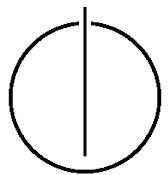


FAKULTÄT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

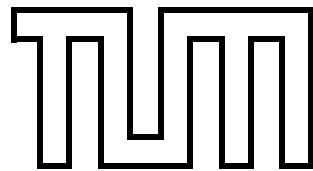
Master's Thesis in Informatics

**Thermal Comfort in Smart
Buildings: Development of a
Personalized Machine
Learning-Based Temperature
Control System**

Patrick Ruoff



b



FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Thermal Comfort in Smart Buildings:
Development of a Personalized Machine
Learning-Based Temperature Control System

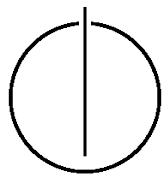
Thermischer Komfort in Smart Buildings:
Entwicklung eines Individualisierten
Machine-Learning-Basierten
Temperaturkontrollsystems

Author: Patrick Ruoff

Supervisor: Prof. Dr. Bernd Brügge

Advisor: Nadine von Frankenberg und Ludwigsdorff, M.Sc.

Date: April 15, 2020



d

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, April 15, 2020

Patrick Ruoff

Acknowledgments

First and foremost, I want to thank my supervisor Professor Bernd Brügge, Professor Vivian Loftness, and Nadine von Frankenberg und Ludwigsdorff for making this master's thesis at the Carnegie Mellon University possible. I consider the last seven months as an extremely insightful time on both an academic and a personal level.

Also, I want to express my honest gratitude for my advisor Nadine von Frankenberg und Ludwigsdorff, whose help and attention to detail were crucial for the quality of this thesis.

Additionally, I am thankful to the German Academic Exchange Service for supporting my stay in the United States financially with the "DAAD IFI" scholarship. I also want to thank Professor Bernd Brügge, Monika Markl, and Nadine von Frankenberg und Ludwigsdorff for their help with the application.

Further, I want to express my gratitude to infraNOMIC and Jochen Gerwig for providing three infrared heating panels along with valuable information about their functionality. In the same way, I am thankful for Professor Brügge's Chair for Applied Software Engineering, which provided three iPhones for this project. Also, I want to thank the Leibniz Supercomputing Centre for giving me access to their cloud service for extensive preprocessing, fitting, and training.

I want to especially thank Till and Leon, who were my most delightful comrades and roommates in the past months and who - along with all the other friends I made at the Intelligent Workplace and CMU - let me enjoy my time in Pittsburgh even more. Also, my thanks go to Paul Schmiedmayer and Sebastian Liedl for their help and work, which I could build on. Last but not least, I want to thank Lena for proofreading and the Global Communication Center for their advice on my academic writing style.

Abstract

Thermal comfort significantly influences a building occupant's performance, well-being, and satisfaction. Various studies have shown that thermal comfort is perceived differently among individuals and depends on diverse factors, such as weather, indoor temperature, and an occupant's metabolism. In practice, the individual differences in thermal comfort are not addressed by current heating, ventilation, and air conditioning (HVAC) systems, which often rely on predefined schedules or set-points to regulate a building's temperature. In this thesis, we design and implement a personalized machine learning-based temperature control system, LATEST (Learning-based Automated Thermal Environment control SysTem). Its central objective is to address each occupant's individual thermal preferences in an automated process. To achieve this, the system integrates thermal actuators for task environments and builds on the approach of conventional HVAC systems, which pre-eminently use indoor temperature and humidity measurements, and extends them by biosignal, outdoor, and temperature control behavior data. These data are processed by state-of-the-art machine learning models to infer an individual occupant's temperature control behavior. The automation is made transparent by providing reasoning for the machine learning model's decision-making.

To evaluate this approach, a field study was conducted at the Robert L. Preger Intelligent Workplace at Carnegie Mellon University in Pittsburgh, Pennsylvania. In particular, the automated system's performance in terms of occupant satisfaction was evaluated against the performance of manual control by the occupants themselves. We found that LATEST can reduce an occupant's command frequency by 79% while increasing thermal comfort to 89%. This study lays a framework for continued work in this area to make automated personalized task thermal controls a reality.

Zusammenfassung

Der thermische Komfort beeinflusst die Leistungsfähigkeit, das Wohlbefinden und die Zufriedenheit von Bewohnern eines Gebäudes erheblich. Eine Vielzahl von Studien haben gezeigt, dass thermischer Komfort von Individuen unterschiedlich wahrgenommen wird und von verschiedenen Faktoren, wie Wetter, Raumtemperatur, und dem Stoffwechsel des Bewohners abhängt. In der Praxis werden die individuellen Unterschiede des thermischen Komforts von gewöhnlichen Heizungs-, Lüftungs- und Klimaanlagen nicht berücksichtigt, da hauptsächlich Zeitpläne oder Sollwerte zur Regelung der Temperatur verwendet werden. In dieser Arbeit entwerfen und implementieren wir ein individualisiertes machine-learning-basiertes Temperaturkontrollsystem, LATEST (Learning-based Automated Thermal Environment control SysTem). Dessen zentrales Ziel ist es, die individuellen thermischen Präferenzen jedes Bewohners in einem automatisierten Prozess zu berücksichtigen. Um dies zu erreichen, integriert das System thermische Aktuatoren für lokale Umgebungen und baut auf dem Ansatz herkömmlicher Systeme auf, die vor allem Raumtemperatur- und Raumluftfeuchtemessungen verwenden, und erweitert diesen um Biosignal-, Außen- und Temperaturregelverhaltensdaten. Diese Daten werden durch modernste Machine-Learning Modelle verarbeitet, um das Temperaturregelverhalten eines einzelnen Bewohners vorherzusagen. Die Automatisierung wird transparent gemacht, indem Begründungen für die Entscheidungsfindung des Machine-Learning Modells gegeben werden.

Um den Ansatz zu evaluieren, wurde eine Feldstudie am Robert L. Preger Intelligent Workplace der Carnegie Mellon University in Pittsburgh, Pennsylvania durchgeführt. Insbesondere wurde die Leistung des automatisierten Systems in Bezug auf die Zufriedenheit der Teilnehmer im Vergleich zur Leistung der manuellen Steuerung durch die Teilnehmer selbst evaluiert. Diese Studie hat gezeigt, dass LATEST die Befehlsfrequenz eines Teilnehmers um 79% reduzieren und gleichzeitig dessen thermischen Komfort auf 89% erhöhen kann. Somit wurde eine Grundlage für weitere Arbeit in diesem Feld geschaffen, um automatisierte personalisierte thermische Steuerung Realität werden zu lassen.

Contents

1	Introduction	2
1.1	Problem	2
1.2	Motivation	5
1.3	Objectives	7
1.4	Outline	8
2	Background	9
2.1	Thermal Comfort	9
2.1.1	Predicted Mean Vote Model	10
2.1.2	Thermal Comfort Factors	11
2.2	Infrared Heating	11
2.3	Machine Learning	12
2.4	Data Preprocessing	14
2.5	Model Selection	18
2.5.1	Performance Metrics	18
2.5.2	Cross-Validation	19
2.6	Machine Learning Interpretability	21
2.6.1	Overview	21
2.6.2	LIME	22
2.6.3	Shapley Values	23
3	Related Work	25
3.1	Machine Learning-Based Thermal Comfort Models	25
3.1.1	Predictive Personal Vote Model	26
3.1.2	Bayesian Network on Air Temperature	27
3.1.3	Linear Regression on Skin Temperature	27
3.1.4	Support Vector Machine	28
3.1.5	Multiple Models on Actuator Control Behavior	28
3.1.6	Neural Networks on Body Shape	29
3.2	Gap in Research	30

4 Requirements Analysis	33
4.1 Overview	33
4.2 Current System	34
4.3 Proposed System	34
4.3.1 Functional Requirements	34
4.3.2 Nonfunctional Requirements	36
4.4 System Models	38
4.4.1 Scenarios	39
4.4.2 Use Case Diagram	42
4.4.3 Activity Diagrams	47
4.4.4 Analysis Object Model	49
5 System Design	53
5.1 Overview	53
5.2 Design Goals	55
5.3 Subsystem Decomposition	57
5.4 Hardware Software Mapping	59
5.5 Persistent Data Management	64
5.6 Access Control	66
6 Object Design	67
6.1 Overview	67
6.2 Data Preprocessing	68
6.3 Model Selection	70
6.3.1 Interval-Stratified K-Fold Cross-Validation	71
6.3.2 Pre-Selection	74
6.3.3 Optimization	76
6.3.4 Model Evaluation	78
7 Case Study	79
7.1 Objectives	79
7.2 Data Collection Phase & Model Selection	80
7.2.1 Design	81
7.2.2 Results	86
7.3 Temperature Control Phase	91
7.3.1 Design	91
7.3.2 Results	93
7.4 Findings	96
7.5 Discussion	97
7.6 Limitations	100

8 Summary	101
8.1 Status	101
8.1.1 Realized Goals	102
8.1.2 Open Goals	103
8.2 Conclusion	104
8.3 Future Work	104
A Case Study	106
A.1 Reminder Sheet	106
A.2 Key Indicators	107
A.3 Phase Transition	108
A.3.1 Randomized Grid Search Performances	108
A.3.2 Optimized Parameters	108
A.4 Final Survey	109

AI Artificial Intelligence

CMU Carnegie Mellon University

FCNN Fully Connected Neural Network

FR Functional Requirement

GBDT Gradient Boosting Decision Trees

GNB Gaussian Naive Bayes Classifier

GSR Galvanic Skin Response

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

HVAC Heating Ventilation and Air-Conditioning

kNN k-Nearest Neighbors Classifier

LIME Local Interpretable Model-Agnostic Explanations

LR Logistic Regression

ML Machine Learning

MLI Machine Learning Interpretability

MRT Mean Radiant Temperature

MSB Microsoft Band 2

NFR Nonfunctional Requirement

PMV Predicted Mean Vote

RF Random Forest

SHAP Shapley Additive Explanations

SVM Support Vector Machine

TCF Thermal Comfort Factors

UML Unified Modeling Language

Chapter 1

Introduction

This chapter introduces the topics of indoor temperature control and automation. The research is motivated by the problem that current indoor temperature control systems are energy-inefficient and do not meet the occupants' thermal comfort needs. The research proposes a new type of temperature control model based on radiant heating panels.

1.1 Problem

The U.S. Energy Information Administration found in their 2015 survey of over 5600 households that space heating accounts to 39% of the total residential energy consumption in North America¹. In commercial buildings, the percentage is 25%². That is the largest share of energy end-use categories. The energy used for indoor temperature control contributes to global climate change, which threatens to cause irreversible global warming. Hence, immediate energy savings are required to prevent crucial harm to the environment and people. Since the amount of energy consumed by indoor temperature control is so large, improvements in efficiency can contribute significantly to the global challenge of climate protection.

A large share of energy consumption is due to meeting comfort for many individuals at the same time. The literature defines comfort as Indoor Environmental Quality (IEQ), which is divided into four factors: thermal comfort, visual comfort, acoustic comfort, and air quality. The most important factor

¹U.S. Energy Information Administration, “Residential Energy Consumption Survey”, access February 24, 2020. www.eia.gov/consumption/residential/data/2015

²U.S. Energy Information Administration, “Commercial Buildings Energy Consumption Survey”, access February 24, 2020. www.eia.gov/consumption/commercial/data/2012

1.1. PROBLEM

of IEQ perceived by occupants is thermal comfort [FW11]. The American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) defines thermal comfort as “that condition of mind that expresses satisfaction with the thermal environment” [ASH13]. IEQ [SS03, LHL⁺06, Par14] and, in particular, thermal comfort [Fan70, dDBD97, Fis02, SFL05] were shown to have a significant influence on health and performance. Despite the mentioned consumption of energy for heating, a 2018 study with nearly 53000 participants found that 43% of occupants in commercial buildings in North America are thermally dissatisfied [HAZA06]. This poses a crucial issue considering that we spend most of our time in enclosed building environments [KNOR01].

Multiple factors influence thermal comfort, and this has been studied in the literature since the 1970s [Fan70]. However, most current heating, ventilation and air-conditioning (HVAC) systems still only allow occupants to control one factor – air temperature [HAZA06]. Other factors influencing thermal comfort include air velocity, clothing insulation, metabolism, activity level [Fan73], or skin temperature [Cho10, CL12]. The body of factors influencing thermal comfort is called thermal comfort factors (TCF) in this thesis. Figure 1.1 depicts some TCF and separates them into two categories: environmental and human factors. On the left of the figure, a conventional HVAC system is modeled. It influences the indoor air temperature only while the other TCF are left ignored.

Though the variety of TCF has been documented for 50 years, technology still does not address these factors. Today’s most widespread HVAC systems rely exclusively on indoor temperature with static schedule-based regulation. This control scheme offers no consideration of the individual occupant’s thermal needs, which causes significant thermal discomfort, waste of energy, and can cause conflicts between occupants [HLZ⁺09]. Other factors on thermal comfort, such as radiant temperature or skin temperature, could be included in the control strategy by conventional sensors.

Though, even systems that try to adapt heating automation to the individual’s needs intelligently face new problems. The Nest Learning Thermostat³ is an intelligent thermostat that learns the occupant’s thermal comfort and adjusts the heating system accordingly. Nevertheless, occupants addressed various flaws. For instance, some intents behind the occupant’s regulation were not understood correctly by the intelligent thermostat, which resulted in poor thermal regulation. Additionally, the occupant’s inability to understand how Nest decides how to act was perceived negatively [YN13].

³Google Store, “Nest Learning Thermostat”, access April 9, 2020.
store.google.com/us/product/nest_learning_thermostat_3rd_gen

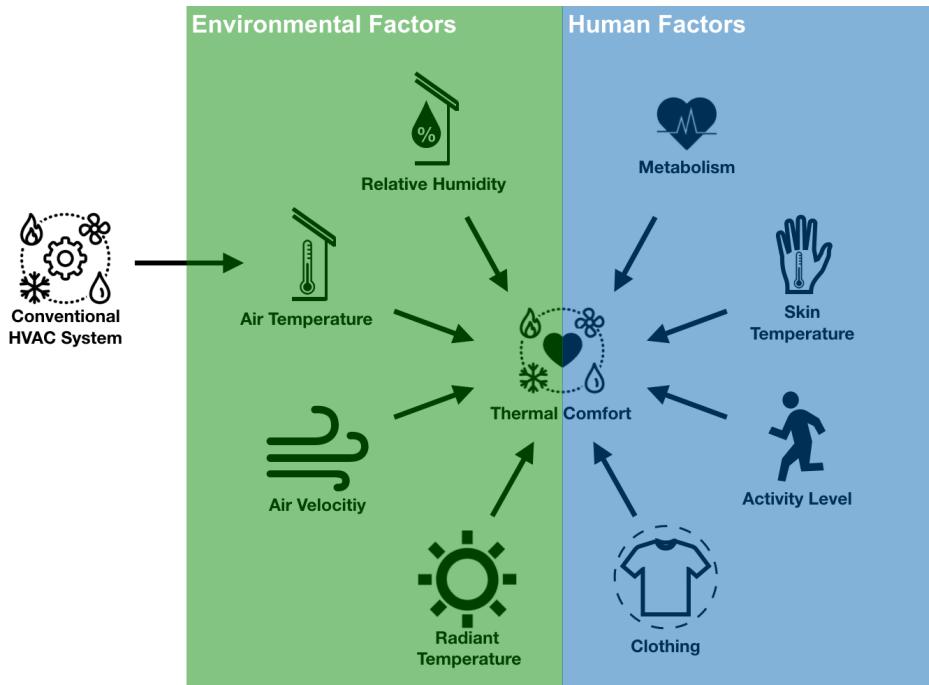


Figure 1.1: Several thermal comfort factors (TCF) grouped into environmental factors and human factors. Conventional HVAC systems exclusively rely on air temperature.

Furthermore, traditional thermostats are controlled centrally without providing control options for individual occupants. In contrast, Loftness et al. argue that means for including individuals into decision-making are necessary [LAC⁺09]. Accordingly, providing occupants with the possibility to control their thermal environment improves their experience and comfort [Fro12]. Yet, the lack of control possibilities for individual occupants is in large part justified by the limitation of conventional HVAC controls, which affect the ambient⁴ environment as a whole. Hence, for conventional HVAC systems, multiple individuals are affected by a change in the control strategy at once. This can lead to broad dissatisfaction and performance loss [CK19]. Besides that, in many cases, these systems have only binary modes of control. They are either switched on or off, while a dimming regulation might be more fitting to reach long-lasting comfort. Other actuators on IEQ, such as fans,

⁴In the literature, the surrounding space is referred to as *ambient* environment that is defined through a floor of a building or a closed space. Generally, there is more than one occupant in an ambient environment. In contrast, *task* environment is the term referred to an occupant's personal space only. For instance, a task light could be a desk lamp while an ambient light would light up a whole floor.

radiant heating panels, window controls, or shading devices are not integrated into conventional HVAC systems.

Additionally, traditional buildings do not implement today's sensing technology although integrating more TCF into HVACs would be possible [Zha19]. Sources of valuable information about the current TCF include solar sensors, skin temperature sensors, heart rate sensors, or human sensors, who can be surveyed via smartphone applications or other interfaces [WBE⁺16, EC12].

1.2 Motivation

The central motivation driving this thesis is the opportunity to enhance the IEQ control experience. This is made possible to the largest part due to inventions in two fields.

First, personalized thermal actuators, such as radiant heaters, are what Hammer and Champy name *disruptive technology* [HC93]. The authors' *old rule* (the issue transformed by the *disruptive technology*) is that an ambient environment can only be affected as a whole. The resulting *new rule* is given by the new ability to control task environments separated from each other. This makes it possible to give every individual occupant control of their personal task environment, ruling out conflicts and enhancing the temperature control experience. Importantly, this also enables reducing the ambient temperature in a shared building while regulating the task environments for every occupant according to their individual needs for saving significant amounts of energy. Figure 1.2 shows two scenes where infrared heating panels are used to heat the task environment of the occupants directly.

Second, the advancements in artificial intelligence (AI) in recent years make it possible to predict more and more complex problems from our lives. Successes include the AI classifying images [CGGS12], processing natural language [CWB⁺11], and outperforming humans playing the board game Go [SSS⁺17]. Previously, these domains were tried to be processed by rule-based approaches to only limited success. Similarly, in thermal comfort research, the latest machine learning methods outperform previous knowledge-based calculations, which promises to cause widespread adjustments in the future [FPWL15, DZAL17, KZS⁺18].

The advancements in both fields make it possible to accurately predict an individual's thermal comfort and hence, automate the temperature control intelligently and individually. With automation comes the occupant's desire to know on which basis their environment is manipulated [YN13]. Recent achievements in machine learning interpretability make it possible to



Figure 1.2: Infrared radiant heating panels by infraNOMIC. On the left picture, a panel with a landscape on its surface is mounted at the wall behind a bed. On the right picture, a panel with a mirror surface is mounted at the ceiling of a massage room. In both cases, the thermal energy is directed to the places where the occupants are.

gain knowledge about the internal decision-making of many machine learning methods [RSG16, LL17]. Hence, these techniques can make machine learning-based systems more transparent to the occupant for building trust in their decisions.

These recent developments in enhancing the temperature control process are further supported by achievements in sensor technology and smart wearables. The latter enable convenient monitoring of human biosignal data. In this work, biosignals are referred to signals from the biosystem of a human body. Wearables, such as smart watches or e-textiles can have integrated sensors that are in contact with the human body and can, therefore, collect biosignal data that were inconvenient to measure before [vF17, Zha19]. Examples are heart rate or blood oxygen levels [KHP⁺18] that can be measured by state-of-the-art smart watches, such as the fitbit versa ²⁵.

Additionally, the concept of smart buildings developed over the last four decades. Smart buildings integrate various networking devices and aim to use energy efficiently. Their main components are sensors, actuators, controllers, a network, an interface, and a central unit for supervising the other components in the system [MLK11]. Hence, smart buildings provide a platform for integrating various sensors and actuators into HVAC systems.

⁵fitbit, “versa 2”, access February 26, 2020.
www.fitbit.com/us/products/smartwatches/versa

1.3 Objectives

For solving the problems from the first section we propose **LATEST**, a Learning based Automated Thermal Environment control SysTem. Its central goal is to enhance the occupant's thermal comfort and temperature control experience while indirectly saving energy. LATEST is personalized to every individual occupant such that it uses the latest machine learning methods to automate temperature control for each occupant individually. It uses local thermal actuators to control every occupant's thermal environment individually. LATEST enhances the occupant's temperature control experience in four ways:

- **Provide Control:** The individual occupant is in control of task thermal actuators via a simple, mobile interface.
- **Intelligent Automation:** LATEST provides automated temperature control that is adapted to an individual's specific thermal comfort needs via machine learning. For that cause, it uses biosignal sensors and a variety of environmental sensors to derive knowledge about the occupant's current thermal comfort.
- **Provide Reasoning:** While controlling the temperature automatically, LATEST provides a reasoning for the occupant to ensure transparency of its decision-making process.
- **Meet Thermal Comfort Needs:** With the semi-automated control of task thermal actuators, LATEST meets the thermal comfort needs of the individual.

The fulfillment of the defined goals is evaluated in a case study. We provide every occupant with a mobile interface connected to one task thermal actuator, a smart wrist band with biosignal sensors and various task and ambient environmental sensors. In the first phase, the occupant and the sensors are monitored while the occupant controls their thermal actuator through the interface. Therefore, data about the occupant's personal temperature control behavior are collected. Then, various machine learning methods are fitted to the data and are compared via interval-stratified k-fold cross-validation and optimized by randomized parameter search techniques using the F1-Score [Chi92]. Hence, an individual model for every occupant will be selected. In a second phase, these personal models control the occupants' thermal actuators automatically. An occupant can interrupt this automated temperature control via the interface and also give feedback about

their thermal comfort to evaluate the performance of the models. At the same time, machine learning interpretability methods provide the occupant with a reasoning for every decision made by the models. A final occupant survey shows how the proposed system's reasonings are perceived.

LATEST is designed, implemented, and evaluated at the Robert L. Preger Intelligent Workplace (IW) of Carnegie Mellon University (CMU) in Pittsburgh, Pennsylvania, USA, and is demonstrated at the Technical University of Munich, Germany. The IW is a smart building living laboratory where a large number of smart building innovations are developed and implemented [HLM⁺97].

1.4 Outline

This thesis covers eight chapters. This introductory chapter gives an overview of the thesis' topics, existing problems, possible solutions, the research objectives and this outline. Chapter 2 provides more information about the fundamental concepts used in this thesis. It contains sections about thermal comfort, machine learning, data preprocessing, model selection and machine learning interpretability. Related work is discussed in Chapter 3. There, a variety of related studies are discussed and a gap in the literature is elaborated, which should be filled by this thesis. Chapter 4 follows the requirements elicitation and analysis templates by Brügge and Dutoit [BD09]. It defines the requirements of the LATEST system and introduces a correct, complete, consistent, and verifiable representation of the system [BD09]. The System Design of the envisioned personalized temperature control system is detailed in Chapter 5, which follows the system design document template by Brügge and Dutoit [BD09]. There, the design goals of the envisioned system are worked out and LATEST is decomposed into smaller subsystems. Chapter 6 covers detailed explanations of the data preprocessing and the model selection process applied in LATEST. This thesis' approach was evaluated in a case study described in Chapter 7. The sections included describe the case study's objectives and the setups and results of its different phases. Additionally, the case study's findings and limitations are discussed. Chapter 8 closes this thesis by providing a summary of the work performed as part of this thesis and gives suggestions for possible future work.

Chapter 2

Background

This chapter covers important background knowledge that is built upon in the rest of this thesis. Section 2.1 covers research of thermal comfort with a focus on Fanger’s Predictive Mean Vote model and thermal comfort factors (TCF). Fundamentals of machine learning (ML) are described in Section 2.3, with details on various ML methods. How data are preprocessed for these ML methods is explained in Section 2.4 and Section 2.5 shows, how ML models are compared. The final section of this chapter introduces the concept of machine learning interpretability and details two of its implementations.

2.1 Thermal Comfort

As explained in the last chapter, ASHRAE defines thermal comfort in their Standard-55 as “that condition of mind that expresses satisfaction with the thermal environment” [ASH13]. Fanger’s research on thermal comfort showed in 1970 that thermal comfort depends on various environmental and physiological factors and differs between individuals [Fan70,Fan73]. His Predictive Mean Vote (PMV) model for predicting thermal sensation is adopted by the ASHRAE 55-Standard. Its output is measured on the ASHRAE 7-point thermal sensation scale, shown in Table 2.1. In the literature, thermal comfort is distinguished from thermal sensation in that a person can feel thermally comfortable while experiencing cool or hot thermal sensation.

In the following sections, the PMV is explained in more detail and findings of TCF research are detailed.

Vote	Comfort Level
+3	hot
+2	warm
+1	slightly warm
0	neutral
-1	slightly cool
-2	cool
-3	cold

Table 2.1: The 7-point thermal sensation scale of the PMV, which is adopted by the ASHRAE Standard-55 [ASH13].

2.1.1 Predicted Mean Vote Model

Thermal comfort was subject to thorough research since Fanger introduced the Predicted Mean Vote model (PMV) in 1970 [Fan70]. The PMV is based on a body of data sets collected in lab environment studies. The model compares several TCF against these data sets and outputs a vote on the ASHRAE 7-point thermal sensation scale. ASHRAE adopted the PMV to meet its guidelines that at least 80% of the occupants in a building should be thermally comfortable [ASH13]. Hence, the PMV is a holistic model aiming to achieve a level of thermal comfort for a group of people.

Considering the thermal environment, the PMV takes into account the TCF air temperature, mean radiant temperature, relative humidity, and air velocity [Fan73]. Mean radiant temperature is the average temperature radiated by the objects surrounding a person. For instance, when moving from the shadow into direct sunlight, an occupant's mean radiant temperature increases, which in turn influences their thermal comfort. On a personal level, the TCF that are considered by the PMV are an occupant's activity level, which is the internal heat production in the body, and the thermal resistance of their clothing, also known as clothing insulation [Fan73].

Even though the PMV is widely adopted, it has various drawbacks [EC12, DTN10]. Since the PMV is based on an existing body of data sets, it is not capable to include new factors when they are found to influence thermal comfort. Several TCF, which are not included in the PMV, are described in the following section. Additionally, the personal TCF included by the PMV, activity level and clothing insulation, were found to be difficult to measure in practice [GK13b].

More novel alternatives to the holistic PMV are personal thermal comfort models. A holistic model aims to predict thermal comfort for all individ-

uals according to certain selected TCF. Their output could be interpreted as personalized according to the input of a combination of personal TCF, such as age, weight, body shape, or gender. In contrast, personal thermal comfort models are specifically designed for and adapted to a particular individual only. In this approach, the model parameters and even the model structure vary for different occupants. Personal thermal comfort models are discussed in detail in the next chapter.

2.1.2 Thermal Comfort Factors

Since Fanger's definition of the PMV in 1970, more factors were found to influence thermal comfort. Figure 1.1 depicts some TCF and separates them into two categories: environmental and human factors. Environmental factors affect every individual in the environment, while human factors affect every human individually.

Air temperature, radiant temperature, relative humidity and air velocity are the most common environmental TCF that are integrated into HVAC systems. However, there are more environmental factors, which influence thermal comfort. For instance, Loftness et al. found in an experiment that outdoor conditions influenced occupants' thermal comfort, despite the same indoor conditions [ZLJ⁺12]. Hence, several recent studies included various outdoor factors into their thermal comfort predictions, such as outdoor air temperature, outdoor relative humidity, cloudiness, or precipitation [KZS⁺18, FQvF⁺19].

Even more human TCF were found and integrated in thermal comfort predictions. As mentioned in the previous section, biosignals from the human body can conveniently be measured and integrated into thermal comfort models. Such biosignals, which were shown to influence thermal comfort, include skin temperature [Cho10, SKJ⁺16, Zha19, FQvF⁺19], skin moisture [vF17], body shape [FQvF⁺19], heart rate [Cho10, vF17, FQvF⁺19], and Galvanic Skin Response (GSR) [GSH67]. Another human TCF is gender, which was discussed in detail by Chang and Kajackaite [CK19].

2.2 Infrared Heating

In this thesis, infrared heating panels are integrated into the temperature control system. Infrared heating is a form of radiant heat transfer, which is not bound to matter and, unlike heat conduction and convection, also occurs in a vacuum. Thermal radiation is that part of the spectrum of electromagnetic radiation that every body emits as a function of its temperature as

soon as it differs from the absolute zero of 0 Kelvin. The most well-known kind of radiant heat is solar radiation, which can be divided into Ultraviolet radiation, visible light and infrared radiation [Kos09].

Conventional heating systems rely on convection heating. Convection is a form of heat transfer based on the transport of particles carrying thermal energy [Kos09]. In the case of water heating, which is most frequently used in conventional heating systems, water is the heat transport medium by convection in a closed pipe circuit between the burner and the inside of the radiators. By conduction, the heat energy passes from the inside to the outside of the radiator. On the outside of the radiator the heat transport medium is air. There, the second kind of convection, called “free convection”, occurs, while the air expands due to heating and moves upwards, while cooler air flows in from below.

Infrared heaters differ from convection heaters, since they do not require matter to transport heat but directly heat an irradiated occupant. In contrast, convection heaters warm up water, air, or both before the thermal energy reaches the occupant. Hence, radiant heating is a better fit to control task environments without affecting the ambient environment.

2.3 Machine Learning

In the last two decades, the advancements in AI and machine learning (ML) coupled with the continuing increase in available computing power, made it possible to predict more and more complex problems from all kinds of domains. Successes include the AI classifying images [CGGS12], processing natural language [CWB⁺11], and outperforming humans in several games [SSS⁺17, VEB⁺17]. Previously, these domains were tried to be processed by rule-based approaches to only limited success. The key difference with supervised ML models is that they are not programmed to work according to explicit instructions but learn patterns of a domain themselves. This is achieved via “training” an ML model on sample data to make predictions about performing a certain task. Hence, every project using an ML model needs to build up training data about the task and train the ML model on them.

There are many supervised ML methods. The ones that are applied in this study are described here.

FCNN Fully Connected Neural Networks are collections of connected nodes, which on their own perform simple mathematical transformations to the input data. These nodes are grouped in layers and their connections are called edges. A neural network is fully

connected, if all nodes of one layer have an edge to every node of the previous and the following layer, which is also referred to as Multi-layered Perceptron in the literature. Variations of neural networks are the central methods behind the successes of AI in recent years [SSS⁺17, CWB⁺11, CGGS12]. Generally, FCNNs can represent any arbitrary complex continuous function under only mild assumptions [LM04]. However, they are considered resource-intensive and require large amounts of data compared to other ML methods.

- GNB** Gaussian Naive Bayes Classifier assumes independence for all input features and a normally distributed likelihood, while both do not hold in general. Nevertheless, Bayesian methods perform accurately and can efficiently handle large data sets [Bis06]. They use maximum likelihood to estimate mean and variance for every feature.
- GBDT** Gradient Boosting Decision Trees combine smaller decision trees into a single predictive model in an iterative process. Over the iterations, each decision tree is designed to correct the residual error of its predecessor [MBBF99].
- kNN** The k-Nearest Neighbors Classifier is an ML method, which, provided an integer k and distance function $d(x, y)$, labels a data point x by the majority class of the k data points that are closest to x in terms of $d(x, y)$. Optionally, the majority is influenced by a weighted distance measure.
- LR** Logistic Regression uses a logistic function to model the probability of a specific class. The idea behind LR is to construct a predictor function that emits a score from a linear combination of weights using a dot product [Bis06].
- RF** Random Forest is an ensemble classifier that produces mean predictions of many decision trees constructed from random subsets of the data set. Using an ensemble of models makes RF less prone to overfitting. It can be used for both classification and regression tasks.
- SVM** Support Vector Machine has a broad range of implementations in human behavior prediction [CV95]. It transforms training examples to a higher-dimensional space with a kernel function and builds a linear model. This model is then used to classify new data by comparing data points distance to a decision boundary of support vectors that was fitted before.

2.4 Data Preprocessing

Before raw sensor measurements and data from actuators are used for training an ML model, they are preprocessed to comply with predefined requirements and to have a format that is suited for the respective ML method. These measurement types and all other generated types, which are used as inputs for ML methods, are called *features*.

In data preprocessing, several adjustments and extensions are applied to the data to improve the expected performance of the ML models. This section describes these adjustments and extensions.

Outlier Handling

Sensors are prone to have outliers in their measurements. While a certain measurement error cannot be ruled out entirely, an outlier represents a strong deviation from the monitored condition. To handle outliers, a moving median filter can be applied to the sensor data. This filter is applied over a series of data points by considering a fixed number of consecutive data points at a time, called a window. The fixed number is called window size. The moving median filter returns the median of the data points in the window. Therefore, the moving median filter is suited to deal with outliers, because an outlier will always have the smallest or largest value in any window, assuming there are not multiple outliers in the same window. Then, the window is moved one data point forward in the time-series and the filter is applied again. This effect of replacing outliers by less extreme values is called smoothing. The window size of the moving median filter can have two unwanted effects. First, if the window size is too small and the window contains a majority of outliers, the median of the window might be an outlier, diminishing the smoothing effect of the filter. Second, if the window size is too large, the filter will replace extreme, non-outlier measurements by less extreme measurements, introducing an attenuation bias [Sto96]. Consequently, the sensor data would have reduced variance, not representing the monitored environmental state.

Figure 2.1 shows a diagram of two series of temperature values over 25 minutes. While the green series, “OriginalTemperature”, has significant outliers in three intervals of time, the yellow series stays almost constant over the full period with only small deviations at two points in time, where the green series shows the most significant outliers. The yellow series, “SmoothedTemperature”, resulted from applying a moving median filter with a window size of 5 on the green series. “OriginalTemperature” has two time intervals where one out of six time points does not deviate from the constant level. “SmoothedTemperature” also has one outlier in each of these time intervals,

2.4. DATA PREPROCESSING

showing that a sequence of outliers cannot always be smoothed fully by a moving median filter with a window size less than double the length of the sequence of outliers.

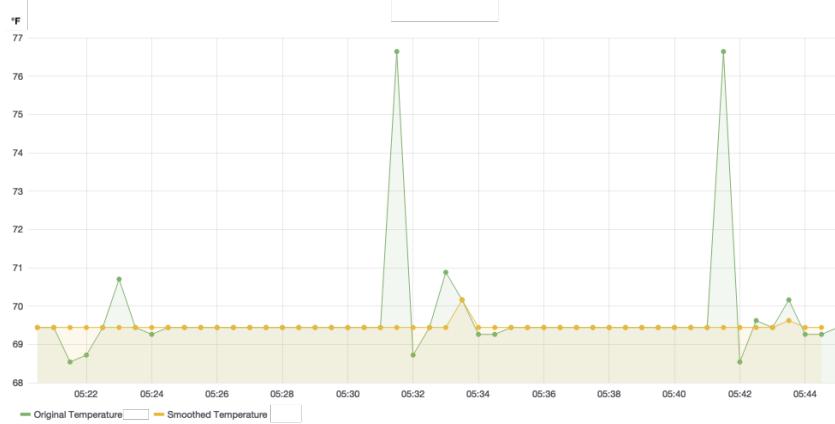


Figure 2.1: A diagram showing two series of temperature data. The series “OriginalTemperature”, colored in green, stays at a constant level of about 20.8°F over the full period of 25 minutes apart from a few outliers, up to 24.7°F . The second series, “SmoothedTemperature” colored in yellow, stays almost constant at the same level, with only two comparably insignificant outliers of less than 0.5°F .

Time Step Fitting

Most ML methods require having exactly one value of every feature for each data point. Therefore, before training an ML model with time-series data of many features with differing frequencies of data points, a data format must be set, which maps all available data points to certain time steps. In most cases, these time steps are set to follow a certain frequency. For instance, when predicting monthly stock trends, there could be one data point each day when the markets are closed. Another scenario would be a thermal actuator, which is controlled every 30 seconds. In this case, each data point is fitted to the preceding point in time, with either a 0 seconds mark or a 30 seconds mark. As a result, the input for the ML models can be taken from every of the specified time steps. Figure 2.2 illustrates this step by showing two diagrams.

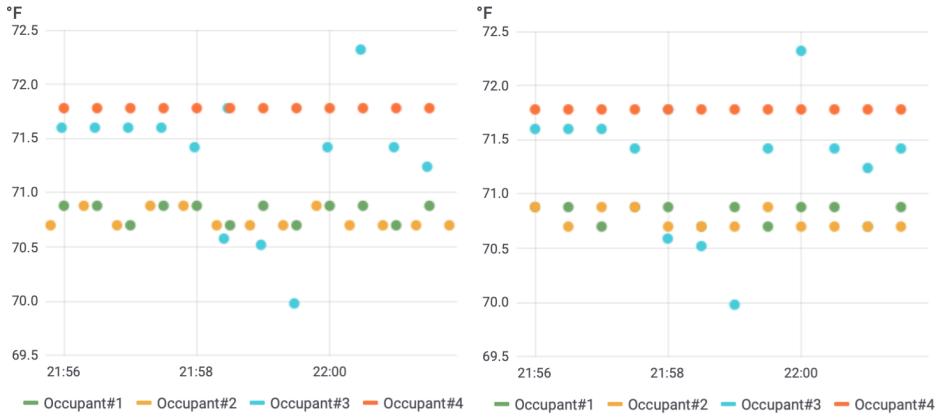


Figure 2.2: Two diagrams, each displaying temperature data for the task environments of four occupants over a time interval of about 6 minutes. The diagram on the left shows data points spread over the time interval with no obvious time format. The diagram on the right shows the same data but the data points were aligned to fit intervals of 30 seconds.

One-Hot Encoding

Categorical features cannot be processed by ML methods. For instance, a feature `TrafficLight`, which represents the state of a traffic light, might have the categories “red”, “yellow”, and “green”. These categories must be made numerical, such that an ML method can process them. The most popular method to implement this is one-hot encoding [PPP17].

One-hot encoding creates a numerical feature for every category that the categorical feature can represent. For instance, the three previously mentioned categories would be represented as three features, namely `TrafficLightRed`, `TrafficLightYellow`, and `TrafficLightGreen`. These generated features’ values in a certain time step are 1, if the categorical feature’s value is their corresponding category, and 0 otherwise. Hence, if at one time step the `TrafficLight` is “green”, only `TrafficLightGreen` is 1 and the other values are 0. It follows that for every data point, exactly one of the generated features for a categorical feature has value 1 while the rest have value 0.

Trend Feature Generation

Most popular ML methods are invariant to the order of their input data points. They base their predictions solely on the current input values and have no built-in concept of time and order. For this reason, order-aware methods, such as long short-term memory neural networks, are superior in

2.4. DATA PREPROCESSING

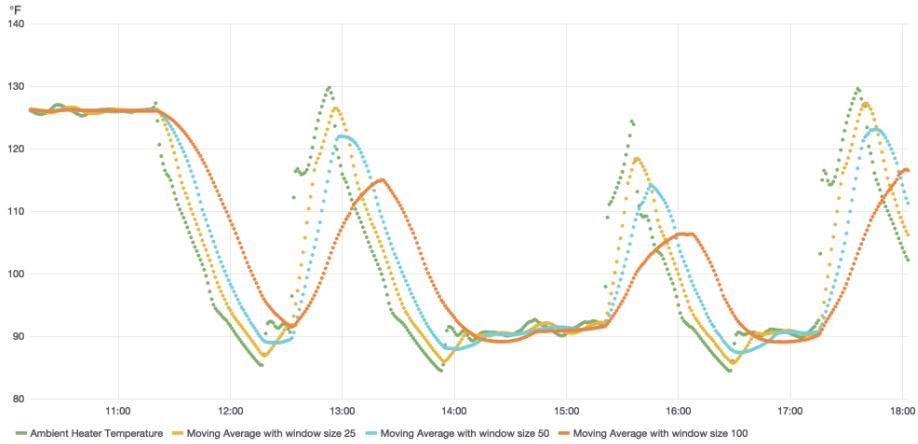


Figure 2.3: A diagram of temperature data of an ambient heater and the corresponding moving averages for window sizes 25, 50, and 100.

tasks in which the order of the inputs is of high importance, such as natural language processing. Similarly, in the domain of thermal comfort, a person might feel differently at a certain ambient temperature level when it just got colder, as opposed to a situation where the temperature just increased to that same level. However, for trend-dependent time-series data, one data point also depends on the trend that lead to this data point. To use this property, features can be generated that include values of preceding data points in the calculation of their values. In this work, these kinds of features are referred to as *trend features*.

One popular example of trend features is the moving average. Its generation is equivalent to the moving median filter explained above, with the only difference being the application of an average instead of a median. Hence, the moving average also has a smoothing effect while it is not as robust to outliers, since outliers influence the average value. Figure 2.3 shows temperature data of an ambient heater and the corresponding moving averages for window sizes 25, 50, and 100. It can be observed that the higher the window size, the stronger the smoothing effect. Other trend features and their application in time-series data can be found in [PSTK15].

Every feature that is generated for training a model, for instance by one-hot encoding or by trend feature generation, must also be generated when this model is deployed.

Data Standardization

All ML methods that are distance-based require standardization of their inputs. Standardization means transforming the data set on a feature-basis to have a mean value of 0 and a variance of 1. Computing a single standardized feature value x' is achieved by subtracting a feature's mean \bar{x} from its value x and dividing the result by the feature's standard deviation σ .

$$x' = \frac{x - \bar{x}}{\sigma}$$

Distance-based ML methods include LR, kNN, SVM, Neural Networks, and clustering algorithms like k-Means Clustering, many of which were introduced in the previous section. In contrast, all tree-based ML methods, which are based on inequalities instead of distance metrics, are invariant to standardization. These include RF and GBDT. Additionally, ML methods, which rely on probability distributions of the features, such as NB, are also invariant to standardization.

2.5 Model Selection

The work with ML models requires means for comparing models. For this comparison, performance metrics and cross-validation are widely used in practice [Bis06]. The following subsections explain these concepts in more detail.

2.5.1 Performance Metrics

This section is limited on metrics for binary classification, since these will be used in the rest of this thesis. For more general explanations, we refer to [Bis06]. The performance metrics for binary classification introduced in this section are defined by the terminology of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negative (FN). These terms are defined according to the total number of data points, which are predicted correctly (= true) or incorrectly (= false), and according to the class they were predicted to belong to, while in this binary case, one class is called positive and the other class is called negative. For instance, for a heater with states “ON” and “OFF”, one could define “ON” as the positive class and “OFF” as the negative class. A data point, which is labeled as “ON” because the heater was turned on at that moment and was classified as “OFF” by a temperature control model, belongs to the False Negatives, since it was incorrectly classified as the negative class.

The most widely applied performance metric is *accuracy*. It is calculated by dividing the number of correctly classified data points, TP and TN, by the number of all data points.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

The use of the popular performance metric accuracy for comparing ML models introduces unwanted effects. For instance, when assessing an ML model with the accuracy metric on a data set where 95% of the data points are of the same target value, the model might learn to return this value for any input. In that case, the accuracy would be 95%, which is generally considered a high value. However, no data point with a different target value than the predominant one is classified correctly. The model learned nothing about the underlying nature of the domain, which would be its desired behavior [Bis06].

To face this issue, other performance metrics can be used. One popular choice for binary and possibly unbalanced data is the f1-score [Chi92, FS10]. The f1-score is the harmonic mean of *precision* and *recall*. *Precision* and *recall* are defined in terms of TP, FP and FN.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Their harmonic mean, the f1-score, is defined as:

$$\text{f1-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

and takes a value in the range from 0 to 1. Since precision and recall have the same weight in the f1-score, misclassifying all data points of one target value, even in a highly unbalanced data set, results in a significantly decreasing f1-score.

2.5.2 Cross-Validation

For comparing the performance of various ML methods on a data set while avoiding overfitting, cross-validation techniques are applied. They are a tool for assessing how the results of an ML model will generalize to an independent data set. Therefore, they split the data set in training and validation sets. The training set is used to train an ML model and the evaluation set is used to assess the performance of the resulting model. For this assessment, all cross-validation techniques require a performance metric to be applied to the target variable. In the case of a temperature control behavior model for a thermal

actuator with on/off control modes, the target variable is the temperature control behavior and its values are “ON” and “OFF”. Since there are only two values, it is called a binary target variable. There are various kinds of cross-validation techniques [RPL10]. Two of them are explained in the following.

K-Fold Cross-Validation

K-fold cross-validation is the most widespread cross-validation technique [RPL10]. When applying k-fold cross-validation to an ML method for a specified integer k , the data set is split into k parts of equal size, called groups. Then, k ML models are trained, while one training is referred to as a fold. In each fold, one group is left out from the data set for training and instead, is used to validate the resulting model’s performance on a given metric. Every group is the validation group exactly once. Typically, the average of the k performance measures is then used to compare the model with other models or of other ML methods or the same ML method with other parameters.

The use of k-fold cross-validation also decreases the risk of overfitting a method’s parameters to the data, since the training and evaluation data change with every fold.

Stratified K-Fold Cross-Validation

For some applications, it is desirable to ensure the same target variable distribution in the training set and the evaluation set. Especially for unbalanced data, if the groups for k-fold cross-validation are chosen at random, it can happen to produce a group that lacks one representation of the target variable. Consequently, this group would be unfit to validate an ML model on that unbalanced data set. A variant of k-fold cross-validation that addresses this issue is stratified k-fold cross-validation. The difference is that when splitting the data set into k groups, this split is conducted such that all the groups have both equal size and the same distribution of the target variable’s values. Hence, for providing equal distributions of target variable’s values across the groups, they are selected such that each one has the same amount of data points for every possible value that the target variable can have. For instance, if the target variable for a heating panel has two states, “ON” and “OFF”, every group will have the same amount of data points with the target variable being “ON” and being “OFF” as every other set, or one more or one less if equality is not possible.

2.6 Machine Learning Interpretability

In recent years, machine learning models were shown to potentially introduce encoding bias, which is responsible for discriminating individuals, strengthening preconceptions, or mistakenly taking random correlation for causation. Racist chatbots [dLSB17] and hackable face detection [SBBR16] are examples of malicious outcomes of learning algorithms.

This unwanted behavior creates a need for ML-based systems to rule out certain biases, such as discrimination [FR14] or danger for people [VA16, AOS⁺16]. However, these objectives cannot be quantified completely unlike the objective of task performance, which most ML models are optimized for. For instance, developers might not be able to produce all tests needed for the safe operation of an autonomous vehicle or to understand all influences that make a credit scoring application discriminate. Yet, apart from the obvious reasons why this is not desired, the latter example is explicitly ruled out by the European Union’s Recital 71 of the General Data Protection Regulation, which is also known as the “Right to Explanation” [Com16].

Therefore, the field of machine learning interpretability (MLI) experienced increasing attention in recent years. In the literature, the term “explainable artificial intelligence” is also widely used for the same field.

2.6.1 Overview

The difficulty in interpreting most ML models lies in their “opaque” nature. Humans cannot interpret how most ML models come up with their decisions. In that sense, Miller defines interpretability as “the degree to which an observer can understand the cause of a decision” [Mil19].

Since the field of ML interpretability is still relatively young, the literature defines various taxonomies. A common taxonomy is listed here:

- **Local Interpretability:** Single decisions can be interpreted. A reasoning can be given, how a certain ML model’s output was predicted according to the corresponding inputs.
- **Global Interpretability:** For global interpretability, an ML model’s decision-making process must be fully understood. Approaches include creating surrogate models that are interpretable copies of the original model and are adapted while training. However, since the surrogates might have differing predictions for certain inputs, global interpretability is only approximated.

- **Model-Specific Interpretability:** MLI methods that only work for certain models are called model-specific. For instance, there is a large number of research projects introducing methods to interpret deep neural networks [BCR97, EBCV09, MV14, YCN⁺15, NYC16]. Specifically, Yosinski et al. propose new means to visualize the activations in every layer of recurrent and convolutional neural networks. They show how intuition about the decision-making of these neural networks can be gained [YCN⁺15]. Another example is given by SVMs. While fitting on the input data, they select a small number of data points, called support vectors, which consequently define the decision boundary of their target value. Hence, studying these support vectors provides a certain degree of interpretability.
- **Model-Agnostic Interpretability:** Model-agnostic methods of MLI are not limited to a certain ML method, like model-specific methods, but work for a range of ML methods. They are of special interest in projects that work with multiple ML methods for the same tasks.

Two popular model-agnostic methods for local interpretability were introduced by the group of Carlos Guestrin at the University of Washington. In 2016, they proposed Local Interpretable Model-Agnostic Explanations (LIME) [RSG16] and SHapley Additive exPlanations (SHAP) in 2020 [LL17]. Both methods apply to any classifier and are detailed in the following.

2.6.2 LIME

LIME’s approach for local interpretability for any ML classification method involves building a linear model around a single prediction of an ML model. Hence, LIME assumes linearity for the space of this prediction, which does not hold in general. As explained in [RSG16], LIME permutes the inputs to one prediction for a specified number of times and lets the ML model predict the outcome to all these permutations. Then, LIME calculates a similarity score between the original prediction and the permutations’ predictions and extracts the features, whose permutations best fit the original model’s prediction. In LIME, the resulting features are then used to fit an interpretable linear model on the permuted data. The resulting model’s feature weights are further compared to return a reasoning, which features have the strongest influence on the original prediction. The influence can be both for the predicted outcome or against it. Figure 2.4 from the University of Washington’s

website¹ shows an example of their Python implementation of LIME, called “lime”². The figure shows the result of applying LIME on an ML model that predicts whether an e-mail is written by an atheist or Christian from the “20 Newsgroups”³ data set. It can be divided into three parts: from left to right are the prediction probabilities, the feature importances, and a visualization of the most important features, which are highlighted in the input e-mail’s context. The feature with the strongest influence is “Posting”. Only after the interpretation with LIME, it was found that this word happens to appear in 21.6% of the training data while only two of them relate to the class atheism. One can further reason if it is understood from a human perspective, why the features “Host”, “NNTP”, or “edu” are also strong indicators for this class.

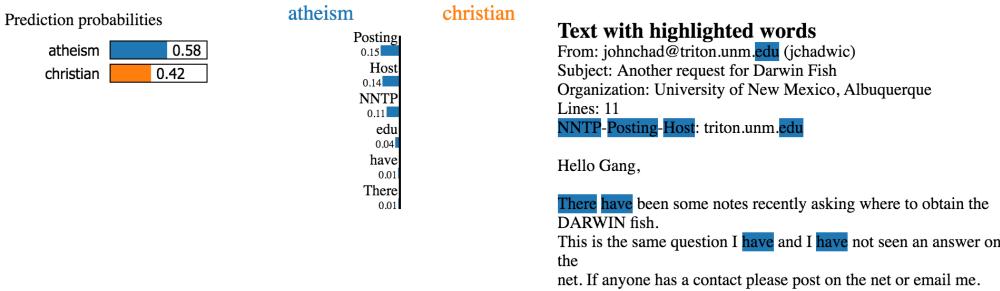


Figure 2.4: The output of applying LIME on an ML model that predicts whether an e-mail’s author is an atheist or Christian.

2.6.3 Shapley Values

SHapley Additive exPlanations (SHAP) are based on LIME and extend the approach by Shapley values and coalitional game theory [LL17, Sha53]. In contrast to LIME, SHAP does not have the local linearity assumption, making its results generally more accurate. SHAP implements the idea that a prediction can be seen as a payout in a game while the features are the players. Shapley values, coined by Lloyd Shapley, are then used to distribute this payout among the players [Sha53]. The detailed explanation of how coalitional game theory is combined with LIME can be found in [LL17]. The

¹University of Washington Computer Science & Engineering, “LIME – Local Interpretable Model-Agnostic”, access March 8, 2020. homes.cs.washington.edu/~marcotcr/blog/lime/

²GitHub, “lime”, access March 8, 2020. github.com/marcotcr/lime

³Apache2 Server, “20 Newsgroups”, access March 8, 2020. qwone.com/~jason/20Newsgroups/

interpretation of SHAP’s result for one feature is how much this feature contributed to the prediction compared to the average of all predictions for the training data.

Figure 2.5 shows the result of applying SHAP on the prediction of a tree-based ML model on the popular “Boston Housing”⁴ data set with the Python module provided by Lundberg et al.⁵ The figure shows which features have the most significant influence on the prediction of the ML model, visually “pushing” the prediction from the base value of 22.34 to 24.41. The strongest feature is “LSTAT”, representing the “lower status of the population”, which seems to have a relatively high value since it makes the housing price increase. On the contrary side to LSTAT, the strongest influences are RM, the average number of rooms per dwelling, and NOX, the nitrogen oxides concentration. Even though these two features have the second and third strongest influences and “push” the prediction downwards, the output still differs from the base in the opposite direction.

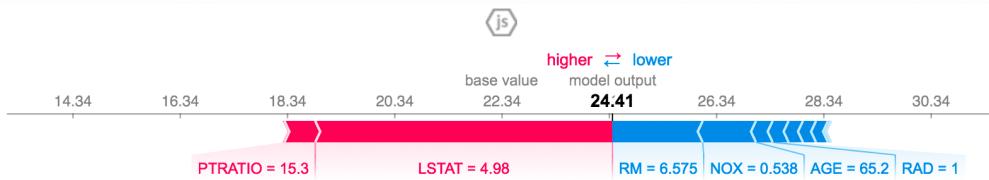


Figure 2.5: The output of applying SHAP on a tree-based ML model that predicts the average housing price in the city of Boston.

A key limitation of both LIME and SHAP is that for interpreting a single prediction, they need to run a multitude of evaluations of the original model, which consumes a relatively high amount of computing power. This is addressed by the Tree SHAP method, which is an optimized method for computing Shapley values for tree-based models and decreases the runtime significantly [LEC⁺20].

⁴Kaggle, “Boston Housing”, access March 8, 2020. www.kaggle.com/c/boston-housing

⁵GitHub, “SHAP”, access March 8, 2020. github.com/slundberg/shap

Chapter 3

Related Work

Personal thermal comfort models are subject to ongoing research since major limitations of the PMV were found. Recent approaches can be divided into various concepts, which are worked out and assessed in this chapter.

Personal thermal comfort models share four general concepts [KZS⁺18]:

- Revolves analysis around an individual occupant.
- Use direct feedback and personal data for adapting the model.
- Prioritize cost-efficient and non-intrusive data collection.
- Implement a data-driven approach to provide flexible testing of diverse modeling methods and potential explanatory variables

From the last bullet point follows that personal thermal comfort models are based on the individual's data. Further, if personal thermal comfort models replace a holistic model, there needs to be one personal model deployed for every occupant in parallel. Recent studies covered a variety of machine learning methods to compare, which methods are best fit to model personal thermal comfort. In this section, several of these studies from the last seven years are described and discussed.

3.1 Machine Learning-Based Thermal Comfort Models

In recent years, rule-based systems in many domains were replaced by machine learning-based approaches [CWB⁺11, CGGS12, SSS⁺17]. The same accounts for the PMV, which was shown to perform worse than state-of-the-art ML methods in many studies [GTBG15, SKJ⁺16, KZS⁺18, FQvF⁺19].

This section covers several studies using various kinds of ML methods and describes the findings most relevant for this project.

3.1.1 Predictive Personal Vote Model

SPOT is a Smart Personalized Office Thermal control system developed at the University of Waterloo [GK13b]. The authors integrated sensors to monitor occupancy and six TCF, namely air temperature, mean radiant temperature (MRT), relative humidity, air velocity, metabolic rate, and clothing insulation, to infer an individual's thermal sensation on the 7-point ASHRAE scale [ASH13]. Figure 3.1 shows how they connected a microcontroller to a Microsoft Kinect¹, an infrared sensor and a laser to infer occupancy, clothing insulation, and metabolic rate.

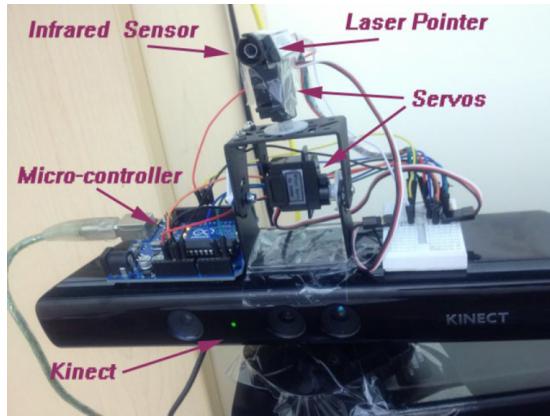


Figure 3.1: The setup proposed by Gao and Keshav featuring a microcontroller, infrared sensor, laser pointer and two servomechanisms [GK13b]. These sensors' measurements are used to predict occupancy, clothing insulation, and metabolic rate.

The authors extend the holistic PMV model by adding a personal part, which results in the Predicted Personal Vote (PPV) model. This personal part is a least-square regression on an individual occupant's preferences for the six TCFs of the model. SPOT directly controls a radiant heater to react on the PPV's prediction of the occupant's current thermal sensation. In a follow-up study named SPOT+, the authors extended the approach by two predictive components: a kNN for predicting future occupancy and a learning-based model predictive control (LBMPC) to predict future temperature levels [GK13a]. Both approaches were evaluated in a field study with

¹Microsoft, “Azure Kinect DK”, access April 3, 2020. www.microsoft.com/en-us/p/azure-kinect-dk/8pp5vxmd9nhq

3.1. MACHINE LEARNING-BASED THERMAL COMFORT MODELS

one participant for SPOT and one of the authors for SPOT+. These studies provided evidence that SPOT can keep an occupant’s thermal sensation level, while SPOT+ can reduce the average thermal discomfort to 0.02 on the 7-point ASHRAE thermal sensation scale. This work is of particular interest, for the use of occupancy detection, task thermal actuators, and control automation, which are important steps for improving the occupant’s temperature control experience. Though their models were deployed to control a thermal actuator in a field study, which many other projects lack, the sample sizes of one participant per case study decrease the informative value of their findings.

3.1.2 Bayesian Network on Air Temperature

Ghahramani et al. propose another approach for modeling personal thermal comfort models using an online learning approach for personalized thermal comfort using a Bayesian Optimal Classifier and Adaptive Stochastic Modeling [GTBG15]. Therefore, the authors model each comfort condition on a 3-point thermal comfort scale (“uncomfortably warm”, “comfortable”, “uncomfortably cold”) by a probability distribution and combine them into a Bayesian Network (BN). This network depends exclusively on the TCF air temperature and relative humidity, while the latter did not prove to have a significant influence on the result. The authors argue that their approach is capable of modeling the change of thermal comfort preferences over time, compared to other approaches. In a field study with 33 subjects, the authors compared their proposed model’s performance to kNN, LR, Decision Tree Classifier, SVM, and PMV when provided 10 data points per subject. Their model scored an average accuracy of 70.1% – the highest score in the study. We assume that their probabilistic approach takes advantage of the small training data size and a more data-intensive study with more TCF covered might show a different result with even higher accuracy.

3.1.3 Linear Regression on Skin Temperature

Sim et al. studied how skin temperature of varying body locations affect thermal comfort by fitting and comparing multiple linear regression models [SKJ⁺16]. These model’s inputs are skin temperature measurements from various body locations and air temperature measurements. In a lab experiment, the authors collected data from 8 test subjects with thermal sensation feedback on a 9-point scale. These data were used to train both holistic and personal thermal sensation models, while the best personal model scored an RMSE of 0.95, which is 0.31 points below the RMSE of the best holistic

model. These findings give further proof of the superiority of personalized models over holistic models. Further, the authors introduced the air temperature gradient, a trend-dependent feature (see Section 2.4), into the model, which proved to increase model performance. Additionally, the authors found that the wrist skin temperature provides the most predictive information on thermal sensation. However, just like [GTBG15], this model also relies on a small selection of TCF only and is still to be deployed and evaluated in a temperature control study.

3.1.4 Support Vector Machine

Jiang and Yao proposed an SVM-approach to model personal thermal comfort [JY16]. In their lab study, they collected thermal sensation votes on the 7-point scale from 20 test subjects, while monitoring four TCF, namely air temperature, mean radiant temperature, relative humidity and air velocity. They found, despite the limited number of covered TCF, the SVM predicts thermal sensation with an average accuracy of 89.9%, outperforming the probabilistic approach from [GTBG15] without making use of trend-dependency. Hence, SVM promises to be well fit for modeling personal thermal comfort.

In the following, two studies are described that made use of this finding by training several ML models with similar properties as SVM (such as Random Forest, Logistic Regression, or Neural Networks) for every occupant and selecting the best performing model.

3.1.5 Multiple Models on Actuator Control Behavior

Kim et al. proposed a novel approach for personal thermal comfort by including the occupant’s control of a “Personal Comfort System” (PCS) chair, displayed in Figure 3.2, as input feature to their ML models [KZS⁺18]. These models are RF, SVM, LR, GBDT, Conditional Inference Tree² (CTree), and a Gaussian Process Classifier³ (GPC) and were trained on a wide range of TCF, including indoor environment, outdoor environment, ambient HVAC activity, PCS control behavior, and one-hot encoded categorical features from survey data. In particular, the authors also included the trend-dependent features (Section 2.4), such as the gradient of the air temperature by hour and the weighted mean of outdoor air temperature by month.

²The Comprehensive R Archive Network, “ctree: Conditional Inference Trees”, access April 3, 2020. cran.r-project.org/web/packages/partykit/vignettes/ctree.pdf

³R Package Documentation, “gausspr: Gaussian processes for regression and classification”, access April 3, 2020. rdrr.io/cran/kernlab/man/gausspr.html

3.1. MACHINE LEARNING-BASED THERMAL COMFORT MODELS



Figure 3.2: The PCS chair developed by the Center for the Built Environment at the University of California, Berkeley [KZS⁺18]. It offers heating and cooling of the occupant’s task environment.

The named models were trained to predict thermal comfort⁴ on a 3-point scale. In a field study with 34 occupants, the authors find that models, which are robust to higher-dimensional data with noise, such as RF, SVM, or LR, rank highest in performance on their data sets, while every model’s performance varies between participants. In particular, they show that these models converge only after training with about 60 survey responses, with three responses per participant per day. Further, they find that models, which use all their monitored TCF and other collected data perform best with a median accuracy of 73% across all test subjects. We conclude that including thermal actuator control behavior into a thermal comfort model is very reasonable, since it is a direct thermal related feedback by the occupant. However, an even more sophisticated approach would predict this thermal actuator control behavior for automation.

From the finding that models, which are robust for higher-dimensional inputs, perform better on larger, higher-dimensional thermal comfort data set, one can conclude that neural networks are a promising choice for future thermal comfort models. This step was conducted in a very recent study detailed in the next section.

3.1.6 Neural Networks on Body Shape

Francis et al. conducted an extensive study in a lab experiment with 77 test subjects [FQvF⁺19]. Their “OccuTherm” system uses a Microsoft Kinect to infer body shape information, such as shoulder circumference, height, and weight. Furthermore, the authors monitored indoor air temperature, outdoor air temperature, outdoor relative humidity, skin temperature, clothing

⁴The authors relate “thermal preference” to the same scale that as is more commonly called thermal comfort scale (“cooler”/“no change”/“warmer”)

insulation, and gender to train various ML models. GSR and activity data were also collected but did not provide valuable information and were not considered for training. As the target variable, thermal comfort was surveyed on a 5-point scale with a mobile application. The authors found that FCNN and RF are the best performing models and benefit from using all available features, supporting the findings from [KZS⁺18]. Hence, body shape proved to be a thermal comfort factor. Furthermore, the authors showed again that holistic models have inferior predictive capabilities to personal models.

3.2 Gap in Research

From the related work and the discussed findings several gaps in research can be worked out, which will be addressed in this thesis.

Several studies identified feedback collection techniques as one obstacle for collecting sufficiently much data. As discussed in the last section, there is a trend towards collecting larger data sets and applying more robust ML methods. As Kim et al. suggested, for every occupant, data for at least 20 days should be gathered before an ML model converges, while occupants tend to forget or decide not to fill out extensive surveys, since it interrupts their current activity and consumes time [EC12, KZS⁺18, FQvF⁺19]. Consequently, researchers aim to reduce the friction related to feedback collection. Francis et al. reason their use of a “reduced 5-point ASHRAE 55 scale (...) in order to reduce the complexity of voting.” Furthermore, Kim et al. suggest that “individuals’ heating and cooling behavior with PCS is a strong comfort predictor and can potentially replace survey feedback as the ground truth for personal comfort models.” Hence, in this study, we will use the occupant’s thermal actuator control behavior for collecting feedback and predicting this behavior to control the same thermal actuator. This results in an approach to collect thermal comfort feedback with little friction and training personal thermal comfort models for temperature control.

Additionally, Francis et al. discuss their choice of the f1-score as metric for evaluating their thermal comfort models, since misclassifying conditions, which are related to label “warmer”, with label “colder”, is worse than misclassifying such conditions with the label “no change”. This difference in weight is not addressed in the f1-score, which treats each misclassifying between the three classes the same. Hence, we propose to introduce thermal preference feedback that can be translated to thermal actuator control states. In particular, we suggest the use of a binary control actuator, so there are only two labels, “ON” and “OFF”, which can accurately be evaluated by the f1-score, even for unbalanced data sets. Therefore, friction of data col-

3.2. GAP IN RESEARCH

lection can be reduced, while training a temperature control model that can be evaluated accurately.

Further, only one of the described projects, conducted by Rabbani and Keshav, evaluated the integration of their personal comfort model in thermal actuator control [GK13b, GK13a]. However, their case studies were limited to one participant per work and they even participated themselves. To fill this gap, we will conduct a case study, to directly evaluate our models' influence on the occupants' change in comfort conditions, while our models automatically control the thermal actuators.

Systems, which try to adapt temperature control automation to the individual's needs by using ML, face new problems. Nest is an intelligent thermostat that learns the occupant's thermal comfort and adjusts an HVAC system accordingly. Nevertheless, occupants addressed various flaws. For instance, some intents behind the occupant's regulation were not understood correctly by the intelligent thermostat which resulted in poor thermal regulation. Additionally, the occupant's inability to understand how Nest decides how to act was perceived negatively [YN13]. Therefore, we propose the application of recent MLI methods, introduced in Section 2.6, to provide a reasoning for how the system works to the occupant.

The findings by Kim et al. and Francis et al. suggest comparing various robust ML methods for every occupant's collected data and deploy the best performing one. Also, both studies found that the more environmental and human TCF are integrated into the model, the better it performs [KZS⁺18, FQvF⁺19]. Additionally, the findings of Patel et al. about the use of trend-dependent features for ML models on time-series data, were shown to apply to the domain of thermal comfort by Ghahramani et al. and Kim et al. [PSTK15, GTBG15, KZS⁺18] Hence, we train several ML methods on a wide range of environmental and human TCF data with trend-dependent features to predict personal thermal actuator control behavior. At the same time, we focus on collecting possibly large data sets for every occupant by using non-intrusive surveys and sensors, such as monitoring temperature control or wearables, as was successfully implemented by various studies [SKJ⁺16, KZS⁺18, FQvF⁺19].

Table 3.1 shows our approach compared to the described related work by integrated TCF, target variable, modeling method, use of trend-dependency, use of interpretability, case study scope, and model evaluation.

Source	Thermal Comfort Factors	Target Variable	Modeling Method	Trend-dependency	MLI	Case Study	Model Evaluation
[GK13b]	AT, MRT, RH, AV, MR, CI	TS 7-point	PPV	N	N	field, 1 subject	N/A
[GK13a]	AT, MRT, RH, AV, MR, CI	TS 7-point	PPV, kNN, LBMPC	N	N	field, 1 subject	N/A
[GTBG15]	AT, RH	TC 3-point	BN, kNN, LR, DT, SVM, PMV	trend-aware model	N	field, 33 subjects	70.1% accuracy
[SKJ ⁺ 16]	ST	TS 9-point	Linear Regression	trend feature	N	lab, 8 subjects	0.95 RMSE
[JY16]	AT, RH, MRT, AV	TS 7-point	SVM	N	N	lab, 20 subjects	89.8% accuracy
[KZS ⁺ 18]	AT, CI, OAT, Cl, Pr, TAT, TRH, AHVAC, TOD, DOW, PCS control	TC 3-point	RF, SVM, LR, GBDT, CTree, GPC	trend features	N	field, 34 subjects	73% median accuracy
[FQvF ⁺ 19]	SC, He, We, TA, OTA, ORH, ST, CI, gender	TC 5-point	FCNN, RF, kNN, SVM, NB, PPV, PMV	N	N	lab, 77 subjects	FCNN: 0.56 RMSE
this research	AT, RH, CO ₂ , OAT, ORH, Cl, WC, AP, WS, ST, GSR, HR, AHVAC, TAT, TRH, CAT, CRH	TP 2-point	FCNN, RF, kNN, SVM, NB, GBDT	trend features	Y	field, 3 subjects	0.97 f1-score

Table 3.1: The concept matrix comparing related work with this research. All abbreviations used are given in Table 3.2.

Abbreviation	Term	Abbreviation	Term	Abbreviation	Term	Abbreviation	Term
AHVAC	ambient HVAC temperature	CRH	corridor relative humidity	ORH	outdoor relative humidity	TP	thermal preference
AP	atmospheric pressure	DOW	day of week	Pr	precipitation	TRH	task relative humidity
AT	air temperature	GSR	galvanic skin response	RH	relative humidity	TS	thermal sensation
AV	air velocity	He	height	SC	shoulder circumference	WC	weather condition
CAT	corridor air temperature	HR	heart rate	ST	skin temperature	We	weight
CI	clothing insulation	MR	metabolic rate	TAT	task air temperature	WS	wind speed
Cl	cloudiness	MRT	median radiant temperature	TC	thermal comfort		
CO ₂	Carbon dioxide levels	OAT	outdoor air temperature	TOD	time of day		

Table 3.2: The abbreviations used for thermal comfort factors and target variables.

Chapter 4

Requirements Analysis

This chapter contains many of the requirements elicitation and analysis methods proposed by Brügge and Dutoit [BD09]. Within the requirements elicitation, the purpose of the system is defined [BD09]. In the process, external entities named actors and their corresponding use cases are identified. Actors are external entities, such as users or external computing services, that interact with the system [BD09]. Use cases represent “general sequences of events” that describe all ways in which an actor interacts with a system [BD09, p. 16]. A brief overview of the system is given in Section 4.1. In Section 4.2 conventional temperature control systems and their limitations are described to motivate why a new solution is needed. The precise functional and nonfunctional requirements of the LATEST system are listed in Section 4.3. Section 4.4 contains system models for LATEST. In particular, there are a detailed description of scenarios of LATEST, its use cases, the corresponding analysis object model, and three activity diagrams modeling its central processes and its separation into two phases of functionality.

4.1 Overview

The envisioned system processes sensor readings to predict how thermal actuators should be controlled for individual occupants to feel thermally comfortable indoors. LATEST uses means of machine learning to predict an occupant’s thermal control behavior. These ML models are trained on data collected by indoor and outdoor environmental sensors as well as biosignal sensors and an occupant’s temperature control data. This approach is successful, if LATEST decreases the occupants’ interaction with the temperature control unit while keeping the occupant as thermally comfortable as they would have been if they controlled the temperature themselves.

4.2 Current System

Current temperature control systems are either manually controlled or temperature based. For the latter conventional thermostats provide the occupant with the option to set a goal-temperature according to their current needs. Most thermostats rely exclusively on indoor temperature measurements to attain the goal-temperature via thermal actuators or air conditioners. These thermostats are often centralized for a whole floor of a building where multiple occupants live. Hence, they do not address the number of the individual's thermal comfort influences discussed in Section 2.1.2. Additionally, both types of systems require the occupant to actively interact with it. This necessity is addressed by the intelligent, automated thermostat Nest. Yet in turn, studies showed that the occupant's inability to understand how Nest works was perceived negatively [YN13].

4.3 Proposed System

LATEST takes into account that thermal comfort varies between individuals. Therefore, the system uses a personal machine learning model for each occupant to predict the occupant's temperature control behavior. These models are based on information that is collected while the occupant controls thermal actuators manually. Subsequent to this *data collection phase* is the *temperature control phase*. In it, LATEST controls the temperature automatically according to the occupant's temperature control behavior. This separation into phases of functionality originates from the necessity of machine learning models to be trained on previously collected data before they can be used to predict. In the subsections below, the corresponding functional and nonfunctional requirements for the proposed system are explained.

4.3.1 Functional Requirements

The proposed system's functional requirements (FRs) are listed in this subsection. FRs describe how the system behaves in its environment without taking its implementation into account [BD09].

- FR1 **Control Temperature Automatically:** In the temperature control phase, the system must control a thermal actuator automatically according to the occupant's previously observed temperature control behavior.

4.3. PROPOSED SYSTEM

- FR2 **Provide Manual Temperature Control:** The occupant must be provided with the ability to control the system’s thermal actuator at any time. In the data collection phase, this functionality provides information about the occupant’s personal temperature control behavior to the system. In the temperature control phase, this functionality enables the occupant to interrupt the automated temperature regulation if they feel thermally uncomfortable.
- FR3 **Control Task Environment:** The system’s thermal actuator must be able to affect the task environment for every occupant separated from other occupants. Hence, the system can control the task environments of two occupants, who share a room in a smart building, individually.
- FR4 **Give Reasoning for Temperature Control:** The envisioned system must provide the occupant with a reasoning for its automated temperature control actions. Therefore, the occupant can build trust in the decisions LATEST takes, since the process is more transparent.
- FR5 **Monitor Biosignals:** LATEST must monitor an occupant’s biosignals. In particular, it monitors each occupant’s heart rate, skin temperature, and GSR, since they are predicting factors for thermal comfort and can be measured non-intrusively [CLL12, CL12, GSH67].
- FR6 **Monitor Indoor Environment:** The envisioned system must monitor the surrounding indoor environment of the occupants. Especially, the surrounding air temperature and relative humidity of the occupants must be measured.
- FR7 **Monitor Outdoor Environment:** LATEST must monitor the outdoor environment surrounding the smart building. In particular, data of air temperature, humidity, and cloudiness must be collected.
- FR8 **Personal Learning:** The envisioned system must learn the temperature control behavior of individual occupants in the context of their personal biosignal data and the indoor and outdoor thermal environment data. Therefore, LATEST uses ML methods on historic data, which are collected in a data collection phase.

- FR9 **Detect Occupancy:** LATEST must detect occupancy to automatically control the temperature when there are occupants.
- FR10 **Turn Off Without Occupancy:** LATEST must turn off the thermal actuator when there are no occupants. Hence, energy can be saved when, for instance, an occupant forgets to turn off a thermal actuator at night.
- FR11 **Run in Parallel:** The envisioned system must run for multiple occupants in parallel.

4.3.2 Nonfunctional Requirements

A system's nonfunctional requirements (NFRs) are its characteristics that are not directly related to its functionality [BD09, p. 148]. Here, the NFRs are classified according to the FURPS+ model [Gra92] excluding the class of *Functionality*, which is covered by the FRs in the previous section. FURPS+ means *Functionality, Usability, Reliability, Performance, and Supportability extended by additional classes*. Since *FURPS+*'s proposal by Grady in 1992, the class of *Reliability* is often replaced by *Dependability* [BD09, p. 126].

- NFR1 **Simplicity (Usability):** The occupant interface is designed such that every functionality can be executed in at most three consecutive actions. Furthermore, it follows the user interface guidelines of the development platform it will be running on. Hence, an occupant who is familiar with the platform's design should be able to understand and use the interface without prior training or reading a manual. A simple interface reduces friction for the occupant to use LATEST when data are collected. The more data are collected and the more granular the occupant's adjustments are, the more accurate the resulting machine learning model will be, which aims to predict the occupant's behavior.
- NFR2 **Safety (Dependability):** LATEST ensures the safety of the occupants with regards to its thermal actuators. If a thermal actuator that is controlled by the envisioned system is a possible source of harm to the occupants, the system's design must rule out any such possibility.
- NFR3 **Robustness to Outliers (Dependability):** LATEST recognizes outliers in sensor readings and smooths their effect if it is

4.3. PROPOSED SYSTEM

appropriate. Hence, the occupant's thermal comfort does not significantly suffer from such technical inaccuracies.

- NFR4 **Robustness to Sensor Failure (Dependability):** In the case of sensor failure, the envisioned system keeps all its main functionality only with eventual losses in occupant comfort.
- NFR5 **Robustness to Invalid User Input (Dependability):** If the occupant forgets to manually turn off their thermal actuators before leaving a space or negligently causes a situation where they or somebody else might be harmed, LATEST should recognize these circumstances and prevent the energy wasted and harm done.
- NFR6 **Availability (Performance):** The envisioned system should be available for the full period of occupancy. Otherwise, if an occupant's thermal comfort is not met, their performance, well-being, and satisfaction might suffer as a consequence [Fan70, dDBD97]. The period of occupancy can vary depending on the application domain. For an application in a smart office, the general working hours are from 9:00 a.m. to 5:00 p.m., which can be extended at both ends by 2 hours to include outliers. For a different application in a smart home the proposed system might need to be available to other specified periods or on-demand.
- NFR7 **Accuracy (Performance):** When controlling the temperature automatically, the envisioned system's predictive performance should result in the occupants being thermally comfortable in at least the same share of time as was when they controlled the thermal actuator manually. Therefore, the occupant's temperature control behavior is accurately automated, resulting in the occupant's satisfaction and comfort.
- NFR8 **Security (Implementation Requirement):** The proposed system should be secure such that an occupant's personal data cannot be accessed by other occupants or persons. The data that are collected includes sensible personal data, such as the occupant's location, thermal comfort, or heart rate. Hence, security and authorization standards have to be ensured.
- NFR9 **Extensibility (Implementation Requirement):** The envisioned system is required to be extensible for various kinds of sensors and actuators, depending on the application domain.

As was stated by Zheng, Dai, & Wang, thermal comfort models have not been widely deployed in practice since they are “intrusive, in the sense that either extra devices need to be installed, or human feedback needs to be collected” [ZDW19]. Hence, for portability to various application domains, the system should be extensible by more and other devices and sensors.

- NFR10 **Scalability (Implementation Requirement)**: According to FR11, the envisioned system runs multiple personal temperature control models, one for each occupant, in parallel. This includes taking measurements from occupancy and biosignal sensors for every occupant, and sensors for the indoor and outdoor environment. Hence, the application’s capabilities and development platform should be scalable to a smart building with up to 120 occupants while taking sensor measurements from every sensor at least every 30 seconds. Additionally, both the envisioned system’s decision about how to control the temperature and the corresponding reasoning should also be generated at least every 30 seconds. Hence, they must take not more than 250 milliseconds to generate on the development platform LATEST runs on.
- NFR11 **Response Time (Performance)**: If an occupant’s interaction with LATEST has a visual reaction, this reaction should feel instantaneous. Hence, it should take no more than 0.1 seconds to take place on the interface [Mil68].

4.4 System Models

This section contains several system models that are based on the requirements from the last section as proposed in [BD09]. These system models aim to give a complete and consistent view of the system in differing levels of abstraction and from different points of view. All models comply with the Unified Modeling Language (UML) as described by Brügge and Du-toit [BD09]. Scenarios provide insights into how occupants interact with the system and motivate the system’s main intentions in a lively manner. There are two scenarios presented in the following subsection. From the requirements and scenarios use cases can be inferred, which are depicted as a use case model in Subsection 4.4.2. Subsection 4.4.4 contains the analysis object model. The separation of the envisioned system’s data collection phase and

temperature control phase, and their included processes are explained in the form of UML activity diagrams in Subsection 4.4.3.

4.4.1 Scenarios

Carroll defines scenarios as “a narrative description of what people do and experience as they try to make use of computer systems and applications” [Car95]. Furthermore, scenarios improve the requirements elicitation by narratively describing a concrete example use of a system from the user’s viewpoint. In that way, they give an insight into the system that is understandable to future users and clients [BD09]. Hence, scenarios are very specific and name concrete instances of people, systems and numbers. The following two scenarios break with the methodology proposed by the literature in that they cover more than just one requirement in one scenario. This way of description is motivated by the strong interconnection between the functional requirements of LATEST defined in Subsection 4.3.1.

Starting a Workday in Museum Entry Hall

<i>Participating actors</i>	<u>Taylor: Occupant</u>
<i>Flow of events</i>	<ol style="list-style-type: none">1. On a cold winter day in Pittsburgh, Taylor arrives at her reception desk in the entry hall of the Carnegie Museum of Natural History.2. LATEST recognizes her arrival through an occupancy sensor and displays a notification on her phone saying “It’s nice having you back here, Taylor! I will take care of your thermal comfort now. Please let me know if you experience any discomfort.”3. Since she just came in from the cold, she does not mind the cool air temperature of 18.6 °C prevailing in the hall.4. LATEST periodically takes sensor measurements from biosignal sensors at Taylor’s smart watch and environmental sensors, which are distributed outdoors, across the hall, and on Taylor’s desk. Through the sensors, LATEST recognizes by intelligent comparison with previously collected data that Taylor will soon feel cool.5. LATEST turns on an infrared heating panel that is directed towards Taylor.6. After four minutes of sitting, Taylor’s body activity slows down and she starts feeling colder.7. Taylor wonders if LATEST knows about her thermal condition from the data that were collected in the data collection phase. She picks up her phone to check the LATEST application.8. There, she sees that the infrared heating panel was turned on already. A message in the “Last event” field says: “Hey Taylor! I just switched on your heating panel for you. From previous occasions I predict that you will feel uncomfortably cold soon.” Only at that moment, Taylor senses the heat from the infrared heating panel and feels thermally comfortable again.

Handling Conflicting Thermal Preferences in Shared Office

<i>Participating actors</i>	Suzy, Till: Occupant
<i>Flow of events</i>	<ol style="list-style-type: none"> 1. Like on most winter days, Till visits the gymnasium before work. After the workout, he and his colleague Suzy simultaneously arrive at their joined desks in the shared office space of CMU's Intelligent Workplace. 2. Till still feels warm from the workout. He doffs his sweatshirt, so he only wears a t-shirt. LATEST recognizes from sensors on Till's smart watch that his skin temperature is unusually high at 32 °C and there is increased moisture on his skin, which indicates a high body activity. At the same time, the sensors on Suzy's smart watch measure normal levels in the context of the environmental sensors in the room and outdoors. Since both are comfortable LATEST does not act. 3. After 40 minutes, Till's body cools down to a level where he almost feels uncomfortably cold at the prevailing indoor air temperature of 20.6 °C and relative humidity of 20%. LATEST recognizes this through the sensors in the Intelligent Workplace and on Till's smart watch. Consequently, LATEST turns on an infraNOMIC infrared heating panel, which is directed towards Till. The thermal energy of the heating panel is transferred via directed infrared rays. Hence, even though Suzy sits right next to Till, Suzy is not affected by them. 4. Till checks the LATEST app on his phone. The "Last event" field says "Hey Till! I just switched on your heating panel. From previous occasions with a similar air temperature and a decreasing body activity, I predict that you will feel uncomfortably cold soon." 5. Only after reading the text, Till realizes the radiant heater's infrared rays starting to warm up his cool feeling body. He puts his sweater back on. 7. LATEST recognizes the change in Till's clothing indirectly through the skin temperature sensor on his smart watch. Four minutes later, it turns off Tills infrared heating panel, so he stays thermally comfortable.

4.4.2 Use Case Diagram

This subsection contains the UML use case diagram of LATEST. According to Brügge and Dutoit a use case diagram consists of actors, use cases, and their relationships [BD09]. Actors are any external entities that interact with the system. Use cases represent the general sequences of these interactions. The use case diagram gives a complete overview of all possible interactions with the system. Hence, all functional requirements from Subsection 4.3.1 are represented in the use cases.

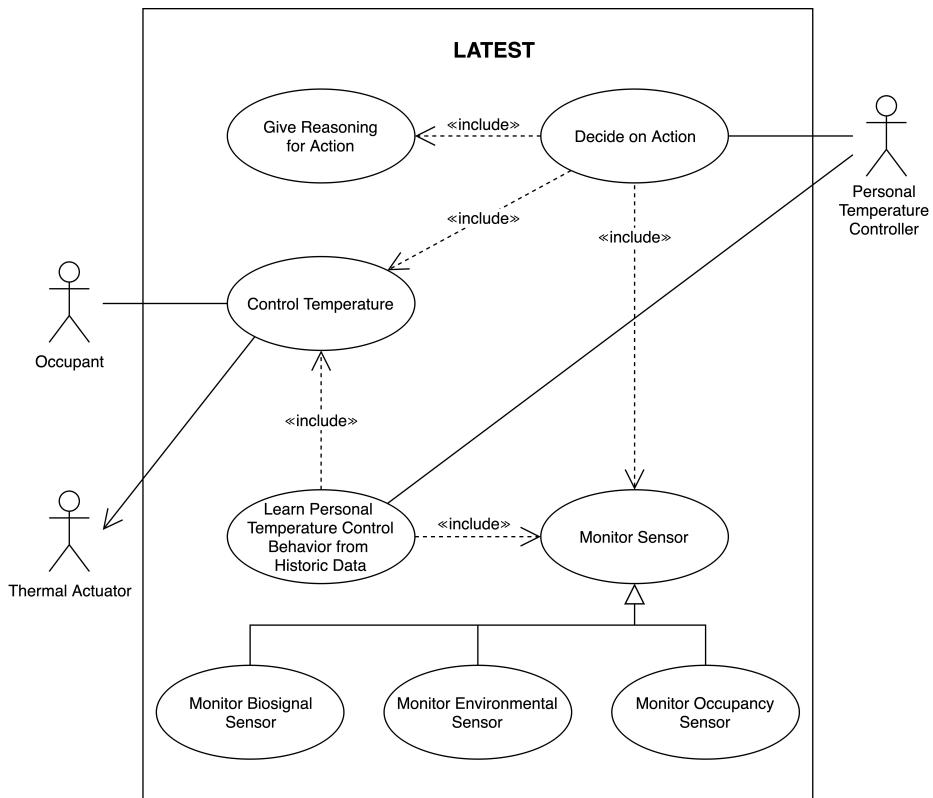


Figure 4.1: UML use case diagram of containing all use cases of LATEST. It depicts the three actors' interactions with the system in the form of use cases. It also illustrates the relationships between these use cases. LATEST works for multiple occupants in parallel, while this use case diagram depicts the use cases for a single occupant only.

Figure 4.1 shows the UML use case diagram of the envisioned system. There are three actors, namely the *Occupant*, the *Thermal Actuator* and the *Personal Temperature Controller*. The illustration shows the use cases for one single *Occupant* but LATEST provides the same functionality for

4.4. SYSTEM MODELS

multiple *Occupants* in parallel (FR11). The *Occupant* can control the temperature with the use of LATEST, which translates the temperature control commands to the *Thermal Actuator*. This use case is called *Control Temperature* (FR2) and is included into the use case *Learn Personal Temperature Control Behavior from Historic Data* (FR8) which, at the same time, includes the use case *Monitor Sensor* (FR5, FR6, FR7, FR9) and sets both sources into context. The *Monitor Sensor* use case is inherited by three more specific use cases *Monitor Biosignal Sensor* (FR5), *Monitor Environmental Sensor* (FR6, FR7), and *Monitor Occupancy Sensor* (FR9), which collect measurements from their specified data type. A more detailed taxonomy for the sensors is given in the following section in Figure 4.5.

The use case *Learn Personal Temperature Control Behavior from Historic Data* uses machine learning methods to train a predictive model of the *Occupant's* temperature control behavior depending on the sensor measurements. It is initialized by the *Personal Temperature Controller*, which uses the resulting model for the second use case it initializes, named *Decide on Action*. As depicted in Figure 4.1, this decision is based on measurements from *Monitor Sensor* and includes two more use cases. First, it also includes the use case *Give Reasoning for Action*, which explains the reasoning behind the *Personal Temperature Controller's* decision to the *Occupant*. The second use case is *Control Temperature* (FR1), which, in this case, represents the automated temperature control defined as FR1 in Section 4.3.1 as opposed to FR2 that it represents when initiated by the *Occupant*. Like before, this use case is directed to the *Thermal Actuator*, which executes the temperature control action. If the *Occupant* controls the temperature manually, the *Personal Temperature Controller* does not decide on new actions for ten minutes. The use case *Decide on Action* includes the functionality to turn off the *Thermal Actuator* in case there is no occupant apparent (FR10).

This representation of use cases was chosen to illustrate the relationships between the actors and LATEST. It aims to show the most important processes and give a complete overview of the functionality of LATEST. The use cases are explained in more detail in the tables below.

Monitor Sensor

<i>Participating actors</i>	Initiated by <i>Occupant</i>
<i>Entry conditions</i>	<ul style="list-style-type: none"> The biosignal sensors, environmental sensors and occupancy sensors are connected to the system.
<i>Flow of events</i>	<ol style="list-style-type: none"> All available sensors take measurements periodically to collect data about the <i>Occupant's</i> biosignals, the environment, and occupancy. The measurement data are persistently stored.
<i>Exit conditions</i>	<ul style="list-style-type: none"> The measurements from all sensors are stored persistently.
<i>Quality requirements</i>	<ul style="list-style-type: none"> Every available sensor is monitored at least every 30 seconds.

Control Temperature

<i>Participating actors</i>	Initiated by <i>Occupant</i> , interacting with <i>Thermal Actuator</i>
<i>Entry conditions</i>	<ul style="list-style-type: none"> The <i>Thermal Actuator</i> is connected to the system. There is an interface for the <i>Occupant</i> to control the temperature.
<i>Flow of events</i>	<ol style="list-style-type: none"> The <i>Occupant</i> gives a command to control the temperature via an interface. The command is executed by the <i>Thermal Actuator</i>. The command and its time of appearance are persistently stored.
<i>Exit conditions</i>	<ul style="list-style-type: none"> The <i>Thermal Actuator</i> executes the <i>Occupant's</i> command.
<i>Quality requirements</i>	<ul style="list-style-type: none"> Every visual reaction to the <i>Occupant's</i> interactions are performed in at most 0.1 seconds.

Learn Personal Temperature Control Behavior from Historic Data

This use case describes how the *Personal Temperature Controller* learns from the *Occupant's* actions for becoming able to predict them. The use case represents the shift from the data collection phase to the temperature control phase.

<i>Participating actors</i>	Initiated by <i>Personal Temperature Controller</i>
<i>Entry conditions</i>	<ul style="list-style-type: none">There is a data set of the <i>Occupant's</i> temperature control commands and sensor data for biosignals, environment, and occupancy available that are expected to cover future conditions.
<i>Flow of events</i>	<ol style="list-style-type: none">LATEST uses machine learning methods to learn a personal temperature control behavior model from the historic data that were collected before.
<i>Exit conditions</i>	<ul style="list-style-type: none">The resulting ML model is available to predict temperature control commands.
<i>Quality requirements</i>	<ul style="list-style-type: none">The system's learned temperature control model is personalized to the <i>Occupant's</i> temperature control behavior.

Decide on Action

This use case describes which processes take place in LATEST to control the temperature automatically. The two included use cases, *Give Reasoning for Action* and *Control Temperature*, are covered by the description as well, since they follow *Decide Action* directly and without exception.

<i>Participating actors</i>	Initiated by <i>Personal Temperature Controller</i> , influences <i>Thermal Actuator</i>
<i>Entry conditions</i>	<ul style="list-style-type: none">The sensors and the <i>Thermal Actuator</i> are connected to the system.There was a personal temperature control behavior model trained previously.The <i>Occupant</i> did not control the temperature manually in the preceding 10 minutes. <p>The sensors' data about the biosignals, the environment, and occupancy are available.</p>
<i>Flow of events</i>	<ol style="list-style-type: none">Based on the latest sensor data, the <i>Personal Temperature Controller</i> decides on an action.A reasoning for the action is generated and displayed to the <i>Occupant</i> in an interface.The action is executed by the <i>Thermal Actuator</i>.
<i>Exit conditions</i>	<ul style="list-style-type: none">The <i>Thermal Actuator</i> executes the temperature control action.The <i>Occupant</i> has the option to inspect the reasoning for the temperature control action in an interface.
<i>Quality requirements</i>	<ul style="list-style-type: none">The generation of both the temperature control action and the corresponding reasoning take less than 250 milliseconds.The latest sensor data are at most 30 seconds old.A new action is generated every 30 seconds. The system's temperature control results in the <i>Occupant</i> being thermally comfortable for at least the same share of time as when they controlled the thermal actuator manually.

4.4.3 Activity Diagrams

This section contains three UML activity diagrams to illustrate the general order of the use cases described in the previous section. The described processes only take place under the assumption that the occupant is present. While these processes run simultaneously for all occupants, here, they are detailed only for one single occupant. The first activity diagram in Figure

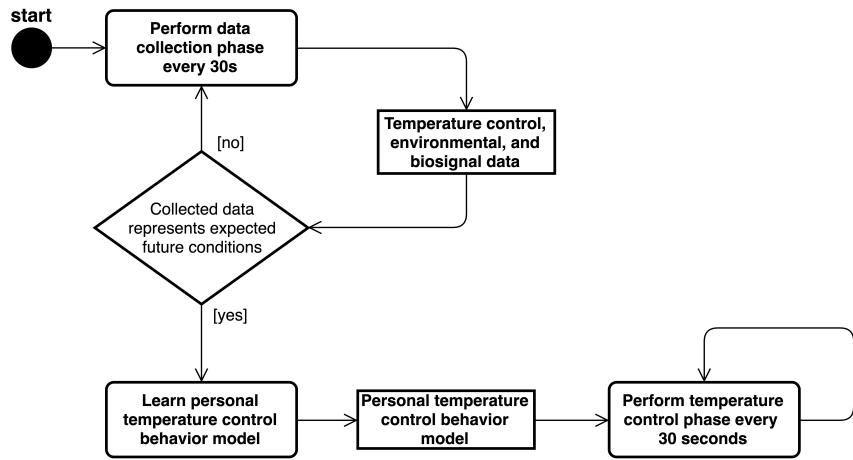


Figure 4.2: The UML activity diagram illustrating the overall functionality of LATEST. It depicts the central transition from the data collection phase to the temperature control phase.

4.2 depicts the separation of the data collection phase and the temperature control phase. This separation into phases of functionality originates from the necessity of ML models to be trained on previously collected data before they can be used for prediction, as explained in Section 2.3. The transition between the two phases is represented by the condition node, verifying if the collected data represent expected future conditions. This requirement is based on the general limitation of machine learning models to make predictions outside the conditions, which they were trained on. For instance, an ML model for predicting thermal comfort, which is trained on data collected in summer, will generally not be applicable in winter.

At first the data collection phase is executed, which results in new temperature control, environmental, and biosignal data every 30 seconds. When the stated condition is met and the data represent the variety of conditions that can be expected in the future, a personal temperature control behavior model is trained. This model is then used in the temperature control phase, which is repeated every 30 seconds.

Data Collection Phase

The goal in this phase is to collect data of the occupant's temperature control behavior in the context of environmental and biosignal sensor measurements. Figure 4.3 shows the order of the processes. At first, the occupancy sensors are used to check if the occupant is there. If they are not, this process is repeated until the occupant returns. When the occupant is there, the second condition is evaluated and, whether the occupant gave a temperature control command, the thermal actuator is adjusted or not. Then, the sensors are monitored to get the environmental and biosignal context data for the occupant's temperature control behavior.

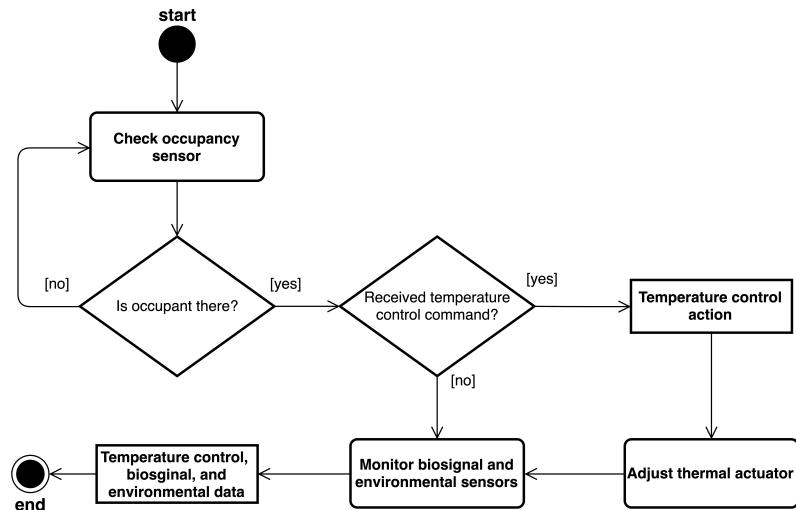


Figure 4.3: The UML activity diagram illustrating the data collection phase of LATEST. The main loop ends once a data set with high variance is at hand.

Temperature Control Phase

In this phase, LATEST performs the temperature control automatically, as long as the occupant does not interrupt. Just like in the data collection phase, at first, the occupancy sensors are used to check if the occupant is there. If they are, it is checked, whether the occupant controlled the temperature manually in the past 10 minutes and thereby interrupted the automated temperature control. If this is not the case, the sensors are monitored so their biosignal and environmental data are used to decide on a temperature control action. For this action, a reasoning is generated before the thermal actuator is adjusted.

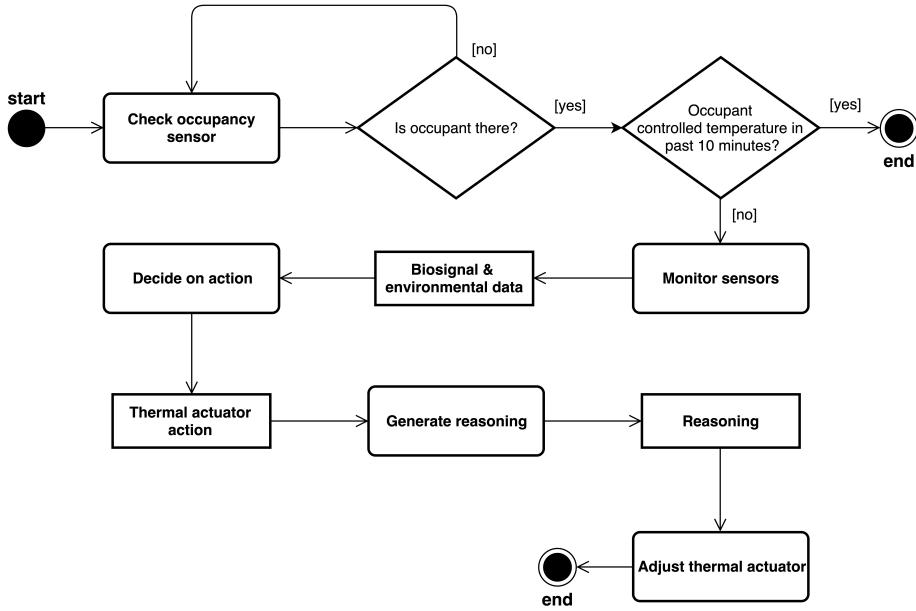


Figure 4.4: The UML activity diagram illustrating the temperature control phase of LATEST. The shown process is repeated periodically after 30 seconds.

The next section describes the analysis objects, which influence the described processes, in more detail.

4.4.4 Analysis Object Model

This and the following sections summarize the analysis of LATEST as proposed by Brügge and Dutoit [BD09]. According to the authors, the analysis process results in a correct, complete, consistent, unambiguous, and verifiable model of the system. Furthermore, analysis takes place on a level of abstraction that identifies general concepts and relationships of the application domain while leaving more granular implementation details unspecified [BD09]. Hence, a more abstract overview is given before the system's specific technical details are described in the next chapter.

This section focuses on the analysis object model (Figure 4.6), which is a UML class diagram showing the most significant objects, attributes, methods and relations of the application domain of LATEST. Before describing the full model, we give an overview of the sensor taxonomy. It depicts which types of sensors are of use in the application domain of LATEST.

Figure 4.5 shows the sensor taxonomy that is used for LATEST. The depicted sensors cover a large number of the Thermal Comfort Factors in-

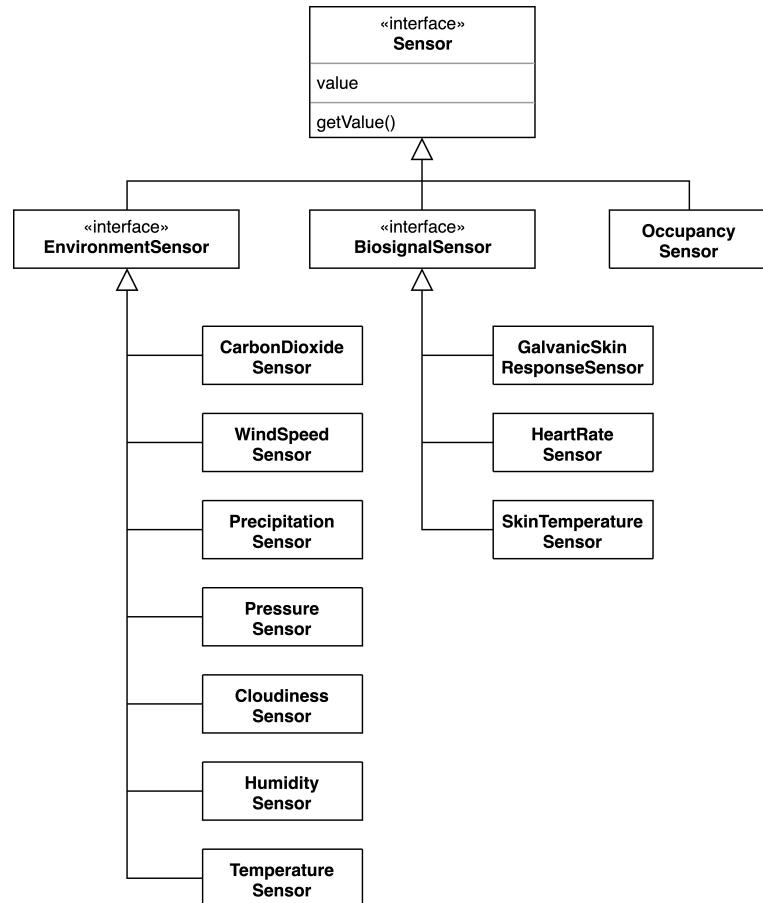


Figure 4.5: The sensor taxonomy implemented in LATEST as a UML class diagram. The sensors are grouped into three general categories, namely occupancy sensors, environmental sensors, and biosignal sensors.

troduced in Section 2.1.2. On top of the model is the *Sensor* interface. It contains a single attribute “value” representing the current measurement value and a method “*getValue()*” to make the “value” externally accessible. The *Sensor* interface is implemented by all sensor classes. Hence, every sensor class implements the same attribute and method. On the second level of the sensor taxonomy there are two more specific interfaces of sensors and the class *Occupancy Sensor* introduced. An *Occupancy Sensor* measures if there is an occupant at a specified place. On the second level of Figure 4.5, the sensor interface on the left side is *Environmental Sensor*. In the scope of this thesis, *Environmental Sensor* includes *CarbonDioxideSensor*, *WindSpeedSensor*, *PressureSensor*, *PrecipitationSensor*, *CloudinessSensor*, *HumiditySensor*, and *TemperatureSensor*. These sensors mea-

sure values that are factors for thermal comfort as described in Section 2.1.2. More Thermal Comfort Factors are measured by sensors that implement the *BiosignalSensor* interface. In this thesis these are *SkinTemperatureSensor*, *GalvanicSkinResponseSensor*, and *HeartRateSensor*. The significance of the biosignals' influence on thermal comfort was discussed in the literature for heart rate [CLL12, Cho10], skin temperature [Cho10], and galvanic skin response [GSH67]. The sensor taxonomy is represented within the *Sensor* interface in the analysis object model in Figure 4.6.

The *Sensor* interface from Figure 4.5 is taken up to represent the sensors that monitor the *Occupant* and the *Environment*. The latter is implemented by the two classes *IndoorEnvironment* and *OutdoorEnvironment*, which represent the different types of environments that are monitored by LATEST. The *IndoorEnvironment* class aggregates the *TaskEnvironment*, which is part of the indoor environment and represents the direct environment surrounding the *Occupant*. In summary, the *Sensor* interface is monitoring four sources: The *Occupant*'s biosignals, the *TaskEnvironment* surrounding the *Occupant*, the *IndoorEnvironment* and *OutdoorEnvironment* environment.

The resulting data are accessed by a *Learner*, which stores them as historic data. Additionally, the *Learner* monitors the *ThermalActuator*, which is controlled by the *Occupant*, and stores the information as historic data. The *Learner* can then learn a personal temperature control model from the collected historic data. The resulting model is accessed by a *PersonalTemperatureController*. The same class also accesses the sensor data from the four sources and feeds them into the model to decide on a temperature control action for the *ThermalActuator*.

The resulting action is then transmitted to an *Interpreter*, which generates a reasoning for the action taken and displays it to the *Occupant*. At the same time, the generated temperature control action is forwarded to the *ThermalActuator*, which executes the command and thereby influences the *TaskEnvironment*.

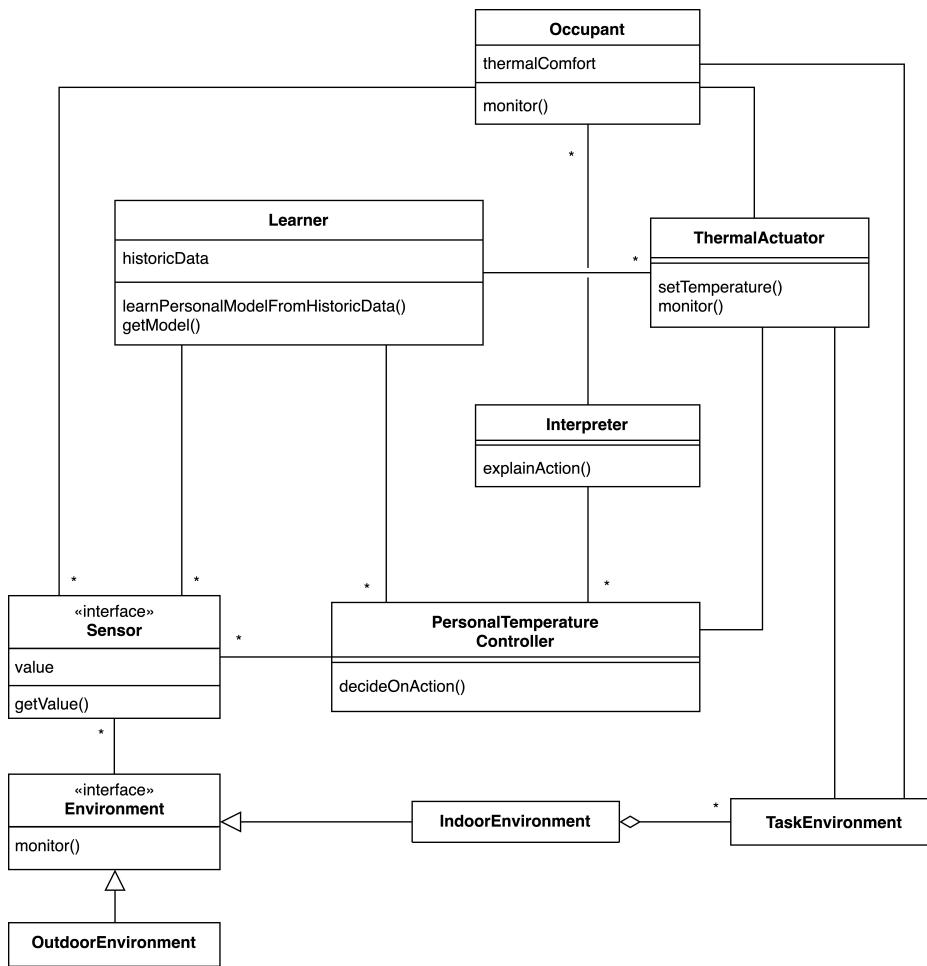


Figure 4.6: The analysis object model of LATEST as a UML class diagram. It depicts the application domain's objects and their attributes, methods and relationships.

Chapter 5

System Design

This chapter details the system design for the envisioned personalized temperature control system. It follows the System Design Document Template from “Object-Oriented Software Engineering. Using UML, Patterns, and Java” by Brügge & Dutoit [BD09]. The system design’s main objective is to work out the design goals of the envisioned system and to decompose it into smaller subsystems. Additionally, the hardware and software platforms on which the system will run are defined. Therefore, system design maps the application domain models, defined in the previous chapter, to the solution domain resulting in a subsystem decomposition that implements specified design goals and a hardware software mapping [BD09].

The following section gives a brief overview of the software architecture and its connection to other chapters. The design goals that should be realized by the system are detailed in Section 5.2. These design goals are implemented by the subsystem decomposition, described in Section 5.3. In Section 5.4, the defined subsystems are mapped to hardware and software components. Section 5.5 details the persistent data management, realized by LATEST. The access control on data managed by LATEST is described in Section 5.6.

5.1 Overview

In Section 4.3.1, FR3 was identified as the system’s ability to affect an individual occupant’s task environment without affecting other occupants’ task environments. Hence, LATEST uses a thermal actuator for every occupant located in their task environment. In Section 4.3.1, FR5 specifies that a means to monitor the occupant’s biosignals must be included in the system. Therefore, LATEST uses wearables for every occupant, to have sensors right at their body, where biosignals can be measured. Further, as described in the

analysis object model in Section 4.4.4, LATEST has environmental sensors in the *TaskEnvironment*, *IndoorEnvironment*, and *OutdoorEnvironment*. The data collected from these sensors are accessed by the *Learner*, which consequently generates a *PersonalTemperatureController* for every *Occupant*. As specified in FR11 and the analysis object model, LATEST implements several components for every occupant individually and runs them in parallel.

These parallel processes are depicted in a simplified form in the high-level design in Figure 5.1. The *Learner*, *Interpreter* and *PersonalTemperatureController* are merged as a *Personal Temperature Controller* for overview reasons. The high-level design gives an overview of the system's architecture for two color-coded occupants, *Occupant A* in orange and *Occupant B* in red. Their respective setups consist of equal components. The high-level design also depicts how *Environmental Sensors* are located in the *Outdoor Environment*, the *Indoor Environment* and the occupants' *Task Environments*. The high-level design introduces the *Occupant Managers*, which function as the managing component between LATEST and the *Occupant*. They transfer the *Occupant's* temperature control commands and their biosignal data to the *Personal Temperature Controller*. In turn, when a *Personal Temperature Controller* generates a temperature control command and a corresponding reasoning, the *Occupant Manager* forwards this reasoning to the Occupant.

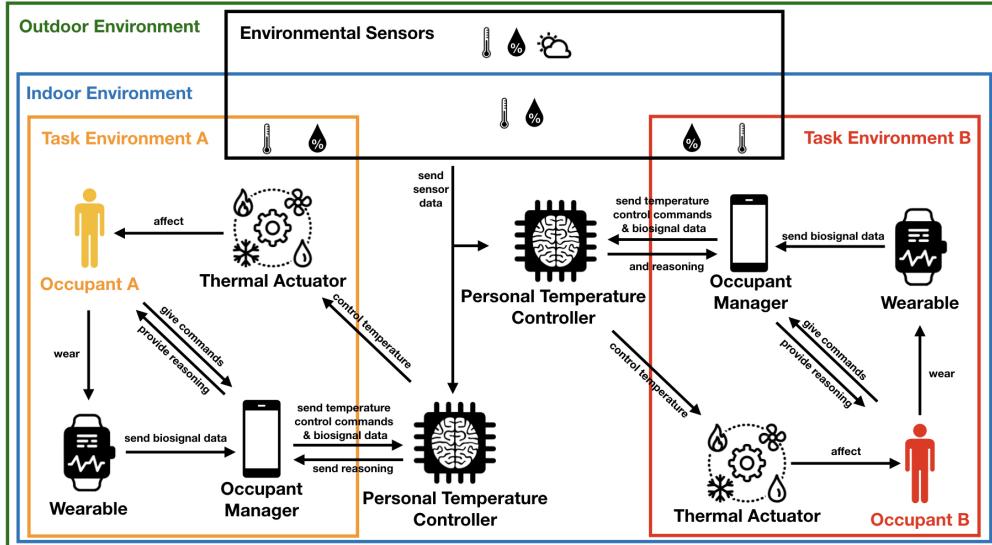


Figure 5.1: The high-level design of LATEST giving an overview of the envisioned system's architecture in the three environments: *Task Environment*, *Indoor Environment*, and *Outdoor Environment*.

5.2 Design Goals

Design goals originate from the NFRs and guide the decisions when multiple NFRs conflict with one another. Their definition is the first step in the system design process [BD09]. The NFRs for LATEST, a personalized temperature control system for multiple occupants are listed in Section 4.3.2. Here, these NFRs are prioritized from 1, being most important, to 11, being least important, to discern the important ones when in conflict. These design goals and their trade-offs influence the subsystem decomposition and hardware software mapping, which are defined in the next sections of this chapter.

1. [NFR2] **Safety (Dependability)**: The safety of the occupants is the most important NFR. Any possible harm caused by the thermal actuators must be ruled out. Safety can conflict with NFR11 “Response Time” since some safety measures might demand additional computations. Yet, these computations should be of proportionately small significance to the other computations, such as the generation of a temperature control decision or a reasoning.
2. [NFR5] **Robustness to Invalid User Input (Dependability)**: If the occupant forgets to manually turn off their thermal actuators before leaving a space or negligently causes a situation where they or somebody else might be harmed, LATEST should recognize these circumstances and prevent the energy wasted and harm done. Since this design goal is safety-related it is of critical priority.
3. [NFR8] **Security (Implementation Requirement)**: The protection of personal data is crucial for the system for legal reasons and to ensure trust from the occupants. The data must be encrypted and authorization management must be in place. Additionally, every occupant can only inspect their own data. Security can conflict with NFR11 “Response Time” since some security measures demand additional computations, such as encryption or authentication. Yet, these computations should be of proportionately small significance to other computations of the system, such as the generation of a temperature control decision or a corresponding reasoning.
4. [NFR6] **Availability (Performance)**: The design goal of availability represents the absolute amount of time the system is functioning correctly. It is a supporting the NFR11 “Response Time”, NFR3 “Robustness to Invalid User Input” and NFR7 “Accuracy” since an offline

system can neither respond nor predict accurately. Hence, availability has a relatively high influence on the occupant's experience.

5. [NFR4] **Robustness to Sensor Failure (Dependability)**: This design goal is related to NFR6 "Availability" since a sensor failure could reduce in a system failure. This design goal is that the proposed system keeps its main functionality in case of sensor failure. Losses in prediction accuracy and resulting occupant comfort are tolerated.
6. [NFR10] **Scalability (Implementation Requirement)**: The envisioned system should run multiple personal temperature control models, one for each occupant, in parallel. This includes occupancy and biosignal sensors for every occupant and sensors for the indoor and outdoor environment. Hence, the application's capabilities and development platform should be scalable to a smart building with up to 120 occupants while taking sensor measurements from every sensor at least every 30 seconds. Additionally, both the envisioned system's decision about how to control the temperature and the corresponding reasoning should also be generated at least every 30 seconds. Hence, they must take not more than 250 milliseconds to generate on the development platform LATEST runs on. This design goal might conflict with NFR7 "Accuracy" since the best performing machine learning model can be slower inferring the occupant's temperature control behavior and generating a corresponding reasoning.
7. [NFR11] **Response Time (Performance)**: Crucial for the occupant's experience is a visual response time that feels to be instantaneous. Hence, every visual reaction to an occupant's interaction should take no more than 0.1 seconds to take place on the interface [Mil68].
8. [NFR7] **Accuracy (Performance)**:
When controlling the temperature automatically, the envisioned system's predictive performance should result in the occupants being thermally comfortable in at least the same share of time as was when they controlled the thermal actuator manually. Therefore, the occupant's experience is improved by a high accuracy. This design goal can reinforce NFR9 "Extensibility", since the predictions should be more accurate the more TCF are monitored. Accuracy-related design goals are of less importance because minor inaccuracies that result in a delay of temperature control are unlikely to be recognized by the occupant [DK07].

9. [NFR9] **Extensibility (Implementation Requirement)**: LATEST should be extensible by various kinds of sensors and actuators for portability and adaptability to other smart buildings. This can include to favor open-source software over commercial software, since open-source software can be accessed without further resources and becomes more portable. This design goal might result in additional development time, which is not considered as an NFR.
10. [NFR1] **Simplicity (Usability)**: The occupant interface is designed such that every functionality can be executed in at most three consecutive actions. A simple interface reduces friction for the occupant to use LATEST when data are collected. The more data are collected and the more granular the occupant's adjustments are, the more accurate the resulting machine learning model will be. Hence, this design goal reinforces NFR7.
11. [NFR3] **Robustness to Outliers (Dependability)**: LATEST recognizes outliers in sensor readings and smooths their effect if it is appropriate. This design goal reinforces NFR7, since the prediction of the temperature control behavior is more accurate without outliers in the input. The effect in increased response time due to additional computations can be neglected.

5.3 Subsystem Decomposition

The subsystem decomposition in Figure 5.2 depicts the subsystems of LATEST with regards to the analysis object model from Section 4.4.4 and the design goals described in the previous section. The naming from the analysis object model is preserved so each of its classes is represented as a subsystem or a component in the subsystem decomposition. The only exception is the *Occupant*-class. Since the occupants are not a part of the system, they are not represented in the subsystem decomposition.

In the center of Figure 5.2 is the subsystem *Occupant Manager* depicted. It includes two components, named *Occupant Interface* and *Data Handler*. The occupant can control the *Heater* component of the *Thermal Actuator* subsystem through the *Occupant Interface*, which transfers the occupant's command over the *Personal Control Unit*'s component *Personal Temperature Controller*. This control functionality represents the first of five sources, which send data to the *Learner* component of the *Personal Control Unit* subsystem.

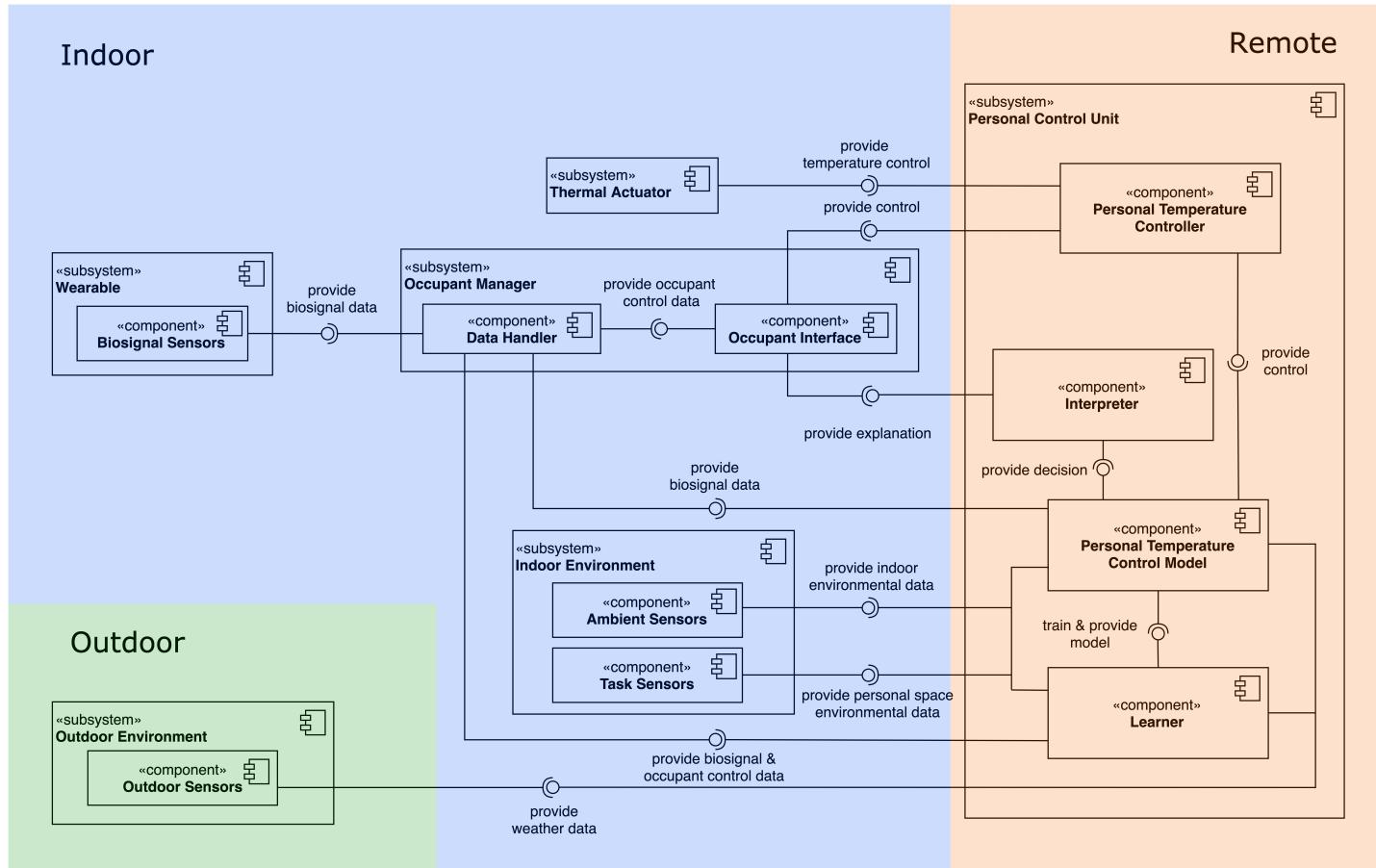


Figure 5.2: The Subsystem Decomposition of LATEST. There are three regions separated by colors: *Indoor* in blue, *Outdoor* in green, and *Remote* in orange.

--- 5.4. HARDWARE SOFTWARE MAPPING

In particular, the occupant's commands on the *Occupant Interface* are provided to the *Data Handler* component, which is also part of the *Occupant Manager* subsystem and forwards the control data to the *Learner*. In the same way, the biosignal data measured on a *Wearable* subsystem by the *Biosignal Sensors* component is provided to the *Occupant Manager's* component, *Data Handler*, which in turn forwards it to the *Learner Component*. The *Occupant Manager* is also used to infer whether the occupant is present or not, representing the *Occupancy Sensor* introduced in Section 4.4.4.

The third and fourth kind of data source depicted in this subsystem decomposition are the *Indoor Environment* subsystem's components, *Indoor Sensors* and *Task Sensors*. They both directly provide their respective environmental data to the *Learner* component at the *Personal Control Unit*. These four data sources are located indoors, as marked by the blue overlay, and send their measurements to the remote *Personal Control Unit*. The fifth and last data source is the *Outdoor Sensors* component included by the *Outdoor Environment* subsystem, which is the only subsystem that is located outdoors. The component provides its weather data directly to the remote *Learner*.

The *Learner* trains a personal temperature control model from the data it collects from the five sources and provides the model to the *Personal Temperature Control Model* component. The *Personal Temperature Control Model* has access to the same five data sources and uses them to decide how the temperature should be controlled for the corresponding occupant. This decision is provided to the *Interpreter* component, which generates a reasoning for the decision and provides it to the *Occupant Interface* for the occupant's inspection. At the same time, the *Personal Temperature Controller* accesses the *Personal Temperature Control Model's* decision who translates it to a command for the *Heater* component of the *Thermal Actuator* subsystem.

5.4 Hardware Software Mapping

In this section, the subsystems introduced in the previous section are mapped to hardware and software components. The resulting model is called hardware software mapping and defines the actual hardware and software components that are used to implement LATEST [BD09].

Figure 5.3 shows the hardware software mapping of LATEST. The subsystem *Personal Control Unit* is realized as a software script on a potentially remote server running Python 3.7.6 on Ubuntu 18.04.1 LTS. The *Personal Control Unit* includes the same four components as in subsystem decomposition: *Personal Temperature Controller*, *Interpreter*, *Personal Temperature*

Control Model, and *Learner*. Their interactions were detailed in Section 5.3. The use of Python is mainly motivated by its broad coverage of implementations of recent ML methods and interpretability methods. State-of-the-art libraries, such as scikit-learn¹, TensorFlow², PyTorch³, lime⁴, or SHAP⁵ offer open-source implementations of the methods that were discussed in Section 2.3. Hence, with Python, a large number of ML methods and ML interpretability methods can be compared, building the basis for the design goals to NFR7 and NFR10.

On the operating system, an *openHAB* instance runs, which acts as a platform to communicate to various kinds of sensors and actuators. *openHAB*⁶ is an open-source home automation platform, which offers support for a large range of devices and protocols and handles workloads of a large number of connected items. Therefore, scalability, availability, and extensibility are provided by design. Data sources and smart devices of various kinds can be added to an *openHAB* instance for which the respective add-ons are offered. That includes most smart device protocols, such as BACnet⁷ or MQTT⁸. Hence, *openHAB* fulfills the design goals of NFR9 and NFR10. Additionally, persistence functionality is offered so selected data can be stored in a database. The changes in sensor and actuator states are forwarded to the database immediately so the sensor's and actuator's live state is always available for external use, realizing the design goal to NFR11.

LATEST runs an *influxDB* instance for its database on the same operating system as the *Personal Control Unit* and *openHAB* are running. *influxDB*⁹ is a database for time-series use cases. Its design trade-offs fit the requirements for LATEST, since *influxDB* is optimized for read and consecutive write operations¹⁰. Consequently, change and delete functionalities are restricted. Since LATEST collects and provides data without changing them, this aspect of *influxDB* does not restrict the performance, supporting the design goal to NFR11. Additionally, *influxDB* is designed for time-series data, which are stored in ascending order making writes of the current timestamp

¹GitHub, “scikit-learn”, access March 2, 2020. github.com/scikit-learn

²GitHub, “TensorFlow”, access March 2, 2020. github.com/tensorflow

³GitHub, “PyTorch”, access March 2, 2020. github.com/pytorch

⁴GitHub, “lime”, access March 2, 2020. github.com/marcotcr/lime

⁵GitHub, “SHAP”, access March 2, 2020. github.com/slundberg/shap

⁶openHAB, “empowering the smart home”, access March 3, 2020. www.openhab.org

⁷BACnet, “Welcome!”, access March 3, 2020. www.bacnet.org

⁸MQTT.org, “Documentation”, access March 3, 2020. mqtt.org/documentation

⁹influxdata, “Real-time visibility into stacks, sensors and systems”, access March 3, 2020. www.influxdata.com

¹⁰influxdata Docs, “InfluxDB design insights and tradeoffs”, access March 3, 2020. docs.influxdata.com/influxdb/v1.7/concepts/insights_tradeoffs/

5.4. HARDWARE SOFTWARE MAPPING

very performant. This property is also beneficial for the use case of LATEST and realizes the design goal to NFR10. On top of that, influxDB provides authentication management so the data are stored securely, realizing the design goal to NFR2. The *influxDB* software component communicates with the *Personal Control Unit* and *openHAB* via HTTP.

The subsystem *Thermal Actuator* is realized by a combination of a plugwise Circle¹¹ and an infraNOMIC Frame Line infrared radiant heating panel¹² with 210 W. This radiant heating panel is safe for operation even considering invalid user input and complies with the UL 499 Standard for Electric Heating Appliances¹³. Consequently, it fulfills the design goals to NFR2, NFR5, and NFR9 detailed in Section 5.2.

Additionally, the heating panel's control modes are “ON” and “OFF”. This functionality encourages the use of plugwise Circle to control the radiant heating panel remotely. The Circle is a *Switch Module* that acts as a current switch, turning the panel on or off via a network protocol, and current sensor, measure the energy used. In LATEST, every plugwise Circle has a wireless connection to the *Personal Control Unit* via a plugwise Stick¹⁴, which is connected to *openHAB* on the *Server* via plug-in. The network protocol used for the wireless connection is Zigbee¹⁵.

The subsystem *Occupant Manager* is realized as a *Mobile Application* running on an Apple iPhone SE¹⁶. This application implements the functionality of both components *Data Handler* and *Occupant Interface*. Hence, it is connected to the *Wearable*, which is realized as a Microsoft Band 2 (MSB) smart watch¹⁷. The use of a smart watch for measuring biosignals is supported by the finding that wrist skin temperature is especially responsive to a person's thermal sensation [CL12]. The MSB uses Bluetooth as a network protocol for connecting to the iPhone. Liedl implemented a mobile Swift application running on iOS, which features connecting the MSB with

¹¹plugwise, “Circle”, access March 2, 2020. www.plugwise.com/en-US/products/circle

¹²infraNOMIC, “Frame-Line, frames made of aluminium: General”, access March 3, 2020. www.infranomic.de/en/products/frame-line/frames-aluminium

¹³UL standards, “Standard for Electric Heating Appliances Purchase UL 499”, access March 3, 2020. standardscatalog.ul.com/standards/en/standard_499_14

¹⁴FCC ID Database, “ZB9-STICK”, access March 2, 2020. fccid.io/ZB9-STICK

¹⁵Zigbee Alliance, “Zigbee The Full-Stack Solution Interlacing All Your Smart Devices”, access March 3, 2020. zigbeealliance.org/solution/zigbee

¹⁶Apple, “Apple Introduces iPhone SE — The Most Powerful Phone with a Four-inch Display”, access March 3, 2020. www.apple.com/newsroom/2016/03/21Apple-Introduces-iPhone-SE-The-Most-Powerful-Phone-with-a-Four-inch-Display

¹⁷Microsoft, “End of support for the Microsoft Health Dashboard applications and services: FAQ”, access March 3, 2020. support.microsoft.com/en-us/help/4467073/end-of-support-for-the-microsoft-health-dashboard-applications

an iPhone [Seb19]. The MSB includes three *Biosignal Sensors*: *Wrist Skin Temperature Sensor*, *Heart Rate Sensor*, and *Galvanic Skin Response*. As detailed in Section 2.1.2, these biosignals have a significant influence on thermal comfort, supporting the design goal to NFR7. The *Mobile Application* communicates with the *Personal Control Unit* via the Hypertext Transfer Protocol (HTTP). The functionality of the *OccupancySensor* introduced in the sensor taxonomy in Section 4.4.4 is implemented by the MSB, which indirectly provides information about the occupant’s location.

In the previous section, two components for the subsystem *Indoor Environment* were identified: *Indoor Sensors* and *Task Sensors*. The *Indoor Sensors* component is realized as three environmental sensors using the BACnet protocol and one sensor, which uses HTTPS for communication. The former are *Indoor Building Temperature Sensor*, *Indoor Building Carbon Dioxide Sensor*, and *Indoor Ambient Heating Temperature Sensor*, which is mounted atop the building’s ambient heating system. The latter is an ESP8266¹⁸ microcontroller located in the corridor of the building. It includes a DHT22 temperature and humidity sensor¹⁹, which communicates to *openHAB* on the *Server* via HTTPS over Wi-Fi. The second component of *Indoor Environment* identified in the previous section, *Task Environment Sensors*, is realized by another ESP8266. The ESP8266 is located at the occupant’s desk and uses HTTPS to transfer measurements taken with a DHT22 temperature and humidity sensor to *openHAB*.

One remaining component identified in the previous section, *Outdoor Sensors*, is realized by a weather station located outdoors. It includes a *Communicator*, which communicates to *openHAB* via HTTPS. The *Communicator* transfers the measurements from six sensors: *Outdoor Humidity Sensor*, *Outdoor Temperature Sensor*, *Outdoor Wind Speed Sensor*, *Outdoor Cloudiness Sensor*, *Outdoor Pressure Sensor*, and *Outdoor Weather Sensor*.

¹⁸Espressif, “ESP8266EX – Low-power, highly-integrated Wi-Fi solution”, access March 3, 2020. www.espressif.com/en/products/hardware/esp8266ex/overview

¹⁹adafruit, “DHT22 temperature-humidity sensor + extras”, access March 3, 2020. www.adafruit.com/product/385

5.4. HARDWARE SOFTWARE MAPPING

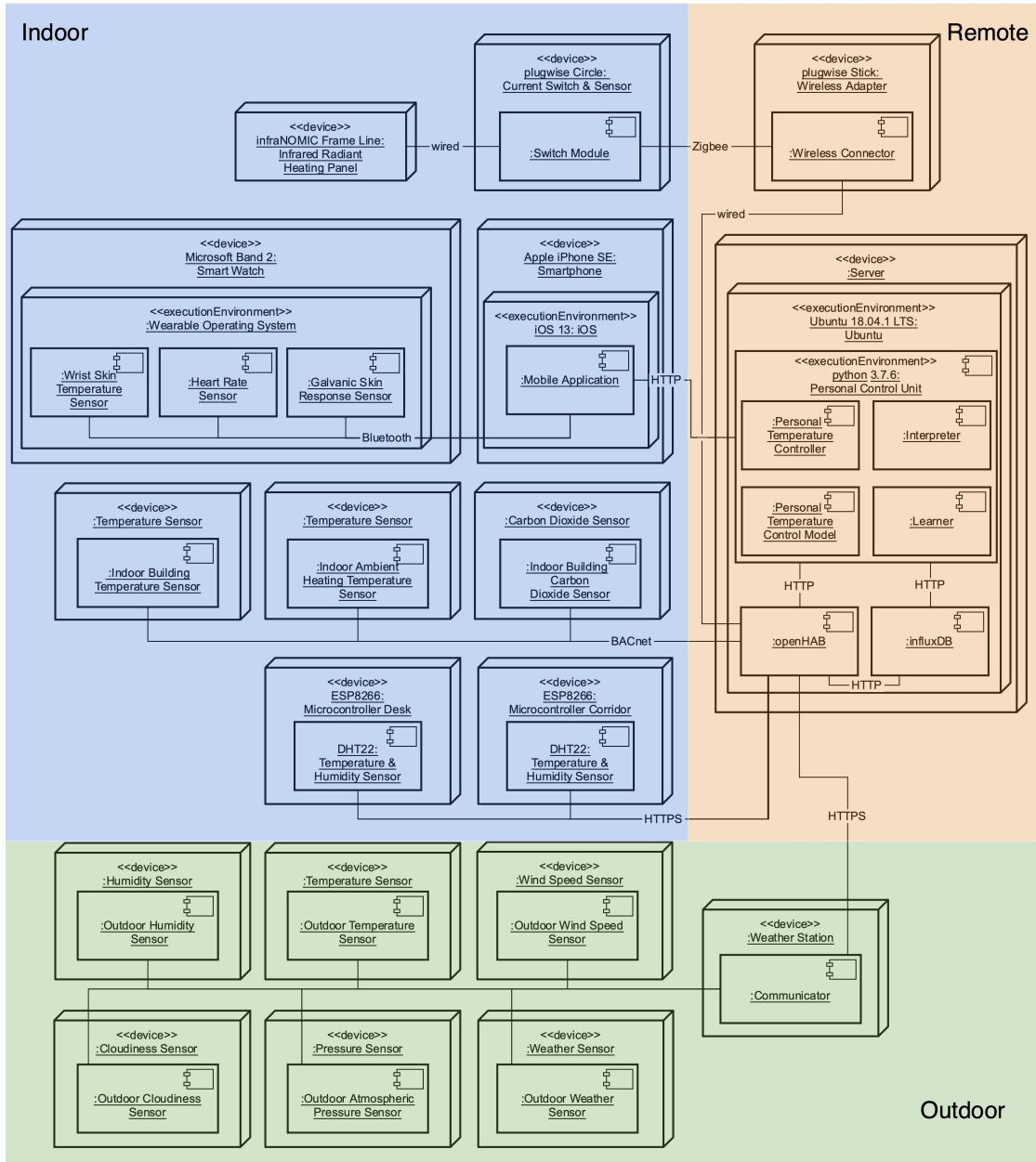


Figure 5.3: The UML deployment diagram depicting the hardware software mapping of LATEST. There are three regions separated by colors: *Indoor* in blue, *Outdoor* in green, and *Remote* in orange.

5.5 Persistent Data Management

This section details the persistent data management of LATEST. It covers the format of the data storage and how the components of the novel temperature control system store it. LATEST stores two kinds of objects: measurements and machine learning models.

Tag	Group	Type
Occupant	THERMAL ACTUATOR	HeatingPanelState
Occupant	BIO SIGNAL	HeartRate, WristSkinTemperature, GalvanicSkinResponse
Occupant	TASK INDOOR ENVIRONMENT	TaskRelativeHumidity, TaskAirTemperature
Indoor	AMBIENT INDOOR ENVIRONMENT	GeneralTemperature, CorridorTemperature, CorridorRelativeHumidity, CarbonDioxideLevel, AmbientHeaterTemperature
Outdoor	OUTDOOR ENVIRONMENT	RelativeHumidity, Temperature, WindSpeed, Cloudiness, AtmosphericPressure, WeatherCondition

Table 5.1: The 17 measurement ids covered by LATEST. Every id consists of a *Tag* and a *Type*. The measurement types are grouped by the second column for a more granular overview.

Measurements are stored in an influxDB instance that runs on a possibly remote server and is connected to an openHAB instance. Measurements are stored to train and evaluate ML models with them. There are groups of measurement types from sensors and actuators that are directly related to a specific occupant. These groups are the activity data from thermal actuators, biosignal sensor data, and data from task sensors, as described in the previous sections. In total, there are six different such measurement types per occupant. All other measurements from sensors are considered environmental. Therefore, every measurement id contains either their corresponding occupant's tag or one of the environment tags: **Outdoor** or **Indoor**. The following table lists all occupant-related and environmental measurement types.

An id is formed by concatenating a measurement’s tag with its type. For instance, the id for the heart rate of an occupant with tag `Occupant2` is `Occupant2HeartRate`. To provide a more detailed overview, the measurement types are organized in Table 5.1 by tag first and group second.

Every measurement consists of three pieces of information:

<i>Id</i>	The id consists of the sensor or actuator type, and a corresponding occupant or environment tag.
<i>Timestamp</i>	The timestamp the measurement was taken.
<i>Value</i>	The value of the measurement.

In influxDB, data points are stored as a time-series grouped by series-id. Hence, every measurement’s id represents one series-id in the influxDB instance of LATEST. As explained in the previous section, influxDB is optimized for writes in ascending time order and read requests on time ranges of an individual series. That fits the requirements of LATEST, since measurements are collected over time and is hence written to the database in ascending time order. Additionally, when the data are accessed by the *Learner* component, it requests the full range of data for a specific occupant and the environment, which is efficiently executed by influxDB as well.

The second kinds of objects that are stored persistently by LATEST are the machine learning models. As detailed in Chapter 6, the various machine learning models are optimized and compared for every occupant’s data. Since the training and optimization process takes considerable resources and should system startups should be fast, the resulting models are serialized and stored on disk with the pickle module²⁰. When a machine learning model for predicting an occupant’s temperature control behavior is deployed, the corresponding file is de-serialized from disk. The resulting Python object representing the machine learning model accesses the latest measurements from the influxDB instance to infer the occupant’s temperature control behavior.

²⁰Python, “pickle – Python object serialization”, access March 2, 2020. docs.python.org/3/library/pickle.html

5.6 Access Control

Since the occupants' personal data are stored on the influxDB, security measures are needed (NFR8). For that cause, influxDB offers authentication to password-secure data. As explained in the previous section, influxDB stores data points as a time-series grouped by a series-id. In turn, multiple series-ids can be grouped by tables, which can be secured by authentication. In LATEST, every occupant has an own table storing all the series of their personal data (i.e. all measurements with their occupant tag). Every such table can be made accessible for read and/or write transactions for specified users. There are users created with predefined passwords for openHAB, the Python script, and every occupant's Swift application. openHAB has read and write privileges to all tables since they need to write task environment data. The Swift applications have read and write privileges only to their corresponding occupant's table since they write their occupant's thermal actuator data and biosignal data. They also have read privileges to the environmental data. The Python script has a read privilege on all tables since it needs to access every occupant's data to train their personal ML model. The influxDB's users and tables are managed by an administrator. Therefore, an intruder would need to have access to the occupant's Swift application or the server running the Python script and openHAB, to access an occupant's personal data. Table 5.2 shows the resulting access matrix for the influxDB instance of LATEST.

Measurements Actors	Occupant-related	Environmental
Swift application	read and write own data	read
openHAB	read and write	read and write
Python script	read and write	read and write

Table 5.2: The access matrix of the influxDB instance of LATEST.

Chapter 6

Object Design

This chapter details the object design for the envisioned personalized temperature control system. The object design follows both the requirement analysis, which results in application objects, and the system design, which defines the system’s subsystem decomposition, hardware software mapping, persistent data management, and access control. Hence, object design closes the “gap between the application objects and the off-the-shelf components by identifying additional solution objects and refining existing objects” [BD09]. In the next chapter, the processes defined in the object design will be experimentally conducted in a case study.

6.1 Overview

The following sections detail the processes that take place after the data collection phase and in preparation for the temperature control phase, as introduced in Section 4.4.3. The processes are described for one occupant, while LATEST provides the same functionality for multiple occupants in parallel.

The first step is to preprocess the collected data, such that their format is suited for the ML methods. Following the preprocessing, the general approach is to train specified ML methods on the collected data and pre-select the best-performing ones before executing a more resource-intensive parameter optimization for finding the model that meets the requirements of LATEST best. One ML method’s parameter optimization returns the parameters that produce the best performance for the collected data concerning the considered set of parameters. Consequently, all ML methods are compared according to their performance for the collected data and on the runtime requirements, specified in Section 4.3.2. At this stage between

the data collection phase and the temperature control phase, the mentioned processes are automated on a process level while the transition between two processes requires active intervention.

6.2 Data Preprocessing

In the data collection phase of LATEST, data about an occupant's control behavior while using a thermal actuator are gathered in the context of various environmental and biosignal sensor measurements. Section 5.5 lists the exact measurement types that are persistently stored by LATEST. These measurement types and all other generated types, which are used as input data, are the features of the ML methods that will be explained in the next section.

In the preprocessing process, several adjustments and extensions are applied to these features to comply with the NFR, specified in Section 4.3.2, and to improve the expected performance of the ML models. This section covers how the preprocessing methods, introduced in Section 2.3, are concretely applied in LATEST and which design choices are taken.

- **Outlier Handling:** As defined in NFR3, LATEST must be robust to outliers in sensor measurements. Hence, a moving median filter is applied to the sensor data. As explained in Section 2.3, the moving median filter can be used for that cause. NFR10 defines that sensor measurements must be taken at least every 30 seconds. Hence, a filter size of 5 is considered reasonable to keep smooth outliers while reducing the risk of introducing an attenuation bias.
- **Time Step Fitting:** As mentioned before, NFR10 defines that at least every 30 seconds a new measurement must be taken from every sensor and a temperature control action with a corresponding reasoning must be generated for every occupant. Hence, there is at least one measurement from every sensor spread across every time interval of 30 seconds. Therefore, the time step fitting method, explained in Section 2.4, is applied to the collected data set. If there are two or more values of the same feature fitted on the same point in time, these values' average is kept. If there is a value missing for a feature in a time step, this value is interpolated between the neighboring given values or, if the feature is categorical, copied from the previous data point.
- **Data Standardization:** As described in Section 2.3, all distance-based ML methods require standardization of their inputs while methods that rely on probability distributions and tree-based methods are

invariant of standardization. For this reason, all data are standardized in LATEST before training the temperature control behavior models.

- **One-Hot Encoding:** The values of the *Outdoor Weather Sensor* from the hardware software mapping in Section 5.4 are categorical, which cannot be processed by ML methods. As specified in Section 5.5, this sensor’s corresponding id is `OutdoorWeatherCondition`. Examples of its categories are “light rain”, “heavy snow”, or “clear sky”. As explained in Section 2.3, these categories must be made numerical, such that ML methods can process them. Therefore, one-hot encoding is applied to this feature. One-hot encoding creates a numerical feature for every category that the categorical feature can represent. For instance, the three previously mentioned categories are represented as three features, namely `OutdoorWeatherConditionLightRain`, `OutdoorWeatherConditionHeavySnow`, and `OutdoorWeatherConditionClearSky`. The original categorical feature is removed from the data.
- **Trend Feature Generation:** As explained in Section 2.3, ML methods on trend-dependent time-series data improve in performance, when trend features are generated. Therefore, LATEST generates trend features for all numerical sensor types. These trend features are moving average, moving weighted average, and Relative Strength Index (RSI) [PSTK15] with three periods each: 25 time steps, 50 time steps and 100 time steps. We showed experimentally that these trend features and periods produce a performance increase, compared to other trend features from [PSTK15] and periods in the given domain.

Every feature that is generated by one-hot encoding or trend feature generation to train an ML model, must also be generated when the model is deployed in the temperature control phase. The features that are used for trend generation are all 17 measurement types listed in Table 5.1 excluding the target variable `OccupantHeatingPanelState` and `OutdoorWeatherCondition`, which is one-hot encoded. Hence, for 15 measurement types trend features are generated for three time periods and three trend features, which results in a total of 135 generated trend features. Additionally, `OccupantHeatingPanelState` is replaced by the features of its one-hot encoding, adding 16 more features. With the 15 original measurements, from which the trend features were generated, that makes a total of 166 features. These features are the input to every occupant’s personalized ML model.

These preprocessing methods are implemented in Python with the pandas library¹, which is the most used library for data processing in Python

¹GitHub, “pandas”, access March 6, 2020. pandas.pydata.org

along with NumPy² and offers efficient implementations for the mentioned preprocessing methods. In LATEST we prefer it over NumPy for its functionality for working with time-series data.

Intervals of Occupancy

Another step of preprocessing is to drop the data for time intervals in which an occupant was not in the environment affected by their thermal actuator. In these intervals, the thermal actuator should not be used, so LATEST does not need to predict how the occupant would control it. These intervals' inverse, the time intervals in which the occupant was in the environment affected by their thermal actuator, are referred to as *intervals of occupancy*. The *intervals of occupancy* are assumed to be independent from one another, while their contained data points might be dependent on one another. Hence, it is assumed that the order in which the *intervals of occupancy* take place can be changed without changing their target values, which will be of use in Section 6.3.

6.3 Model Selection

NFR7 from Section 4.3.2 requires the temperature control behavior model to have high accuracy. Therefore, a model selection process is conducted. In this process, multiple ML methods are trained on the collected and preprocessed data. The models that perform best on an interval-stratified k-fold cross-validation are then parameter-optimized with a randomized grid search. In the last step, the optimized model that performs best and complies with the runtime requirements from NFR10 is selected to be deployed in the temperature control phase.

For the comparison of temperature control behavior models, the training data can be expected to be relatively unbalanced as they are in other thermal comfort data sets [FQvF⁺19]. A data set is unbalanced when the distribution of its targets values is highly uneven. For LATEST, this is likely to be the case for some occupants, since a share people are either too warm in most of the cases, or too cold – compared to other occupants they share an ambient environment with [CK19]. Additionally, the target variable in LATEST is binary. As explained in Section 2.3, for unbalanced data with binary target variable, as is the case in LATEST, the f1-score is a fitting performance metric.

²NumPy.org, “NumPy”, access March 6, 2020. numpy.org

6.3.1 Interval-Stratified K-Fold Cross-Validation

For the cross-validation techniques explained in Section 2.3 there are no requirements on which data points belong to the same set, other than equaling out the target value distributions for the stratified k-fold cross-validation. However, one central requirement for cross-validation techniques is that the training set and validation set are independent. Since thermal comfort data are trend-dependent, meaning every data point is dependent on the preceding and following data point in the time-series, these dependent data points must be taken into the same group within cross-validation. For instance, if an occupant did not want to turn a heater on 30 seconds ago, it is highly probable that they do not want to turn on the heater now, since, in the domain of thermal comfort, conditions change over longer stretches of time. Therefore, both data points (the current one and the one from 30 seconds ago) are taken into the same group, so a classifier, which has the information of one of the data points, is not validated using the other data point.

Therefore, we propose an interval-stratified k-fold cross-validation. It is founded on the assumption that the *intervals of occupancy*, as defined in the previous section, are independent of one another while the data points of a single *interval of occupancy* are dependent³. Hence, the *intervals of occupancy* are not split up into multiple groups in interval-stratified k-fold cross-validation. At the same time, it is ensured that the resulting groups are of equal size and have an equal target variable distribution. For both requirements, if equality is not possible, the groups are constructed to be as close in size and in target variable distribution as possible.

The following three figures show an example of applying an interval-stratified 2-fold cross-validation on time-series data. Figure 6.1 shows the data as they could be found in a time-series database. For the cross-validation, the key information is “Occupancy” and “Heating ON/OFF”, the heating status, in the top of the figure. In LATEST, the heating status is the target variable and “Occupancy” defines the *intervals of occupancy*. Additionally, there are four colored, unspecified features displayed, which represent the TCF data the ML model is trained on.

Figure 6.2 depicts the data of Figure 6.1, separated into four *intervals of occupancy*, which are color-coded in yellow, green, red, and blue. Additionally, the time intervals of the consecutive heating states are displayed

³It is noted that hold-out cross-validation, which is popular for comparing ML models on time-series data, is built around the assumption that one data point is dependent to all preceding data points [Tas00]. Hence, for this kind of cross-validation, the evaluation set must follow the training set chronologically. This assumption does not hold for the *intervals of occupancy*.

in-place as integers. An interval-stratified 2-fold cross-validation distributes these four intervals into 2 groups, such that both the size and the target variable distribution for every group are as close to each other as possible.

The split to the interval-stratified 2-fold cross-validation is shown in Figure 6.3. “Group1” contains three *intervals of occupancy* with a total of 410 data points, 280 with heating status “ON” and 130 with heating status “OFF”. “Group2” holds the largest interval with 390 data points, of which 270 belong to the target value “ON” and 120 to “OFF”. Hence, both the group sizes and the target value distributions are as close to each other across all groups as possible, without splitting an interval. Hence, all data points are independent between the groups, meeting all the requirements specified for the interval-stratified k-fold cross-validation.

Table 6.1 shows the training and validation sets for the two folds. In the first fold, the model is trained on the data of “Group1” and validated with the data in “Group2”. In the second fold, “Group1” is the validation set for a model trained on the data of “Group2”. In practice, the k for a k-fold cross-validation is set to a value of at least 4, resulting in a validation set size of at most 25%.

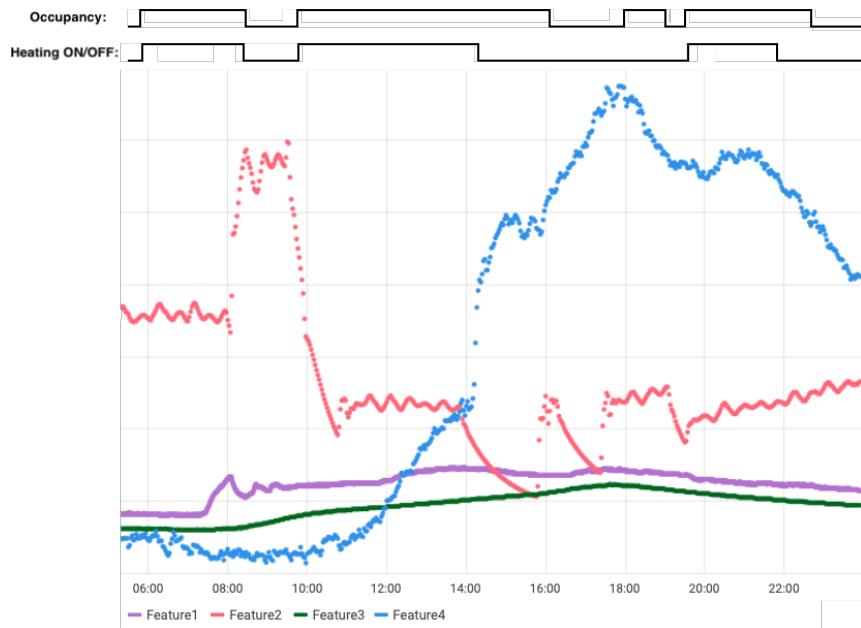


Figure 6.1: A diagram showing data of four unspecified features with the corresponding occupancy and heating status.

6.3. MODEL SELECTION

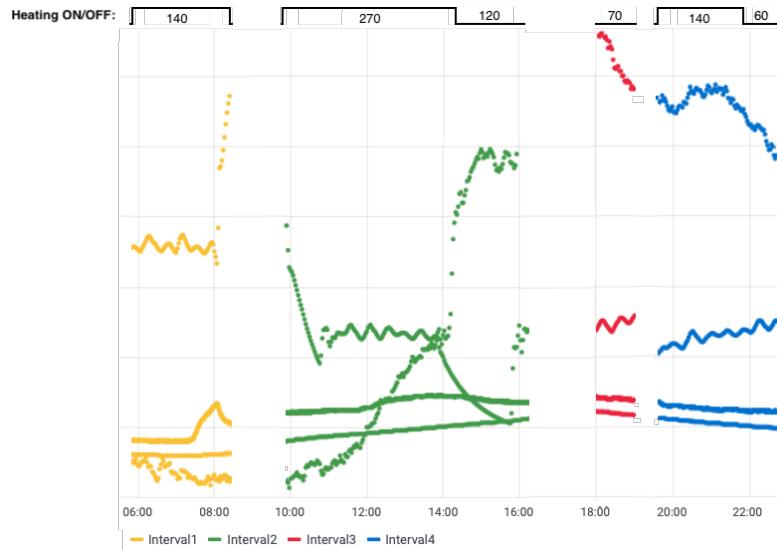


Figure 6.2: A diagram showing the data of Figure 6.1 separated into four *intervals of occupancy*, which are color-coded in yellow, green, red, and blue. In the top of the diagram, the target variable, heating activity is depicted, switching between “ON” and “OFF”. The respective lengths of consecutive heating states are displayed in-place as integers.

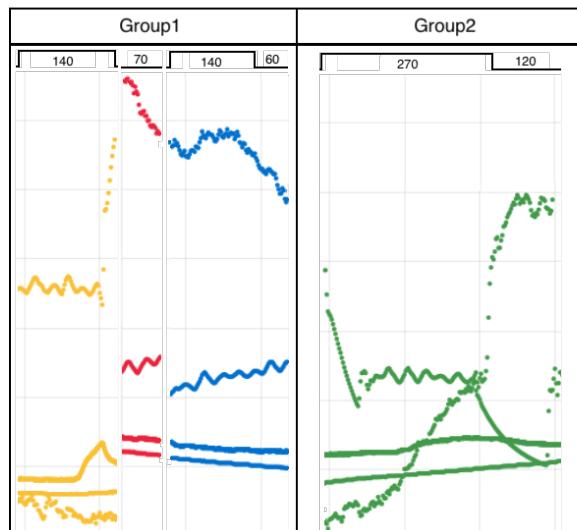


Figure 6.3: The groups to the interval-stratified 2-fold cross-validation for the data depicted in Figure 6.1.

	Training Set	Validation Set
Fold1	Group1	Group2
Fold2	Group2	Group1

Table 6.1: The training and validation split of an interval-stratified 2-fold cross-validation on the data shown in Figure 6.1 for the groups displayed in Figure 6.3.

6.3.2 Pre-Selection

The goal of the pre-selection in LATEST is to train a prototype for every ML method on an occupant’s collected data, without investing large resources in parameter optimization or model adaptation while validating the ML methods’ initial performance. Following this, the ML methods, whose prototypes have no promising performance, are not optimized in the more resource-intensive randomized grid search. This is considered the case for ML methods, whose model’s average f1-score in an interval-stratified k-fold cross-validation is smaller by more than 0.3 than the highest average f1-score of any other ML model trained.

Since in the domain of thermal comfort and temperature control, there is not one predominant ML method, LATEST trains and compares various ML methods for every occupant’s collected data. LATEST compares the following ML methods, which were introduced in Section 2.3 and were successfully applied in the literature described in Chapter 3. The exact implementations used are provided in the footnotes.

- Fully Connected Neural Networks⁴
- Gaussian Naive Bayes Classifier⁵
- Gradient Boosting Decision Trees⁶
- k-Nearest Neighbors⁷

⁴fastai Documentation, “tabular”, access March 6, 2020. docs.fast.ai/tabular.html

⁵scikit-learn API, “GaussianNB”, access March 6, 2020. scikit-learn.org/stable/modules/generated/sklearn.naive.bayes.GaussianNB.html

⁶scikit-learn API, “GradientBoostingClassifier”, access March 6, 2020. scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

⁷scikit-learn API, “KNeighborsClassifier”, access March 6, 2020. scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

- Logistic Regression⁸
- Random Forest⁹
- Support Vector Machine¹⁰

The most popular open-source frameworks for working with neural networks in Python are Google’s TensorFlow and PyTorch. They offer a wide range of functionality and adaptability and, having other frameworks built on top of them, they make training neural networks from a high-level possible. Examples of these higher-level frameworks are Keras¹¹ for TensorFlow or fastai¹² for PyTorch. They make it possible to build prototypes of neural networks without much development effort and while providing the ability to adapt implementation details. Hence, they meet the requirements of LATEST of creating a prototype for pre-selection and for optimizing the parameters afterward. We found that the combination of fastai and PyTorch fits this use case better, since fastai requires less specification for prototyping with built-in options for preprocessing methods, such as handling missing values, one-hot encoding, or standardization. Additionally, it provides a module specifically implemented for comparatively small data sets, called tabular, which provides many implementations needed for this use case.

The implementation used for all ML methods but the FCNNs are taken from the scikit-learn library, which is used in a large number of research projects [BLB⁺11]. The scikit-learn library also offers popular implementations for standardization¹³ and optimization¹⁴. Additionally, their pipeline module¹⁵ helps to efficiently integrate standardization into training and inference.

The initial parameters are oriented at the default values in the respective implementations. A rationale for these parameters can be taken from

⁸scikit-learn API, “LogisticRegression”, access March 6, 2020. scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

⁹scikit-learn API, “RandomForestClassifier”, access March 6, 2020. <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

¹⁰scikit-learn API, “SVC”, access March 6, 2020. scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

¹¹Keras, “Keras: Deep Learning for humans”, access March 6, 2020. keras.io

¹²fastai, “Making neural nets uncool again”, access March 6, 2020. www.fast.ai

¹³scikit-learn API, “StandardScaler”, access March 6, 2020. scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

¹⁴scikit-learn API, “3.2. Tuning the hyper-parameters of an estimator”, access March 6, 2020. scikit-learn.org/stable/modules/grid_search.html

¹⁵scikit-learn API, “Pipeline”, access March 6, 2020. scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html

the respective implementation’s documentation. Table 6.2 lists these initial parameters in the third column for every applied ML method. The parameter identifiers and their value-codes are adopted from their implementations.

6.3.3 Optimization

In LATEST, the scikit-learn implementation of the randomized grid search¹⁶ is used for parameter optimization. This optimization technique draws parameter values from specified probability distributions and trains a model for the resulting parameters. Since specifying fixed values in a conventional grid search is prone to the developer’s bias towards reasonable parameter options, randomized grid search was shown to provide results more efficiently [BB12]. For sampling integer features the geometric distribution *geom* is used:

$$\text{geom}(k \mid p) = (1 - p)^{k-1} p.$$

The exponential distribution *expo* is used for continuous features:

$$\text{expo}(x \mid \lambda) = \lambda e^{-\lambda x}.$$

Table 6.2 shows the exact distributions for all parameters in the columns “Distributions”. The parameter lists written in square brackets mark the options of a discrete uniform distribution. Hence, every option is equally likely to be drawn as a training parameter. While setting the parameter distributions, the iteration count of the randomized grid search must be adapted to the available computational resources. In this thesis, we train 200 models for every considered ML method.

¹⁶scikit-learn API, “RandomizedSearchCV”, access March 6, 2020. scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

Method	Parameter	Initial Value	Distribution
FCNN	tot_epochs	30	[30, 50, 100]
	layers	[200, 200, 150, 100]	[200, 200, 150, 100]
	max_lr	1e-4	[3.3e-4, 1e-4, 3.3e-5]
GNB	var_smoothing	1e-9	$expo(3.3e+9)$
GBDT	loss	'deviance'	['deviance', 'exponential']
	learning_rate	0.1	$expo(3)$
	n_estimators	100	$geom(6e-3)$
	subsample	1.0	[0.7, 0.9, 1.0]
	max_features	None	['auto', 'sqrt', 'log2']
kNN	criterion	'friedman_mse'	'friedman_mse'
	n_neighbors	5	$geom(6e-3)$
	weights	'uniform'	['uniform', 'distance']
	algorithm	'auto'	['auto', 'ball_tree', 'kd_tree', 'brute']
	leaf_size	30	$geom(2.5e-4)$
LR	P	2	[1, 2]
	penalty	'l2'	['l2', 'none']
	C	0.1	$expo(1e-2)$
	fit_intercept	True	[True, False]
	class_weight	None	[None, 'balanced']
RF	solver	'lbfgs'	['newton-cg', 'lbfgs', 'sag', 'saga', 'liblinear']
	n_estimators	100	$geom(6e-3)$
	criterion	'gini'	['gini', 'entropy']
	bootstrap	True	['True', 'False']
	max_features	'auto'	['auto', 'sqrt', 'log2']
SVM	class_weight	None	[None, 'balanced', 'balanced_subsample']
	C	1.0	$expo(1e-2)$
	kernel	'rbf'	['rbf', 'linear', 'sigmoid']
	gamma	'scale'	$expo(10)$
	max_iter	5000	[5000]

Table 6.2: The parameters and their distributions used for training the prototypes and optimizing the parameters. The table shows the parameters for all the considered ML methods' implementations.

6.3.4 Model Evaluation

In Section 5.2, the possibly conflicting NFR10 for scalability and NFR7 for accuracy were ranked for priority. In the model evaluation stage, these NFR are evaluated for the optimized models. Since NFR10 is ranked more important than NFR7, the single model between all ML methods with the highest average f1-score and which meets the runtime requirement of NFR7 will be deployed in the temperature control phase. This runtime requirement is generating a temperature control prediction with the corresponding reasoning in at most 250 milliseconds. How the reasoning is created is detailed in the following.

Reasoning Generation

FR4 specifies that LATEST gives a reasoning, when automatically controlling the occupant’s temperature. This requirement is motivated by providing a more transparent application, in which the occupants can build trust, since they know that the actions taken by LATEST are reasonable. For this use case, local interpretability suffices, since single predictions can be interpreted, for which there are more profound methods implemented [RSG16, LL17]. Additionally, LATEST needs model-agnostic interpretability for being able to interpret predictions by all its ML methods. In Section 2.6, the local, model-agnostic MLI methods LIME and Shapley values were introduced. Both fit for this use case and have Python implementations available, namely lime¹⁷ and SHAP¹⁸.

As explained in Section 2.6, LIME assumes linearity for the space of the ML model’s prediction, which does not hold in general. However, SHAP, which is built upon LIME, does not have that assumption making it generally more accurate [LL17]. Additionally, there is an implementation for SHAP specialized for tree-based ML methods that is highly performant. Hence, when the tree-based ML methods have a high performance, their reasoning can be generated very fast to ensure meeting the runtime requirements mentioned in the previous section. For these reasons, SHAP is used over LIME in LATEST.

¹⁷GitHub, “lime”, access March 7, 2020. github.com/marcotcr/lime

¹⁸SHAP, “SHAP”, access March 7, 2020. shap.readthedocs.io/en/latest

Chapter 7

Case Study

This chapter describes the objectives, design, results, and findings of a case study that was conducted as part of this thesis at the Intelligent Workplace (IW) at Carnegie Mellon University. The case study's subject is a personalized temperature control system with radiant heating panels, connected to a mobile application. The study was conducted with three test subjects in two phases: the data collection phase was performed over 6 weeks and the temperature control phase lasted 2 weeks. This separation of phases is required since ML models can generalize to unseen conditions only to a certain degree. In general, ML models are not expected to perform well in significantly changed conditions compared to the conditions they were trained on. Consequently, to provide comparable environmental conditions¹, the data collection phase was conducted between November 13, 2019, to February 7, 2020, excluding the Christmas break from December 16, 2019, to January 19, 2020, and the temperature control phase was performed from February 18, 2020, to March 6, 2020.

For this field study, LATEST was set up at the subjects' desks in their customary work environment.

7.1 Objectives

In the first phase of this case study, a data set of TCF was collected, while monitoring the test subjects' control behavior of a task infrared heating panel. This data collection was designed such that the resulting data represent various environmental conditions and are comparable to the conditions that were expected to be present for the temperature control phase.

¹Climate-Data.org, “Pittsburgh Climate”, access March 9, 2020. en.climate-data.org/north-america/united-states-of-america/pittsburgh-846

Another objective was to collect as much data as possible in the specified period, since larger data sets generally enhance the predictive performance of ML models. Therefore, subjects demand a simple, non-intruding way of providing feedback [KZS⁺18].

In Section 1.3, four objectives for enhancing occupants' temperature control experience have been introduced. Here, these objectives for the temperature control phase of LATEST are specified concretely.

- **Provide Control:** LATEST should enhance the subjects' temperature control experience by providing them the control of their task temperature.
- **Automate Temperature Control:** In the temperature control phase, LATEST should reduce the number of the subjects' commands by at least 30% compared to the data collection phase.
- **Provide Reasoning:** When controlling the temperature automatically, LATEST should provide a corresponding reasoning to the occupant to enable transparency, which improves the subjects' temperature control experience. This objective is evaluated in a survey at the end of the case study.
- **Meet Thermal Comfort Needs:** In the temperature control phase, LATEST should achieve the subjects' thermal comfort levels from the data collection phase or increase them.

Further, it is one objective of this project to gain evidence of which conditions should be met for the transition from the data collection phase to the temperature control phase. The conditions were determined by data set size, total time of thermal actuator use, and coverage of represented conditions in the data.

7.2 Data Collection Phase & Model Selection

This case study followed a quantitative analysis of TCF data, temperature control data, and thermal comfort and thermal sensation feedback data. Therefore, LATEST was extended by the functionality of collecting thermal comfort and thermal sensation feedback data. These data were used to evaluate the temperature control phase of LATEST in the second part of this case study.

7.2.1 Design

Three test subjects were randomly recruited at the IW to participate in this case study: two Ph.D. students and a professor. Two of them were female and one was male with an average Body Mass Index of 21.7. Two of the subjects were between 30 and 39 years old and the third subject's age lay between 50 and 59 years. The subjects' average shoulder circumference was 99 centimeters. The subjects were asked to choose their clothing according to a constant level of thermal resistance for the period of the case study. Additionally, they were advised not to change their clothes to influence their thermal comfort but to state their sensation as feedback instead and use the system to enhance any discomfort.

For this case study, the three subjects' desks at the open space of the IW were equipped with the following items. The setup implements the hardware software mapping of LATEST detailed in Section 5.4. Also, photographs of the desks' setup for two of the test subjects are shown in Figure 7.1.

- **Air Temperature and Relative Humidity Sensor:** The desks were equipped with ESP8266 microcontrollers and DHT22 temperature and relative humidity sensors at places that are relatively protected by expected air circulation. They measured the task air temperature and relative humidity at the level of the subjects' trunk.
- **Chair:** Every subject worked at their desk sitting down.
- **Infrared Radiant Heating Panel:** Under every desk, an infrared heating panel was placed, facing the subject. The distance to the chair's position was about 70 centimeters. The subjects were advised that the heating panels can get very hot and must not be touched with bare skin. These heating panels were plugged in with a plugwise Circle, which was connected to a server running the personal control unit of LATEST via a plugwise Stick.
- **iPhone SE:** Every subject got an iPhone SE for this case study. They ran the LATEST mobile application, which was connected to the same server as the plugwise Circles.
- **Microsoft Band 2:** Every subject got a Microsoft Band 2 for this case study. It was connected to the respective subject's iPhone SE via Bluetooth.
- **Reminder:** Every subject's setup included a printed reminder in DIN A5 format. This reminder can be found in Appendix A.1.

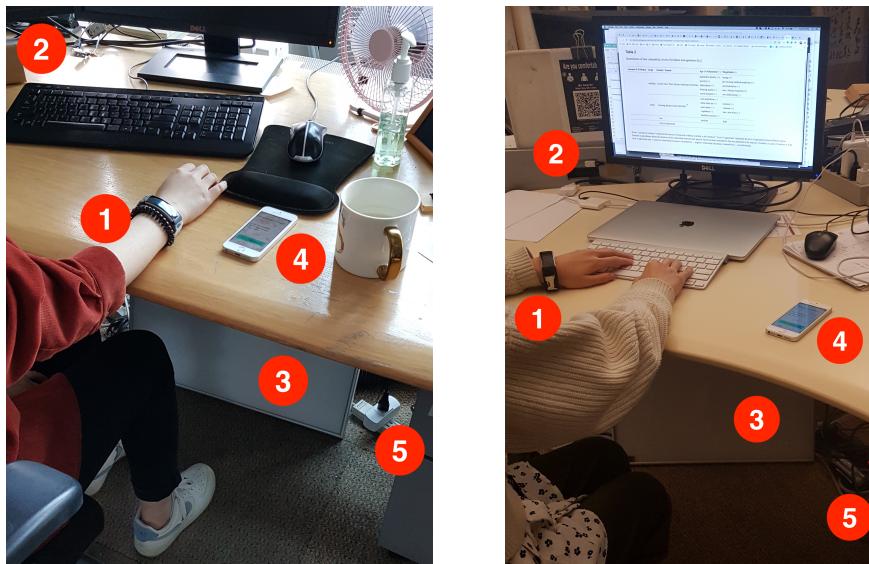


Figure 7.1: Two photographs of the setup of the data collection phase of this case study. There are red labels for (1) the Microsoft Bands at the subjects' wrists, (2) the fixed, white DHT22 air temperature and relative humidity sensors, (3) the infrared heating panel, (4) the iPhone SE running the LATEST mobile application, and 5 the plugwise Circle, controlling and monitoring the current to the radiant heating panel.

Apart from the subjects' desks, the IW was equipped with several sensors described in the following.

- **Ambient Temperature and Relative Humidity Sensor:** At the corridor connecting the three desks, an ESP8266 microcontroller with DHT22 temperature and relative humidity sensor was placed to measure the ambient temperature and humidity.
- **Weather Station:** The weather station, accessible by the OpenWeather API² with Station ID 5180905, is located at a distance of about 750 meters and provides measurements for the outside temperature, relative humidity, wind speed, atmospheric pressure, cloudiness, and a classification of the general weather condition.
- **Indoor Ambient Heating System:** At the IW, there was a water-based ambient radiant heating system, which was not controlled by

²OpenWeather, “Weather API”, access March 10, 2020. openweathermap.org/api

7.2. DATA COLLECTION PHASE & MODEL SELECTION

LATEST but by the building control system. The current temperature of the water in the pipe was monitored by a sensor, connected to openHAB via BACnet. The water was pumped through mullions at the window sides of each desk area. The mullions were located around the full floor at a distance of about 150 cm to each other, being the primary influence of ambient radiant temperature.

- **Ambient Temperature and Carbon Dioxide Sensor:** There were two sensors for air temperature and carbon dioxide at the top of the ceiling. They were located on the opposite side of the room as compared to the microcontroller at the corridor.

Within this setup, all sensor types of LATEST detailed in Section 5.5 were monitored every 30 seconds. The resulting measurements were sent to and stored persistently on a server. This server ran openHAB, influxDB, and the Python script for the full period of the case study. The only exceptions to this were the Christmas break and in the transition between the data collection phase and the temperature control phase.

Figure 7.2 shows the interface of the data collection version of the LATEST mobile application. There were two questions asked with one slider for the subject’s answer. The slider values were adapted from the 7-point thermal sensation scale of ASHRAE Standard 55. We used a reduced 5-point scale for thermal sensation, displayed at the top of the screen, and a 3-point scale for thermal comfort (“cooler”, “no change”, “warmer”) located below, which was successfully applied in other studies, described in Chapter 3. With these reduced scales, the application met the simplicity requirement NFR1 and minimized friction for giving feedback, which has been observed to be an issue for long-term data collection [KZS⁺18]. When a subject pressed the green “Send Feedback” button on the left side of the figure, an animation of a confirmation label was displayed, saying “Successfully saved – Thanks for your feedback!” This confirmation label disappeared after 1.5 seconds. At the same time, the subject’s feedback, which was represented by the two sliders, was sent to the server to be persistently stored for the evaluation of this case study. The slider states were color-coded, which can be seen when comparing the two sides of the figure. Additionally, on the interface on the right side, a push notification can be seen at the top. This push notification was sent if a subject did not give feedback in the previous 30 minutes, to get the subject’s attention for providing feedback.

The mobile application put the test subject in control of their infrared heating panel. When a subject stated that they would like the temperature to be adjusted to a warmer level with the lower slider and pressed the

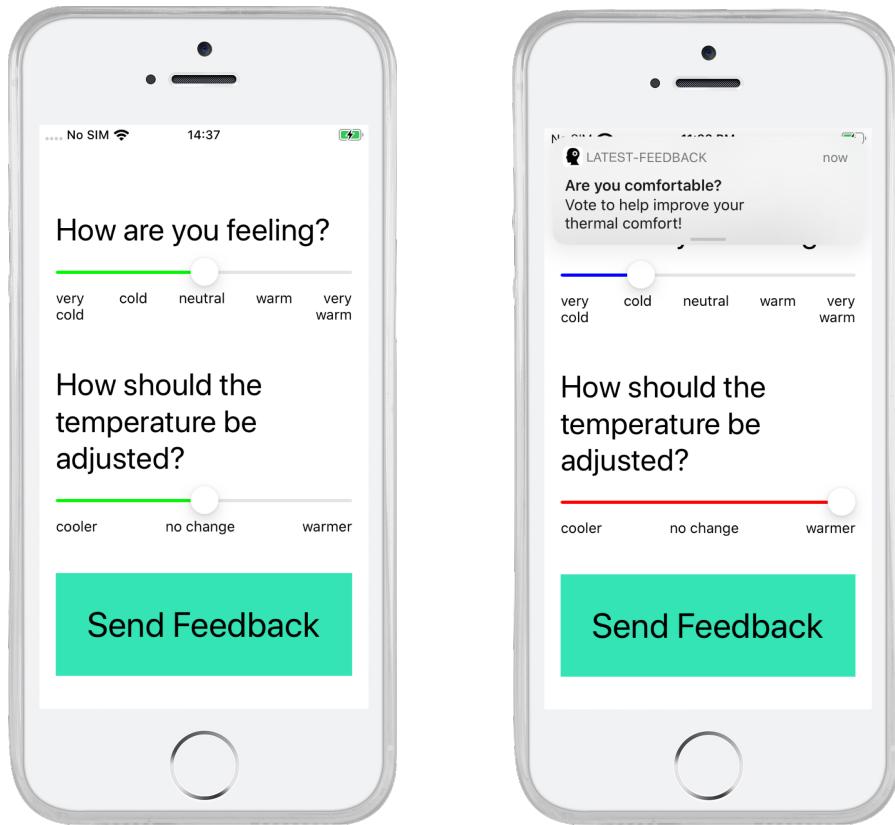


Figure 7.2: The interface of the LATEST mobile application for data collection. The display can be separated into three parts: the question for the subject's thermal sensation with a corresponding 5-point scale slider, the question for the subject's thermal comfort with a corresponding 3-point scale slider, and the green "Send Feedback" button.

"Send Feedback" button, the heating panel was turned on. Similarly, when a subject gave feedback that they liked it to be cooler, their heating panel got switched off via the plugwise Circle, if it was turned on before. The interface aimed to provide a simple means for the subjects to give feedback about their thermal sensation and thermal comfort, while they controlled their infrared heating panel. Therefore, there were only two sliders and one button included, which complied with NFR1. Additionally, to reduce even more friction for giving feedback for this case study, the iPhones were set to never lock their displays. Therefore, the subjects did not need to unlock the phones for giving feedback and had a visual cue of the lighted displays.

The LATEST mobile application was based on and was adapted from

7.2. DATA COLLECTION PHASE & MODEL SELECTION

the FETCh application, developed by Sebastian Liedl [Seb19]. As explained in Section 5.3, the occupancy – the information whether a subject is at their desk – was inferred by the LATEST mobile application. The subjects were advised to run the LATEST mobile application on their iPhones only when they are sitting at their desks. When the app was running, the iPhones requested biosignal measurements from the Microsoft Band every 30 seconds. The subjects were asked to give feedback regularly and to adjust the thermal actuator as soon as they felt uncomfortable.

The case study was designed such that after February 7, 2020, the data collection ended, due to the previously described shift in environmental conditions. Then, the transition to the temperature control phase was executed, which is described in Section 6.3. This transition featured the data pre-processing, training, model pre-selection, optimization, and model evaluation, which were described in detail in the previous chapter. Figure 7.3 depicts all processes and their outputs as a UML activity diagram. At the end of the transition, the best fitting ML model had the highest average f1-score while meeting the runtime requirement of NFR7, which generated a temperature control prediction with the corresponding reasoning in at most 250 milliseconds.

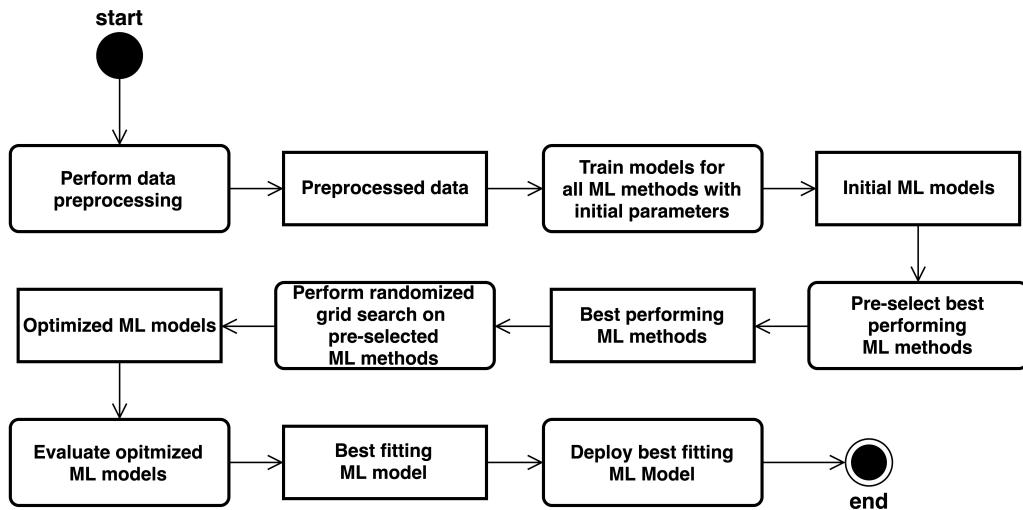


Figure 7.3: The processes executed in the transition to the temperature control phase, as described in Chapter 6.

7.2.2 Results

The setup detailed in the previous section was used to collect a data set consisting of thermal comfort and thermal sensation feedback data, temperature control data, biosignal data, and environmental data for three test subjects. In the specified time frame from November 13, 2019, to February 7, 2020, the data set was being collected, whose key indicators are shown in Table 7.1 and Table 7.2. Additional key indicators can be found in the Appendix in Table A.1.

Time of:	Occupancy	Heating	“too warm”	“comfortable”	“too cold”
<i>Subject 1</i>	30.2 h	0.8 h/2.7%	1.2 h/3.9%	28.1 h/93.0%	0.9 h/3.0%
<i>Subject 2</i>	20.6 h	4.2 h/20.4%	1.2 h/5.9%	16.7 h/81.1%	2.7 h/12.9%
<i>Subject 3</i>	171.7 h	61.1 h/35.6%	8.0 h/4.7%	137.6 h/80.1%	26.1 h/15.1%

Table 7.1: The results of the data collection phase expressed in the total time of occupancy, time of heating, and times of being in each of the three possible thermal comfort categories for every test subject. For the key indicators the corresponding shares of the subject’s total time of occupancy is shown as well. The time between two successive thermal comfort votes was assumed to belong to the former vote.

	Temperature Control Commands			Thermal Comfort Feedback	
	Count “ON”	Count “OFF”	Command Frequency	Total Count	Feedback Frequency
<i>Subject 1</i>	2	2	0.13 per hour	79	2.62 per hour
<i>Subject 2</i>	9	4	0.63 per hour	71	3.45 per hour
<i>Subject 3</i>	24	17	0.24 per hour	711	4.14 per hour

Table 7.2: The subjects’ commands to the system and the feedback gathered in the data collection phase.

The professor and the two Ph.D. students, who participated as test subjects in this case study, had different time schedules and thermal preferences. Therefore, the data sets of the three subjects were of differing sizes and unbalanced to different degrees, even though the ambient temperature is comparable across the parts of the open space at the IW. For *Subject 2*, data for 12 days were lost since their Microsoft Band 2 did not send any measurements. Similarly, on eight days within the data collection phase, the MSB of *Subject 1* did not work properly, resulting in loss of the data for these days. Furthermore, the test subjects occasionally forgot to turn off their mobile

7.2. DATA COLLECTION PHASE & MODEL SELECTION

applications and left behind their MSBs when leaving their desks. In these cases, the MSBs still sent mostly constant measurements every 30 seconds. To face this issue, ranges of constant measurements in skin temperature or heart rate over 10 minutes were excluded from the intervals of occupancy, defined in Section 6.3.

The remaining data were used to perform model selection. The specified ML models were trained on the collected data with the “Initial Values” from Table 6.2. The ML models were compared via their average f1-score in an interval-stratified k-fold cross-validation. Since the data for *Subject 1* were highly unbalanced with 97.3% belonging to target variable “Heater OFF”, k was set to 5 for all subjects for keeping comparability across subjects. Figure 7.4 shows the results of training the selected ML models expressed in the average f1-score across the folds.

As specified in the previous chapter, the ML methods, whose model’s average f1-score in the interval-stratified k-fold cross-validation is smaller by more than 0.3 than the highest average f1-score of any other ML model trained, are not included in the parameter optimization process. This was the case for the FCNN for all subjects and GNB for *Subject 3*. Hence, these methods were not optimized for the respective subjects in the randomized grid search that followed.

The parameter distributions for the randomized grid search are listed in Table 6.2. The results of the grid search with 200 iterations are depicted as boxplots in Figure 7.5. The boxes range from the first to the third quartile while outliers are not counted and depicted as circles. The maximum numbers of the f1-scores per ML method and test subject are shown in the Appendix in Table A.2.

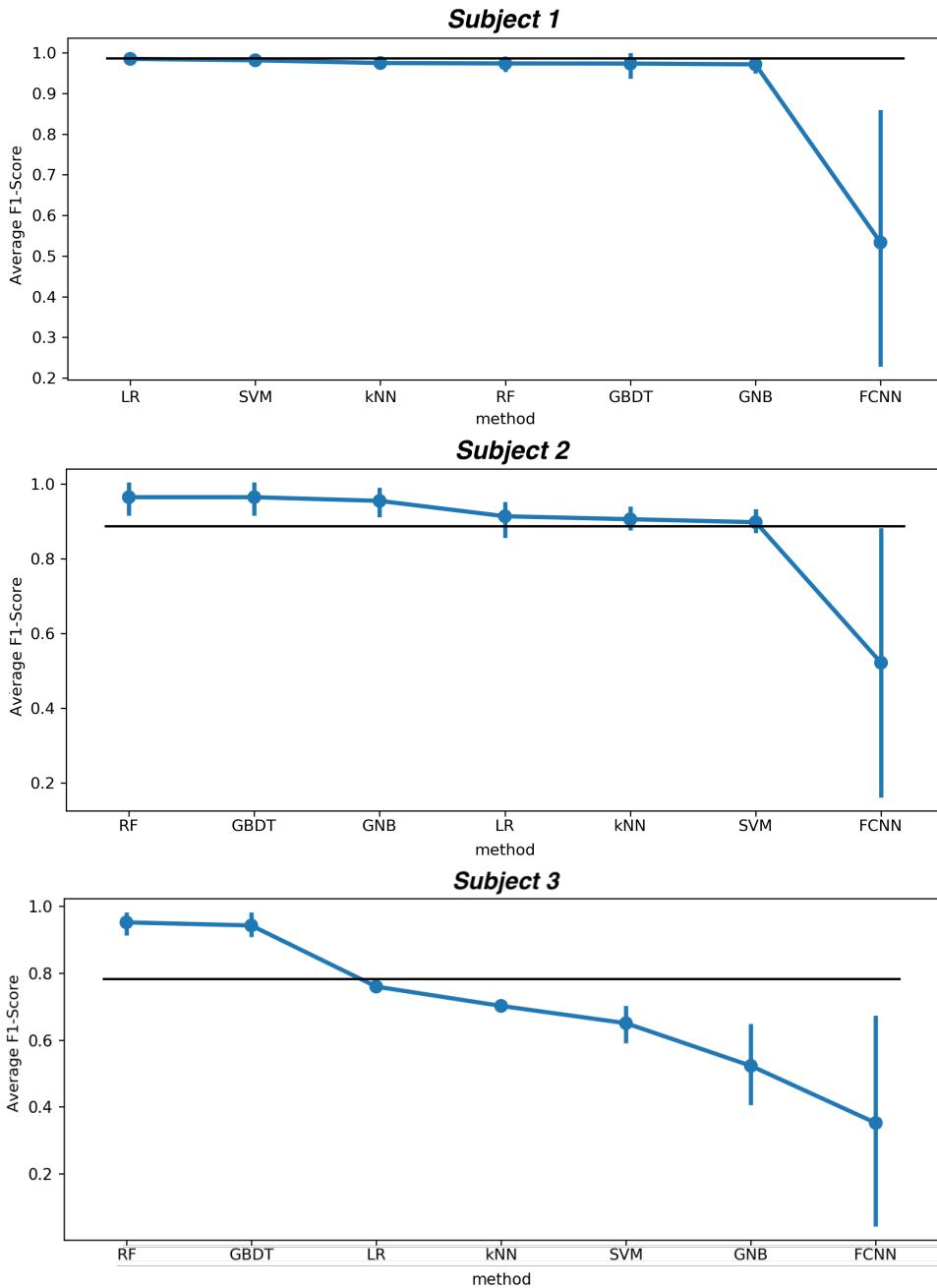


Figure 7.4: The average f1-score of the initial models. The depicted variance shows the variance in f1-score in the interval-stratified 5-fold cross-validation. The black horizontal line represents the f1-score of a naive model that always outputs the target value that is represented most often. The methods are ordered from left to right in descending average f1-score.

7.2. DATA COLLECTION PHASE & MODEL SELECTION

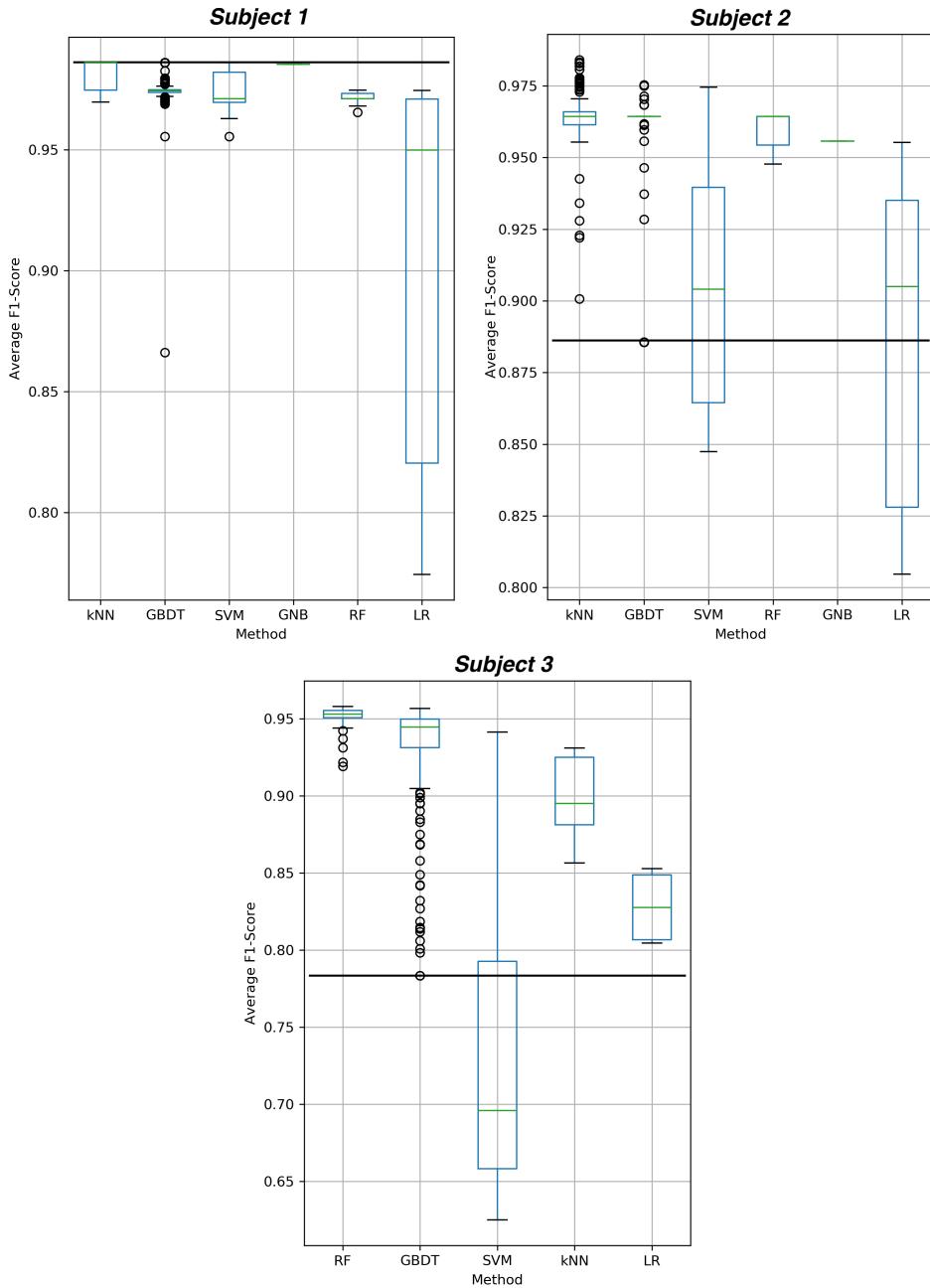


Figure 7.5: The results of a randomized grid search expressed as average f1-scores of interval-stratified 5-fold cross-validations. For each cross-validation there were 5 models trained, which was performed 200 times per ML method for every test subject. The black horizontal line represents the f1-score of a naive model that always outputs the target value that is represented most often. The methods are ordered from left to right in descending maximum average f1-score.

For *Subject 1* and *Subject 2* a kNN model had the highest average f1-score and for *Subject 3* it was an RF model. To evaluate whether these models complied with NFR7, a runtime test was conducted. 20 predictions and their corresponding reasonings with a SHAP interpreter were generated per model and their average runtime was compared against the 250-millisecond limit. The average runtime is shown in Table 7.3. Since the kNN models could not meet the requirement, the second-highest ML methods, which are GBDT in both cases, were additionally evaluated for *Subject 1* and *Subject 2*. For the tree-based models a more efficient implementation for SHAP was used, which was not applicable for other ML models [LEC⁺20]. Since the kNN did not comply with NFR7, the tree-based models RF and GBDT were deployed for the temperature control phase.

	Method	Average Runtime
<i>Subject 1</i>	kNN	341.5 ms
	GBDT	3.4 ms
<i>Subject 2</i>	kNN	298.4 ms
	GBDT	3.2 ms
<i>Subject 3</i>	RF	7.8 ms

Table 7.3: The average runtimes of generating one prediction with a corresponding SHAP reasoning in an experiment of 20 generations. Tree-based models were faster by a significant factor due to their implementation.

The implementations of RF and GBDT in scikit-learn offer a metric of feature-importance, which provides a weight of the influence of every feature while training the model. These weights sum up to 1 for all features. Figure 7.6 shows these feature importances per test subject aggregated by selected feature types. For instance, the moving average of air temperature measured at the desk of *Subject 2* with window size 25, which corresponds to the id `Subject2TaskAirTemperatureAverage-25`, is added three times: for temperature, trend-25, and task.

7.3. TEMPERATURE CONTROL PHASE

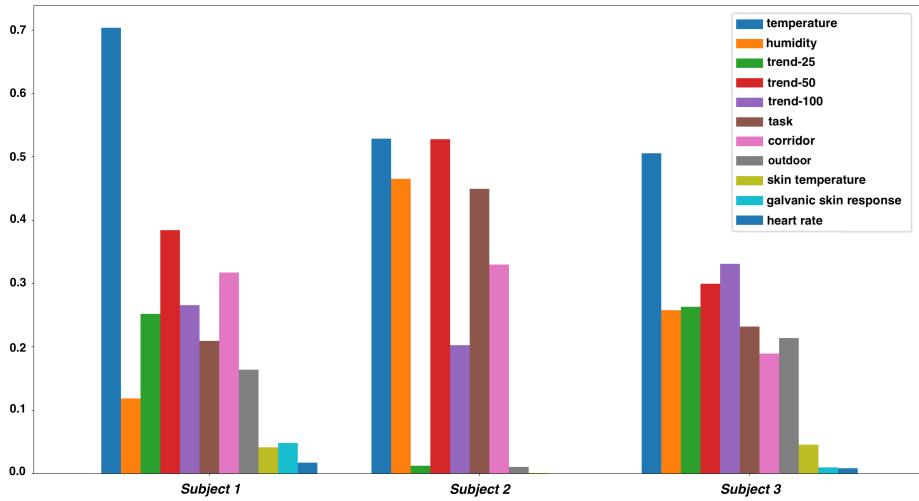


Figure 7.6: The feature importances of the deployed models aggregated by certain feature types for every subject. The value of these feature importances weight the influence a feature type had while a model was trained.

7.3 Temperature Control Phase

The temperature control phase followed the model selection process. This part of the case study evaluated the effects of LATEST on the occupant in terms of automation, usability, and accuracy while LATEST controlled the task temperature for every subject automatically. Both a real-time quantitative survey and a follow-up qualitative survey were conducted.

7.3.1 Design

The general setup of this part of the case study was the same as in the data collection phase. The differences were the automated control of the thermal actuator via ML models and the functionality of the LATEST mobile application. On a server, a personal temperature control ML model was deployed for every occupant, as described in Section 5.4. As explained in Section 6.2, these ML models require a live generation of trend features and one-hot features, which were used to train the respective models. Additionally, the LATEST mobile application was adapted as described in the following.

For the temperature control phase, the LATEST mobile application was extended by the “Info Screen”. This made the screen from Figure 7.2 the “Home Screen”, which kept the same interface while the functionality of controlling the infrared heating panel via the provided feedback button was

CHAPTER 7. CASE STUDY

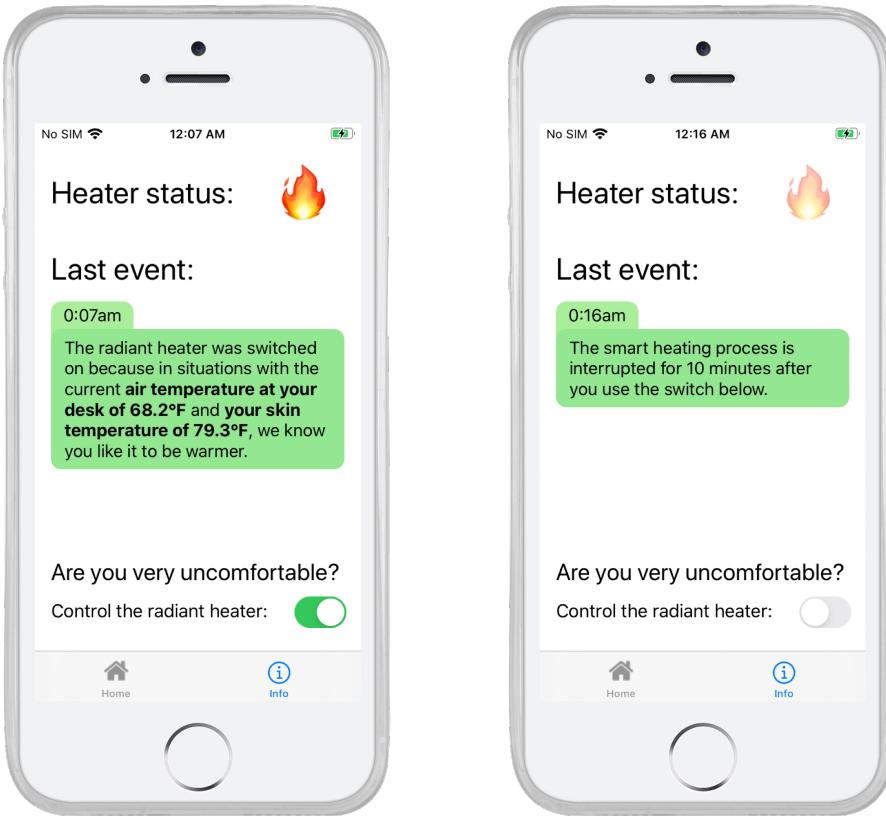


Figure 7.7: The interface of the temperature control version of the LATEST mobile application.

removed. Figure 7.7 displays two instances of the “Info Screen”. In the temperature control phase, the switch at the bottom of the screen provided the functionality for the subject to manually control the infrared heating panel. In the situation of the screen on the left side, the “Heater Status” is marked as on, which is indicated by a fire symbol that is greyed out if the heater is turned off – as in the screen on the right side.

Additionally, the “Info Screen” featured a text label, detailing a message with the corresponding time of its occurrence. On the screen on the left side, a reasoning message explains that the heating panel was just automatically switched on by LATEST. In the background, a tree-based ML model decided to switch the heating panel on under the current TCF conditions. A SHAP interpreter then processed the local prediction space and found that the task air temperature and the subject’s skin temperature were the strongest influences on this prediction. Both influences were in favor of the prediction and

7.3. TEMPERATURE CONTROL PHASE

were displayed in bold with their current values to provide transparency to the subjects. The message on the screen on the right side shows the situation when the subject was thermally very uncomfortable and used the switch to manually turn off their heating panel. In this case, the automated control was interrupted for 10 minutes.

With this setup, a quantitative analysis was conducted where the subjects gave feedback about their thermal comfort and thermal sensation while their task temperature was controlled automatically by ML models. The occupants could always interrupt this automated process.

After conducting the temperature control phase, a final survey was filled out by every test subject. The qualitative survey with nine questions was conducted in Google Forms³. The questions we found most important and their answers are shown in the following section. The remaining results of the survey can be found in Section A.4.

7.3.2 Results

The setup detailed in the previous section was used to collect a data set consisting of thermal comfort feedback data, thermal sensation feedback data, and temperature control data for three test subjects. In the specified time frame from February 18, 2020, to March 6, 2020, the data set was being collected, whose key indicators are shown in Table 7.5 and Table 7.6. Additional key indicators can be found in the Appendix in Table A.1. Table 7.4 shows the overall mean and standard deviation of the outside temperature and outside relative humidity for the full period of both phases of the case study.

Data Collection Phase			Temperature Control Phase	
	Mean	Standard Deviation	Mean	Standard Deviation
Temperature	36.0 °C	9.8 °C	35.8 °C	10.5 °C
Relative Humidity	75.7%	15.8%	62.9%	21.5%

Table 7.4: The mean and standard deviation of the measured outside temperature and outside relative humidity for both phases of this case study.

To finish the case study, the test subjects filled out a final survey after the temperature control phase was conducted. The answers to three of the questions are shown in Table 7.7 and Figure 7.8. These questions are of

³Google, “Create Forms”, access March 23, 2020. www.google.de/intl/en/forms/about

CHAPTER 7. CASE STUDY

Time of:	Occupancy	Heating	“too warm”	“comfortable”	“too cold”
<i>Subject 1</i>	33.3 h	0.0 h/0.0%	0.0 h/0.0%	33.3 h/100.0%	0.0 h/0.0%
<i>Subject 2</i>	12.5 h	0.0 h/0.0%	0.0 h/0.1%	10.7 h/86.0%	1.7 h/13.8%
<i>Subject 3</i>	74.6 h	10.1 h/13.5%	0.1 h/0.1%	66.4 h/89.0%	8.1 h/10.8%

Table 7.5: The results of the temperature control phase expressed in the total time of occupancy, heating, being too warm, being thermally comfortable, and being too cold for every test subject. For the key indicators the corresponding shares of the subject’s total time of occupancy are shown as well. The time between two successive thermal comfort votes was assumed to belong to the former vote. The displayed values are rounded.

	Manual Temperature Control Commands			Thermal Comfort Feedback	
	Count “ON”	Count “OFF”	Command Frequency	Total Count	Feedback Frequency
<i>Subject 1</i>	0	0	0.0 per hour	13	0.39 per hour
<i>Subject 2</i>	8	1	0.44 per hour	43	3.45 per hour
<i>Subject 3</i>	5	2	0.05 per hour	262	3.51 per hour

Table 7.6: The subjects’ commands to the system and the feedback gathered in the temperature control phase.

special interest for the study since they evaluate the system’s requirements. Their results are discussed in the following sections.

The first question shown in Table 7.7 asked for the test subjects’ evaluation of NFR11. The requirement of simplicity of the LATEST mobile application, which is defined in NFR1, was rated in the second question of Table 7.7. The question and answers listed in Figure 7.8, show how the system’s reasonings were perceived in this case study.

7.3. TEMPERATURE CONTROL PHASE

Question	Options	Count
<i>"How did the app's visual reactions feel for you?"</i>	"instantaneous" "low latency" "high latency"	3 0 0
<i>"How easy was it for you to understand how the app works?"</i>	"I understood how the app works instantly." "I figured the app out after a while." "I had trouble understanding how the app works without a manual."	3 0 0

Table 7.7: Two multiple-choice questions from the final survey and the participants' responses. The first question evaluates how the visual reactions of the mobile application were perceived by the test subjects, which was answered unanimously with "instantaneous". The second question regarded to the simplicity of the app. All three participants stated that they "understood how the app works instantly".

How did you perceive the reasoning written in the message field? Did it help you to understand on what basis the system decides to control the heating panel?

3 responses

Not much. There were variables that were out of my comprehension.

Did not notice the message field

yes, it did help, but I still hoped it could be turned on when I was cold

Figure 7.8: The final survey's question about how the reasonings were perceived and the three answer texts.

7.4 Findings

Table 7.4 shows the aggregated outside conditions prevailing while the two phases of this case study. The outside temperature varied by only 0.2°C in the mean and 0.7°C in the standard deviation between the two phases. Furthermore, the average relative humidity differed by 12.8% between the two phases. These statistics confirm our assumption that similar conditions prevail across the months between November and March in Pittsburgh, which proves that this period is well suited for heating-related research.

In the data collection phase, the amount of data for the subjects varied significantly. As explained in Section 7.2.2, the schedules of the test subjects allowed them to work at their desks to differing amounts of time. For instance, *Subject 1* was using the system for 33.3 h in the two weeks of the temperature control phase and 30.2 h in the 5 weeks of the data collection phase, resulting in a longer evaluation period than the training period. Adding to that, the amount of data collected for *Subject 1* and *Subject 2* was reduced significantly due to the malfunction of their MSBs. Kolar faced issues with the same hardware in their 2019 study [Kol19]. Consequently, the collected data set for *Subject 3*, whose MSB worked without major issues, was larger by a factor of 5 and 8 for *Subject 1* and *Subject 2*, respectively.

For *Subject 1* a very unbalanced data set was collected since they only heated in 2.7% of the time in the data collection phase. That is because *Subject 1* was comfortable with the ambient thermal environment in 93% of the data collection phase – even without using the task heater. As a consequence, the GBDT, which predicted *Subject 1*'s temperature control behavior, did not turn on the radiant heating panel for once in the 33.3 h of the temperature control phase, which is expected when training a model on such unbalanced data. This absence of heating was not perceived negatively by *Subject 1*, who stated to be comfortable in 100.0% of the temperature control phase. Therefore, the command frequency was reduced from 0.13 per hour to 0.0 per hour.

As mentioned before, the data set for *Subject 2* was crucially reduced due to hardware malfunction. Consequently, the training data for their temperature control behavior model covered only 20.6 h of system usage. In contrast to *Subject 1*, this data set was less unbalanced with a share of 20.4% heating. However, the GBDT that was deployed did not turn on the radiant heating panel at all in the relatively short temperature control phase of 12.5 h. This decision behavior might be explained by *Subject 2* having been relatively comfortable with the environmental and biosignal conditions in this period. That is, *Subject 2* stated to be thermally comfortable in 86.0% of the time in the temperature control phase, an increase of 4.9% compared to the data col-

lection phase, while at the same time the command frequency decreased by more than 30% to 0.44 commands per hour. Additionally, *Subject 2* stated to feel “too cold” in only 13.8% of the time, which is a relatively small increase of 0.9% considering the task heater was not used at all in the temperature control phase.

For *Subject 3* the largest data set was collected with 171.7 h and the highest feedback frequency of 4.14 feedbacks per hour. At the same time, the data collected were least unbalanced with a heating share of 35.6%. In the temperature control phase, the deployed RF increased the share of time *Subject 3* was thermally comfortable by 8.9%, while decreasing the command frequency to almost one-fifth of the original 0.24 commands per hour. These findings are the most interesting ones in this thesis. They show that with sufficient data, the approach is capable to successfully automate personal temperature control by improving thermal comfort while reducing occupant commands.

Additionally, evidence for the importance of the implemented TCF and feature types on temperature control behavior was collected. Figure 7.6 shows the deployed models’ aggregated feature importances. Even though single feature types significantly differ between the three models, general tendencies can be observed.

For all test subjects, temperature-related features were most important. Furthermore, the trend features’ aggregated importances surpass the importances of non-trend features: 0.90 for *Subject 1*, 0.74 for *Subject 2*, and 0.89 for *Subject 3*. Hence, the implementation of these trend features proved to critically enhance the models’ performances. Evaluating the influence of the outdoor factors, the models’ feature importances differ between the subjects: 0.16 for *Subject 1*, 0.01 for *Subject 2*, and 0.21 for *Subject 3*. In contrast, the biosignal factors **SkinTemperature**, **GalvanicSkinResponse**, and **HeartRate** only have small to insignificant aggregated influence on the models’ predictions: 0.11 for *Subject 1*, 0.00 for *Subject 2*, and 0.06 for *Subject 3*.

7.5 Discussion

In the scope of this case study, *Subject 1* was thermally comfortable even without using the system. Their data suggests that for individuals, who are generally thermally comfortable in their environment, LATEST does not decrease these thermal comfort levels. Though, no finding about how the system improves these individuals’ temperature control experience can be made for the absence of discomfort.

In general, the data sets for both phases for *Subject 2* were very small,

which resulted in poor informative value about the main objectives of this case study. We found that hardware malfunction is a critical weakness of the approach that should be handled in future applications.

However, *Subject 3* provided a high-quality data set for both phases of the case study over a total of 246.3 h with a feedback frequency of 3.95 feedbacks per hour and heating share of 28.9%. The findings for *Subject 3* provide evidence that the LATEST system can improve an individual's temperature control experience. This is shown by the four objectives defined in Section 7.1 in the following.

The first objective was to provide the occupant with the control over their task temperature. This objective is met by providing the unrestricted control of the task thermal actuator by the LATEST mobile application. However, the third response to a question in the final survey, displayed in Figure 7.8, suggests that one test subject was not aware of this functionality in the temperature control phase, when the reasonings were displayed. This conflicts with the responses to the second question in Table 7.7, which indicates that all three test subjects thought they understood the system's functionality instantly. Hence, we conclude that the system's interface could be further improved to make the option of controlling the task thermal actuator more obvious to the occupant. This might be achieved by placing the switch for controlling the task heating panel, displayed in Figure 7.7, to the top of the "Info Screen".

The second main objective is the automation of the temperature control. This functionality reduced the frequency of temperature control commands for *Subject 3* by 79.2% to 0.05 commands per hour. Hence, the desired goal of reducing this frequency by 30.0% was surpassed significantly.

Third, a reasoning for the automated temperature control was provided, to enhance transparency in the system's decision-making. As shown in the excerpt of the final survey in Figure 7.8, this functionality did not fully achieve the desired effect. When asked for the gain of the reasoning in the message field, one test subject stated, "Not much. There were variables that were out of my comprehension". This suggests that the usage of the message field with a reasoning containing the two most important variables and their values does not completely serve the desired purpose of providing transparency. To improve this in future work, a table containing the most important influences, their current values, and the magnitude of their influences as Shapley values might be considered.

The fourth objective is meeting the individual thermal comfort needs. For *Subject 3*, the time being thermally comfortable was increased from 80.1% in the data collection phase to 89.0% in the temperature control phase. We consider this 8.9% increase in comfort levels along with the reduction of the

subject's command frequency by 79.2% as the major finding of this thesis. When sufficient data about an occupant, who occasionally uses their task thermal actuator, are collected without hardware malfunction, LATEST can automate their personal temperature control to the occupant's comfort.

An additional objective of this case study was to gather evidence for favorable conditions of the transition between the data collection phase and the temperature control phase. Since the collected data set for *Subject 2* did not prove to be sufficiently large, which it did for *Subject 3*, we suggest that the transition should not be done with a smaller data collection period than 25 hours. Generally, the data collection phase should take place as long as possible, while covering the expected conditions for the temperature control phase, which extends the findings from Kim et al. [KZS⁺18]

Further, the findings for *Subject 1* suggest that the ML model deployed, will not provide valuable knowledge from severely unbalanced data (>97%) but will keep the heating panel in the predominant state. Hence, it should be pursued to collect data of the underrepresented state. For instance, in future work, the data collection could be influenced by changing the ambient temperature control system's settings for a specified amount of time.

Also, as described in the previous section, we did not find that biosignal factors have significant importance for the deployed models' predictions. In contrast, the implemented trend features were found to be of critical importance, proving their informative value for RF and GBDT in personal thermal comfort.

Furthermore, evidence was gathered about which ML methods are fitting for the domain of thermal comfort and temperature control.

In the model pre-selection (Figure 7.4), the FCNNs had by far the worst average f1-scores and the highest corresponding variances. Hence, the collected data and chosen FCNN architecture produced a different result than what was observed by Francis et al. [FQvF⁺19] We assume that the collected data sets were too small for applying FCNNs and that a more resource-intensive optimization would be necessary along with more complex architectures, such as recurrent neural networks, long short-term memory, or convolutional neural networks.

In the same stage, for the training data of *Subject 3*, the two tree-based methods, RF and GBDT, were the only models that scored a higher average f1-score than a naive guesser, which would always turn the heater off independent from the input. Since we consider this data set to have the highest informative value in this case study, we find that these two methods are well fit in the domain of temperature control. This is further supported by the evidence displayed in the three box plots in Figure 7.5. In the Figure, the optimization results of RF and GBDT score among the highest average

f1-scores with comparably small variances across all three test subjects. Additionally, as detailed in Section 7.2.2, their SHAP implementations are faster by a factor of 100. Therefore, we recommend the use of these tree-based ML methods in comparable temperature control projects with an interpretability component.

7.6 Limitations

The main limitation of this case study was the sample size. The collected data sets for both *Subject 1* and *Subject 2* were reduced significantly in size due to hardware malfunction, making them insufficiently small for the applied ML methods to converge. Consequently, these hardware malfunctions decreased the informative value of this case study. However, we do not know of any other more reliable commercial smart bands that measure skin temperature or GSR. Furthermore, our research budget did not fit additional hardware. Therefore, after conducting the data collection phase, when one MSB was not working, *Subject 1* and *Subject 2* exchanged the remaining MSB according to their schedules. The impact of the malfunctioning of the MSBs increased since they were used to infer occupancy. In future work, occupancy sensors should be considered.

Another limitation was the runtime of generating Shapley values for kNN. Even though kNN models scored the highest f1-scores for *Subject 1* and *Subject 2*, they were not deployed because they did not meet the runtime requirement. If a faster Shapley value calculation for kNN, similar to [LEC⁺20], will be available in the future, deploying kNN models is expected to further improve the system’s accuracy.

Additionally, even the largest data set for *Subject 3*, which was not decreased in size by hardware malfunction, did not prove fit for the application of FCNNs, which could be expected considering the findings of Francis et al. [FQvF⁺19]. To address this in future work, we emphasize collecting larger data sets and train more complex architectures, such as long short-term memory or convolutional neural networks.

Furthermore, the reasonings shown in the message field did only partially improve the occupant’s experience, which we conclude by the answers to a question in the final survey, depicted in Figure 7.8. However, we did not observe the issue described by Yang and Newman, who found that temperature control automation leads to the occupant’s expectation of transparency [YN13]. If more sophisticated ML interpretability methods are available in the future, true transparency of the ML model’s decision-making might be provided to the occupant.

Chapter 8

Summary

Thermal comfort was shown to significantly influence an occupant's performance, well-being, and satisfaction. A variety of factors influence an individual's thermal comfort, such as air velocity, clothing insulation, or skin temperature [Fan70, Cho10, CL12]. However, most current heating, ventilation and air-conditioning (HVAC) systems still only allow occupants to control one factor – air temperature. Therefore, even though we spend large amounts of energy on IEQ regulation, occupants are still uncomfortable [HAZA06].

In this thesis, we propose LATEST, a personalized machine learning-based temperature control system. Its central objective is to automate personalized temperature control while meeting thermal comfort for multiple occupants simultaneously. Therefore, it makes use of task thermal actuators and state-of-the-art machine learning methods. Furthermore, LATEST gives reasonings for its actions to provide transparency of its decision-making processes. In a field case study, we found that the approach can reduce occupant command frequency by 79% while increasing thermal comfort levels by 9%.

8.1 Status

The central objective of this thesis was to improve the individual's temperature control experience by automating temperature control while keeping thermal comfort levels. To realize this goal, LATEST was designed to implement the FR and NFR that were specified in Chapter 4. Which of these requirements were implemented successfully, is explained in Subsection 8.1.1. Subsection 8.1.2 contains descriptions of the goals that are still open or were only partially achieved.

8.1.1 Realized Goals

In the case study described in the previous chapter it was detailed, how all FR of the system were implemented. In particular, LATEST controlled the temperature automatically (FR1) according to an individual test subject's previously observed temperature control behavior (FR8) and gave a reasoning for every made decision (FR4). This automation could be interrupted by the test subject's temperature control commands (FR2). All these temperature control commands were executed by task infrared heating panels (FR3). Furthermore, LATEST monitored biosignal (FR5), indoor environmental (FR6), and outdoor environmental (FR7) conditions. Additionally, occupancy was detected indirectly by the test subject's use of their MSB (FR9), which resulted in switching off the thermal actuators, when no occupant was detected (FR10). Each of these functionalities was running in parallel for multiple occupants at once (FR11).

The systems most important identified design goals (Section 5.2) were the NFRs “Safety” and “Robustness to Invalid User Input”. These design goals were achieved by the use of infraNOMIC Frame Line infrared heating panels as thermal actuators, which comply with the UL 499 Standard for Electric Heating Appliances¹. Each of these panels integrates three α Therm Thermal Protectors ST-22², which shut the panels off in case of overheating. Additionally, the system implements the functionalities to shut off the heating panel if there is no occupant (FR10) and to shut off all heating panels at 9 pm. Hence, in case of a hardware defect or if an occupant forgot to turn off the heating panel before they left, no harm can be caused and energy loss is limited.

Furthermore, “Availability” was identified as the fourth most important design goal. It is realized by the use of highly available software: Ubuntu, iOS, Swift, Python, influxDB, and openHAB. The system’s availability further depends on the Wi-Fi connection between the environmental sensors and the server. In the case study, the system and all its components were available over the full period.

As explained in Section 7.2.2, in the case study, the design goal of “Scalability” was reached by selecting the tree-based ML models over kNN for two test subjects because the inference and reasoning generation of kNN did not meet this requirement.

Further, the design goal of “Response Time” was quantitatively eval-

¹UL standards, “Standard for Electric Heating Appliances Purchase UL 499”, access March 3, 2020. standardscatalog.ul.com/standards/en/standard_499_14

² α Therm, “Thermal protector ST-22 series”, access March 30, 2020. www.alphatherm.de/datenblaetter/Thermoschalter/ST-Serie-en.pdf

uated in the case study’s final survey. The result can be seen in the first question of Table 7.7, showing that all three test subjects stated, they felt the applications visual reactions to be instantaneous. This design goal was not measured qualitatively.

The design goal of “Accuracy” was also quantitatively evaluated in the case study. Though the less accurate GBDT were chosen over kNN for *Subject 1* and *Subject 2* in favor of decreasing runtimes, all three test subjects were thermally comfortable for an increased share of time in the temperature control phase.

Additionally, the design goal of “Extensibility” was achieved by using extensible open-source software. As explained in Section 5.4, openHAB is extensible by design³, offering compatibility for a large number of sensors and actuators. The same accounts for influxDB and the scikit-learn and pandas libraries used in the Python script. By their use, preprocessing more and different kinds of sensor and actuator data is possible without adjustments.

Furthermore, the design goal of “Robustness to Outliers” was achieved by applying a moving median filter on the sensor data, as explained in Section 6.2.

8.1.2 Open Goals

Some goals were only partially reached. One of them is the design goal of “Security”, which was ranked of high importance. As explained in Section 5.6, the personal data stored on the influxDB instance, can only be accessed by an occupant’s own mobile Swift application, the openHAB instance, and the python script. The iPhones running the Swift applications are password-secured. The openHAB instance can only be accessed via HTTPS and a password-secured user. The python script runs on a server, which was accessed via Secure Shell. However, definite security cannot be reached. If an attacker gets into the openHAB instance or the server, running openHAB, influxDB, and the python script, they are, in theory, able to access all data for every occupant.

Furthermore, the design goal of “Robustness to Sensor Failure” was only partially reached. As explained in Section 6.2, missing sensor data were interpolated or filled with the previous measurement. Consequently, in the case study (Chapter 7), we did not lose any data to missing environmental measurements. Yet, the simultaneous failure of all three biosignal sensors of the MSB resulted in loss of occupancy information. In future work, this issue can be avoided by using a more reliable smart watch or additional occupancy

³openHAB, “Add-on Reference”, access March 30, 2020. www.openhab.org/addons

sensors.

Also, the goal of simplicity of the user interface was only partially reached. The mobile application was designed, such that every functionality can be executed by at most three actions of the occupant. This is supported by the responses to a question of the case study's final survey, listed as the first question in Table 7.7, which shows that all three test subjects stated, they understood the system's functionality instantly. However, the third response to a different question in the final survey, displayed in Figure 7.8, suggests that one test subject was not aware of the functionality of controlling the heating panel manually in the temperature control phase. Hence, we conclude that the system's interface could be further simplified to make the option of controlling the task thermal actuator more obvious to the occupant. This might be achieved by placing the switch for controlling the task heating panel, displayed in Figure 7.7, to the top of the "Info Screen".

8.2 Conclusion

A large share of the available energy is consumed by temperature control without meeting thermal comfort for many individuals. This is due to current HVAC controls' lack of addressing the individual's thermal comfort needs and the range of TCF, which they do not integrate. This thesis contributes to solving this problem by showing that it is possible to meet individual thermal comfort with the combination of state-of-the-art ML controls and task thermal actuators, while putting the individual in control of their task environment. The case study we conducted provides evidence that centralized, ambient HVAC controls can be replaced by personalized, task controls and that this process can be automated while meeting the individual's thermal comfort needs. Hence, we achieved one step towards a more efficient and convenient distribution of energy for temperature control, which will result in saving energy while increasing thermal comfort conditions and decreasing the need for human interaction.

8.3 Future Work

To reach the goal described in the previous section, more research in this field is needed.

A crucial step would be to conduct a similar study with more participants and more interconnected environments. In many office buildings, people share open spaces with desks close to each other. It is of interest to find

8.3. FUTURE WORK

whether infrared heating panels can control an individual's task environment without having an unwanted effect on their neighbor's task environment. For this study, we propose to use heating panels with more than 210 watts, which we found insufficient in some situations.

Additionally, to evaluate the energy saved by this approach, a future study should include the management of the ambient temperature control system. Such a study may be divided into two parts. In one part, the ambient temperature is controlled centrally without task thermal actuators, while in the other part, the ambient system's activity is decreased and every occupant has control over their personal task thermal actuator, which is automated later on as was done in this project. If such a study shows that in the first part significantly more energy is consumed, while thermal comfort levels are lower than in the second part, a potent argument addressing direct financial interests for developing and deploying decentralized temperature control systems can be made.

Furthermore, this body of work addressed a range of TCF which could be further expanded for both actuators as well as sensors to further increase comfort accuracy. Additionally, when integrating both heaters and coolers a study over multiple seasons could be conducted, which would produce large data sets and is expected to prove a high degree of personalization and high comfort accuracy. Hence, future work includes the extension of the approach by task coolers, such as personal fans or air conditioners, monitoring more TCF, such as clothing insulation or radiant temperature, and expanded study periods.

Appendix A

Case Study

The following contains additional resources and key indicators from the case study. Figure A.1 shows the reminder sheet, which was printed and handed out to all test subjects. Next, Table A.1 contains key indicators from all collected data for both phases and each test subject. Table A.2 shows the maximum f1-scores in the randomized grid search for every test subject. The parameters of the ML models that were deployed are shown in Table A.3. Lastly, Section A.4 contains questions and responses of the final survey, which were not discussed in Chapter 7.

A.1 Reminder Sheet

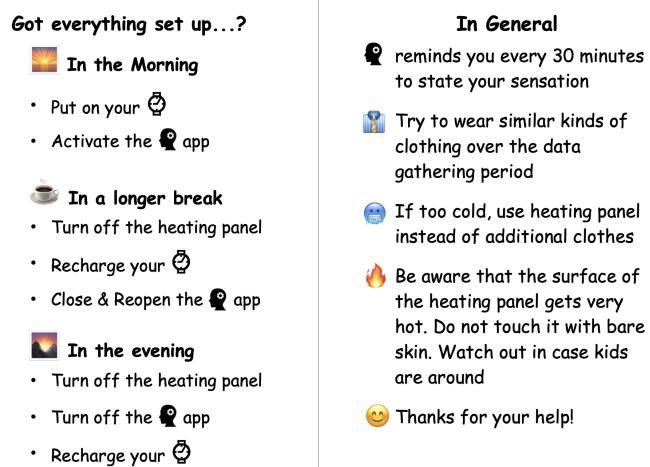


Figure A.1: A reminder that was printed in DIN A5 format and handed out to every test subject for the case study.

A.2 KEY INDICATORS

A.2 Key Indicators

Data Collection Phase															
Subject 1	CO ₂	AT	Cl	ORH	AP	OAT	WS	TRH	TAT	CRH	CAT	AHVAC	GSR	HR	ST
mean	799.7	72.7	68.9	70.3	30.0	38.1	7.5	30.4	73.1	31.2	71.1	95.4	71139.2	78.6	30.7
std	138.1	1.5	36.5	18.1	0.3	9.1	3.9	3.9	1.7	4.8	1.4	16.0	128800.7	14.7	1.4
min	514.9	69.8	1.0	29.0	29.3	23.1	0.6	23.2	67.0	22.2	67.0	65.1	188.0	37.0	22.6
25%	708.9	71.3	75.0	57.0	29.7	32.5	4.6	27.4	71.7	28.5	70.3	86.6	2500.0	72.0	30.0
50%	787.7	72.8	90.0	71.0	30.0	36.1	6.9	30.0	73.0	30.6	71.2	90.2	6515.0	75.8	31.0
75%	872.5	73.9	90.0	86.0	30.2	42.2	10.2	32.0	74.3	33.0	71.9	105.2	26863.0	81.0	31.6
max	1176.9	77.0	90.0	100.0	30.7	62.3	21.9	44.4	76.8	49.0	74.1	127.3	340330.0	175.0	34.0
Subject 2	CO ₂	AT	Cl	ORH	AP	OAT	WS	TRH	TAT	CRH	CAT	AHVAC	GSR	HR	ST
mean	797.1	73.3	54.4	65.7	30.0	44.0	9.1	34.7	71.9	33.0	70.7	84.4	139201.2	76.0	31.6
std	107.7	0.9	38.7	18.3	0.2	5.2	3.5	6.8	2.3	7.0	2.0	13.3	162247.7	6.9	1.5
min	526.4	70.4	1.0	38.0	29.6	32.6	2.0	25.3	65.3	24.1	65.3	62.1	276.0	61.0	25.0
25%	720.0	73.1	5.0	51.0	29.7	40.9	6.9	28.5	71.0	26.7	70.3	75.7	1346.0	72.0	30.8
50%	791.7	73.4	75.0	62.0	30.0	42.7	9.1	33.5	72.5	33.0	71.5	88.0	5622.0	75.0	31.7
75%	873.0	74.0	90.0	80.0	30.1	49.3	11.4	39.4	73.4	35.9	72.1	89.5	340330.0	79.0	32.7
max	1079.1	74.5	90.0	100.0	30.2	53.8	21.9	51.7	74.8	48.0	74.6	120.1	340330.0	156.0	34.1
Subject 3	CO ₂	AT	Cl	ORH	AP	OAT	WS	TRH	TAT	CRH	CAT	AHVAC	GSR	HR	ST
mean	779.1	72.4	66.9	71.0	29.9	38.4	8.1	34.9	70.8	32.3	70.7	89.7	200671.6	71.8	30.7
std	126.5	1.6	36.7	16.4	0.2	8.6	4.0	5.4	1.8	5.8	1.8	15.8	160013.6	5.9	1.6
min	504.5	64.2	1.0	30.0	29.2	14.4	0.2	24.5	64.0	22.1	63.8	58.1	230.0	36.0	23.7
25%	699.1	71.3	40.0	58.0	29.7	32.9	4.6	31.2	70.1	28.7	69.9	83.5	9835.0	69.0	29.6
50%	771.9	72.6	90.0	71.0	30.0	36.9	8.0	34.7	71.0	32.2	71.0	86.8	340330.0	72.0	30.6
75%	859.5	73.2	90.0	86.0	30.1	43.2	11.4	37.5	71.9	34.0	71.7	97.2	340330.0	74.0	31.9
max	1178.9	78.2	90.0	100.0	30.7	62.3	21.9	72.4	76.2	52.0	75.2	124.2	340330.0	155.0	35.0
Temperature Control Phase															
Subject 1	CO ₂	AT	Cl	ORH	AP	OAT	WS	TRH	TAT	CRH	CAT	AHVAC	GSR	HR	ST
mean	745.3	71.8	69.0	64.6	30.0	36.5	8.8	26.5	72.0	26.0	70.4	89.4	34966.6	73.8	30.4
std	106.4	0.9	34.3	21.8	0.2	10.3	5.0	4.2	1.1	5.7	2.6	11.8	93391.7	6.2	1.5
min	547.3	70.6	1.0	20.0	29.7	22.8	1.6	18.7	70.5	17.0	60.2	71.1	956.0	47.0	22.6
25%	667.8	71.2	40.0	46.0	29.7	26.6	5.8	23.9	71.0	22.2	69.4	83.7	2706.5	70.0	29.6
50%	731.9	71.4	90.0	68.0	29.9	29.6	8.0	25.1	71.5	23.7	70.5	88.6	4550.0	74.0	30.4
75%	798.8	72.3	90.0	81.0	30.4	46.8	11.4	30.2	72.6	29.9	72.1	90.2	8923.0	77.0	31.6
max	957.3	74.3	90.0	100.0	30.5	52.1	23.0	35.5	74.8	39.4	80.5	125.5	340330.0	102.0	33.2
Subject 2	CO ₂	AT	Cl	ORH	AP	OAT	WS	TRH	TAT	CRH	CAT	AHVAC	GSR	HR	ST
mean	697.6	72.7	53.6	47.4	30.0	45.8	12.2	27.3	72.7	26.0	71.9	79.2	7847.6	74.3	30.8
std	56.4	0.6	37.2	14.6	0.2	8.6	5.9	2.7	1.7	3.1	1.4	7.5	43888.1	6.8	1.2
min	606.1	70.9	1.0	20.0	29.7	28.7	1.9	19.0	69.2	16.7	64.0	69.7	198.0	56.0	25.9
25%	644.3	72.4	1.0	37.0	29.7	39.1	6.9	26.2	71.5	24.5	71.4	71.6	491.0	70.0	29.9
50%	700.1	72.5	75.0	43.0	29.8	48.6	11.4	27.7	72.8	26.2	72.1	81.4	623.0	72.0	30.7
75%	740.3	73.1	90.0	57.0	30.3	53.0	17.2	29.5	73.5	28.2	72.8	85.6	1167.0	79.5	31.6
max	807.9	73.9	90.0	87.0	30.4	56.7	25.2	31.7	78.2	33.7	80.0	114.5	340330.0	110.0	33.5
Subject 3	CO ₂	AT	Cl	ORH	AP	OAT	WS	TRH	TAT	CRH	CAT	AHVAC	GSR	HR	ST
mean	695.0	72.3	60.87574	63.0	29.9	40.2	9.5	31.7	70.6	28.1	71.3	81.7	180918.8	69.8	29.2
std	96.8	0.9	38.0	19.2	0.2	10.4	4.8	5.4	1.1	6.4	1.8	11.4	161231.7	4.3	2.6
min	502.3	69.3	1.0	27.0	29.4	16.3	1.0	24.1	66.5	17.4	59.3	67.2	358.0	52.0	22.9
25%	628.3	71.5	1.0	45.0	29.7	29.8	5.8	27.7	69.8	23.7	70.3	72.1	13321.0	67.0	28.4
50%	700.1	72.4	90.0	63.0	29.8	40.8	9.1	31.5	70.6	27.2	71.7	82.5	239661.0	71.0	29.5
75%	759.1	72.7	90.0	76.0	30.3	49.0	11.4	33.9	71.4	30.5	72.3	85.7	340330.0	72.0	31.1
max	958.1	74.4	90.0	100.0	30.5	56.7	25.2	46.0	73.4	47.5	81.5	122.0	340330.0	123.0	33.5

Table A.1: The key indicators of the data collected for each test subject in both phases. The used abbreviations are defined in Table 3.2. The values for air temperature and the temperature of the ambient HVAC system are in degree Fahrenheit.

A.3 Phase Transition

A.3.1 Randomized Grid Search Performances

	GNB	GBDT	kNN	LR	RF	SVM
<i>Subject 1</i>	0.985	0.986	0.986	0.974	0.975	0.986
<i>Subject 2</i>	0.956	0.975	0.984	0.955	0.964	0.975
<i>Subject 3</i>	N/A	0.957	0.931	0.853	0.958	0.941

Table A.2: The maximum f1-scores of the randomized grid search for each ML method and test subject. For *Subject 3*, GNB was not optimized because it did not meet the performance requirement of pre-selection.

A.3.2 Optimized Parameters

Subject	Method	Parameter	Optimized Value
<i>1</i>	GBDT	loss	'deviance'
		learning_rate	0.0
		n_estimators	76
		subsample	0.9
		max_features	'sqrt'
		criterion	'friedman_mse'
<i>2</i>	GBDT	loss	'deviance'
		learning_rate	0.92407
		n_estimators	350
		subsample	0.9
		max_features	'sqrt'
		criterion	'friedman_mse'
<i>3</i>	RF	n_estimators	119
		criterion	'entropy'
		bootstrap	True
		max_features	'sqrt'
		class_weight	'balanced_subsample'

Table A.3: The parameters of the ML models that were deployed in the case study.

A.4 Final Survey

This section contains the questions and responses of the final survey of the case study, which are not shown in Chapter 7.

At what share of the time do you think the system heated correctly?

3 responses

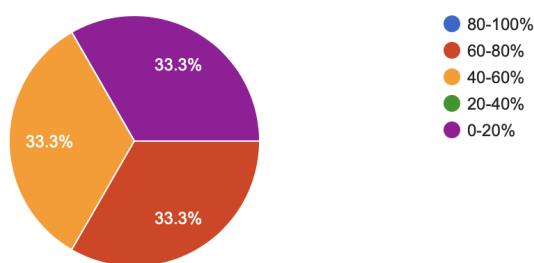


Figure A.2: Multiple-choice question towards the share of time in which the system heated correctly as reflected after 12 weeks of case study. The responses stand in stark contrast to the direct feedback collected while the case study was running, which is shown in Table 7.1 and Table 7.5

In the second part of the study, the heating plate was automatically controlled by LATEST. How did you perceive the accuracy of its heating and absence of heating?

3 responses

There were occasions where I could feel more comfortable if the plate was turned on.

Did not notice much difference

there were a lot of the late afternoon when i felt cold, but it didn't turn on at all

Figure A.3: Freely-formulated question toward the share of time in which the system heated correctly as reflected after 12 weeks of case study. The responses reflect the responses from the question in Figure A.2

APPENDIX A. CASE STUDY

Data Collection: What do you think would need to change in the environment so you would use the system more often?

3 responses

More flexible location of the heating plate

Maybe if the ambient temperature is lower

nothing

Figure A.4: Question towards possible changes of the environment to increase usage and feedback frequency.

Data Collection: What do you think would need to change in the application so you would use the system more often?

3 responses

For me it was fine. I used it a lot with just clicking the send button.

Maybe if there is a more clear indicator of whether the system is on, or in high power, I could feel more comfortable using it.

make the heater stay on for longer

Figure A.5: Question towards possible changes of the system to increase usage frequency.

A.4. FINAL SURVEY

"I used the system to control the infrared heating plate.."

3 responses

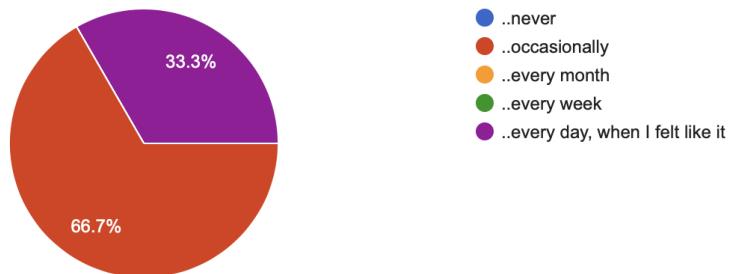


Figure A.6: Question towards the self-reflected usage frequency. The responses reflect the observed usage frequencies in Chapter 7.

Do you have any general remarks?

3 responses

It was interesting being one of the subjects for the experiment.

Thanks for having me in the experiment.

there are so many confounding variables during the experiments. Maybe a memo note can be added for each data entry, such as holding a cup of coffee, wearing a sweater today. I don't know if it's aware of the outdoor climate or just the indoor, knowing the outdoor climate maybe helpful for the application to learn better.

Figure A.7: Question towards general remarks by the test subjects.

List of Figures

1.1	Several thermal comfort factors (TCF) grouped into environmental factors and human factors. Conventional HVAC systems exclusively rely on air temperature.	4
1.2	Infrared radiant heating panels by infraNOMIC. On the left picture, a panel with a landscape on its surface is mounted at the wall behind a bed. On the right picture, a panel with a mirror surface is mounted at the ceiling of a massage room. In both cases, the thermal energy is directed to the places where the occupants are.	6
2.1	A diagram showing two series of temperature data. The series “OriginalTemperature”, colored in green, stays at a constant level of about 20.8 °F over the full period of 25 minutes apart from a few outliers, up to 24.7 °F. The second series, “SmoothedTemperature” colored in yellow, stays almost constant at the same level, with only two comparably insignificant outliers of less than 0.5 °F.	15
2.2	Two diagrams, each displaying temperature data for the task environments of four occupants over a time interval of about 6 minutes. The diagram on the left shows data points spread over the time interval with no obvious time format. The diagram on the right shows the same data but the data points were aligned to fit intervals of 30 seconds.	16
2.3	A diagram of temperature data of an ambient heater and the corresponding moving averages for window sizes 25, 50, and 100.	17
2.4	The output of applying LIME on an ML model that predicts whether an e-mail’s author is an atheist or Christian.	23
2.5	The output of applying SHAP on a tree-based ML model that predicts the average housing price in the city of Boston.	24

LIST OF FIGURES

3.1	The setup proposed by Gao and Keshav featuring a microcontroller, infrared sensor, laser pointer and two servomechanisms [GK13b]. These sensors' measurements are used to predict occupancy, clothing insulation, and metabolic rate.	26
3.2	The PCS chair developed by the Center for the Built Environment at the University of California, Berkeley [KZS ⁺ 18]. It offers heating and cooling of the occupant's task environment.	29
4.1	UML use case diagram of containing all use cases of LATEST. It depicts the three actors' interactions with the system in the form of use cases. It also illustrates the relationships between these use cases. LATEST works for multiple occupants in parallel, while this use case diagram depicts the use cases for a single occupant only.	42
4.2	The UML activity diagram illustrating the overall functionality of LATEST. It depicts the central transition from the data collection phase to the temperature control phase.	47
4.3	The UML activity diagram illustrating the data collection phase of LATEST. The main loop ends once a data set with high variance is at hand.	48
4.4	The UML activity diagram illustrating the temperature control phase of LATEST. The shown process is repeated periodically after 30 seconds.	49
4.5	The sensor taxonomy implemented in LATEST as a UML class diagram. The sensors are grouped into three general categories, namely occupancy sensors, environmental sensors, and biosignal sensors.	50
4.6	The analysis object model of LATEST as a UML class diagram. It depicts the application domain's objects and their attributes, methods and relationships.	52
5.1	The high-level design of LATEST giving an overview of the envisioned system's architecture in the three environments: <i>Task Environment</i> , <i>Indoor Environment</i> , and <i>Outdoor Environment</i>	54
5.2	The Subsystem Decomposition of LATEST. There are three regions separated by colors: <i>Indoor</i> in blue, <i>Outdoor</i> in green, and <i>Remote</i> in orange.	58
5.3	The UML deployment diagram depicting the hardware software mapping of LATEST. There are three regions separated by colors: <i>Indoor</i> in blue, <i>Outdoor</i> in green, and <i>Remote</i> in orange.	63

LIST OF FIGURES

6.1	A diagram showing data of four unspecified features with the corresponding occupancy and heating status.	72
6.2	A diagram showing the data of Figure 6.1 separated into four <i>intervals of occupancy</i> , which are color-coded in yellow, green, red, and blue. In the top of the diagram, the target variable, heating activity is depicted, switching between “ON” and “OFF”. The respective lengths of consecutive heating states are displayed in-place as integers.	73
6.3	The groups to the interval-stratified 2-fold cross-validation for the data depicted in Figure 6.1.	73
7.1	Two photographs of the setup of the data collection phase of this case study. There are red labels for (1) the Microsoft Bands at the subjects’ wrists, (2) the fixed, white DHT22 air temperature and relative humidity sensors, (3) the infrared heating panel, (4) the iPhone SE running the LATEST mobile application, and 5 the plugwise Circle, controlling and monitoring the current to the radiant heating panel.	82
7.2	The interface of the LATEST mobile application for data collection. The display can be separated into three parts: the question for the subject’s thermal sensation with a corresponding 5-point scale slider, the question for the subject’s thermal comfort with a corresponding 3-point scale slider, and the green “Send Feedback” button.	84
7.3	The processes executed in the transition to the temperature control phase, as described in Chapter 6.	85
7.4	The average f1-score of the initial models. The depicted variance shows the variance in f1-score in the interval-stratified 5-fold cross-validation. The black horizontal line represents the f1-score of a naive model that always outputs the target value that is represented most often. The methods are ordered from left to right in descending average f1-score.	88
7.5	The results of a randomized grid search expressed as average f1-scores of interval-stratified 5-fold cross-validations. For each cross-validation there were 5 models trained, which was performed 200 times per ML method for every test subject. The black horizontal line represents the f1-score of a naive model that always outputs the target value that is represented most often. The methods are ordered from left to right in descending maximum average f1-score.	89

LIST OF FIGURES

7.6	The feature importances of the deployed models aggregated by certain feature types for every subject. The value of these feature importances weight the influence a feature type had while a model was trained.	91
7.7	The interface of the temperature control version of the LATEST mobile application.	92
7.8	The final survey's question about how the reasonings were perceived and the three answer texts.	95
A.1	A reminder that was printed in DIN A5 format and handed out to every test subject for the case study.	106
A.2	Multiple-choice question towards the share of time in which the system heated correctly as reflected after 12 weeks of case study. The responses stand in stark contrast to the direct feedback collected while the case study was running, which is shown in Table 7.1 and Table 7.5	109
A.3	Freely-formulated question toward the share of time in which the system heated correctly as reflected after 12 weeks of case study. The responses reflect the responses from the question in Figure A.2	109
A.4	Question towards possible changes of the environment to increase usage and feedback frequency.	110
A.5	Question towards possible changes of the system to increase usage frequency.	110
A.6	Question towards the self-reflected usage frequency. The responses reflect the observed usage frequencies in Chapter 7.	111
A.7	Question towards general remarks by the test subjects.	111

List of Tables

2.1	The 7-point thermal sensation scale of the PMV, which is adopted by the ASHRAE Standard-55 [ASH13].	10
3.1	The concept matrix comparing related work with this research. All abbreviations used are given in Table 3.2.	32
3.2	The abbreviations used for thermal comfort factors and target variables.	32
5.1	The 17 measurement ids covered by LATEST. Every id consists of a <i>Tag</i> and a <i>Type</i> . The measurement types are grouped by the second column for a more granular overview.	64
5.2	The access matrix of the influxDB instance of LATEST.	66
6.1	The training and validation split of an interval-stratified 2-fold cross-validation on the data shown in Figure 6.1 for the groups displayed in Figure 6.3.	74
6.2	The parameters and their distributions used for training the prototypes and optimizing the parameters. The table shows the parameters for all the considered ML methods' implementations.	77
7.1	The results of the data collection phase expressed in the total time of occupancy, time of heating, and times of being in each of the three possible thermal comfort categories for every test subject. For the key indicators the corresponding shares of the subject's total time of occupancy is shown as well. The time between two successive thermal comfort votes was assumed to belong to the former vote.	86
7.2	The subjects' commands to the system and the feedback gathered in the data collection phase.	86

LIST OF TABLES

7.3	The average runtimes of generating one prediction with a corresponding SHAP reasoning in an experiment of 20 generations. Tree-based models were faster by a significant factor due to their implementation.	90
7.4	The mean and standard deviation of the measured outside temperature and outside relative humidity for both phases of this case study.	93
7.5	The results of the temperature control phase expressed in the total time of occupancy, heating, being too warm, being thermally comfortable, and being too cold for every test subject. For the key indicators the corresponding shares of the subject's total time of occupancy are shown as well. The time between two successive thermal comfort votes was assumed to belong to the former vote. The displayed values are rounded.	94
7.6	The subjects' commands to the system and the feedback gathered in the temperature control phase.	94
7.7	Two multiple-choice questions from the final survey and the participants' responses. The first question evaluates how the visual reactions of the mobile application were perceived by the test subjects, which was answered unanimously with "instantaneous". The second question regarded to the simplicity of the app. All three participants stated that they "understood how the app works instantly".	95
A.1	The key indicators of the data collected for each test subject in both phases. The used abbreviations are defined in Table 3.2. The values for air temperature and the temperature of the ambient HVAC system are in degree Fahrenheit.	107
A.2	The maximum f1-scores of the randomized grid search for each ML method and test subject. For <i>Subject 3</i> , GNB was not optimized because it did not meet the performance requirement of pre-selection.	108
A.3	The parameters of the ML models that were deployed in the case study.	108

Bibliography

- [AOS⁺16] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016.
- [ASH13] Standard-55 ASHRAE. *Thermal Environmental Conditions for Human Occupancy*. American Society of Heating, Refrigerating and Air-Conditioning Engineers, 2013.
- [BB12] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(1):281–305, February 2012.
- [BCR97] Jose Benitez, Juan Castro, and Ignacio Requena. Are artificial neural networks black boxes? *IEEE Transactions on Neural Networks*, September 1997.
- [BD09] Bernd Brügge and Allen Dutoit. Object-oriented software engineering using uml, patterns, and java. *Learning*, 5(6):7, 2009.
- [Bis06] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer New York, 2006.
- [BLB⁺11] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [Car95] John Carroll, editor. *Scenario-Based Design: Envisioning Work and Technology in System Development*. John Wiley & Sons, Inc., USA, 1995.

BIBLIOGRAPHY

- [CGGS12] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in Neural Information Processing Systems 25*, pages 2843–2851. Curran Associates, Inc., 2012.
- [Chi92] Nancy Chinchor. Muc-4 evaluation metrics. In *In Proceedings of the Fourth Message Understanding Conference*, pages 22–29, 1992.
- [Cho10] Joon-Ho Choi. Cobi: Bio-sensing building mechanical system controls for sustainably enhancing individual thermal comfort. *Ph.D. Dissertation*, 2010.
- [CK19] Tom Chang and Agne Kajackaite. Battle for the thermostat: Gender and the effect of temperature on cognitive performance. *PLOS ONE*, 14, May 2019.
- [CL12] Joon-Ho Choi and Vivian Loftness. Investigation of human body skin temperatures as a bio-signal to indicate overall thermal sensations. *Building and Environment*, 58, September 2012.
- [CLL12] Joon-Ho Choi, Vivian Loftness, and Dong-Won Lee. Investigation of the possibility of the use of heart rate as a human factor for thermal sensation models. *Building and Environment*, 2012.
- [Com16] European Commission. Recital 71. *General Data Protection Regulation*, 2016.
- [CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.
- [CWB⁺11] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537, November 2011.
- [dDBD97] Richard de Dear, Gail Brager, and Cooper Donna. Developing an adaptive model of thermal comfort and preference - final report on rp-884. *ASHRAE Transactions*, 1997.
- [DK07] David DeGroot and Larry Kenney. Impaired defense of core temperature in aged humans during mild cold stress. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology*, 292(1), 2007.

BIBLIOGRAPHY

- [dLSB17] Carolina de Lima Salge and Nicholas Berente. Is that social bot behaving unethically? *Communications of the ACM*, 60(9):29–31, August 2017.
- [DTN10] Noël Djongyang, René Tchinda, and Donatien Njomo. Thermal comfort: A review paper. *Renewable and Sustainable Energy Reviews*, 14(9):2626–2640, 2010.
- [DZAL17] Changzhi Dai, Hui Zhang, Edward Arens, and Zhiwei Lian. Machine learning approaches to predict thermal demands using skin temperatures: Steady-state conditions. *Building and Environment*, 114:1–10, 2017.
- [EBCV09] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *Technical Report, Université de Montréal*, January 2009.
- [EC12] Varick Erickson and Alberto Cerpa. Thermovote: Participatory sensing for efficient building hvac conditioning. In *Proceedings of the Fourth ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, BuildSys’12, New York, NY, USA, 2012.
- [Fan70] Povl Fanger. *Thermal comfort: Analysis and applications in environmental engineering*. Danish Technical Press, 1970.
- [Fan73] Povl Fanger. Assessment of man’s thermal comfort in practice. *Occupational and Environmental Medicine*, 30(4):313–324, 1973.
- [Fis02] William Fisk. How ieq affects health, productivity. *Ashrae Journal*, 44, May 2002.
- [FPWL15] Asma Farhan, Krishna Pattipati, Bing Wang, and Peter Luh. Predicting individual thermal comfort using machine learning algorithms. pages 708–713, August 2015.
- [FQvF⁺19] Jonathan Francis, Matias Quintana, Nadine von Frankenberg, Sirajum Munir, and Mario Bergés. Occutherm: Occupant thermal comfort inference using body shape information. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys’19, New York, NY, USA, 2019.

BIBLIOGRAPHY

- [FR14] Keith Frankish and William Ramsey. *The Cambridge handbook of artificial intelligence*. Cambridge University Press, 2014.
- [Fro12] Monika Frontczak. Human comfort and self-estimated performance in relation to indoor environmental parameters and building features. *Ph.D. Dissertation*, 2012.
- [FS10] George Forman and Martin Scholz. Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. *SIGKDD Explorations*, 12:49–57, January 2010.
- [FW11] Monika Frontczak and Paweł Wargoński. Literature survey on how different factors influence human comfort in indoor environments. *Building and Environment*, 46(4):922–937, 2011.
- [GK13a] Peter Gao and Srinivasan Keshav. Optimal personal comfort management using spot+. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys’13, New York, NY, USA, 2013.
- [GK13b] Peter Gao and Srinivasan Keshav. Spot: A smart personalized office thermal control system. pages 237–246, January 2013.
- [Gra92] Robert Grady. *Practical Software Metrics for Project Management and Process Improvement*. Prentice-Hall Inc., Upper Saddle River, NJ, USA, 1992.
- [GSH67] Adolf Gagge, Jan Stolwijk, and James Hardy. Comfort and thermal sensations and associated physiological responses at various ambient temperatures. *Environmental Research*, 1(1):1–20, 1967.
- [GTBG15] Ali Ghahramani, Chao Tang, and Burcin Becerik-Gerber. An online learning approach for quantifying personalized thermal comfort via adaptive stochastic modeling. *Building and Environment*, 2015.
- [HAZA06] Charlie Huizenga, Shiva Abbaszadeh, Leah Zagreus, and Edward Arens. Air quality and thermal comfort in office buildings: Results of a large indoor environmental quality survey. *HB 2006 - Healthy Buildings: Creating a Healthy Indoor Environment for People, Proceedings*, 3, January 2006.
- [HC93] Michael Hammer and James Champy. *Reengineering the Corporation: A Manifesto for Business Revolution*. Harper Business, 1993.

BIBLIOGRAPHY

- [HLM⁺97] Volker Hartkopf, Vivian Loftness, Ardeshir Mahdavi, Stephen Lee, and Jayakrishna Shankavaram. An integrated approach to design and engineering of intelligent buildings—the intelligent workplace at carnegie mellon university. *Automation in Construction*, 6(5):401–415, 1997.
- [HLZ⁺09] Tyler Hoyt, Kwang Lee, Hui Zhang, Edward Arens, and Tom Webster. Energy savings from extended air temperature set-points and reductions in room air mixing. *International Conference on Environmental Ergonomics*, pages 608–611, 2009.
- [JY16] Lai Jiang and Runming Yao. Modelling personal thermal sensations using c-support vector classification (c-svc) algorithm. *Building and Environment*, 99, January 2016.
- [KHP⁺18] Yasser Khan, Donggeon Han, Adrien Pierre, Jonathan Ting, Xingchun Wang, Claire Lochner, Gianluca Bovo, Nir Yaacob-Gross, Chris Newsome, Richard Wilson, and Ana Arias. A flexible organic reflectance oximeter array. *Proceedings of the National Academy of Sciences*, 115(47), 2018.
- [KNOR01] Neil Klepeis, William Nelson, Wayne Ott, and John Robinson. The national human activity pattern survey (nhaps): A resource for assessing exposure to environmental pollutants. January 2001.
- [Kol19] Oliver Kolar. An incremental learning approach for personalized thermal comfort in smart offices. *Master’s Thesis*, 2019.
- [Kos09] Peter Kosack. Beispielhafte vergleichsmessung zwischen infrarotstrahlungsheizung und gasheizung im altbaubereich. Technical report, University of Kaiserslautern, 2009.
- [KZS⁺18] Joyce Kim, Yuxun Zhou, Stefano Schiavon, Paul Raftery, and Gail Brager. Personal comfort models: Predicting individuals’ thermal preference using occupant heating and cooling behavior and machine learning. *Building and Environment*, 129:96–106, 2018.
- [LAC⁺09] Vivian Loftness, Azizan Aziz, Joon-Ho Choi, Kevin Kamp-schroer, Kevin Powell, Mike Atkinson, and Judith Heerwagen. The value of post-occupancy evaluation for building occupants and facility managers. *Intelligent Buildings International*, 1:249–268, January 2009.

BIBLIOGRAPHY

- [LEC⁺20] Scott Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2, January 2020.
- [LHL⁺06] Vivian Loftness, Volker Hartkopf, Khee Lam, M Snyder, Ying Hua, Y Gu, Joon-Ho Choi, and X Yang. Sustainability and health are integral goals for the built environment. *HB 2006 - Healthy Buildings: Creating a Healthy Indoor Environment for People, Proceedings*, 1:1–17, January 2006.
- [LL17] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc., 2017.
- [LM04] Grzegorz Lewicki and Giuseppe Marino. Approximation by superpositions of a sigmoidal function. *Applied Mathematics Letters*, 17:1147–1152, December 2004.
- [MBBF99] Llew Mason, Jonathan Baxter, Peter Bartlett, and Marcus Frean. Boosting algorithms as gradient descent. *NIPS*, pages 512–518, 1999.
- [Mil68] Robert Miller. Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I*, New York, NY, USA, 1968. Association for Computing Machinery.
- [Mil19] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.
- [MLK11] Boran Morvaj, Luka Lugaric, and Slavko Krajcar. Demonstrating smart buildings and smart grid features in a smart energy city. In *Proceedings of the 2011 3rd International Youth Conference on Energetics (IYCE)*, July 2011.
- [MV14] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. *Computing Research Repository*, 2014.
- [NYC16] Anh Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features

BIBLIOGRAPHY

- learned by each neuron in deep neural networks. *Computing Research Repository*, 2016.
- [Par14] Ken Parsons. *Human Thermal Environments: The Effects of Hot, Moderate, and Cold Environments on Human Health, Comfort, and Performance*. CRC Press, Inc., USA, 3 edition, 2014.
 - [PPP17] Kedar Potdar, Taher Pardawala, and Chinmay Pai. A comparative study of categorical variable encoding techniques for neural network classifiers. *International Journal of Computer Applications*, 175:7–9, October 2017.
 - [PSTK15] Jigar Patel, Sahil Shah, Priyank Thakkar, and Ketan Kotecha. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications*, 42:259–268, January 2015.
 - [RPL10] Juan Rodriguez, Aritz Perez, and Jose Lozano. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):569–575, March 2010.
 - [RSG16] Marco Ribeiro, Sameer Singh, and Carlos Guestrin. “why should i trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2016.
 - [SBBR16] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, 2016.
 - [Seb19] Liedl Sebastian. Fetch: A feedback approach for determining thermal comfort in smart buildings. *Bachelor’s Thesis*, 2019.
 - [SFL05] Olli Seppanen, William Fisk, and Q Lei. Effect of temperature on task performance in office environment. January 2005.
 - [Sha53] Lloyd Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

BIBLIOGRAPHY

- [SKJ⁺16] Soo Sim, Myung Koh, Kwang Joo, Seungwoo Noh, Sangyun Park, Youn Kim, and Kwang Park. Estimation of thermal sensation based on wrist skin temperatures. *Sensors*, 16:420, March 2016.
- [SS03] Jonathan Samet and John Spengler. Indoor environments and health: Moving into the 21st century. *American Journal of Public Health*, 93(9):1489–93, 2003.
- [SSS⁺17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–359, October 2017.
- [Sto96] Thomas Stoker. Smoothing bias in the measurement of marginal effects. *Journal of Econometrics*, 72(1):49–84, 1996.
- [Tas00] Leonard Tashman. Out-of-sample tests of forecasting accuracy: an analysis and review. *International Journal of Forecasting*, 16(4):437–450, 2000. The M3- Competition.
- [VA16] Kush Varshney and Homa Alemzadeh. On the safety of machine learning: Cyber-physical systems, decision sciences, and data products. *Computing Research Repository*, 2016.
- [VEB⁺17] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander S Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. Starcraft ii: A new challenge for reinforcement learning, 2017.
- [vF17] Nadine von Frankenberg. E-textiles in the context of smart buildings. *Master’s Thesis*, 2017.
- [WBE⁺16] Daniel Winkler, Alex Beltran, Niloufar Esfahani, Paul Maglio, and Alberto Cerpa. Forces: Feedback and control for occupants to refine comfort and energy savings. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, New York, NY, USA, September 2016.

BIBLIOGRAPHY

- [YCN⁺15] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *Computing Research Repository*, 2015.
- [YN13] Rayoung Yang and Mark Newman. Learning from a learning thermostat: Lessons for intelligent systems for the home. pages 93–102, September 2013.
- [ZDW19] Zimu Zheng, Yimin Dai, and Dan Wang. Duet: Towards a portable thermal comfort model. In *BuildSys’19: Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 51–60, 2019.
- [Zha19] Chenlu Zhang. A bio-sensing and reinforcement learning control system for personalized thermal comfort and energy efficiency. *Ph.D. Dissertation*, 2019.
- [ZLJ⁺12] Song Zhen, Vivian Loftness, Kun Ji, Sam Zheng, Bertrand Lasternas, Flore Marion, and Yuebin Yu. Advanced, integrated control for building operations to achieve 40% energy saving. U.S. Department of Energy, September 2012.