". In: *Geophysical Data Analysis: Discrete Inverse Theory (Third Edition)*. Ed. by William Menke. Third Edition. Boston: Academic Press, 2012, pp. 163–188. ISBN: 978-0-12-397160-9. DOI: https://doi.org/10.1016/B978-0-12-397160-9.00009-6. URL: https://www.sciencedirect.com/science/article/pii/B9780123971609000096.

[72] Shivlal Mewada, P. Sharma, and Shreeya Gautam. "Exploration of fuzzy system with applications". In: Dec. 2017, pp. 479–498. DOI: 10.4018/978-1-5225-3232-3.ch025.

[73] Milos Miljanovic. "Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction". In: *Indian Journal of Computer Science and Engineering* 3 (Feb. 2012).

[74] Melanie Mitchell. *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998. ISBN: 0262631857.

[75] Elena Mocanu et al. "Unsupervised energy prediction in a Smart Grid context using reinforcement cross-building transfer learning". In: *Energy and Buildings* 116 (2016), pp. 646–655. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2016.01.030. URL: https://www.sciencedirect.com/science/article/pii/S0378778816300305.

[76] J. Mockus. *Bayesian Approach to Global Optimization: Theory and Applications*. Mathematics and its Applications. Springer Netherlands, 2012. ISBN: 9789400909090. URL: https://books.google.it/books?id=VuKoCAAAQBAJ.

[77] Kristine Monteith et al. "Turning Bayesian model averaging into Bayesian model combination". In: *The 2011 International Joint Conference on Neural Networks*. 2011, pp. 2657–2663. DOI: 10.1109/IJCNN.2011.6033566.

[78] Riccardo Moriconi, Marc Peter Deisenroth, and K. S. Sesh Kumar. "High-dimensional Bayesian optimization using low-dimensional feature spaces". In: *Machine Learning* 109.9 (2020), pp. 1925–1943. ISSN: 1573-0565. DOI: 10.1007/s10994-020-05899-z. URL: https://doi.org/10.1007/s10994-020-05899-z.

[79] Jay I. Myung, Daniel R. Cavagnaro, and Mark A. Pitt. "A tutorial on adaptive design optimization". In: *Journal of Mathematical Psychology* 57.3 (2013), pp. 53–67. ISSN: 0022-2496. DOI: https://doi.org/10.1016/j.jmp.2013.05.005. URL: https://www.sciencedirect.com/science/article/pii/S0022249613000503.

[80] Alberto Hernandez Neto and Flávio Augusto Sanzovo Fiorelli. "Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption". In: *Energy and Buildings* 40.12 (2008), pp. 2169–2176. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2008.06.013. URL: https://www.sciencedirect.com/science/article/pii/S0378778808001448.

[81] Fernando Nogueira. *Bayesian Optimization: Open source constrained global optimization tool for Python*. 2014–. URL: https://github.com/bayesian-optimization/BayesianOptimization.

[82] Dario Passos and Puneet Mishra. "A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks". In: *Chemometrics and Intelligent Laboratory Systems* 223 (2022), p. 104520. ISSN: 0169-7439. DOI: https://doi.org/10.1016/j.chemolab.2022.104520. URL: https://www.sciencedirect.com/science/article/pii/S0169743922000314.

[83] Subodh Paudel et al. "A relevant data selection method for energy consumption prediction of low energy building based on support vector machine". In: *Energy and Buildings* 138 (2017), pp. 240–256. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2016.11.009. URL: https://www.sciencedirect.com/science/article/pii/S0378778816314700.

[84] Subodh Paudel et al. "Support vector machine in prediction of building energy demand using pseudo dynamic approach". In: June 2015, pp. 1–13. DOI: https://doi.org/10.48550/arXiv.1507.05019.

[85] Gilles Plessis et al. "Using design of experiments methods to develop low energy building model under Modelica". In: *Proceedings of Building Simulation 2011: 12th Conference of International Building Performance Simulation Association* (Jan. 2011).

[86] R. Polikar. "Ensemble based systems in decision making". In: *IEEE Circuits and Systems Magazine* 6.3 (2006), pp. 21–45. ISSN: 1558-0830. DOI: 10.1109/MCAS.2006.1688199.

[87] Jesús A. Pulido-Arcas, Alexis Pérez-Fargallo, and Carlos Rubio-Bellido. "Multivariable regression analysis to assess energy consumption and CO2 emissions in the early stages of offices design in Chile". In: *Energy and Buildings* 133 (2016), pp. 738–753. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2016.10.031. URL: https://www.sciencedirect.com/science/article/pii/S0378778816312671.

[88] J. R. Quinlan. "Induction of decision trees". In: *Machine Learning* 1 (1986), pp. 81–106. DOI: 10.1007/BF00116251. URL: https://doi.org/10.1007/BF00116251.

[89] J.R. Quinlan. "Simplifying decision trees". In: *International Journal of Man-Machine Studies* 27.3 (1987), pp. 221–234. ISSN: 0020-7373. DOI: https://doi.org/10.1016/S0020-7373(87)80053-6. URL: https://www.sciencedirect.com/science/article/pii/S0020737387800536.

[90] Aowabin Rahman, Vivek Srikumar, and Amanda D. Smith. "Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks". In: *Applied Energy* 212 (2018), pp. 372–385. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2017.12.051. URL: https://www.sciencedirect.com/science/article/pii/S0306261917317658.

[91] Imran Rahman et al. "On the performance of accelerated particle swarm optimization for charging plug-in hybrid electric vehicles". In: *Alexandria Engineering Journal* 55.1 (2016), pp. 419–426. ISSN: 1110-0168. DOI: https://doi.

org/10.1016/j.aej.2015.11.002. URL: https://www.sciencedirect.com/science/article/pii/S1110016815001921.

[92] L.A. Rastrigin. "Automation and Remote Contro". In: 24.11 (1963), 1337–1342. URL: https://archive.org/details/sim_automation-and-remote-control_1963-11_24_11/page/n1/mode/2up.

[93] Sameen Rj. "Artificial intelligence: In modern dentistry". In: *Journal of Dental Research and Review* 7 (Mar. 2020). DOI: 10.4103/jdrr.jdrr_2_20.

[94] Joseph Rocca. *Ensemble methods: bagging, boosting and stacking. Understanding the key concepts of ensemble learning*. 2019. URL: https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205 (visited on 04/23/2019).

[95] D. E. Rumelhart, J. L. McClelland, and PDP Research Group, eds. *Parallel Distributed Processing. Volume 1: Foundations*. Cambridge, MA: MIT Press, 1986.

[96] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall, 2010.

[97] Oluwarotimi Williams Samuel et al. "Chapter 5 - Computational Intelligence Enabling the Development of Efficient Clinical Decision Support Systems: Case Study of Heart Failure". In: *Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications*. Ed. by Arun Kumar Sangaiah, Michael Sheng, and Zhiyong Zhang. Intelligent Data-Centric Systems. Academic Press, 2018, pp. 123–133. ISBN: 978-0-12-813314-9. DOI: https://doi.org/10.1016/B978-0-12-813314-9.00005-0. URL: https://www.sciencedirect.com/science/article/pii/B9780128133149000050.

[98] Robert E. Schapire and Yoram Singer. "Improved Boosting Algorithms Using Confidence-rated Predictions". In: *Machine Learning* 37.3 (1999), pp. 297–336. ISSN: 1573-0565. DOI: 10.1023/A:1007614523901. URL: https://doi.org/10.1023/A:1007614523901.

[99] Günther Schrack and Mark Choit. "Optimized relative step size random searches". In: *Mathematical Programming* 10.1 (1976), pp. 230–244. ISSN: 1436-4646. DOI: 10.1007/BF01580669. URL: https://doi.org/10.1007/BF01580669.

[100] M. Schumer and K. Steiglitz. "Adaptive step size random search". In: *IEEE Transactions on Automatic Control* 13.3 (1968), pp. 270–276. DOI: 10.1109/TAC.1968.1098903.

[101] Kenan Šehić et al. "LassoBench: A High-Dimensional Hyperparameter Optimization Benchmark Suite for Lasso". 10 pages, 10 figures. 2021. URL: https://hal.science/hal-03562043.

[102] Pengyuan Shen and Huilong Wang. "Archetype building energy modeling approaches and applications: A review". In: *Renewable and Sustainable Energy Reviews* 199 (2024), p. 114478. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2024.114478. URL: https://www.sciencedirect.com/science/article/pii/S1364032124002016.

[103] V. Siddharth et al. "Automatic generation of energy conservation measures in buildings using genetic algorithms". In: *Energy and Buildings* 43.10 (2011), pp. 2718–2726. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2011.06.028. URL: https://www.sciencedirect.com/science/article/pii/S0378778811002775.

[104] Wei-Lee Woon Sokratis Papadopoulos Elie Azar and Constantine E. Kontokosta. "Evaluation of tree-based ensemble learning algorithms for building energy performance estimation". In: *Journal of Building Performance Simulation* 11.3 (2018), pp. 322–332. DOI: 10.1080/19401493.2017.1354919. eprint: https://doi.org/10.1080/19401493.2017.1354919. URL: https://doi.org/10.1080/19401493.2017.1354919.

[105] Nivethitha Somu, Gauthama Raman M R, and Krithi Ramamritham. "A deep learning framework for building energy consumption forecast". In: *Renewable and Sustainable Energy Reviews* 137 (2021), p. 110591. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2020.110591. URL: https://www.sciencedirect.com/science/article/pii/S1364032120308753.

[106] Keartisak Sriprateep et al. "Heterogeneous ensemble machine learning to predict the asiaticoside concentration in centella asiatica urban". In: *Intelligent Systems with Applications* 21 (2024), p. 200319. ISSN: 2667-3053. DOI: https://

doi.org/10.1016/j.iswa.2023.200319. URL: https://www.sciencedirect.com/science/article/pii/S2667305323001448.

[107] Lukas G. Swan and V. Ismet Ugursal. "Modeling of end-use energy consumption in the residential sector: A review of modeling techniques". In: *Renewable and Sustainable Energy Reviews* 13.8 (2009), pp. 1819–1835. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2008.09.033. URL: https://www.sciencedirect.com/science/article/pii/S1364032108001949.

[108] Abebe Tilahun Tadie, Zhizhong Guo, and Ying Xu. "Hybrid Model-Based BESS Sizing and Control for Wind Energy Ramp Rate Control". In: *Energies* 15.23 (2022). ISSN: 1996-1073. DOI: 10.3390/en15239244. URL: https://www.mdpi.com/1996-1073/15/23/9244.

[109] Giovanni Tardioli et al. "Data Driven Approaches for Prediction of Building Energy Consumption at Urban Level". In: *Energy Procedia* 78 (2015). 6th International Building Physics Conference, IBPC 2015, pp. 3378–3383. ISSN: 1876-6102. DOI: https://doi.org/10.1016/j.egypro.2015.11.754. URL: https://www.sciencedirect.com/science/article/pii/S1876610215024868.

[110] ULSS 2 Marca Trevigiana. *DELIBERAZIONE DEL DIRETTORE GENERALE n.1348 del 28/06/2024*. 2024. URL: https://www.aulss2.veneto.it/mys/apridoc/iddoc/7561.

[111] Athanasios Tsanas and Angeliki Xifara. "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools". In: *Energy and Buildings* 49 (2012), pp. 560–567. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2012.03.003. URL: https://www.sciencedirect.com/science/article/pii/S037877881200151X.

[112] Geoffrey K.F. Tso and Kelvin K.W. Yau. "Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks". In: *Energy* 32.9 (2007), pp. 1761–1768. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2006.11.010. URL: https://www.sciencedirect.com/science/article/pii/S0360544206003288.

[113] Ramu Vankudoth, P Shireesha, and T Devi. "A Model of System Software Components Using Genetic Algorithm and Techniques". In: 6 (Oct. 2016).

[114] Vladimir Vapnik, Steven Golowich, and Alex Smola. "Support Vector Method for Function Approximation, Regression Estimation and Signal Processing". In: *Advances in Neural Information Processing Systems*. Ed. by M.C. Mozer, M. Jordan, and T. Petsche. Vol. 9. MIT Press, 1996, pp. 281–287. URL: https://proceedings.neurips.cc/paper/1996/file/4f284803bd0966cc24fa8683a34afc6e-Paper.pdf.

[115] Vladimir Naumovich Vapnik. *The Nature of Statistical Learning Theory, Second Edition*. Statistics for Engineering and Information Science. Springer, 2000. ISBN: 978-0-387-98780-4.

[116] Fazli Wahid and DoHyeun Kim. "A Prediction Approach for Demand Analysis of Energy Consumption Using K-Nearest Neighbor in Residential Buildings". In: *International Journal of Smart Home* 10 (Feb. 2016), pp. 97–108. DOI: 10.14257/ijsh.2016.10.2.10.

[117] Yongjie Wang et al. "Physics-guided LSTM model for heat load prediction of buildings". In: *Energy and Buildings* 294 (2023), p. 113169. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2023.113169. URL: https://www.sciencedirect.com/science/article/pii/S0378778823003997.

[118] Yuanyuan Wang et al. "Application of residual modification approach in seasonal ARIMA for electricity demand forecasting: A case study of China". In: *Energy Policy* 48 (2012). Special Section: Frontiers of Sustainability, pp. 284–294. ISSN: 0301-4215. DOI: https://doi.org/10.1016/j.enpol.2012.05.026. URL: https://www.sciencedirect.com/science/article/pii/S0301421512004387.

[119] Zeyu Wang and Ravi S. Srinivasan. "A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models". In: *Renewable and Sustainable Energy Reviews* 75 (2017), pp. 796–808. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2016.10.079. URL: https://www.sciencedirect.com/science/article/pii/S1364032116307420.

[120] Zeyu Wang et al. "Random Forest based hourly building energy prediction". In: *Energy and Buildings* 171 (2018), pp. 11–25. ISSN: 0378-7788. DOI: https:

//doi.org/10.1016/j.enbuild.2018.04.008. URL: https://www.sciencedirect.com/science/article/pii/S0378778818311290.

[121] B. Wauman, H. Breesch, and D. Saelens. "Evaluation of the accuracy of the implementation of dynamic effects in the quasi steady-state calculation method for school buildings". In: *Energy and Buildings* 65 (2013), pp. 173–184. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2013.05.046. URL: https://www.sciencedirect.com/science/article/pii/S0378778813003356.

[122] Matthieu "Weber and Ferrante" Neri. ""Contiguous Binomial Crossover in Differential Evolution"". In: *"Swarm and Evolutionary Computation"*. Ed. by Leszek "Rutkowski et al. "Berlin, Heidelberg": "Springer Berlin Heidelberg", "2012", "145–153". ISBN: "978-3-642-29353-5".

[123] Yixuan Wei et al. "A review of data-driven approaches for prediction and classification of building energy consumption". In: *Renewable and Sustainable Energy Reviews* 82 (2018), pp. 1027–1047. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2017.09.108. URL: https://www.sciencedirect.com/science/article/pii/S136403211731362X.

[124] Darrell Whitley. "A Genetic Algorithm Tutorial". In: *Statistics and Computing* 4 (Oct. 1998). DOI: 10.1007/BF00175354.

[125] Wojciech "Wieczorek and Zbigniew J." Czech. ""Selection Schemes in Evolutionary Algorithms"". In: *"Intelligent Information Systems 2002"*. Ed. by Mieczysław A. "Kłopotek, Sławomir T. Wierzchoń, and Maciej" Michalewicz. "Heidelberg": "Physica-Verlag HD", "2002", "185–194".

[126] Bill Wilson. *The Machine Learning Dictionary*. 2012. URL: http://www.cse.unsw.edu.au/~billw/mldict.html#activnfn (visited on 12/20/2019).

[127] P.H. Winston. *Artificial Intelligence*. Addison-Wesley series in computer science. Addison-Wesley, 1984. ISBN: 9780201082593. URL: https://books.google.it/books?id=VGYJqiAz6_sC.

[128] David H. Wolpert. "Stacked generalization". In: *Neural Networks* 5.2 (1992), pp. 241–259. ISSN: 0893-6080. DOI: https://doi.org/10.1016/S0893-

6080(05)80023-1. URL: https://www.sciencedirect.com/science/article/pii/S0893608005800231.

[129] Li Xuemei et al. "Building Cooling Load Forecasting Model Based on LS-SVM". In: *2009 Asia-Pacific Conference on Information Processing*. Vol. 1. 2009, pp. 55–58. DOI: 10.1109/APCIP.2009.22.

[130] T. Yalcinoz and U. Eminoglu. "Short term and medium term power distribution load forecasting by neural networks". In: *Energy Conversion and Management* 46.9 (2005), pp. 1393–1405. ISSN: 0196-8904. DOI: https://doi.org/10.1016/j.enconman.2004.07.005. URL: https://www.sciencedirect.com/science/article/pii/S019689040400192X.

[131] Roman Yampolskiy. "Why We Do Not Evolve Software? Analysis of Evolutionary Algorithms". In: *Evolutionary Bioinformatics* 14 (Dec. 2018), p. 117693431881590. DOI: 10.1177/1176934318815906.

[132] Junjing Yang, Mattheos Santamouris, and Siew Eang Lee. "Review of occupancy sensing systems and occupancy modeling methodologies for the application in institutional buildings". In: *Energy and Buildings* 121 (2016), pp. 344–349. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2015.12.019. URL: https://www.sciencedirect.com/science/article/pii/S0378778815304540.

[133] Shuangming Yang, Bernabe Linares-Barranco, and Badong Chen. "Heterogeneous ensemble-based spike-driven few-shot online learning". In: *Frontiers in neuroscience* 16 (2022), p. 850932.

[134] Zhun Yu et al. "A decision tree method for building energy demand modeling". In: *Energy and Buildings* 42.10 (2010), pp. 1637–1646. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2010.04.006. URL: https://www.sciencedirect.com/science/article/pii/S0378778810001350.

[135] Fan Zhang et al. "Time series forecasting for building energy consumption using weighted Support Vector Regression with differential evolution optimization technique". In: *Energy and Buildings* 126 (2016), pp. 94–103. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2016.05.028. URL: https://www.sciencedirect.com/science/article/pii/S0378778816303899.

[136] Hai xiang Zhao and Frédéric Magoulès. "A review on the prediction of building energy consumption". In: *Renewable and Sustainable Energy Reviews* 16.6 (2012), pp. 3586–3592. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2012.02.049. URL: https://www.sciencedirect.com/science/article/pii/S1364032112001438.

[137] Hamed Zibaei and Saadi Mesgari. "Improved discrete particle swarm optimization using Bee Algorithm and multi-parent crossover method (Case study: Allocation problem and benchmark functions)". In: (Mar. 2024). DOI: 10.21203/rs.3.rs-4143898/v1.