A proposed method for unsupervised anomaly detection for a multivariate building dataset

MASTER THESIS

Author:
Roberto G. SÁNCHEZ A.[1]

Supervisor:
Dr. Julien NEMBRINI [2]

Professor:
Dr. Denis LALANNE [3]

UNIVERSITY OF FRIBOURG

Human-IST Group
Department of Informatics

July 21, 2017

[1] robertogonzalo.sanchezalban@unifr.ch, University of Fribourg
[2] julien.nembrini@unifr.ch, University of Fribourg
[3] denis.lalanne@unifr.ch, University of Fribourg

Declaration of Authorship

I, Roberto G. SÁNCHEZ A., declare that this thesis titled, "A proposed method for unsupervised anomaly detection for a multivariate building dataset" and the work presented in it, is my own original work. I declare that I have acknowledged the work of others by providing detailed references of said work. Furthermore, I declare that no part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution. I am aware of the University's regulations concerning plagiarism, including those regulations concerning disciplinary actions that may result from plagiarism.

Signed: _____

Date: _____

"Practice makes the Master, the more you practice the better you get. Do not forget that God is your best coach."

Roberto Sánchez A.

University of Fribourg

Abstract

Faculty of Science
Department of Informatics

Master of Science in Computer Science

A proposed method for unsupervised anomaly detection for a multivariate building dataset

by Roberto G. SÁNCHEZ A.

Ubiquitous devices employed in building facilities are allowing us to acquire a diverse amount of data relative to the internal systems of buildings. This is contributing to the growing awareness of the gap that exists between the desired performance of a building and its actual performance. Automated fault detection and diagnostic (AFDD) systems have been showed to be effective at detecting the root cause of performance problems. This master thesis is interested in finding motif cluster (typical patterns) and discord clusters (atypical/abnormal patterns), two types of patterns used by some AFDD approaches. Our approach attains to discover daily patterns in a multivariate fashion for a studied building dataset by using the Gaussian Hidden Markov models. The discovered motif cluster profiles define the typical performance of the building, while the discovered discord cluster profiles spot potential performance problems of the building. Three proposed models create a label data frame that summarize all the daily patterns in a table allowing the researcher to do further aggregation about the motif/discord cluster profiles. The proposed models where tested in a case study where the North-East and South-West ventilation systems of the studied building were compared. The results provide information about the pattern evolution across different seasons and years, as well as the dynamics between various variables. In addition, anomalous daily profiles were spotted as a multivariate pattern in the North-East ventilation system, that demonstrate how powerful this approach is. Finally, this approach had good feedback from the building experts and the potential of our approach motivates further research.

Acknowledgements

I would like to thank all the persons who helped me throughout this master thesis and made the project possible. In particular,

Dr. Denis Lalanne for creating a space of research, and the opportunity to exploit my talents in his group of research.

Dr. Julien Nembrini, my supervisor, who inspired me through several ideas and his critical thinking during the project. Thanks for each suggestions that was an important contribution to this work. I appreciate his time and patience.

Esther, Sven and Theresa for their help with the proofreading and grammar corrections.

Colleagues and friends of the faculty for their encouragement (José, Marin, Minh).

My family, for their support during my whole residence in Fribourg, even if they were not there... their prayers could touch me. Hilda, Gonzalo, Jessy and Pablo.

Renate, my love, who was supporting me all the time during this master thesis, you put a smile in my heart.

Abraham, Issac and Jacob's God, for being my guide in everything I do.

Contents

List of Figures

List of Tables

List of Abbreviations

AFDD	**A**utomated **F**ault **D**etection and **D**iagnostic
pdf	**P**robability **D**ensity **F**unction
OcP	**Oc**cupant **P**resence
HVAC	**H**eating **V**entilation and **A**ir **C**onditioning
TABS	**T**hermally **A**ctive **B**uilding **S**tructure
IAQ	**I**ndoor **A**ir **Q**uality
ASHRAE	**A**merican **S**ociety of **H**eating, **R**efrigeration and **A**ir **C**onditioning **E**ngineers
MVPA	**M**ulti**V**ariate **A**nalysis
HMM	**H**idden **M**arkov **M**odel
GaHMM	**Ga**ussian **H**idden **M**arkov **M**odel
IoT	**I**nternet **o**f **T**hings
SAX	**S**ymbolic **A**ggregate appro**X**imation
PCA	**P**rincipal **C**omponent **A**nalysis

1 Introduction

1.1 Motivation

The expansion of the use of ubiquitous devices (sensors and actuators) in buildings have allowed to improve building performance and improve the occupant experience during these last two decades **de2017occupancy**, **abdallah2015developing**, **dong2009sensor** However, it is estimated that as much as 30% of energy consumed by commercial buildings is due to an unappropriated use of internal systems like HVAC, TABS, lighthing and others in existing building automation systems (BASs) [1]. BASs provide an automatic control of the internal systems in buildings and have the capacity to collect underlying data for postoperative analysis. Some of the objectives of postoperative analysis are the improvement of the overall performance of the building, comfort-enhancement for occupants and improving energy efficiency **miller2015automated**, **capozzoli2015fault**

Nowadays, thanks to the availability of data coming from the internal systems of buildings (e.g. HVAC, TABS, lightning, etc.), there is a growing awareness of the gap that exists between the original building design and the actual performance of the building **miller2015automated** Nevertheless, there is still a need for tools to assist the improvement of the building performance. Automated fault detection and diagnostic (AFDD) systems have been shown to be effective at detecting the root cause of performance problems **kim2017review** By reviewing the literature, one can see that the use of artificial intelligence and data mining techniques for AFDD **capozzoli2015fault** is still minimal in this domain. We believe therefore, that the potential of artificial intelligence and data mining techniques can be exploited in the goal of providing more intuitive and powerful tools to detect performance problems in buildings.

1.2 Objectives

There is not yet an uniform consensus for building performance assessment. Some approaches such as building benchmarking and the use of performance metrics have served to evaluated buildings in specific areas like energy consumption. In other cases, it is still unclear what indoor variables need to be measured to carry out a building performance assessment **owens2012measuring**, **web_NIOSH** Normally the behavior(performance) of a building might be ruled by recommendations and accepted industry practices (e.g. ASHRAE's standards, GBPN's policies, etc). For example, one can find standards related to the indoor environmental quality (IEQ) where a range of building variables are monitored (e.g. CO_2, noise, temperature, etc.) and defined according to the level of acceptability judged by its occupants, or by the technical parameters of underlying systems **owens2012measuring** However, one cannot find so much information about the internal dynamics of the building. We notice that there is a lack in building science literature defining the actual behavior and the dynamics between monitored variables of a building. This is because most of the AFDD studies cannot make generalized affirmations about the performance of buildings since each building is unique. Therefore, each AFDD is based on diagnostic methods such as: Historic-Based, qualitative and quantitative-based approaches are attuned to the particular building in question.**katipamula2005methods**

[1] 2012 Commercial Building Energy Consumption Survey (CBECS)

One of the objectives of this thesis is to propose a methodology that assists the stakeholders (designers, architects and occupants) in defining the typical "behavior" [2] of the building, so that in this way, experts can benefit from this gain of information and, finally improve the buildings performance, occupant comfort and find opportunities to save energy.

This thesis is devoted to proposing a methodology for conducting unsupervised fault detection using machine learning algorithms in a multivariate building dataset. We are interested in defining the "behavior" of the building across months, seasons and years. This, in our opinion, can be done by finding the typical patterns of the variables. Once we have the typical patterns (i.e. daily profile) of the building, we can detect those days where the building's variables fluctuate very different from its typical patterns, and therefore, potential performance problems can be detected. In other words, this work seeks to find the most common patterns that appear in measured variables in a building, and discovering also the common interrelationship that exists between them. One last point of interest of this work is to find a practical application of the proposed method for fault preventive analysis, and predictive analysis using the discovered typical patterns.

1.3 Project Outline

The first chapter of this thesis introduces the motivations and the objectives of performing anomaly detection in buildings. The second chapter reviews the state of the art of Automated Fault Detection and Diagnostics AFDD's and the relating topics that are needed for the our proposition, that is the *GaHMM - profile, seasonal and interactional* models, explained in section **??**. The third chapter describes the studied building and his correspondent multivariate dataset. The fourth chapter explains the modeling process, the implementation of the *GaHMM - profile, seasonal and interactional* models. This chapter includes an evaluation part where the reader can appreciate the clustering quality of our proposed models. The fifth chapter presents the results of each of our proposed model, those models are tested in case of study for finding anomalies in the ventilation systems of the building. Finally, we include expert building feedback at the last chapter. Our proposed methodology could serve for further research according to the expert remarks.

[2] This term expresses the idea that the indoor environment of a building changes depending on external conditions, the interaction with humans, maintenance operations, or any other phenomena that produces this change.

2 State of the Art (Literature Review)

This chapter reviews several topics that are relevant for this work. The first section reviews the presence of occupants in a building since it is an important concept to keep in mind during the analysis of time series of the multivariate building dataset. This concept allows the researcher to be aware of the interaction between human beings and a building. The following sections review all the necessary concepts that are used in our proposed methodology.

2.1 Important definitions

Some important definitions that are used in this master thesis:

- **Behavior of the builiding** this term expresses the idea that the indoor environment of a building changes depending on external conditions, the interaction with human beings, maintenance operations, or any other phenomena that produces this change.
- **Daily profile** is a vector of 24 values that describe the fluctuation of a variable during the day, each value for each hour.
- **Cluster daily profile** is a group of days that have similar daily profile for the studied variable. The cluster profile is defined by a mean vector $P_x = \{\mu_{x_i} \forall i \in [0, 23]\}$ and a standard deviation vector $STD_x = \{\sigma_{x_i} \forall i \in [0, 23]\}$ of length $N = 24$.
- **Motif cluster profile** is a collection of days that have a typical/common daily profile. This cluster profile appears often in the time series data.
- **Discord cluster profile** is a collection of days that have non-typical/ abnormal daily profiles.
- **Label data frame** is a big table that summarizes all the generated labels for each proposed model. It create a mapping between the discovered patterns and a code of identification.

2.2 Occupant presence

Occupant presence (OcP) in buildings has been investigated for decades. In **artf_light_1980** **artf_light_1980** proposed a method for predicting the use of artificial lighting based on the switching behavior of occupants. His approach uses a logistic curve as a probability density function (*pdf*) for deciding the likelihood of whether a switching activity will occur during the course of a day, according to the occupant presence. In **page_2008** **page_2008** et al. [**page_2008**] summarized similar approaches for heating, cooling and ventilation systems, and they proposed a generalized stochastic model for simulation of occupant presence. **page_2008** et al. [**page_2008**] presented a scheme (figure **??**) where it is possible to see the means of interaction between occupants and buildings. They claimed that occupant presence is an input to all other models (i.e. water, electrical appliances, lighting utilities, etc.) and the OcP model will be central to the family of other stochastic models.

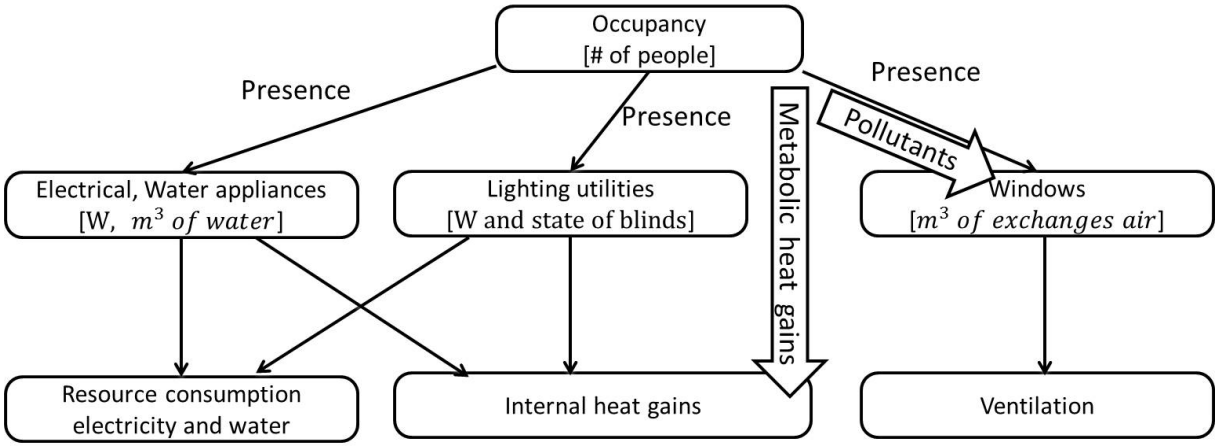


FIGURE 2.1: Outputs of the occupancy model and their later use by stochastic models of occupants' behaviour. **page_2008**

Most of the OcP models that are summarized in **page_2008**'s work fit their parameters according to the related presence data. This process, called "calibration process", was often made by hand or deduced by heuristic techniques. In most cases, the evaluation methods for these models was the measurement of their capacity to reproduce, realistic behavior of the occupants after the calibration process. Having a realistic simulation of occupant presence is one of the most important aspects for the OcP models, this is because, a lack of an accurate estimation of occupant presence will imply a miscalculation of resources such as water, electric energy and others. Thus, the over- or underestimation of presence is undesirable in the majority of the cases **page_2008**, **wang_2005**, **profile_comp_2001**

wang_2005 wang_2005 wang_2005 put into evidence one example of OcP overestimation. In their simulation, they observed peaks of presence in the first hour of the day (which implies overestimation of presence) due to their assumption of the normal distribution of the arrival time in the morning. **wang_2005** were aware of the absent periods of the occupants, and they found that the absence intervals were exponentially distributed and that the coefficient of the exponential distribution for a single office was treated as a constant over the day. To overcome the overestimation problem of presence, the authors mention the use of empirical distributions for daily events like the arrival in the morning.

2.2.1 Occupancy diversity factors

Nowadays, one of most the common ways to estimate occupant presence is by using the diversity profiles approach (also called occupancy schedules or occupancy profiles) **profile_comp_2001**, **davis2010occupancy**, **duarte2013revealing**(2001, 2010, 2013). This approach consists of daily profiles that are composed of 24-hour representative values. These representative values are a more realistic interpretation of occupant presence because a daily profile does not make assumptions about the probability distribution of daily events such as arrival times in the morning (as it was described before in the example of **wang_2005**). However, this approach could miscalculate the presence of occupants throughout the year when there is a repetition of a subset of profiles that do not consider the temporal variations such as seasonal habits, irregular days (late arrivals or early departures), differences in behavior between weekdays and weekends, and atypical behavior like the presence of people in office buildings on a weekend **davis2010occupancy**

To have a more realistic representation of occupant presence, the OcP models would like to be the closest possible to the real behavior of the occupants. However, for simplicity, it is common that the diversity factor for each workday (i.e. usually from Monday to Friday) is treated identically, and weekends treated with a different profile. A typical profile is lower during the absence of the occupants and increases its value during the expected occupancy periods. A maximum factor value is achieved when the maximum expected presence of occupants occurs, or in the case of some types of HVAC equipment, when weather is over extreme design condition. Duarte et al. 2013 **duarte2013revealing**

Davis et al. **davis2010occupancy** show examples of OcP profiles of different kinds of buildings. In their work, we can find remarkable differences between workdays for each building type. One example is the library building, where we can see a highest occupancy level on Mondays and a lowest occupancy level on Fridays. For this building, the authors suggested an individual OcP modeling for the following three groups: Monday through Thursday, Friday, and weekend schedules. These groups have what will be called in this work as behavioral coincidence. [1]

2.2.2 Deterministic and stochastic occupancy factors

duarte2013revealing duarte2013revealing mention the idea of deterministic and stochastic occupancy factors. According to their criteria, while a deterministic OcP model identifies or creates a standard workday profile which is the same for the whole workweek, a different weekend profile is created for Saturday and Sunday. The deterministic model assumes no change in occupancy schedules throughout the year. On the other hand, a stochastic model uses various probabilistic methods to capture the random nature of individuals' behavior. Both OcP models are valid for estimating occupancy presence, the former is simple but overestimates occupancy presence when it does not consider variations over all the entire year. The latter might include an OcP miscalculation when there is no consideration of long absence periods such as business trips, vacations and other such things.

2.2.3 Recent studies

Today, more buildings have ways in which to measure different kinds of variables of interest at their facilities. The evolution of electronic devices allows ubiquitous sensing by using different technologies, as for example Wireless Sensor Network (WSN). This electronic evolution is part of the Internet of Things (IoT) paradigm (2013, 2014 [**gubbi2013internet**, **zanella2014internet**]) and offers the ability to measure, infer and understand environmental indicators from different settings. This generation of enormous amounts of data however, must also be analyzed to give a good grasp of the process of interest **gubbi2013internet** The IoT concept aims at making the Internet even more immersive and pervasive, so that little by little, the access is easier to devices such as microntrolers, home appliances, surveillance cameras, monitoring sensors actuators, and others. This implies new opportunities for new application in different domains **zanella2014internet** One application is **Smart Cities** where there are important issues such as the optimization of the use of resources in the urban context, structural health of buildings, waste management **zanella2014internet** For example, there is a tendency to believe that in most urban centers around the world, through processing, visualizing, and uploading sensor data from large architecture, when measurements and models are shared between buildings with control systems, will allow one building to shade another or mitigate

[1] Note that this term will be used in this work. This term indicates a similarity between two or more OcP profiles

the so-called urban canyon effect [2] **cuff2008urban** The canyon effect could affect various local conditions (e.g. temperature, air quality, wind and others) of the closest neighborhood, such that in some cases it implies high temperature for the buildings that are inside of the canyon, or bad air quality among others [**cuff2008urban**, **andreou2013thermal**]. Sharing the common patterns that describe the behavioral environment of the buildings in the canyon-like environment could mitigate adverse changes on the local conditions in urban canyons **cuff2008urban**

In the context of Smart Buildings, new studies benefit from the data acquisition facilities that IoT paradigm provides. In a more general vision, the estimation of OcP for estimating resources is part of Smart cities as well. In fact, the consumption of resources, public services, the use of public spaces and the interaction of several systems like traffic and others are linked with presence of people (i.e. urban sensing) in the smart cities context (2007, 2006, 2006)**abdelzaher2007mobiscopes**, **burke2006participatory**, **campbell2006people**). Several studies using different sensors have been carried the last decade. **benezeth2011towards**, **huang2017occupancy** 2011, 2017 **benezeth2011towards**, **huang2017occupancy** explain some of these approaches, the challenges and problem for detecting people in indoors environments. Some of the problems are related to the lack of data analysis in primary sensors. For example, PIR (Passive Infrared Sensor) (2013,2013) **duarte2013revealing**, **nguyen2013energy** cannot differentiate the number of users or know whether if the user is a human or other entity such as a pet or any other animal. Other example is the inconveniences when noise sensors cannot detect low levels of noise coming from the occupants (i.e. when people are quiet) (2013, 2014)**uziel2013networked**, **kelly2014application benezeth2011towards**, **huang2017occupancy** summarize two proposals for overcoming the current limitations, the first one recommends the use of multiple low-cost, non-intrusive, environmental occupancy sensors, privileging the use of an independent distributed detectors network combined with a probabilistic data analysis **benezeth2011towards**, **huang2017occupancy** The second one recommends the use of more advanced devices such as video cameras which implies a large data storage and privacy concerns **benezeth2011towards** Furthermore, sophisticated vision algorithms are needed and they deal with multiple issues like background substraction, tracking and recognition 2013, **sid2013detection** **huang2017occupancy** 2017, **huang2017occupancy** propose an approach based on a hybrid sensor (i.e. CO_2 sensor and light sensor) to detect the OcP, where this combination of two sensors creates a more robust sensor for detecting OcP. The results indicate that this hybrid combination leads to more accurate occupancy detection than only using a CO_2 sensor. In literature, one observe that there is a tendency to use the second proposal of **benezeth2011towards**'s work to detect presence, because is considered more robust. We apply this proposal by using the available data of the CO_2, *exhausted air temperature, intake air temperature, status of the blind system* of the studied building. We use these variables in order to see the interaction of variables when there is OcP. Since several studies uses CO_2 levels as an input to estimate OcP **huang2017occupancy**, **labeodan2015occupancy**, **nassif2012** we use the mentioned variables to create multivariate samples [3] in our proposed *GaHMM interactional model* explained in section **??**.

2.3 Automated Fault Detection and Diagnostic

kim2017review 2017 **kim2017review** provided a summary of AFDD studies published since 2004. They pointed out that 118 new studies in the past decade were identified and reviewed in their work. The latter work proposes a classification of AFDD methods based on the compilation of previous articles **katipamula2005methods1**, **katipamula2005methods2** Basically, there

[2] Terminology for places where the street is rounded by buildings on both sides creating a canyon-like environment.

[3] Multivariate samples are explained in section **??**.

are three big groups of AFDD approaches: *quantitative methods, qualitative methods, and process history-based*. The process history-based AFDD methods is the most popular approach because they rely on historical data to train models and because of their reduced modeling complexity **kim2017review**, **katipamula2005methods1** Our interest is focused on process history-based approaches, since a subcategory of this group (i.e. Black box methods) applies pattern recognition techniques to explain a relationship between inputs and outputs of a process or a system **kim2017review** The final idea of these approaches is to compare the performances of the building over a period of time to what is expected, in this way incorrect operation or unsatisfactory performances can be detected **capozzoli2015fault**, **katipamula2005methods1** In this domain of detecting unsatisfactory performances, data mining techniques can be used for this purposes. However, there are few papers that use artificial intelligence and data mining techniques (mostly used in building energy consumption fault detection **capozzoli2015fault**, **miller2015forensically**). Therefore, we observed that there still room for applying machine learning, and data mining in this domain. This master thesis attends to contribute to the AFDD literature, with time-dependent cluster task and multivariate pattern recognition (i.e. the three proposed models in section **??**).

2.4 Data mining process

Here we include the data mining techniques to use in our approach.

2.4.1 Multivariate pattern analysis

Univariate analysis considers one single dependent variable (DV) being measured, and analyses whether the variation of DV is associated with different conditions of interest (i.e. independent variables IV). Each measure is considered as a sample, and the measures of this variable can be represented as a vector. In contrast, multivariate analysis (MVPA) considers multiple dependent variable (DVs) (depending on the nomenclature, it can be called as features or voxels) that are measured and analyzes the relationship with the independent variables (IVs). A sample in MVPA, is therefore, a vector of N values where N is the number of features. In the end, the measurements can be presented by a matrix, that is a two-dimensional, where there are M samples and N features, and the matrix is sized M x N (**baur2007multivariate** 2007 **baur2007multivariate**). This concept is largely used in our proposition, this help us to defined the observed samples that are explained in section **??**.

Depending on the terminology, a pattern can be described as a vector containing the observations of features for a single sample. In a simplistic and generic sense, MVPA includes any analysis where the outcome is dependent on the variability and/or consistency of measurements across samples by features matrix **baur2007multivariate baur2007multivariate** proposed the following typical steps for doing multivariate data analysis:

1. Framing the research question in such a way that it can be modeled mathematically.
2. Selecting the right statistical model. Every multivariate model searches for certain patterns in data. It might miss other patterns. Using different multivariate methods therefore may lead to different results. Among the theoretical questions multivariate analysis can address: **a)** identifying latent classes; **b)** causal analysis; **c)** identifying patterns in time; **d)** network analysis; and **e)** multilevel analysis. [4]
3. Verifying that assumptions and prerequisites for the chosen statistical procedure are met.

[4] Most multivariate procedures can be viewed as a special case of general linear models (GLM).

4. Preparing data for the specific analysis.
5. Computing the model using statistical algorithms and methods.
6. Analysis of the results.

Some of the previous steps are applied in the following sections. In this study, we applied MVPA because we believe that we can explain in a more meaningful way the building performance/behavior by considering all the variables together (e.g. CO_2 levels, blind height, cooling energy, etc.) as a single sample. This is observed in models *GaHMM - seasonal* and *GaHMM - interactional*, section **??**.

2.4.2 Hidden Markov Model

This section discusses the perils and advantages of using Hidden Markov Model **(HMM)** and in particular, describe one extension of the HMM called the Guassian Hidden Markov Model **(GaHMM)**. [5] The mathematical notations and details that this work adopts for HMM are in **pfundstein2011hidden**'s work , 2011 **pfundstein2011hidden** However, some definitions for later purposes are listed here:

- $\mathbb{O} = (O_1, O_2, ..., O_T)$ is the observed sequence, each observation (i.e. $O_{n\in[1,T]}$) is a sample that can be a number or a vector. T is the total number of the samples.
- $\mathbf{K}$ is the number of hidden states for an HMM model. $\mathbf{K} \in \mathbb{N}^+$.
- $\mathbb{S} = (s_1, s_2, ..., s_T)$ is the hidden state sequence given an observed sequence. Depending on the problem, we can be interested in finding the most likely sequence of hidden states $\mathbb{S}$ that generates/emits the observed sequence $\mathbb{O}$. The hidden sequence $\mathbb{S}$ can be determined by Viterbi's algorithm. [6]
- $\pi = (\pi_1, ..., \pi_{\mathbf{K}})$ is the initial state probabilities where $\pi_i = P(s_{i=1})$ and $\sum_{i=1}^{\mathbf{K}} \pi_i = 1$.
- $a_{i,j}$ is the transition probability for going from the hidden state s_i to the hidden state s_j. It can be denoted as $a_{i,j} = P(s_{t+1=j}|s_{t=i})$.
- $\mathbb{A} = \{a_{i,j} \mid i \in [1, \mathbf{K}]; j \in [1, \mathbf{K}]\}$ is the transition probability matrix of the hidden states where $\sum_{j=1}^{\mathbf{K}} a_{i,j} = 1$.
- $b_{k,t}|t \in [1, T]$ is the probability that a sample O_t is emitted in state s_k. Or in other words that, given a hidden state s_k, the observed sample O_t was emitted in time t for this hidden state.
- $\mathbb{B} = \{b_{k,t}|k \in [1, K]; t \in [1, T]\}$ is the observation/emission probability matrix where $b_{k,t} = P(O_t|s_k)$. Typically, a multivariate Gaussian distribution is assumed, but other distributions can be used as well.
- $\lambda = \{\pi, \mathbb{A}, \mathbb{B}\}$ is the parameter vector that specify an HMM model.
- The **limited horizon assumption** claims that the probability of being in a state at time t (s_t) depends only on the state at time $t-1$ (s_{t-1}). The reasoning underlying this assumption is that the state s_t represents enough summary of the past to reasonably predict the future. Formally:

$$P(s_t|s_{t-1}, s_{t-2}, ..., s_1) = P(s_t|s_{t-1}) \tag{2.1}$$

[5] Some authors use GHMM for Generalized Hidden Markov Model. To disambiguate the term, we use GaHMM for Gaussian Hidden Markov Model.

[6] The words 'emits' and 'generates' is used indiscriminately in HMM literature.

- The **stationary process assumption** claims that the conditional distribution over a next state given a current state does not change over time. In other words, it is assumed that state transition probabilities are independent of the actual time at which the transitions takes place. Formally:

$$P(s_{t_1+1=j}|s_{t_1=i}) = P(s_{t_2+1=j}|s_{t_2=i}) \quad t_1, t_2 \in [2, T] \quad \wedge \quad t_1 \neq t_2 \tag{2.2}$$

- The **output independence assumption** is that the current output (observation O_i) is statistically independent of the previous outputs (observations $O_{i-1}, O_{i-2}, ..., O_1$). Formaly:

$$P(\mathbb{O}|\mathbb{S}, \lambda) = \prod_{t=1}^{T} P(O_t|s_t, \lambda) \tag{2.3}$$

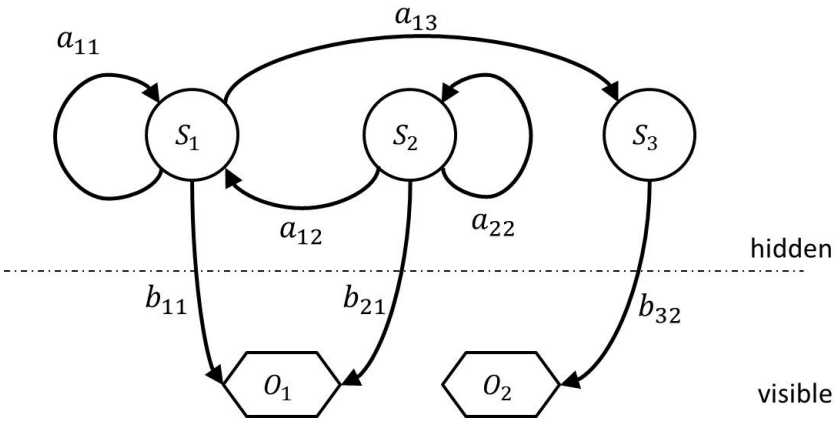


FIGURE 2.2: **HMM Visual representation:** It shows the transition probabilities between three hidden states $\{S_1, S_2, S_3\}$ and the emission probability transitions $\{b_{11}, b_{21}, b_{32}\}$. Note that the transition probabilities equal to zero are not included, for example: a_{23}. Only the observed sequence $O_{t\in[1,T]}$ is observable, the states $\{S_1, S_2, S_3\}$ that produce the samples are not observable. The emission probability transition $b_{k,t}$ can be any probability function, for example a Gaussian distribution.

Figure **??** shows the visual representation of the HMM. It says there is an internal Markov process, that itself cannot be "observed" directly which is called as the "hidden Markov process" (i.e. three hidden states $\{S_1, S_2, S_3\}$ with their correspondent probability transitions a_{ij}). This Markov chain is the responsible for generating observed samples O_1, O_2 according to the probabilities specified by the matrices $\mathbb{A}$, $\mathbb{B}$ and π. Once one hidden state is reached, it is said that the state S_k emits an observed sample O_t under the probability $b_{k,t}$. The sequence of hidden states $\mathbb{S}$ that generated the observed sequence $\mathbb{O}$ can be revealed by solving the decoding problem that is explained in section **??**.

2.4.3 Gaussian Hidden Markov Model

To overcome certain predefined weaknesses of the standard HMM, new ways to combine HMM with other approaches or new models have been proposed (**ghahramani2001introduction** 2001 **ghahramani2001introduction**). One extension of the HMM model for continuous observation values is the Gaussian Hidden Markov Model (GaHMM). Here we borrow some of its concepts for a better description of this extension **bilmes1998gentle**, **murphy2002dynamic**

A Hidden Markov Model is a probabilistic model of the joint probability of a collection of random variables $\{O_1, O_2, ..., O_T\}$ and $\{S_1, S_2, ..., S_{\mathbf{K}}\}$. Where the O_t variables are either continuous or discrete observations and the S_k variables are hidden and discrete. The term discrete for the hidden states implies that we only have $\mathbf{K}$ possible categorical states. Two assumptions make this model tractable, which are defined in equations **??** and **??**. Furthermore, it is assumed that the underlying 'hidden' Markov chain defined by $P(S_t|S_{t-1})$ is time-homogeneous (i.e. it respects the stationary process assumption, equation **??**). The hidden Markov chain is represented by the time-independent stochastic matrix $\mathbb{A}$ and the special case when $t = 1$ is described by the initial state distribution π. Now, if the observations O_i are discrete symbols, we can represent the observation model as the matrix $\mathbb{B}$ where $b_{k,t} = P(O_t|S_k)$, but if the observations O_i are a vector in $\mathbb{R}^{\mathrm{L}}$, it is common practice to use a continuous probability density function, instead of a set of discrete probabilities. GaHMM represents $b_{k,t}$ using a Gaussian distribution:

$$P(O_t = y|S_t = x) = \Gamma(y; \mu_x, E_x) \tag{2.4}$$

Where $\Gamma(y; \mu, E)$ is the Gaussian density function with a mean vector μ_x and the variance covariance matrix E evaluated at y:

$$\Gamma(y; \mu, E) = \frac{1}{(2\pi)^{L/2}\sqrt{\| E \|}} exp\Big[-\frac{1}{2}(y-\mu)^{\intercal}E^{-1}(y-\mu)\Big] \quad [7] \tag{2.5}$$

A more flexible representation of the GaHMM model is a mixture of $\mathbf{K}$ Gaussians:

$$P(O_t = y|S_t = x) = \sum_{m=1}^{\mathbf{K}} P(M_t = m|S_t = x)\Gamma(y; \mu_{(m,x)}, E_{(m,x)}) \tag{2.6}$$

where M_t is a hidden variable that specifies which mixture component to use, and $P(M_t = m|S_t = x) = C(x, m)$ is the conditional prior weight of each mixture component **murphy2002dynamic** In other words, this is the sum of all mixture components with their correspondent distribution, so that all together it represents the probability of an observed sample O_t occurring at time t when the hidden state is equal to x. At the end, each hidden state S_i is defined by a mean vector μ_x with his distribution represented by variance-covariance matrix E_x.

The three problems to solve using HMM

HMM is used to solve three kind of problems that are: the learning, evaluation and decoding problem. The solution of these problems is the core of our proposition. In the modeling process, explained in section **??**, one finds the libraries that solve each of them. Basically, our proposition find the best HMM models to discover all the possible patterns into the observed sample $\mathbb{O}$. The theoretical details are explained briefly in this section, and the whole implemented process in section **??**. The following explains the general idea of the process:

- Learning/ training process: The HMM model is fitted with the observed samples $\mathbb{O} = (O_1, O_T)$. The definition of the observed samples is very important since it defines the kind of HMM to use.
- Evaluation process: Here one uses the log probability of $P(\mathbb{O}|\mathbb{S}, \lambda)$ (i.e. equation **??**), to select the best trained model. The model that fits the best the parameters of λ is the one who has the greatest probability.

[7] Note that $\intercal$ is the symbol for the vector/matrix transpose and L is the cardinality of μ_x.

- Decoding process: When one obtain the best model, there is a perfect matching between the observed samples $\mathbb{O}$ and the sequences of states $\mathbb{S}$. One says that each hidden state S_k emits one sample O_i. This is graphically explained in section **??**. We use this mechanism to cluster similar observed samples.

Here the theoretical details. Using the Markov assumptions (i.e. **??**, **??**, **??**), one can answer questions about the observed sequence, the hidden sequence, or the model parameters λ. These related questions are known as the three basic problems of HMM. This work does not go into detail about these three questions, but the reader is invited to refer to an excellent tutorial in **bilmes1998gentle** and more related literature of HMM in **haussler1996generalized**, **ghahramani2001introduction**, **stamp2004revealing**, **ramage2007hidden**, **pfundstein2011hidden**

The Learning Problem Given an observed sequence $\mathbb{O}$ how can we find the HMM that best fits? HMM has different ways to tune the parameters $\lambda = \{\pi, \mathbb{A}, \mathbb{B}\}$. There are two approaches for the training process, the generative training algorithms and discriminative training algorithms **dymarski2011hidden** Usually, the solution space of HMMs is coded as a function of λ and one can consider two main optimization criteria as being: Maximum Likelihood (ML) and Maximum Mutual Information (MMI). This optimization of parameters λ is usually done by gradient algorithms in order to find the maximum likelihood. The Maximum Likelihood Estimation (MLE) can be found by using the Expectation-Maximization algorithm (EM). This algorithm is able to deal with derivatives of the likelihood function with respect to all the unknown values of λ by picking arbitrary values for one set of unknown parameters, and then using the previous set, to estimate a second set of parameters, and then apply this procedure recursively until the convergence of parameters. In general, this process generates multiple solutions, so there is no guarantee that the global maximum will be found **dymarski2011hidden** Our proposed method uses EM algorithm to tune the λ parameters, and the best trained model (i.e. the one that get the maximum likelihood) is elected by doing a k-fold cross validation process. This process is explained in **??**.

The Evaluation Problem In general, one of the benefits of HMM is its evaluation property. The question that involves the observed sequence and the parameters of the model is known as the evaluation problem. The question is formulated as follows: What is the probability that the given observations $\mathbb{O} = (O_1, O_2, ..., O_T)$ can be generated by a Hidden Markov Model with parameter $\lambda = \{\pi, \mathbb{A}, \mathbb{B}\}$. In other words: $p(\mathbb{O}|\lambda) = ?$. The solution to this problem is the use of the forward or backward algorithm that finds $p(\mathbb{O}|\lambda)$ in about K^2T multiplications **stamp2004revealing** The solution to this question can be used to evaluate a trained HMM, or it can also suffice just to know whether or not an observed sequence can be generated/emitted by a given HMM, the latter being used in classification problems. In this study, since we are conducting unsupervised fault discovery and are interested in evaluating our trained models and using the best ones, we can perform fault detection.

The Decoding Problem In some cases, one is interested in finding the "most likely" state sequence of the Markov process, given an observed sequence $\mathbb{O} = (O_1, O_2, ..., O_T)$. Literature defines "most likely" in at least in two ways: 1. "most likely" is defined as the state sequence with the highest probability from among all possible state sequence of length T. 2. "most likely" is defined as the state sequence that maximizes the expected number of correct states **stamp2004revealing** This study is interested in finding the whole state sequence with maximum likelihood.

Therefore, the problem is defined as: To find an optimal sequence for the underlying hidden Markov model given a HMM with parameter $\lambda = \{\pi, \mathbb{A}, \mathbb{B}\}$ and an observed sequence $\mathbb{O}$.

Stated differently, we want to know the "most likely" hidden state sequence that emitted the observed sequence $\mathbb{O}$. To solve this problem the Viterbi algorithm is used. By using Viterbi algorithm, one can find the sub-sequences of an observation sequence O that best matches to a given hidden Markov model. For the present study, this is the way in which the typical daily patterns are discovered and clustered. Finding the most likely hidden state sequence is the way in which we tag days where there is similar pattern across the entire time series.

Advantages and perils of using Hidden Markov Model

An HMM is a generative, probabilistic model. This model generates distributions by using the available information from the observed sequence. Because of its capacity for detecting sequences, this model is often used for recognition problems that involve sequence recognition such as speech recognition, gesture recognition, information extraction, recognition of Human Genes in DNA and others **ramage2007hidden**, **seymore1999learning**, **haussler1996generalized** **seymore1999learning** 1999 **seymore1999learning** stated the advantages and perils of HMMs as follow:

> "HMM offers the advantages of having strong statistical foundations that are well-suited to natural language domains, handling new data robustly, and being computationally efficient to develop and evaluate due to the existence of established training algorithms. The disadvantages of using HMMs are the need for an a priori notion of the model topology and, as with anystatistical technique, large amounts of training data".

Later, **ghahramani2001introduction** 2001 **ghahramani2001introduction** showed that HMMs are a kind of Bayesian Network because it is possible to derive the HMM algorithms from more general algorithms for Bayesian networks. He also explained how to overcome some weakness of HMM models (due to the unconstrained transition matrix $\mathbb{A}$ and the exponential number of states in the model) by creating more general models for HMM such as factorial HMMs and tree-structured HMMs.

2.4.4 Hierarchical clustering

Hierarchical clustering seeks to build a hierarchical structure of observed objects in a recursive fashion. Two methods are identified: **1. Agglomerative.** This is a "bottom up" approach, each object starts in its own cluster. Then clusters are successively merged until the desired cluster structure is obtained. **2. Divisive** This is a "top down" approach, all objects belong to one cluster. Then the cluster is divided into sub-clusters, which are successively divided into their own sub-clusters. This process continues until to reach the desired structure **maimon2007soft** Since the *GaHMM profile* model described in section **??** discovers the existing daily patterns into the time series of the variables, we need an strategy to group cluster profiles in a "bottom up" fashion, creating in this way groups of cluster profiles. There are abundant literature about similarity distances and linkage methods [8] for hierarchical clustering, the reader is invited to see more details about the set of metrics and linkage methods in **saraccli2013comparison**, **mullner2011modern**, **maimon2007soft** We describe in section **??**, the use of the hierarchical agglomerative clustering for grouping the discovered cluster profiles.

[8] Rules that serve as criteria for joining similar objects

2.4.5 Symbolic Aggregate Approximation (SAX) transformation

SAX allows the representation of time-series data in words of a finite alphabet A. This approach was developed by **keogh2005hot** 2007 **lin2007experiencing**, **butler2015sax**, **keogh2005hot** The SAX transformation follows this process: The normalized timeseries, Z(t) [9], is first broken down into N individual non-overlapping subsequences. This step is known as chunking, and the period length N is based on a context logical specific period [10] **lin2007experiencing** In the next step, each sunsequence of the time series is divided into W equal sized segments. The mean of the points in each small segments is calculated and an alphabetic character from A is assigned according to the table in **?? lin2007experiencing** To do this asignation, each mean values that falls in zones between the vertical breakpoints, $B = \beta_1, ..., \beta_a - 1$ is substituted by a symbol. Figure **?? lin2007experiencing** exemplify the concept, where the time series (blue line) is transformed into the correspondent word "baabccbc". This approach is used in DayFilter approach **miller2015automated** which is considered as one of the way to detect daily profiles in the state of the art **kim2017review** The SAX method can convert time series data with an equivalent symbolic representation for identifying relevant patterns by comparing strings. **miller2015automated** 2015 **miller2015automated** presented how the SAX method can be implemented to detect the schedules using total building power measurement. **miller2015automated** performed pattern discovery over two different power measurement time series data of the energy consumption of two buildings. His analysis imply among other issues, the identification of discord profiles that implied high corruption of energy **kim2017review**, **miller2015automated** The SAX approach is compared against our proposed *GaHMM profile* model in section **??**.

β_i \ a	3	4	5	6	7	8	9	10
β_1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	-1.28
β_2	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	-0.84
β_3		0.67	0.25	0	-0.18	-0.32	-0.43	-0.52
β_4			0.84	0.43	0.18	0	-0.14	-0.25
β_5				0.97	0.57	0.32	0.14	0
β_6					1.07	0.67	0.43	0.25
β_7						1.15	0.76	0.52
β_8							1.22	0.84
β_9								1.28

FIGURE 2.3: A lookup table that contains the breakpoints that divide a Gaussian distribution in an arbitrary number (from 3 to 10) of equiprobable regions

2.5 Data Visualization

This master thesis does not develop a new data visualization artifact for presenting the discovered information within the multivariate dataset. Developing a new visualization goes beyond our objectives. However, our work uses visualization principles and tools in order to communicate to the reader complex structures of patterns, the interaction between variables, the evaluation of models and the respective results of each section. This task is not a easy task to do since one has to look for visualization that fits the best with the data. In fact, the use of

[9] z-scored normalization is applied

[10] Since the interest of this work is to find daily pattern, therefore $N = 24$

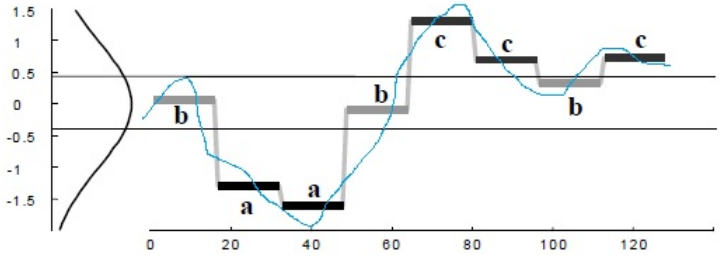


FIGURE 2.4: A time series is discretized by first obtaining a PAA approximation and then using predetermined breakpoints to map the PAA coefficients into SAX symbols. In the example above, with n = 128, w = 8 and $A = \{a, b, c\}$, the time series is mapped to the word baabccbc

principles, concepts, techniques and theories for data visualization come from multiple backgrounds: programming, web design, semiotic or psychology (**aparicio2014** 2014 **aparicio2014**). Therefore, choosing one visualization could become a complex situation because it could imply several criteria from different disciplines. In this work, the combination of data mining, and data visualization can be considered as art and science **aparicio2014**, **kohavi2001data** because there is not strict rules to define which visualization is the most appropriate, and the trial-and-error process is one of the common approaches to use in this domain **kohavi2001data** Propositions such as Exploratory data analysis (EDA) aims to looking at data for finding descriptive patterns, trends or any hint that help to generate hypothesis of interest of the researcher. However, if this process is made by hand in a high dimensional dataset, it becomes impractical in some cases because of the filtering process **witten2016data** We believe that the use of data mining techniques in combination with the appropriated visualizations are a powerful tool for knowledge discovery. The reader will see the use of existing visualization artifacts for different purposes. For example, box plots **williamson1989box** are used for a visual evaluation of the cluster quality of cluster profiles in section **??**. Other concept such as the calendar visualizations **van1999cluster** are needed in order to see the distribution of clusters in section **??**, and others uses. We expect by the use of data visualizations be able to discover the behavior of the building and represent it in the best possible way.

2.5.1 Hierarchical Edge Bundles

Hierarchical Edge Bundles visualization was proposed by **holten2006hierarchical**'s work (2006, **holten2006hierarchical**). This visualization is a compound graph that is based on visually bundling the adjacency edges, i.e., non-hierarchical edges, together. In this way, two or more nodes are joined by using polylines that are bended using a B-spline curve for more readability. This tree visualization based technique can be used in conjunction with other visualizations to express different concepts. Figure **??** shows an example where this visualization displays adjacency relations between nodes. Colors in the linkage line provide more information about the connection between nodes.

In our case, we use this visualization to represent the connection between the variables (time series of the measurements) in the building, this can be appreciated as an information visualization who explains the existing linear correlation between variables (i.e. nodes). This visualization was used for the interactional model where each node is a variable of the building dataset and each linkage line represent the linear correlation between them, refer section **??** for more details. Other variations and more layouts of this visualization are explained in

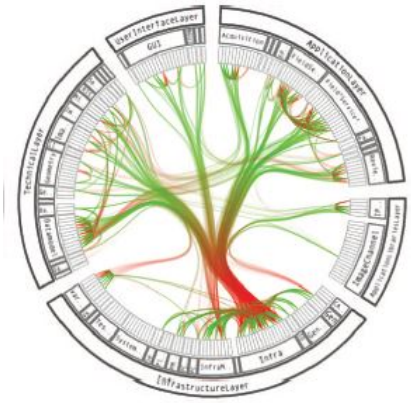


FIGURE 2.5: Radial layout construction of hierarchical edge bundles

holten2006hierarchical [11]

[11] Library available on https://bl.ocks.org/mbostock/7607999

3 Monitored Building Dataset

This chapter is devoted to the description of the multivariate building dataset and the way in which we manage the time series data. The first section describes the provided raw data, the next section describes the general information of the office building, and finally, we explain how we dealt with the provided time series using a Big-Data database.

3.1 Building Dataset - Case Study

The studied dataset was provided by Synergy BTC AG [1] which is a consulting agency with focus on Software-as-a-Service for buildings, located in Bern, Switzerland. A visual interaction tool using this dataset was proposed by **roman2015 roman2015 roman2015** Relevant information relative to this building was taken from that study, and is presented in the following sections.

3.2 Office Building Information

The monitored building is located in eastern Switzerland. Figure **??** shows a model of the building. Table **??** summaries general information of the building. Additional information referring to the internal systems and zones are described in this section.

Information on building use and system technology	
Gross floor area	9560 m^2
Number of floors:	3,
Location:	Eastern Switzerland, indutrial area with
Use:	Office and administrative building

Space (zoning) per floor
13 office areas (mostly open space)
6 meeting rooms
5 border zones (traffic area, toilet, stairs)

Building Shell	
Outside Walls	U = 0.24 W/m2K, massive
Glazing	Uf = 1.1 W/m2K, Ug = 0.65, g = 0.40
Interior walls:	U = 2.0 W/m2K, lightweight

TABLE 3.1: General information of the building.

Zoning map per floor Figure **??** shows the room layout for a floor of the office building. It is divided into three different zones: meeting rooms (red), office areas (green) and border zones (blue). For every floor, the zoning is the same.

Mechanical Ventilation System The studied building has two mechanical ventilation systems. The yellow area for the North-East zone and the blue area for the South-West zone (see figure **??**).

[1] http://www.synergy.ch/

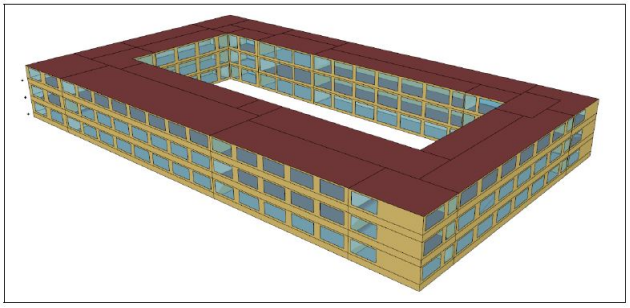


FIGURE 3.1: Model of the building **roman2015**

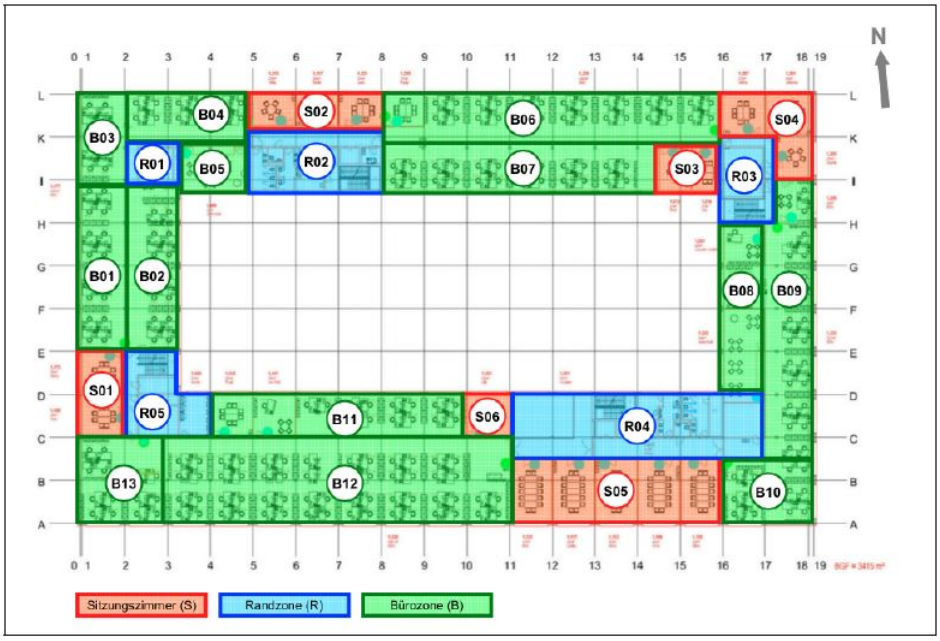


FIGURE 3.2: Zoning for each floor **roman2015**

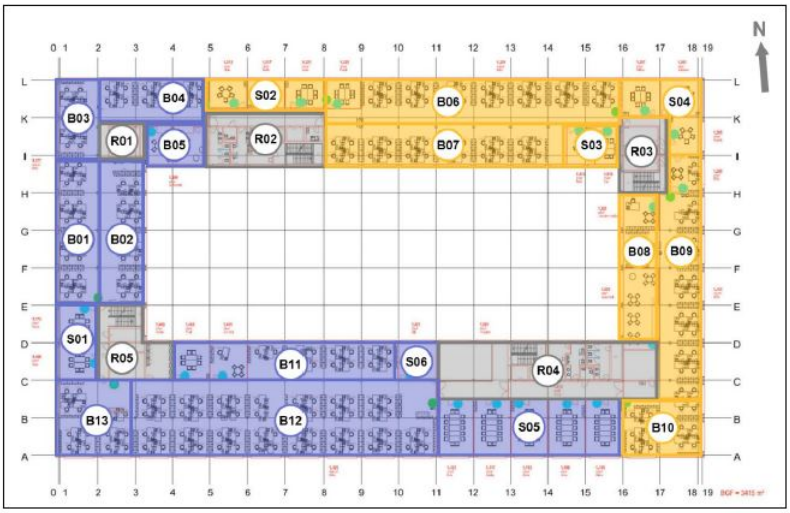


FIGURE 3.3: Mechanical ventilation systems per floor **roman2015**

Heating and Cooling Systems The studied building has two thermally-activated building systems (TABS). The red area for the North-East zone and the green area for the South-West zone (see figure **??**). Each TABS supplies all floors of the corresponding building zone. There are some differences in the zoning of the mechanical ventilation systems and the heating and cooling system. For example, room B10 is in the north/east zone of the ventilation system, but in the south/west zone of the heating and cooling system.

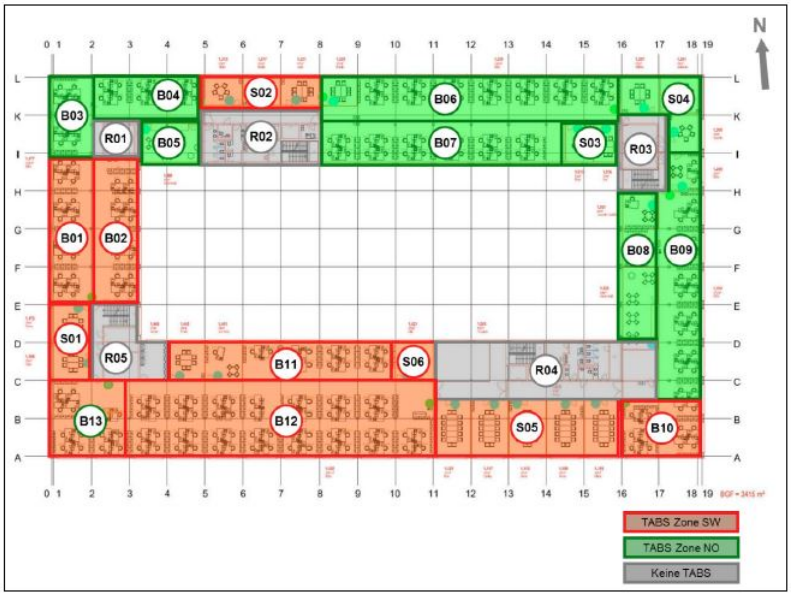


FIGURE 3.4: TAB systems per each floor **roman2015**

Humidity and temperature in rooms Temperature and humidity are measured in 4 rooms of each floor. Therefore 12 variables named with the name of room are available in this dataset according to figure **??**.

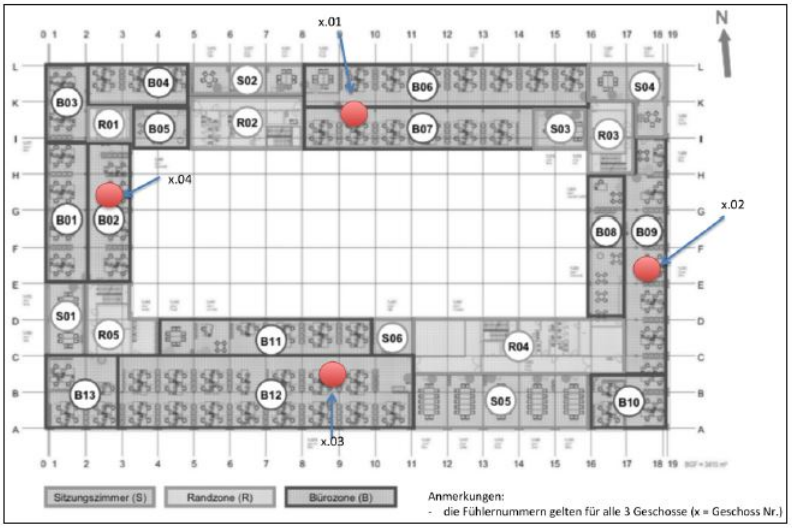


FIGURE 3.5: Mechanical ventilation systems per floor **roman2015**

3.3 Dataset handling

The provided dataset has 64 variables with a time range of 3 years, from mid of 2012 until mid of 2015 with hourly time resolution. This implies 25943 registers per variable giving more than 1.6 million measurements. These 64 variables were associated according the provided information explained in section **??**. Table **??** summarizes the different categories of variables in the building dataset. The complete list of variables is included in the annex **??**. Each category collects variables of the same type, same internal subsystems or that belong to the same spatial information zone according to section **??**. Additionally, the dataset has been enriched with outdoor weather data. The buildings nearest weather station is Zurich-Kloten, for which the data has been acquired from the *Bundesamt fur Meteorologie und Klimatologie Meteo Schweiz*. The variables are outdoor temperature (°C), precipitation (mm) and sunshine hours (min). More information about how the weather temperature changes in Switzerland is in **rebetez2008monthly**

Variable name	Unit
CO2 exhaust air	ppm
Humidity exhaust air	%
Room humidity	%
Temperature exhaust air	°C
Temperature intake air	°C
Room Temperature	°C
Outdoor Temperature	°C
Blinds status	0/1
Cooling TABS	kWh
Heating TABS	kWh
Sunshine presence per hour	minutes (min)
Precipitation level	mm
Blinds height	% (100% closed, 0% open)
Blinds angle	% (100% closed, 0% open)

TABLE 3.2: Variable categories within the Data-set

3.3.1 Building Zones Category

We divided the building dataset into coming from one of two main spatial zones in the building (i.e. North-East and South-West zone), according to the information provided by Synergy BTC AG. These zones are associated to two different mechanical ventilation and TABS systems. We associated each variable (i.e. time series of the measurements) to the according zone and created a metadata table that is included in **??**. Field *breakout_group* uses the codes A_* for the North-East zone and the codes B_* for the South-West zone.

tagname	alias	orientation	category	breakout_group	alias_breakout_group	units
V005_vent01_CO2	CO2 Ventilation NE	NE	CO2	A	CO2 Ventilation NE	ppm
V022_vent02_CO2	CO2 Ventilation SW	SW	CO2	B	CO2 Ventilation SW	ppm
V037_tabs_cold_SW	Cooling tabs SW	SW	Cooling	B_1	Cooling SW	kWh
V075_tabs_cold_NO	Cooling tabs NE	NE	Cooling	A_1	Cooling NE	kWh
V034_tabs_warm_SW	Heating tabs SW	SW	Heating	B_2	Heating SW	kWh

TABLE 3.3: Extract of the proposed metadata table.

3.3.2 Building measures - Time Series

Script **??** shows how the time series are stored in the database. This script represents a register where variable is the key and the numeric value is the measure of the variable at time equal to timestamp. Our approach adopts a Big-Data database that uses JSON-like documents with schemas. Each register is a JSON document with keys like the timestamp, name of the variable, epoch[2], and others. The flexibility and scalability that Big-Data offers, allows us to retrieve high volumes of data quickly and store complex structures like nested JSON objects.

$$
\begin{array}{l}
\{ \\
\quad timestamp \quad : \quad 2012-06-23 \quad 01:00:00 \\
\quad epoch \quad : \quad 1340406000.0 \\
\quad variable1 \quad : \quad 142.25 \\
\quad variable2 \quad : \quad 123.45 \\
\quad ... \\
\quad variablen \quad : \quad 123.45 \\
\}
\end{array}
\tag{3.1}
$$

In our approach, we include the epoch time to the existing dataset to manipulate the time series. This is one simple way to query the time series using ranges of time in a time series dataset. For example, script **??** indicates a NOSQL expression [3] for asking values of a time series within the time range defined by the key *epoch*. *db.timeseries* is the collection of JSON documents that contains the time series of the dataset. The reserved words *$gte, $lte* define the time range (i.e. greater than equal to 1412204098 and lower than equal to 1412204099) of the requested variables *variable_1, ..., variable_n*.

$$
\begin{array}{rl}
db.timeseries.find(\{ & \\
\quad epoch : \{ & \\
\qquad \$gte \quad : \quad 1412204098, & \\
\qquad \$lte \quad : \quad 1412204099 & \\
\qquad \} & \\
\}, \{ & \\
\quad variable\_1 \quad : \quad True, & \\
\quad ..., & \\
\quad variable\_n \quad : \quad True & \\
\}) &
\end{array}
\tag{3.2}
$$

Another proposition from the last script uses a list of timestamps as is showed in script **??**[4]. **An important remark** to do is that the NOSQL queries do not guarantee a chronological order of the requested elements by default. Methods to order the requested data are needed, this is a very important aspect to consider when one uses time series, especially for the training process as it is explained in section **??**.

$$
\begin{array}{rl}
db.timeseries.find(\{ & \\
\quad timestamp : \{ & \\
\qquad \$in \quad : \quad [\text{'2012-12-21', '2012-12-23', ...}], & \\
\qquad \} & \\
\}, \{ & \\
\quad variable\_1 \quad : \quad True, & \\
\quad ..., & \\
\quad variable\_n \quad : \quad True & \\
\}) &
\end{array}
\tag{3.3}
$$

Finally, we show in figure **??** an example of JSON objects stored in the Big-Data database. It shows the power of this approach where key/value pairs are nested. This provides flexibility to our data models, facilitating the storage and the retrieve of complex objects.

[2] Unix epoch (or Unix time or POSIX time or Unix timestamp) is the number of seconds that have elapsed since January 1, 1970 (midnight UTC/GMT), not counting leap seconds (in ISO 8601: 1970-01-01T00:00:00Z).

[3] This is MongoDB syntax, manual available on https://docs.mongodb.com/tutorials/

[4] One common framework that facilitates similar operations for manipulating the time series is included as *rs_common_framework_v4.py*.

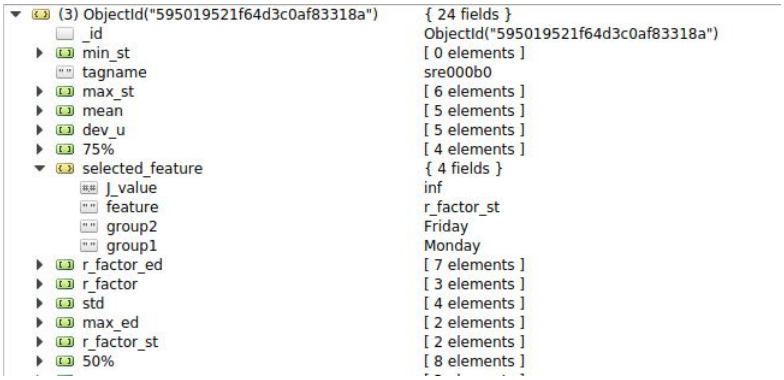


FIGURE 3.6: Example of JSON objects stored in the Big-Data database.

4 Methodology

This chapter is devoted to the description of the software framework developed to solve the problem proposed in **??**. This work proposes a solution based on Big-Data database, python scripts and exploratory visualizations that allows a solution using Gaussian Hidden Markov Model and hierarchical agglomerative clustering. Finally, at the end of this chapter, we evaluate each of the proposed models to observe the different results that we achieve with each one.

4.1 Software Framework

Identifying the problem scenario We consider that the unsupervised fault detection by using machine learning in a multivariate building dataset is a problem that falls in different domains. Regarding the variety of the time series, we consider this problem as a Big-Data problem, since IoT [1] is allowing to collect data from ubiquitous sensors, therefore criteria of volumen, variety and velocity are present **george2014big**, **gubbi2013internet** Regarding the discovery process of daily profiles, each daily profile is a sub-sequence of the whole trend, and finding the common patterns in the whole trend is analogous to the sub-sequence analysis of DNA sequences, for instance **haussler1996generalized**, **ghahramani2001introduction**, **stamp2004revealing**, **ramage2007hidden**, **pfundstein2011hidden** Finally, regarding data mining and visualization mechanisms, we consider that this problem requires effective techniques for knowledge discovery and expressive data visualization artifacts that fits the data **witten2016data**, **aparicio2014**, **kohavi2001data**

Proposed solution We use MongoDB [2] database for having a smart manage of the dataset, the measures and calculations are stored as JSON-like documents allowing flexibility to save/retrieve the data as it was explained in section **??**. The architecture of the proposed solution is a traditional three-tier architecture powered by Python's scripts: a big-data database, an application web server and the front-end tier that is the browser's user. Figure **??** shows the flow of information. The data processing step is done by a transversal script [3] that provides the primitive methods for the all the different modules that are connected. The proposed solution allows the following data mining actions:

a Raw Data Screening process: Filter values that do not belong to measuring process of the variable.

b Correlation analysis of variables: Construct a matrix of linear correlation of the variables of the multivariate dataset. [4]

c Feature selection process: Calculate the gain of information of an arbitrary feature.

[1] IoT is including building and industrial control systems. There is still a question: 'Will we have a smart BAS in the future or is it just part of the IOT?'

[2] For more information: https://www.mongodb.com/what-is-mongodb

[3] script in: /Thesis_project/lib/rs_common_framework_v4.py

[4] This process is explained inside of the interactional model, section **??**

d Modeling process: The training process of the models, the best models are stored for latter uses. [5]

e Visualization process: Two visualization mechanism are proposed. **1.** *Jupyter notebook* [6] is an open-source web application that allows to create and share documents that contain live code, equations, visualizations and explanatory text. **2** Flask web server [7] is an open-source web framework that is based on Werkzeug, Jinja 2. This micro-framework is compatible with other libraries like *queue.js*, *jQuery.js*, *d3.js*, *bootstrap.js* that are used to generated personalized information visualizations.

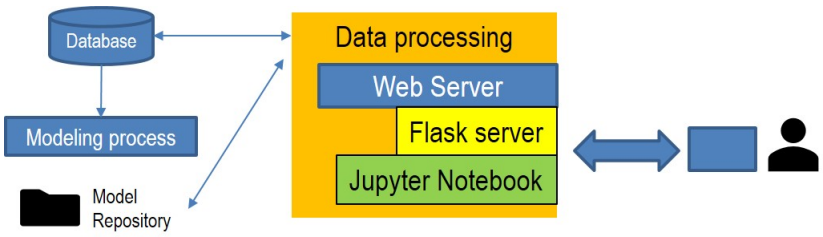


FIGURE 4.1: Software framework schema

Figure **??** shows the implemented collection of JSON documents that were created for each proposed module according to the last list of data mining actions, each of them are explained in details in the next sections.

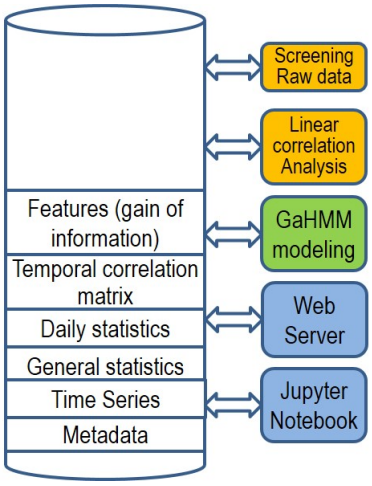


FIGURE 4.2: Implemented collections for saving time series and results

4.2 Raw Data Screening process

The raw data screening process is an important step to do before performing any action over the time series dataset. Aspects like the definition of limits and the quality of the data are important issues to know before starting the knowledge extraction process **miller2015forensically** The raw data screening process aims to clean the dataset in order to offer a good quality

[5] GaHMM models are saved as *.pkl* files in folder: */Thesis_project/HMM_models/Final_models*. The description of each model is in *description_model.txt*.

[6] We use the python version. More available information in http://jupyter.org/.

[7] Web server is implemented using Flask for Python. http://flask.pocoo.org/

for the next phases of the knowledge extraction process. Filter out outliers before applying data mining techniques is commonly applied **miller2015forensically**, **capozzoli2015fault**, **lin2003symbolic**, **kohavi2001data**, **lin2007experiencing** One typical approach is by using the 3σ's (also called six sigma) rule **wiborg2014applied**, **miller2015automated** or the use of Interquartile range analysis **bickel2015mathematical** In our approach we use the 3σ's and propose a new way to spot outliers by using the concept of quartiles. These two approaches are explained after reviewing literature about unsupervised outlier detection.

4.2.1 Unsupervised Outlier detection

zimek2014ensembles 2014 **zimek2014ensembles** explain the challenges and some popular approaches used for unsupervised outlier detection. The fact that, there is no consensus on the definition of an outlier makes this task hard to do and evaluate. Clearly, the definition of an outlier is subjected to nature of the data. For example, for a time series an outlier may be related to the frequency, amplitude of the variable, or any other criteria such as the number of peaks. Therefore, there is no general outlier detection algorithm, each algorithm is able to detect outliers according to the particular criteria that the researcher is interested in. Nevertheless, **zimek2014ensembles** 2014 **[zimek2014ensembles]**, explain the idea of integrating various different outlier detection results, and in this way, the collection of approaches will detect the all most likely outliers.

We apply these two approaches to automatically set the limits of the variables without having any previous knowledge of each variable. These two methods determine the upper and the lower limit (**UpL**, **LoL**) of a variable, such that we can spot outlier values that are outside of the range $[LoL, UpL]$ [8]. An easy way to spot unwanted values is by plotting the variable of interest as a trend line, and apply a visual filter. This approach becomes impractical when we deal with large time series and a big number of variables. Therefore, we need more practical ways to find the variable limits and spot unwanted values. We apply the proposed methods over variables that might not have fixed limits, so that a statistical analysis allows the automatic definition of **UpL** and **LoL**, after which we can spot extreme atypical values. In other cases, where the limits are explicit (e.g. relative humidity %) the definition of the limits is not needed, but these approaches are able to spot outliers anyway.

Both approaches are based on: *a)* six sigma and *b)* percentile analysis. The first approach fits very good when the measures of a variable follow a Gaussian distribution and belong to a controlled process. The second approach is more general and can be fitted to measures that do not necessarily follow a Gaussian distribution.

Six sigma approach Details of this approach are shown in figure **??**. Basically, this approach spots values that are significantly different from the suggested/expected trend,[9] that is the detected values z do not belong to the range limited by a Lower Control Limit (LCL) and an Upper Control Limit (UCL), i.e. $z \notin [LCL, UCL]$. This approach is based on the three sigma rule, and expresses a conventional heuristic that nearly all values are taken to lie within three standard deviations of the mean **wiborg2014applied**, **miller2015automated** In other words, almost all the possible values for a variable that follows a normal distribution are in an interval of six sigma: $[\mu - 3\sigma, \mu + 3\sigma]$. Where μ is the mean of the variable, σ is the standard deviation, and therefore $LCL = \mu - 3\sigma$ and $UCL = \mu + 3\sigma$.

[8] These outlier values can be associated with noise, inaccuracies, mistaken measures, unwanted deviation due to instrument decalibration and others.

[9] This can be subjective depending on the field, it can be that having peaks in the trend is a normal part of the process

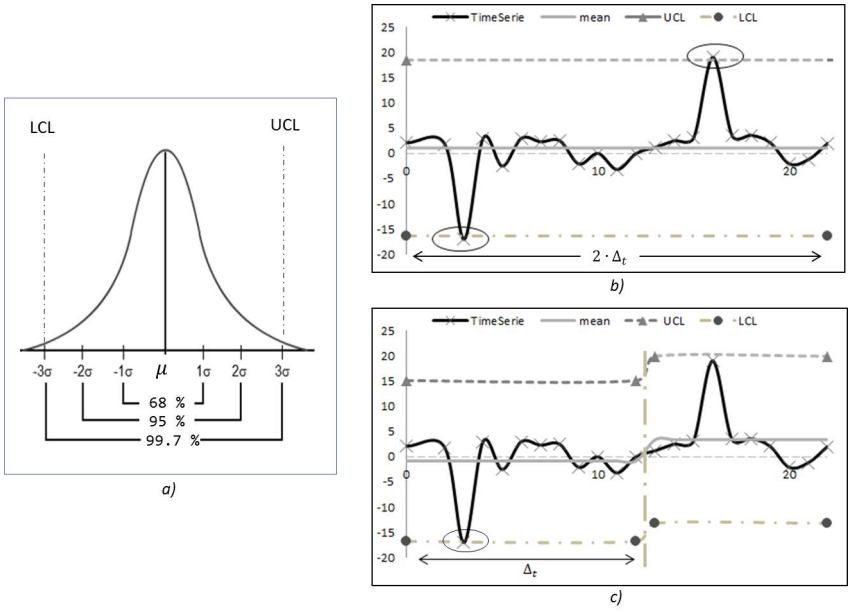


FIGURE 4.3: *a)* **The 3σ's rule**: There is a 99.7% probability that the variable values belong to the interval $[LCL, UCL]$. *b)* Six sigma approach for a time series with fixed LCL and UCL limits. Two spotted values (in the ellipse) are considered as outlier values. *c)* Six Sigma approach with a windowing of Δt size. The LCL and UCL limits are not fixed and change according to the values inside of the window.

In Figure **??**.*b)*, one can observe that the 6σ's approach detects two outlier values for a window of size $2\Delta_t$, while in Figure **??**.*c)*, the same approach for a window of size Δ_t only detects one outlier value in the first window and the other, in the second window, is not detected. We observe in the experiments that the window size affects the detection of outlier values. We notice that the bigger the window is, the more chance there is to detect general outliers, but in contrast, the chance of detecting local outliers is low. We also observe that the smaller the windows is, the less general outliers we can detect. Therefore, having a good definition of the window size is critical for this approach.

Percentile analysis approach shows the use of percentile analysis for spotting outlier values (marked inside the ellipses). This proposed approach determines the LoL and UpL limits by a linear regression over extreme selected (or rather "selected extreme..."?) percentiles. In this example, we select

Figure **??**.*c)* shows the use of percentile analysis for spotting outlier values (marked inside the ellipses). This proposed approach determines the *LoL* and *UpL* limits by a linear regression over selected extreme percentiles. In this example, we select the percentiles $\mathbf{P}_l = [P_5, P_{10}, P_{15}, P_{20}, P_{25}]$ for the LoL limit, and the percentiles $\mathbf{P}_u = [P_{75}, P_{80}, P_{85}, P_{90}, P_{95}]$ for the UpL limit. Two trends are defined using the mentioned percentiles:

- The up_le trend conformed by $up\_le(x) = \mathbf{P}_u$ and $x = [0.75, 0.8, 0.85, 0.9, 0.95]$.
- The lo_le trend conformed by $lo\_le(x) = \mathbf{P}_l$ and $x = [0.05, 0.1, 0.15, 0.2, 0.25]$.

When we apply a linear regression over each trend, we find the predicted values for the points $x = 1 \rightarrow up\_le(x)$ and $x = 0 \rightarrow lo\_le(x)$. These values define the UpL and LoL limits respectively. These can be observed in Figure **??**.*b)* as up_le and lo_le. Finally, we assume that

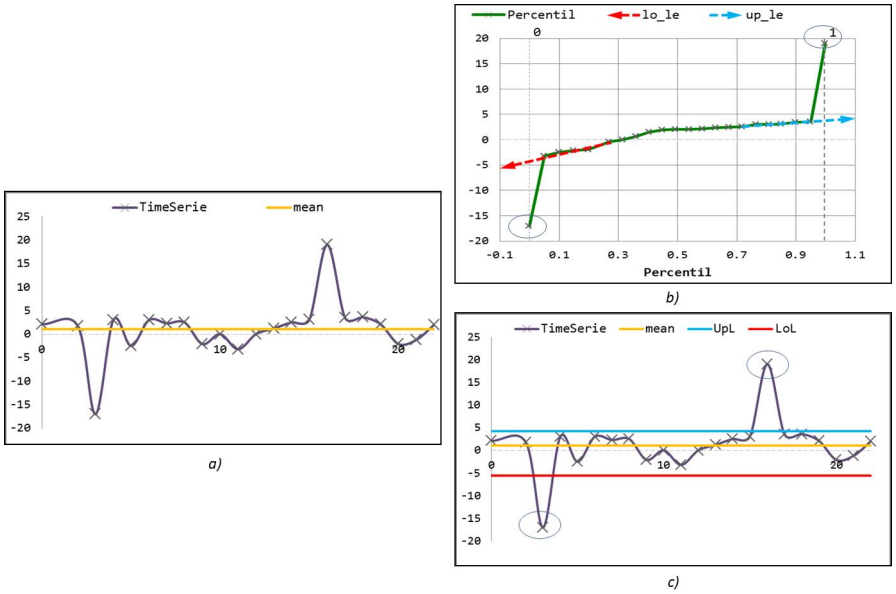


FIGURE 4.4: *a)* Simple time series, similar to Figure **??**. *b)* Percentile curve of the time series. Trends *lo_le* and *up_le* are the linear regression of the extreme percentile values. *c)* The percentile analysis approach detects two outlier values (in the ellipse), these values do not belong to the interval $[LoL, UpL]$.

if there are outlier values (using amplitude as a criteria), they would not follow the linear trend and they would be either in the 5^{th} percentile or the $95 - 100^{th}$ percentile.

In contrast to the six sigma approach, note that percentile analysis does not have symmetric limits to the mean. Observe how the UpL and LoL are not symmetric to the mean μ in figure **??**.*c)*. This property allows the use of this approach for distributions that are not necessarily Gaussian.

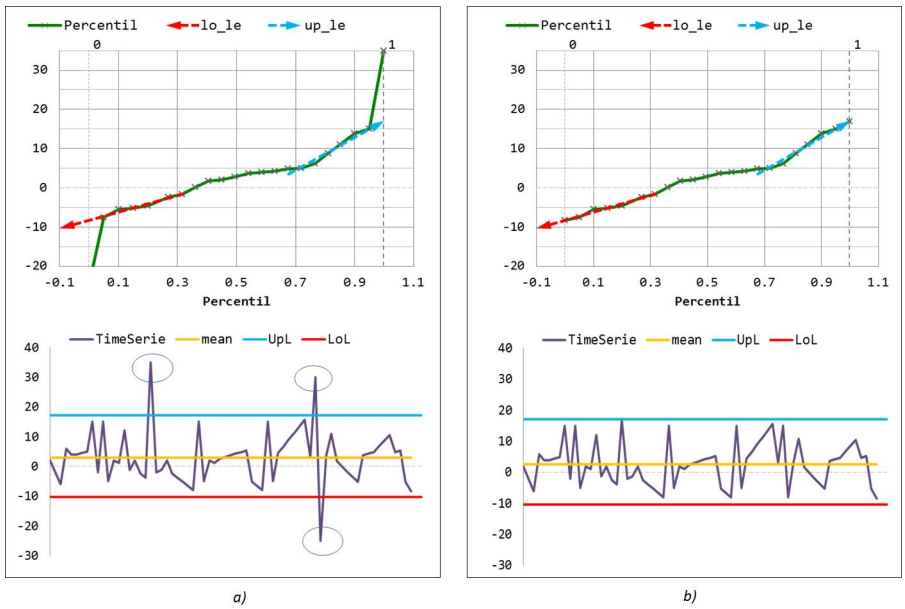


FIGURE 4.5: *a)* Three outliers detected using the percentile analysis approach. *b)* The same time series presented without the three detected outliers.

To exemplify the behavior of the percentile analysis, observe Figure **??** where any value outside the interval $[LoL, UpL]$ is detected as outlier.

Comparison between six sigma and percentile analysis

To evaluate both approaches, we perform several experiments [10] over theoretical signals where we define the limits of the variable. The random variable is defined by $2A \cdot np.random.rand(n, 1) - A$ [11], therefore the variable is limited in the range $[-A, A]$. We introduce outliers on purpose, to know if both approaches detect these outliers. Figure **??** shows in **a)** the random signal in range [-10, 10] with added outilers, and **b)** the detection of outliers by using both approaches.

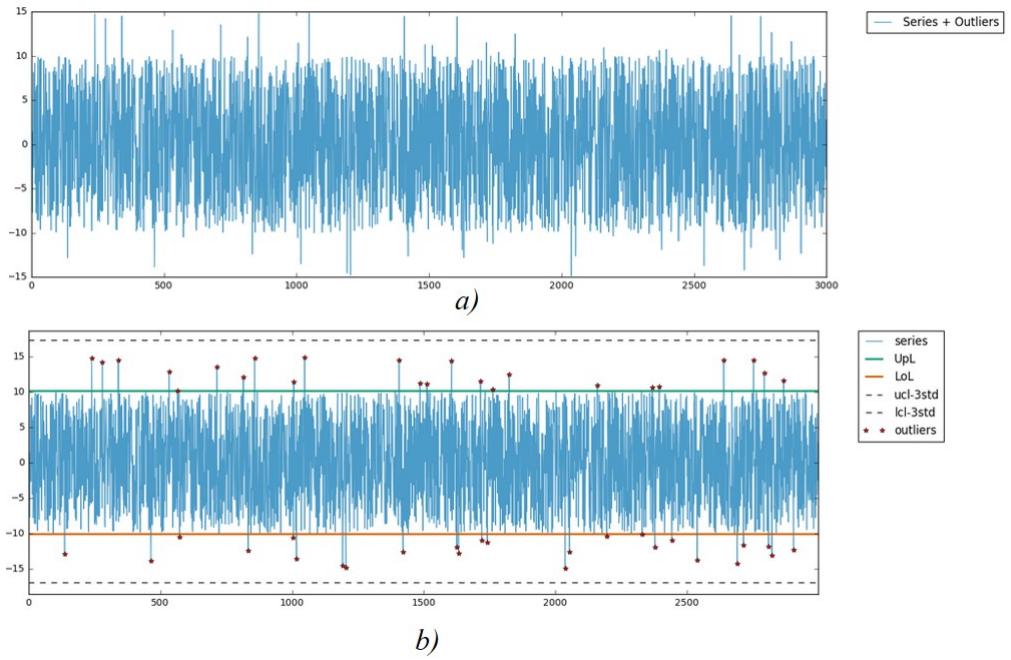


FIGURE 4.6: *a)*A random signal with range of [-10, 10] with added outliers. *b)* Detection of outliers by using six sigma and percentile analysis approach.

One observe how the limits of the six sigma approach (i.e. ucl-3std and lcl-3std) do not detect any outlier (i.e. 0%) while the percentile approach detects 100% of them. In the following experiment in figure **??**,we tested the quantity of outliers that this approach can detect with an accuracy greater than 95%. We include the results of six sigma approach to make comparison between the two approaches. The tested time series has length of 3000, and one observes the detection precision decreases less than 95% when the original time series contains more than 300 inserted outliers.

This happens because the extreme values begin to represent more than 10% of the total points of the series. Are these points part of the process?. Figure **??** shows this situation.

We use the percentile analysis to depurate the time series from values that do not correspond to the underlying process of the variable. This is important because outlier values could create havoc in the training process, visualization of results and other further steps. For example, the non convergence of the tuning parameters of the model during the training process. [12].

[10] The complete set of experiments are included as digital annex in */Thesis_project/iPythonBooks/Outlier_detection*

[11] Create an array of the given shape and populate it with random samples from a uniform distribution over [0, 1)

[12] We save the detected values of this approach in collection: *detection outlier*. Script: *1. detection_using_quartiles.py*

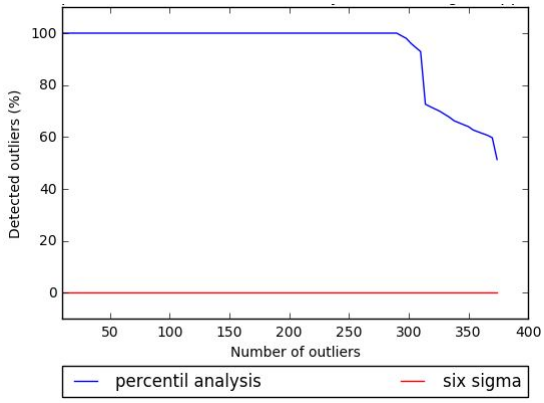


FIGURE 4.7: Testing the number of outliers that percentile analysis and six sigma approach can detect.

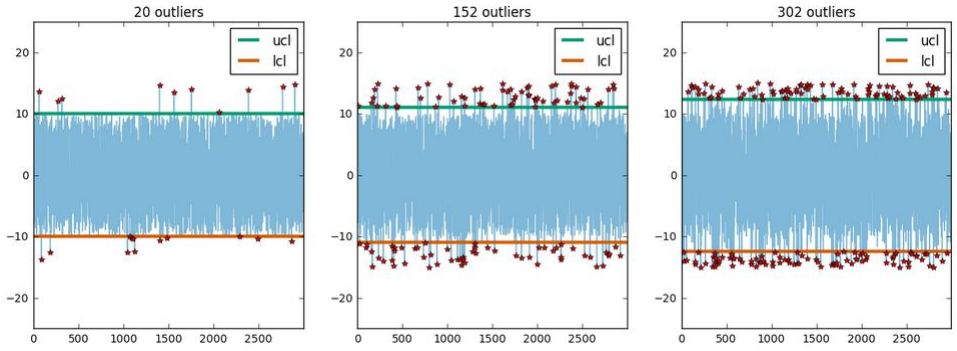


FIGURE 4.8: Percentile analysis approach spotting outliers.

The six sigma approach is used as well when we apply the SAX process according to section **??**. Table **??** shows the number of points that were detected as outliers for a set of variables. One observes that each approach has different results for each particular variable. We observe that the percentile analysis approach has best detected outliers, since there is no assumption of a Gaussian distribution of the time series as *six-sigma* does.

4.3 Feature Selection

This module assists the selection of features for the creation of multivariate samples as it was explained in section **??**. We want to choose the best features from different variables such that one sample can express information in a more high level, making more robust and integral our proposed models. This is the approach of the *GaHMM seasonal* model that is explained in section **??**. One common approach to feature selection is the use of the Kullback–Leibler distance, this approach allows the measure of information when one uses an arbitrary feature. We borrowed and adapted the concepts that are presented in (**eguchi2006interpreting** 2006) **eguchi2006interpreting** to explain the feature selection that was performed in this study. If we let P and Q be two probability distributions of one feature x over two different clusters of data. Then let $p(x)$ and $q(x)$ be their respective probability functions. The Kullback–Leibler distance is thus defined by:

$$D(P,Q) = \int p(x) log \frac{p(x)}{q(x)} dx \tag{4.1}$$

	Number of detected points	
Variable	# Percentil Analysis	# Six Sigma
V004_vent01_hum_out	148	362
V023_vent02_temp_out	105	279
V005_vent01_CO2	114	473
V022_vent02_CO2	141	474
V006_vent01_temp_out	87	264
V012_vent01_temp_in	294	531
V021_vent02_hum_out	193	403
V029_vent02_temp_in	324	604
V037_tabs_cold_SW	87	152
V074_tabs_warm_NO	310	280
V075_tabs_cold_NO	164	132
V099_blinds_height_N_o	35	13
V102_blinds_height_N_i	33	13
V105_blinds_height_O_o	54	17
V108_blinds_height_O_i	34	12
V111_blinds_height_S_o	48	16
V112_blinds_angle_S_o	30	12
V114_blinds_height_S_i	50	14
V115_blinds_angle_S_i	35	10
V117_blinds_height_W_o	26	11
V118_blinds_angle_W_o	25	11
V120_blinds_height_W_i	38	12
V121_blinds_angle_W_i	39	11

TABLE 4.1: Number of points (individual measurements) spotted as outliers.

Two basic properties of D can be observed from this formula:

a. non-negativity $D(Q, P) \geq 0$, and a especial case: if $P == Q$ then $D(Q, P) = 0$

b. asymmetry $D(P, Q) \neq D(Q, P)$

Since this distance is asymmetry, it is normal practice to use the symmetric K–L distance (also known as J-distance) **coetzee2005correcting**

$$J(p, q) = D(p, q) + D(q, p) \tag{4.2}$$

It follows then, that if $J(p, q) = 0$ the two clusters characterized by the feature x are similar. Stated in another way, the two clusters should be a single cluster (or a feature x does not provide any information helpful to the discrimination of these two groups). In contrast, if $J(p, q)$ is high then surely the feature x describes two different clusters. Thus, one is interested in finding the features that return a high J value on latent clusters. This selection can be done by sophisticated and robust methods (**sui2013information** 2013 **sui2013information**). However, this work uses a simplified version for feature selection, since we are only interested in having an approximate ranking of features, due to the evaluation process for GaHMM scoring the clustering quality of each model, so then we can be sure that the feature selection was good. The final objective of our proposition [13] is a collection of JSON documents where one can rank the features from the highest "gain of information" until the lowest one. This ranked list of feature is created by the following steps (see figure **??** to illustrate the process):

1 Features are calculated in a daily fashion using the script *2.statistics daily.py*. The feature data space (i.e. daily feature collection) is created according to the feature list in annex **??**.

2 Random groups of data in the feature space are created. In our case, 7 groups of data were created by using the weekday name label [14].

3 Chose one feature from the feature list and perform the K-L distance $J(p, q)$ over all the created groups in a pairwise fashion. Save the best K-L distance into the JSON document collection *'feature selection'*.

[13] Script *2.entropy_calculation.py* implements this proposition.

[14] This choice is arbitrary. It was done because we observed that variables like CO_2, temperature and others change between working days and weekends. Therefore it was considered convenient to create the random clusters by using the names of the day. Other ways to create random groups can also be done.

4 Order features according to the best J value. (i.e. ranked list).

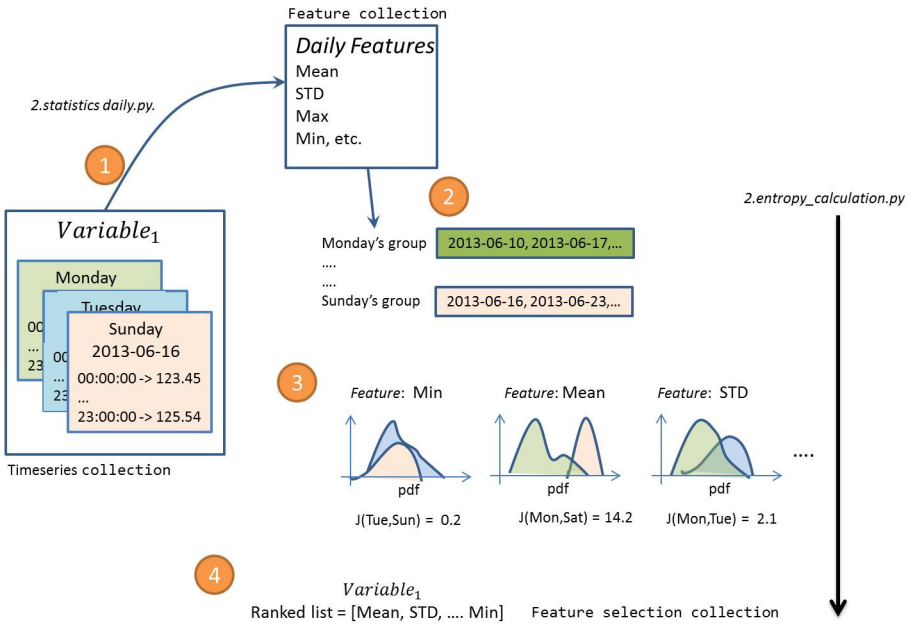


FIGURE 4.9: Schema of the implemented feature selection

By always conserving the same random groups for each interaction, it is possible to measure (approximately) the gain of information that each feature provides. Other analyses like the correlation between features can be done **sui2013information** but this is outside of the scope of this study. One can see that each variable (e.g. CO_2, humidity, temperature, etc.) has its own ranked list (i.e. some features that are important for a set of variable V_1, are less important for another set of variable V_2). Since we keep an invariable definition of the random groups, at the end, one can obtain a ranking of the features according his respective gain of information.

Ripple Factor

We calculate 18 features in daily fashion for each variable, the complete list of features are in annex **??**. We propose one feature that help us to gain information for the seasonal model explained in section **??**. Here, we explain its concept: Given a time series of length N, a new feature called ripple factor captures the shape of the trend that crosses the mean μ_x of the series. The feature has three key values to converge:

a) The feature converges to 1 when at most points of the time series are above the mean.

b) The feature converges to 0, crossing the mean on many occasions.

c) The feature converges to -1 when most of the point of the time series are below the mean.

This behavior is shown in figure **??**. To achieve the previous description, we use a common ratio (i.e. 1/N) for all the points that belong to the series, and create a power series that maps the position of each series point **p** to the exponent of each term according to the condition $\mathscr{C}_f$. In this way, all the points that are above the mean are used to form the term A_b and the rest of the points are used for the term B_e. Finally, the difference between these two terms divided by the size of the series is the mathematical definition of our feature. The condition $\mathscr{C}_f$, the range and mathematical definition of the proposed feature are specified in equation **??**.

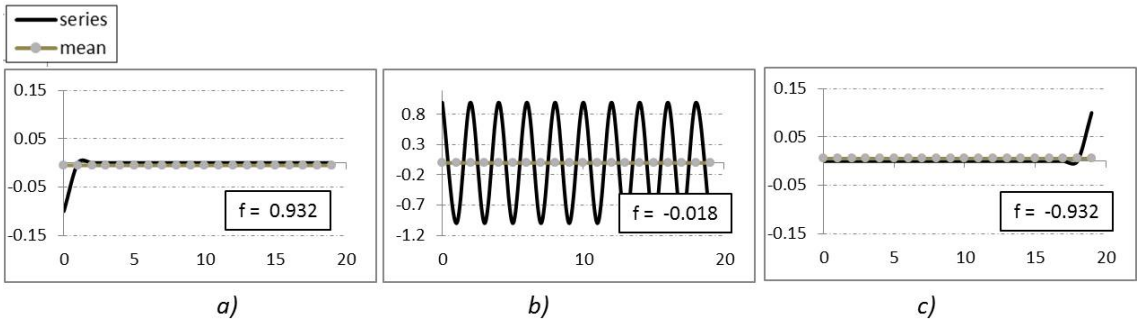


FIGURE 4.10: Behavior of Ripple Factor feature

$$
\begin{aligned}
&\text{Time serie} && : \quad X(n) = [x_0, x_1, x_2, ..., x_p], \quad |X| = N \\
&\text{Feature range} && : \quad \mathscr{R}_f\{X(n)\} \in (-1, 1), \quad \mathscr{R}_f\{X(n)\} \in \mathbb{R}^+ \\
&\text{Feature condition} && : \quad \mathscr{C}_f\{X\} = \begin{cases} A_b = \sum\limits_{x_p >= \mu_x} 2^{\,p\frac{1}{N}}, & \forall p \in [0, N-1] \\ B_e = \sum\limits_{x_p < \mu_x} 2^{\,p\frac{1}{N}}, & \forall p \in [0, N-1] \end{cases} \\
&\text{Feature definition} && : \quad \mathscr{R}_f\{y(n)\} = ln(2) \cdot \frac{A_b - B_e}{N}
\end{aligned}
\tag{4.3}
$$

The limits of this feature are found when either A_b or B_e is equal to zero (but not simultaneously). For the upper limit, this occurs when all the points are above or equal to the mean. Figure **??** shows how the limits behave according to the size of the series. Furthermore, we show in equation **??** the upper and lower limit of this feature.

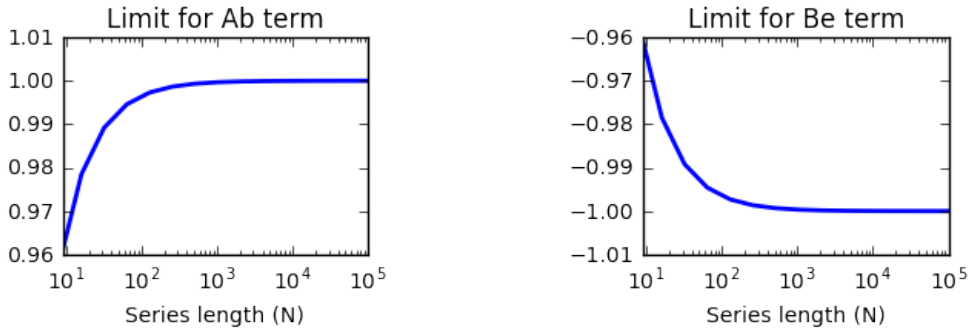


FIGURE 4.11: Limits for the Ripple Factor feature according to the length of the time series.

$$
\begin{aligned}
\text{Upper limit when } B_e = 0 \quad &: \quad \lim_{N\to\infty} ln(2)\cdot\frac{A_b - B_e}{N} = ln(2)\cdot\lim_{N\to\infty}\frac{\sum_{p=0}^{N-1} 2^{\,p\frac{1}{N}}}{N} \\
&: \quad ln(2)\cdot\lim_{N\to\infty}\frac{1}{N(2^{\,\frac{1}{N}}-1)} = 1 \\
\text{Lower limit when } A_b = 0 \quad &: \quad \lim_{N\to\infty} ln(2)\cdot\frac{A_b - B_e}{N} = -ln(2)\cdot\lim_{N\to\infty}\frac{-\sum_{p=0}^{N-1} 2^{\,p\frac{1}{N}}}{N} \\
&: \quad -ln(2)\cdot\lim_{N\to\infty}\frac{1}{N(2^{\,\frac{1}{N}}-1)} = -1
\end{aligned}
\tag{4.4}
$$

The following part discusses several examples that use this feature. We propose comparable examples where the mean is equal to zero in all the cases. The capacity of the ripple factor to discriminate time series that have similar statistics is appreciated in figure **??**. The three time series have the same mean, variance, percentiles, and maximum and minimum value, but the ripple factor for each one is different, so that, we can discriminate for example a sine signal from a cosine signal.

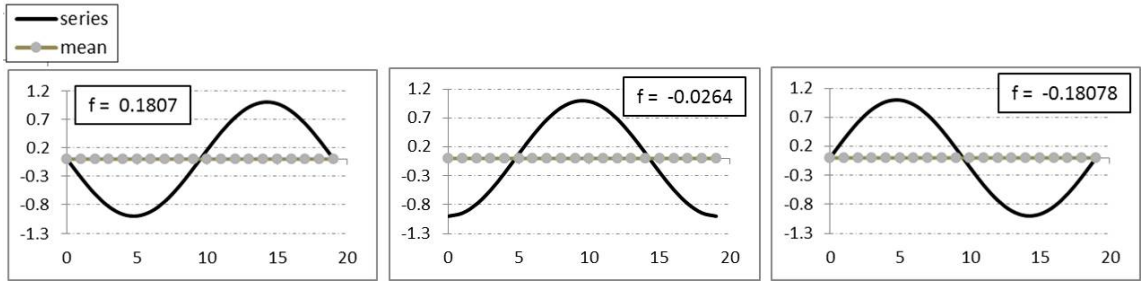


FIGURE 4.12: Ripple factor for sine and cosine signal.

In figure **??**, we appreciate how this feature changes depending on the shape of each particular signal. The examples show how this feature changes as soon as the oscillations approach the mean. The closer the oscillations are to the mean, the nearer to zero the feature is.

Extension of the ripple factor feature:

The last behavior explained in figure **??** allows to extend the concept of the ripple factor to one that is most general. If we change the definition of the feature condition $\mathscr{C}_f$, we can substitute the parameter of comparison (in the original definition μ_x) for any other numeric value $v \in \mathbb{R}$. This change is useful when for some reason we need to know whether or not most of the oscillations are close to the value v

Discussion of the Ripple Factor:

As we see in figure **??** the feature is sensitive to the length of the series. Nevertheless, for this study the feature was calculated using a fixed window of one day (i.e. sub-sequences of same size), therefore the limits are well defined. To know how much these limits differ, we consider a time series of size $N = 10$ and one of size $N = 50$. Their limits are 0.965 and 0.993

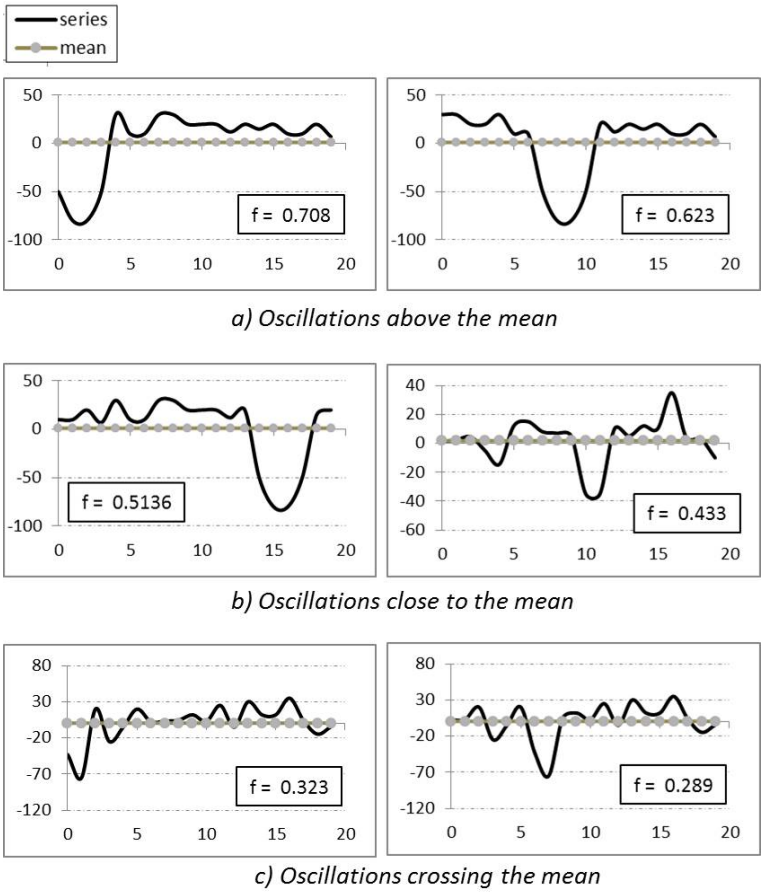


FIGURE 4.13: Ripple factor for sine and cosine final.

respectively and the difference is only 0.028. To know whether this difference is significant, an evaluation must be done. The mentioned evaluation goes beyond the purposes of this study and we only use this feature because the length of sub-sequences is fixed (therefore the limits are well defined) and because using the Kullback–Leibler divergence, this feature obtains a good gain of information being selected for the *GaHMM-seasonal* profile, explained in section **??**.

Another relevant aspect of this feature is the fact that it does not depend on the amplitude of the treated signal. This fact allows the signal to be amplified or reduced without changing the ripple factor. This can be an advantage when the shape of the series is an important characteristic to analyze. Finally, this feature is complementary to other statistics such as the the mean, standard deviation and others, but not a substitute for them.

4.4 GaHMM Modeling

Hmmlearn **gahmm_manual** [15] is the python library used in this project. We use the internal methods that this library offers in order to perform the training process. We recall here the training process that was explained in section **??**, and we add information about the employed library.

[15] http://hmmlearn.readthedocs.io/en/stable/

- Learning/ training process: The HMM model is fitted with the observed samples $\mathbb{O} = (O_1, O_T)$. The definition of the observed samples is very important since it defines the kind of HMM to use. Each proposed model define the samples according to the underlying problem that each model wants to solve. In all the cases we use a *GaHMM* model since we use sequential continuous values. The definition of the observed samples are explained in each model. The library method to use for the training process is called: **fit** method **gahmm_manual** this implements the solution that was explained in section **??**.
- Evaluation process: For evaluation purposes, the log probability of $P(\mathbb{O}|\mathbb{S}, \lambda)$ (i.e. equation **??**) is used to select the best trained model. The model that fits the best the parameters of λ is the one who has the greatest probability. The library method that calculates $P(\mathbb{O}|\mathbb{S}, \lambda)$ is the **score** method using the forward/backward algorithm (section **??**).
- Decoding process: When one obtain the best model, there is a perfect matching between the observed samples $\mathbb{O}$ and the sequences of states $\mathbb{S}$. One says that each hidden state S_k emits one sample O_i. To know the likely sequence of hidden states the **predict** method solves the decoding problem by using Viterbi algorithm. (section **??**).

4.4.1 GaHMM Training process

The training process is done by solving the HMM evaluation problem (section **??**). This evaluation is done by the *score* method **gahmm_manual** where the log probability of $p(\mathbb{O}|\lambda)$ is calculated. The log probability has practical advantages, and is useful for finding the best trained model. This work adopts the use of the log probability of $p(\mathbb{O}|\lambda)$ for finding the best model. However, since its value is not easy to understand, this document reports probability in the interval of [0,1]. For this purpose, given that each observation could have been drawn by each every hidden state with a certain probability, this allows us to perform a time-dependent clustering task **[pfundstein2011hidden]**. In this way, given the observed sequence $\mathbb{O} = (O_1, O_2, ...O_T)$ and the correspondent sequence of hidden states $\mathbb{S} = (S_1, S_2, ..., S_T)$ [16], we calculate the average of $p(\mathbb{O}|\mathbb{S})$ as follows:

$$\overline{p}(\mathbb{O}|\mathbb{S}) = \frac{1}{T}\sum_{i=1}^{T} p(O_i|S_i) \tag{4.5}$$

Where T is the total number of observations, S_i is the likely hidden state that emits the observation O_i, and $p(O_i|S_i) \in [0, 1]$ is the probability that the observed sample O_i was emitted by the hidden state S_i. The hidden state sequence $\mathbb{S}$ that matches with the observed sequence is found by the **predict** method **gahmm_manual** of the GaHMM library. It follows that the bigger the log probability is, the better the average $p(\mathbb{O}|\mathbb{S})$ is. Therefore, the equation **??** shows a direct relationship with the log probability and his range is in the interval of [0,1].This is the way in which we will report the results throughout the document.

Cross validation process

As is explained in section **??**, the EM algorithm is a gradient-based optimization method that can be stuck in local optima. Hence, it is generally recommended to run several instances of the model with various initialization values, and then choose the best one. This process of training leads to problems of over-fitting in some cases. To avoid this, and to guarantee a well-trained model, the cross-validation technique is performed in most of the cases. Since our proposition deals with time series, the k-fold cross-validation technique is not necessarily valid if we do not consider the time correspondence (**bergmeir2015note** 2015) **bergmeir2015note**

[16] Recall: There are K hidden states for a HMM, therefore $\mathbb{S}$ is a permutation of K values with length equal to T

GaHMM model	Script	Observed sample type
Profile	4.hmm_learning_per_variable_k-fold.py	Univariate sample (daily profile of the selected variable)
Seasonal	4.hmm_learning_k-fold.py	Multivariate sample (features of several variables)
Interactional	4.hmm_learning_k-fold.py	Multivariate sample (linear correlation between variables)

TABLE 4.2: Implemented scripts for training

The traditional k-fold cross validation performs a random partitioning of the original sequence into k equal sized blocks. One inconvenient is that the samples are treated as independent sequences and not as a part of a sequence of events. Fitting the HMM with a random order of samples O_i causes a poor performance of the model. Our proposed approach for cross-validation follows the indications of **bergmeir2015note**'s work, therefore the random selection is avoided. In short, the partial observed sequences for training $\mathbb{O}_{train}$ and testing $\mathbb{O}_{testing}$ respect the original order of the time series. Figure **??** shows the schema for k-fold cross validation of $k = 5$.

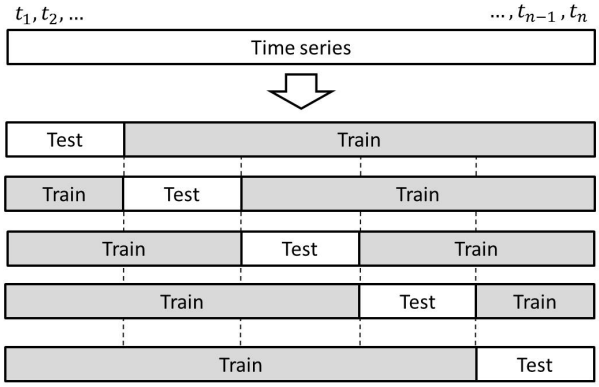


FIGURE 4.14: k-fold cross validation with $k = 5$. Note that the order of the time series is not altered, no random selection is done.

Fixing parameters of GaHMM and cross validation process

Fortunately, the GaHMM library **gahmm_manual** proposes some parameters by default. Since the Gaussian Model implements a monitor routine for checking the internal convergence, there was no need to change the default parameters for: *min_covar, means_prior, means_weight, n_iter, tol*. Nevertheless, in order to guarantee the convergence of the trained model, each hidden state uses a diagonal covariance matrix (i.e. *covariance_type= "diag"*). Additionally, as it was explained in section **??**, Viterbi algorithm was chosen by default. The only parameter that was modified for each training round was the number of components, which is the number of hidden states to use (i.e. *n_components* parameter). Regarding the cross validation process, we found by doing experiments that select k in the interval of [10, 15] is a good choice. If $k < 10$, the final clusters tend to be mixed, while $k > 20$, the over-fitting effect is evident.

Scripts for training Two python scripts were created for training the three different models explained in section **??**. Table **??** presents the context were each script is used. Details about the models are in the next section, here the structure of each training script is explained.

Both training scripts vary the *n_components* parameter in an interval of $[n_{min}, n_{max}]$, in which $n\_components = n$ the k-fold cross validation process is executed, afterwards the best model is chosen (for us, this is called a round). At the end, the best model M_j among all the rounds is picked as the final model. This process is explained in the following pseudo-codes:

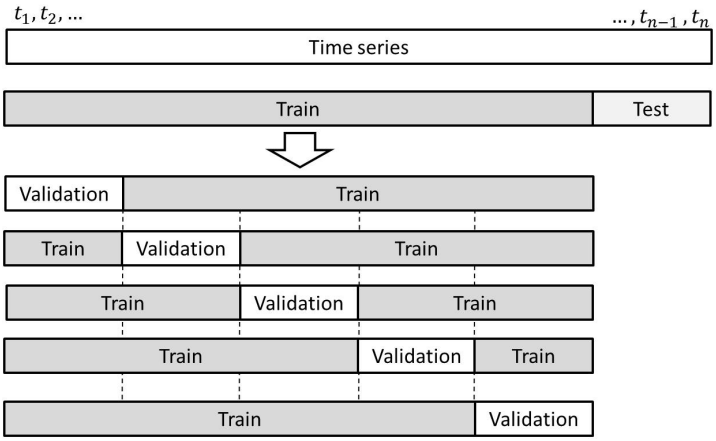


FIGURE 4.15: k-fold cross-validation with $k = 5$ for a *GaHMM* model. The testing set is separated to be used at the end of the k-fold cross validation process (Script: *4.hmm_learning_k-fold.py*).

Pseudo-code for script *4.hmm_learning_k-fold.py*:

1 Extract information from the *metadata* collection for the running of the corresponding training process. (i.e. variables, categories, daily vector to use, etc.)

2 Establish the timeline to work with, the training set and the testing set [17]. Here the observed sequence $\mathbb{O}$ is defined.

3 Define the minimum and maximum number of hidden states for the training process. $n \in [n_{min}, n_{max}]$. Initialize $n = n_{min}$.

4 Initialize a Gaussian Hidden Markov model with $n\_components = n$.

5 Perform the k-fold cross-validation over the training data set, see figure **??**. At each step of the cross-validation process, the best *GaHMM* model (M_i) is chosen by using the validation set [18].

6 Test the best model (M_i) that was found in the k-fold cross-validation process using the test set. Compare the current model (M_i) with the last best model (M_j) and elect the best model [19].

7 Increase the number of hidden states $n = n + 1$. If $n < n_{max}$ then go to step 4, otherwise the training process is finished.

Here the pseudo-code for script *4.hmm_learning_per_variable_k-fold*:

1 Extract information from the *metadata* collection for running the correspondent training process. (i.e. variables, categories, daily vector to use, etc.)

2 Establish the timeline to work with, and the training set [20]. Here the observed sequence $\mathbb{O}$ is defined.

[17] The testing set differs from the validation set. The testing set is a reserved part of the whole data that is used just at the end of the whole training process. Its purpose is to test if the trained model is capable also of working on data that it has not seen before

[18] For this purpose, the log probability of $p(\mathbb{O}|\lambda)$ of the trained model is used.

[19] M_j is the best model to have been found among all the rounds. When $n = n_{min}$ then $M_j = M_i$.

[20] Since we are interested in finding all possible daily patterns across multiple years, at the end of the cross validation process, we test each model with all the available data.

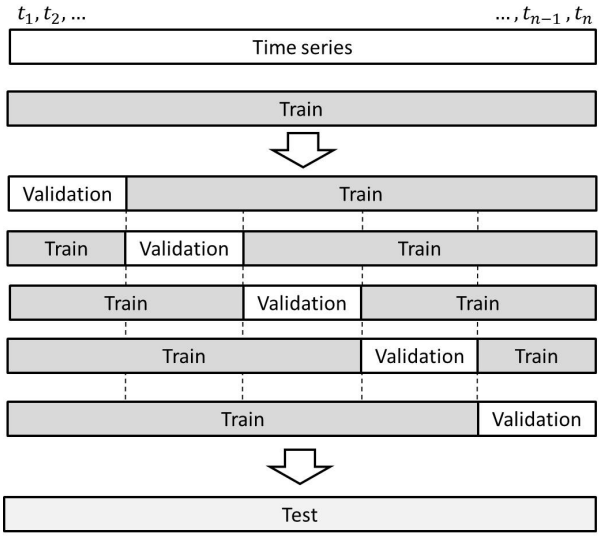


FIGURE 4.16: k-fold cross validation with $k = 5$ for a *GaHMM* model. The testing set is the entire time series. This allows the verification of the fitness of the model across the entire time series (Script: *4.hmm_learning_per_variable_k-fold.py*).

3 Define the minimum and maximum number of hidden states for the training process. $n \in [n_{min}, n_{max}]$. Initialize $n = n_{min}$.

4 Initialize a Gaussian Hidden Markov model with $n\_components = n$.

5 Perform the k-fold cross-validation over the training data set, see figure **??**. At each step of the cross-validation process, the best *GaHMM* model is chosen by using the validation set [21].

6 Test the best model M_i that was found in the k-fold cross validation process using the test set. Compare the current model M_i with the last best model M_j and elect the best model.

7 Increase the number of hidden states $n = n + 1$. If $n < n_{max}$ then return to step 4, otherwise the training process is finished.

4.4.2 Implemented GaHMM models

4.4.3 GaHMM - profile model

We propose the use of the *GaHMM - profile* model for performing a time-depending clustering task over a time series of interest. This implies that each variable in the multivariate building dataset has his own model. To exemplify the time-depending clustering task, we can take the time series of CO_2 levels of the North-East ventilation system. The time series is divided in a daily fashion, consequently 1081 observed samples were created for the whole timeline ($\approx$ 3 years). Figure **??** shows how each sub-sequence of the entire trend (i.e. daily profile) is considered as an observed sample O_i.

The observed sequence $\mathbb{O}$ is defined as a matrix of 24×1081 size. Each sample O_i is a vector of length $L = 24$ where each hour of the day has a corresponding value. Once the observed sequence is defined, the training process for the *GaHMM - profile* is performed according to section **??** (*4.hmm_learning_per_variable_k-fold*). The best *GaHMM - profile* model is found by maximizing the likelihood of $p(\mathbb{O}|\lambda)$. At the end of the process, once the best model is found,

[21] For this purpose, the log probability of $p(\mathbb{O}|\lambda)$ of the trained model is used.

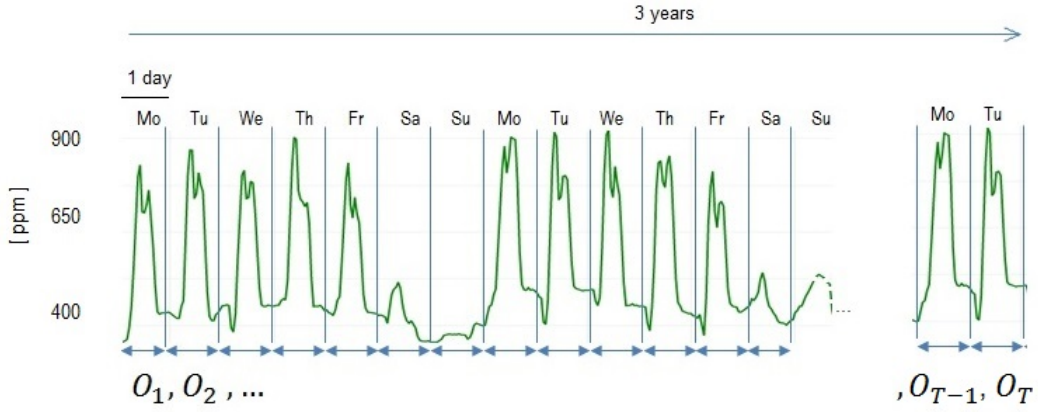


FIGURE 4.17: The CO_2 time series for the North-East ventilation system is split in a daily fashion. Each daily profile is an observed sample O_i.

we assume that there is a hidden state S_k that is responsible for generating a sample O_i. In this way, we can find the best match between the sequence of hidden states $\mathbb{S}$ and the observed sequence $\mathbb{O}$. Using the Viterbi's algorithm we can discover the hidden state sequence $\mathbb{S}$ and, in this way, perform the time-depending clustering. Figure **??** shows a hypothetical example where the CO_2 time series is disaggregated by daily profiles O_i and the observed sequence is matched with the hidden state sequence $\mathbb{S} = [2, 5, 0, 4, 2, 4, 1, 6, 3, 3, 0, 2, 4, 1]$. Then using the identification ID of the hidden states, we can cluster days with similar daily profiles. For practical purposes, we name each cluster with the number of the associated hidden state (i.e. $ID = S_k$).

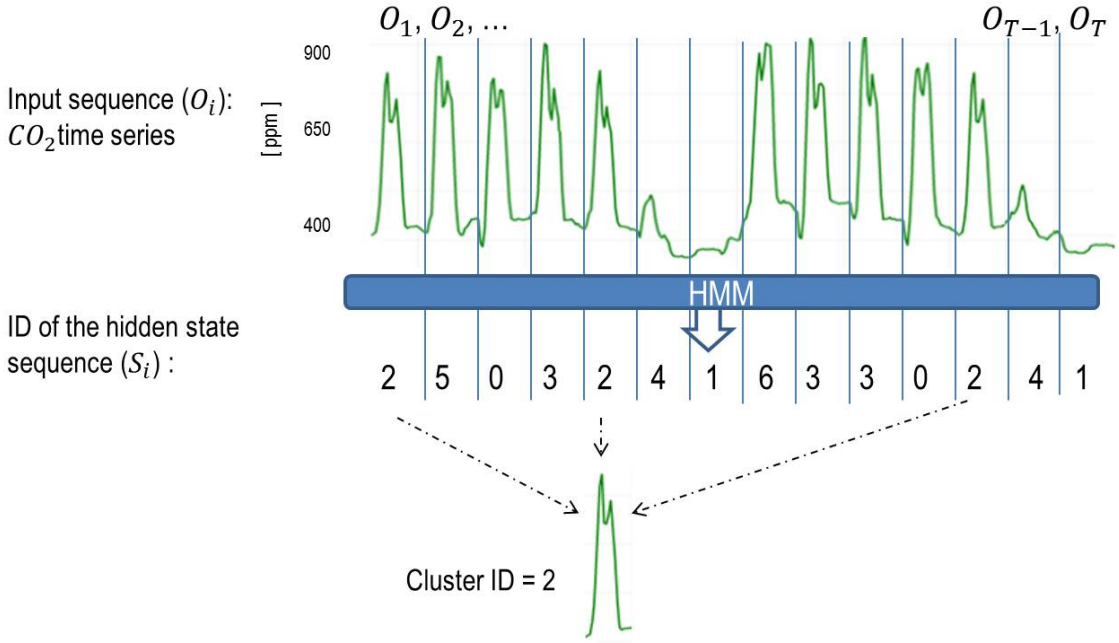


FIGURE 4.18: Hypothetical example: The CO_2 time series is "emitted" by a $GaHMM$ model of 7 hidden states. $\mathbb{S} = [2, 5, 0, 4, 2, 4, 1, 6, 3, 3, 0, 2, 4, 1]$. At the bottom-center the daily profile for cluster $ID = 2$ corresponding to the hidden state $S_k = 2$.

The results of this model are discussed in section **??** where the model is applied over the

CO_2 time series in a case study. The quality of the cluster is reflected directly in the log probability of $p(\mathbb{O}|\lambda)$ and the average of $p(\mathbb{O}|\mathbb{S})$. However, to get a visual evaluation of the model, each cluster can be plotted out using box plots. A good cluster is one who has few (if any) outliers (i.e. fliers) outside of the whiskers. Examples of these daily profile clusters are in annex **??**.

4.4.4 GaHMM - seasonal model

Several factors affect the way in which a building performs during the day. One of them is without a doubt, the outdoor conditions **miller2015forensically**, **miller2015automated**, **djamila2017indoor** The main purpose of this model is to cluster days according to the similarity of the outdoor conditions. In this way, we can use the information for understanding the dynamics of the building across seasons. For this purpose, the available **weather variables** (i.e. *Outdoor temperature of the building, sunshine presence, precipitation and weather temperature*) are used for finding seasonal groups in the dataset.

Implementation A daily sample O_i for this model is defined as a vector that contains selected features from the different weather variables. For practical purposes, this vector is herein called as S_{vector}. Features from the weather variables are calculated in a daily fashion [22]. Afterwards the entropy value for each feature is calculated by using the Kullback–Leibler divergence distance [23] according to section **??**). This last value and the training process of the *GaHMM* allows us to perform the feature selection in an indirect fashion. The final seasonal model uses only six features of a list of nineteen features (annex **??**). These six selected features [24] present in general, the maximum values of entropy among the list of features. Table **??** shows the gain of information of each individual feature, the bigger the value is, the more information we gain. This information can be retrieved from the *'feature selection'* collection. In this way, each feature can be ranked according its gain of information. Thus, F_1 gains more information with and F_{19} gains the less. In this way, from a collection of m variables, a daily sample O_i contains f features of each variables V, that is:

$$\begin{aligned} O_i = S_{vector} &= [F_1^1, F_2^1, ...F_f^1, F_1^2, F_2^2, ...F_f^2, ..., F_1^m, F_2^m, ...F_f^m] \\ where: & \\ & F_1^i, F_2^i, ...F_f^i \in V_i \quad i \in [1, m] \end{aligned} \tag{4.6}$$

In the end, the observed sequence $\mathbb{O}$ is a matrix of $fm \times 1081$ size. Each sub-sequence of features O_i for each day during the entire timeline (1081 days). Since the observed sequence is defined, the training process is performed according to section **??**. The clustering quality of this model is evaluated on section **??**, and the results of the best trained model are exposed and discussed in section**??**.

4.4.5 GaHMM - interactional model

This model attempts to cluster days where the interaction between variables of the building dataset behave similarly. For this purpose, we use the Pearson correlation coefficient (r) to measure the linear correlation between two variables X_1 and X_2 [25]. r varies between — 1 and +1, which represent perfect negative and perfect positive linear relationships, respectively. When r = 0, there is no linear correlation between the variables **cohen2013applied** To perform

[22] script 2.*statistics daily.py*

[23] Script: 2.*entropy_calculation.py*

[24] The selection feature is done in an indirectly fashion by the training process, see in section **??**. The best model is reached by using six features among all, observe table **??** where the best features are in orange color

[25] X_1 and X_2 are time series of the building dataset. It could be the entire time series or just a selected period.

	Sunshine presence		*Outdoor temperature*		*Weather Temperature*		*Precipitation*	
	f	*J(f)*	*f*	*J(f)*	*f*	*J(f)*	*f*	*J(f)*
F_1	r_factor_st	inf	r_factor_ed	1.00	r_factor_ed	1.00	max_ed	inf
F_2	r_factor_ed	0.23	r_factor	0.51	r_factor	0.98	(max-min)*std	inf
F_3	min_me	0.01	(max-min)*std	0.24	(max-min)*std	0.59	25%	4.53
F_4	max_st	0.01	r_factor_st	0.21	std	0.42	min_me	1.98
F_5	max_ed	0.01	dev_u	0.17	dev_u	0.31	min	1.82
F_6	mean	0.01	std	0.16	r_factor_st	0.31	min_ed	0.64
F_7	50%	0.01	min_ed	0.10	max_ed	0.16	max_me	0.13
F_8	dev_u	0.01	max_ed	0.08	min	0.15	dev_u	0.12
F_9	r_factor	0.01	75%	0.07	min_st	0.15	std	0.11
F_{10}	75%	0.01	max	0.06	max_st	0.13	min_st	0.09
F_{11}	r_factor_u	0.00	50%	0.06	75%	0.12	max	0.09
F_{12}	max_me	0.00	max_st	0.06	min_me	0.11	50%	0.07
F_{13}	max	0.00	max_me	0.06	25%	0.09	75%	0.04
F_{14}	(max-min)*std	0.00	mean	0.05	max_me	0.09	mean	0.02
F_{15}	std	0.00	min_st	0.05	min_ed	0.09	max_st	0.02
F_{16}	min	0.00	min_me	0.04	max	0.09	r_factor_ed	0.02
F_{17}	min_ed	0.00	25%	0.04	50%	0.07	r_factor_st	0.01
F_{18}	25%	0.00	min	0.03	mean	0.07	r_factor_u	0.01
F_{19}	min_st	0.00	r_factor_u	0.02	r_factor_u	0.05	r_factor	0.01

TABLE 4.3: Entropy value *J* for the *weather variables*. It shows the gain of information for each variable according to section **??**. Definition of each feature in annex **??**.

a multivariate analysis of variables in the building data set, a Pearson correlation matrix is created using a set of variables $V = [X_1, X_2, .., X_n]$, as follows:

$$R = \begin{pmatrix} r(X_1, X_1) & r(X_1, X_2) & ... & r(X_1, X_n) \\ r(X_2, X_1) & r(X_2, X_2) & ... & r(X_2, X_n) \\ ... & ... & ... & ... \\ r(X_n, X_1) & r(X_n, X_2) & ... & r(X_n, X_n) \end{pmatrix} \tag{4.7}$$

Where $X_i = [x_1, x_2, ..., x_T]$ is an individual time series of length T of the set of variables V and the individual Pearson correlation coefficient (r) between two time series X and Y is calculated by:

$$r(X, Y) = \frac{\sum_{k=1}^{T}(x_k - \overline{x})(y_k - \overline{y})}{\sqrt{\sum_{k=1}^{T}(x_i - \overline{x})^2}\sqrt{\sum_{k=1}^{T}(y_i - \overline{y})^2}} \tag{4.8}$$

In our approach, the R matrix is calculated in a daily fashion, so that each time series has a length $T = 24$. For the whole timeline, 1081 matrices of size $n \times n$ are saved in the 'correlation_matrix_daily' collection by the script *2.correlation_matrix_v1_daily*. We define two sets of variables V, one for each part of the building. That is V_1 for the North-Eastern part and V_2 for the South-Western part of the building. In addition, the weather variables are included in both sets. This matrix allows the visualization of the average Pearson correlation coefficient $\overline{r}$ between variables by using a Hierarchical Edge Bundles [26] visualization **??**, see figure **??**.

$$\overline{r}(X, Y) = \frac{1}{L}\sum_{i=0}^{L} r(X_i, Y_i) \tag{4.9}$$

The average value of the Pearson correlation coefficient is calculated by equation **??** where X_i and Y_i are a daily time series, and L is total number of daily time series. Figure **??** shows in red and green colors, the average values of r during the entire timeline. The green color implies a positive correlation while the red implies a negative correlation between variables. The reason for a pair of variables having a $\overline{r}$ value close to zero, is because there is either

[26] Web server (*Flask_project.py*): `http://127.0.0.1:5000/correlation`

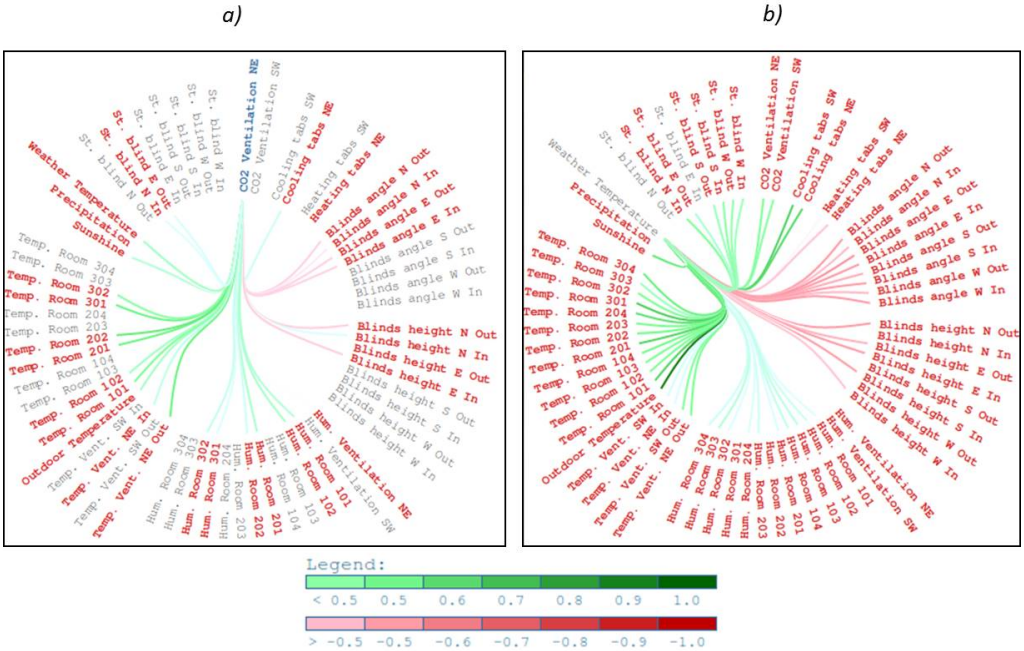


FIGURE 4.19: Average value of the Pearson correlation coefficient for variables of the building dataset. *a)* Variables correlated with the measures of CO_2 levels of the North-East ventilation system. *b)* Variables correlated with the weather temperature measures.

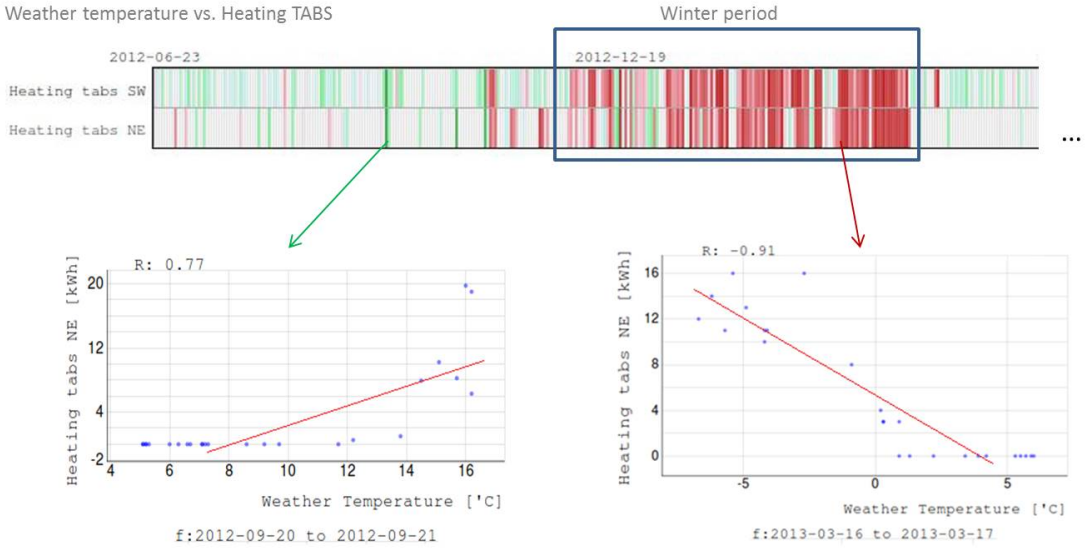


FIGURE 4.20: Average value of the Pearson correlation coefficient for variables of the building dataset. *a)* Variables correlated with the measures of CO_2 levels of the North-East ventilation system. *b)* Variables correlated with the weather temperature measures.

no correlation between them or because the correlation changes frequently depending on the season and/or the day of the week (i.e. going from positive to negative and vice versa). This is the case for instance of the correlation between weather temperature (°C) and heating TABS energy consumption (kWh). Figure **??** refers to the aforementioned case. It is clear to see how the correlation between these two variables changes during the winter period. Furthermore,

the fact that the heating TABS of the South-West part of the building has more slightly positive correlation than the North-East part (when not in the winter period), is remarkable. This is the reason why the $\overline{r}$ value for the south-west TABS is close to zero (see **??** *.b)*).

The *GaHMM - interactional* model uses the information from the matrix R. Each observed samples O_i is defined by a vector called R_{vector}. This vector contains the r value for variables of interest.

$$O_i = R_{vector}(X_i) = [r(X_i, X_1), r(X_i, X_2), ..., r(X_i, X_n)] \tag{4.10}$$

For our proposed model, X_i refers to the CO_2 levels of the North-East or South-West ventilation systems. The variables $X_1, ..., X_n$ correspond to different zones of the building (i.e. North-East or South-West). In this way, two R_{vector} are defined for each zone in a daily fashion. Since the observed sample O_i is defined, the training process can be performed according to section **??**. The evaluation and results of this model are exposed and discussed on sections **??**, **??**.

4.5 Hierarchical Agglomerative Clustering

In this section, we adapt the concepts that were explained in section **??**, therefore in our context, an object is called a cluster profile. Since clustering is the grouping of similar objects, one can use different measures to determine whether two objects are similar or dissimilar. There are two main type of measures used to estimate similarity: distance measures and similarity measures **maimon2007soft** Normally, distance measures are used for determining the similarity between two objects in the HAC approach. Here a brief list of some of them: *Euclidean distance, squared Euclidean distance, city-block (Manhattan) distance, Chebychev distance, power distance, Mahalanobis distance*. We do not go in details about each dissimilarity/similarity distance, the reader can find details in **maimon2007soft**, **saraccli2013comparison**, **mullner2011modern** However we consider important to add the definition of the euclidean distance (equation **??**), for exemplification purposes:

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_i - q_i)^2 + \cdots + (p_n - q_n)^2} \tag{4.11}$$

$d(p, q)$ is the euclidean distance of two daily profiles p and q, where p_i corresponds to a measure value that belongs to the daily profile p. The Hierarchical agglomerative clustering creates a distance matrix D using a distance measure. For example, table **??** is the distance matrix of 5 hypothetical daily profiles, where profiles 3 and 5 are the closest ones. This implies the first hierarchical cluster (i.e. $id = 35$).

	1	2	3	4	5
1	0				
2	9	0			
3	3	7	0		
4	6	5	9	0	
5	11	10	2	8	0

FIGURE 4.21: Distance matrix D of 5 hypothetical daily profiles, using the euclidean distance. First interaction.

Now the distance between any cluster profile against the hierarchical cluster $id = 35$ should be defined. This new similarity distance between hierarchical nodes is what is called as linkage method. There are different methods for linkage: *Single link, Complete link, Group average, Ward's method, etc.*. The reader is invited to refer more information about linkage methods in

maimon2007soft, **saraccli2013comparison**, **mullner2011modern** We include in equation **??** the group average linkage method, for illustration purposes:

$$d(A,B) = \frac{1}{|A| \cdot |B|} \sum_{p \in A} \sum_{q \in B} d(p,q) \tag{4.12}$$

$d(A,B)$ implies the distance that exists between hierarchical clusters A and B using the average linkage method. A hierarchical cluster collapse similar objects where dissimilarity distance is the smallest. Table **??** shows the case when the profile cluster 3 and 5 are collapsed in one hierarchical cluster 35. The distances within the distance matrix D are calculated by using equation **??**.

	35	1	2	4
35	0			
1	11	0		
2	10	9	0	
4	9	6	5	0

FIGURE 4.22: Distance matrix D of 5 hypothetical daily profiles, using the euclidean distance. Second interaction.

In a new interaction, the new hierarchical cluster is $ID = 24$ since the distance between profile 2 and 4 is the shortest one. New interactions are performed until the total reduction of the distance matrix D. The result of the hierarchical clustering can be displayed in a tree-like structure, called a dendrogram (see figure **??**), with one cluster at the top containing all the objects, and each branch groups similar objects.

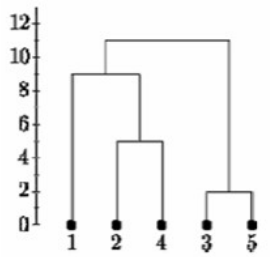


FIGURE 4.23: Resulting dendrogram from the distance matrix D.

This is the process to perform a hierarchical agglomerative clustering with cluster profiles. We indicate in a practical way, how this clustering is performed in section **??**. In addition, a measure (i.e. *Cophenetic correlation coefficient*) for knowing how faithfully a dendrogram preserves the pairwise distances between the original unmodeled data points is used. This measure allows to know which distance metric and linkage method is the most appropriated for the data **saraccli2013comparison**

4.6 Evaluation

4.6.1 Evaluation of the *GaHMM - profile* model

Our approach trains a *GaHMM-profile model* for each variable of the data set. This training process is done according to **??** [27]. The best model is elected by using the log probability of

[27] Script: *4.hmm_learning_per_variable_k-fold.py*

$p(\mathbb{O}|\lambda)$ and the results are presented as the average of $p(\mathbb{O}|\mathbb{S})$. A box plot is proposed to see the quality of the cluster profiles. Figure **??** shows **a)** a profile cluster of a poor trained *GaHMM - profile* (i.e. $\overline{p}(\mathbb{O}|\mathbb{S}) = 0.885$, $n_{component} = 9$) of the CO_2 measures of the North-East ventilation system, one can see the amount of outliers (i.e. fliers) outside of the whiskers, in contrast with **b)** a profile cluster of the best trained *GaHMM - profile* (i.e. $\overline{p}(\mathbb{O}|\mathbb{S}) = 0.993$, $n_{component} = 33$) where there are only 4 outliers.

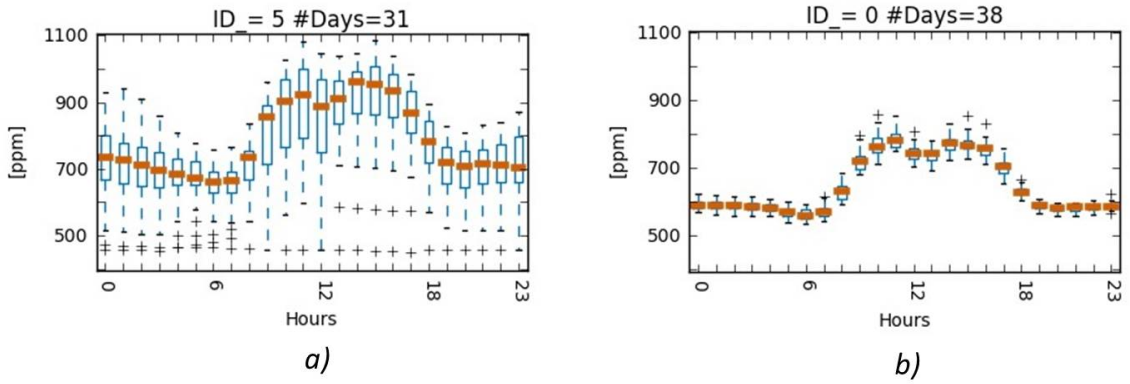


FIGURE 4.24: CO_2 cluster profile of a) *GaHMM - profile* with $\overline{p}(\mathbb{O}|\mathbb{S}) = 0.885, n_{component} = 9$ b) *GaHMM - profile* with $\overline{p}(\mathbb{O}|\mathbb{S}) = 0.993, n_{component} = 33$.

It follows then that, cluster profiles which belong to the best *GaHMM - profile* models have tiny standard deviations for each hour of the day and there are no outliers (the number of outliers is less than 10% of the whole samples). In this way, one obtains a good representation of the existing daily profiles. This is how each *GaHMM - profile* model is evaluated. Table **??** shows the evaluation of the models that are used in a case study (section **??**). Additionally, one can see the quality of clusters in annex **??** and the digital folder annex: iPythonBooks/Diversity of profiles.

Variable	TABS Cooling NE	TABS Heating NE	Exhaust Air Temperature	Intake air temperature	CO_2 NE
$\overline{p}(\mathbb{O}|\mathbb{S})$	0.9997	0.9982	0.9963	0.9976	0.9932
$n_{components}$	33	32	31	33	33

TABLE 4.4: Example of evaluation of *GaHMM - profile* models for different variables

4.6.2 Evaluation of the *GaHMM - seasonal* model

A *GaHMM - seasonal* model is trained according to section **??** [28] and **??**. To evaluate a *GaHMM - seasonal* model, we propose the use of log probability of $p(\mathbb{O}|\lambda)$, average of $p(\mathbb{O}|\mathbb{S})$ and a visualization of the S_{vector} in 3D space by using PCA. Table **??** shows the evaluation of the best trained models for different number of features f and different number of hidden states n according to sections **??** and **??**. The selected variables for this table are the outdoor temperature of the building and the weather temperature. One can observe that the average of $p(\mathbb{O}|\mathbb{S})$ improves as we add new features but this diminishes after $f = 6$. The tendency is that the more features and variables we use to train the model, the more hidden states we need to describe the clusters. The observed sample O_i becomes more scattered and therefore more clusters are needed, however even if we increase the number of hidden states, we do not get the same results that were achieved by the model with $f = 6$ and $n = 2; 3; 4$.

[28] script *4.hmm_learning_k-fold.py*

Since the best results are achieved with $n = 2, 3, 4$, we can choose to group the seasonal patterns in 2 or 3 or 4 clusters. We decide to group the patterns in four latent groups. They were named as *summer, winter, coldest transition and hottest transition* due to the distribution of each cluster, this can be observed in figure **??**. Notice that names spring and autumn were not included in the list because when one sees the monthly distribution of the clusters, one can observe that the seasonal period of spring and autumn share some patterns in common. This last fact is explained further on section **??**.

	f = 3	f = 4	f = 5	f = 6	f = 7	f = 9	f = 10	...	f = 18
n = 2	**0.99109447**	0.99153911	0.99178444	**0.99859934**	0.98443497	0.98467069	0.97706896		0.9924231
n = 3	0.98247032	**0.99248157**	**0.99339644**	**0.99963577**	0.9746971	0.97469422	0.969422		0.9974565
n = 4	0.97436977	0.98552845	0.98819893	**0.99889142**	**0.98574802**	**0.98505652**	0.96177504		0.9956667
n = 5	0.97249754	0.98488397	0.98415621	0.99113528	0.98121257	0.9791666	**0.9791666**		**0.9992562**

TABLE 4.5: Average of $p(\mathbb{O}|\mathbb{S})$ for different *GaHMM* models using f features of outdoor temperature and weather temperature variables.

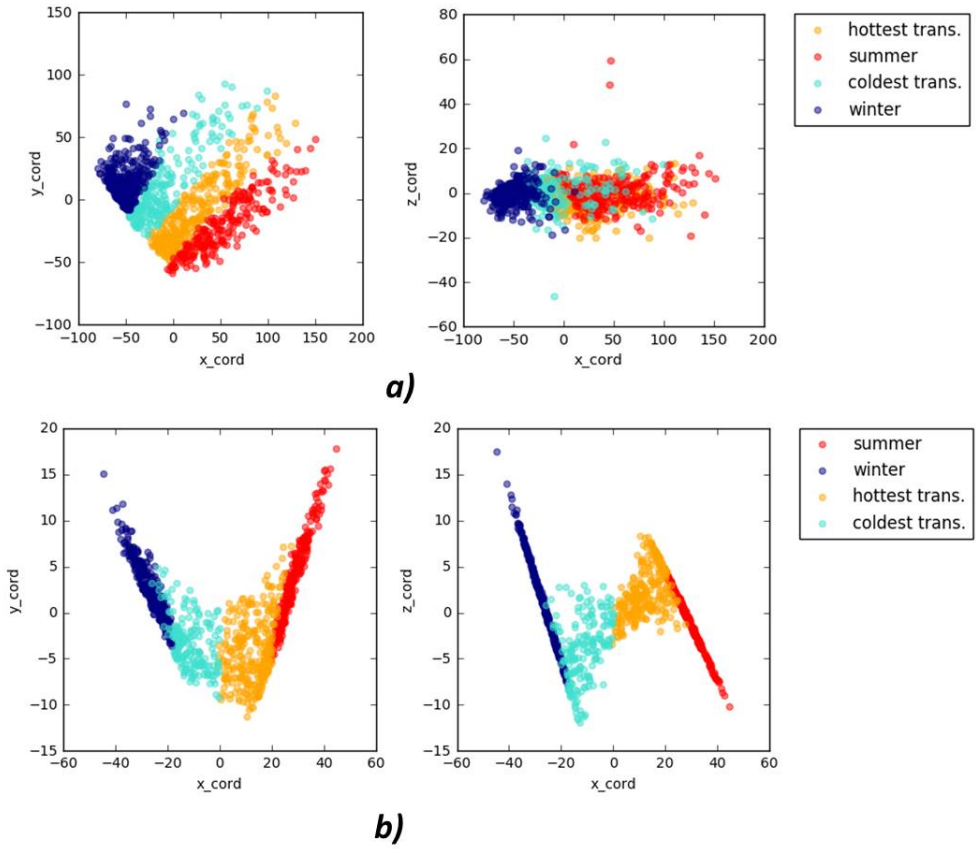


FIGURE 4.25: Dimensional reduction using PCA for visualization: a) Visualization of S_{vector} with 18 features b) Visualization of S_{vector} with 6 features. Colors according to the trained *GaHMM* seasonal models.

One can observe the last fact by using the PCA procedure to transform the S_{vector} to a new coordinate system, in this way, one can appreciate the quality of the clusters and the dispersion of the S_{vector}. Figure **??** shows the difference when **a)** we use a S_{vector} of 18 features according to annex **??** and **b)** when we use a S_{vector} of 6 features that have a good gain of information[29]

[29]The gain of information (i.e. entropy) was calculated and saved in 'feature_selection' collection

according to **??**. Figure **??**.*b)* shows how the 6 features represent a good clustering quality. The results of the best GaHMM seasonal models are explained in section **??**.

4.6.3 Evaluation of the *GaHMM - interactional* model

A *GaHMM - interactional* model is trained according to section **??** [30] and **??**. To evaluate a *GaHMM - interactional* model, we propose the use of log probability of $p(\mathbb{O}|\lambda)$, average of $p(\mathbb{O}|\mathbb{S})$ and a visualization of the R_{vector} in 3D space by using PCA. Table **??** shows the evaluation of the best trained models for different variables in V_1 and the different number of hidden states n according to sections **??** and **??**. Each category is a group of different variables as is shown in annex (**??**, *breakout_group*).

Selected Categories	*n=2*	*n=3*	*n=4*	*n=5*
[A_6_1, A_6_2, A_6_3]	0.988713	0.977466	0.957416	0.960756
[A_3, A_4_1, A_4_2, A_6_1, A_6_2]	0.999950	1.000000	0.999999	0.993509
[A_5_1, A_5_2, A_6_1, A_6_2, A_6_3]	0.988981	0.976075	0.970061	0.977153
[A_3, A_5_1, A_5_2, A_6_1, A_6_2, A_6_3]	0.995121	0.997511	0.972120	0.979629
[A_3, A_4_1, A_4_2, A_5_1, A_5_2, A_6_3]	0.998842	0.997909	0.999975	0.985505
[A_3, A_4_1, A_4_2, A_6_1, A_6_2, A_6_3]	0.999599	0.998505	0.982654	0.984053
[A_3, A_4_1, A_4_2, A_5_1, A_5_2, A_6_1, A_6_2]	0.999978	0.999708	0.998038	0.985093
[A_4_1, A_4_2, A_5_1, A_5_2, A_6_1, A_6_2, A_6_3]	0.992985	0.974719	0.963989	0.991043
[A_3, A_4_1, A_4_2, A_5_1, A_5_2, A_6_1, A_6_2, A_6_3]	0.993445	0.984509	0.974876	0.965273
[A_1, A_2, A_3, A_4_1, A_4_2, A_5_1, A_5_2, A_6_1, A_6_2, A_6_3]	0.999715	0.998432	0.997369	0.987582

TABLE 4.6: $\overline{p}(\mathbb{O}|\mathbb{S})$ for each *GaHMM interactional* model. The code of the selected categories are in annex **??**. These categories belongs to the North-East part of the building.

The selected categories $[A\_3, A\_4\_1, A\_4\_2, A\_6\_1, A\_6\_2]$ (table **??**) is a group of variables that have a relevant linear correlation with the CO_2 levels of the North-East part of the building according to the evaluation in the table **??**. This fact is visible when one applies dimensionality reduction using PCA to project the R_{vector} in a 3D space. Figure **??** shows the PCA transformation where one appreciates the difference when **a)** R_{vector} is defined by categories *[A_1, A_2, A_3, A_4_1, A_4_2, A_5_1, A_5_2, A_6_1, A_6_2]*, and **b)** R_{vector} is defined by categories *[A_3, A_4_1, A_4_2, A_6_1, A_6_2]*. It is remarkable to see a good clustering quality in the case b) where all the three cluster are clearly defined. Clusters: [*Regimen NC, Regimen WC, Regimen PC*] refers to Regimen of Negative Correlation, Weak Correlation and Positive Correlation. These clusters and the corresponding results are described in section **??**. As additional information, we observe the same behavior for the South-West part of the building (i.e. V_2 and categories B_*).

[30] script *4.hmm_learning_k-fold.py*

Code	Variables
A_3	[Blinds angle N Out, Blinds angle N In, Blinds angle E Out, Blinds angle E In]
A_4_1	[Blinds height N Out, Blinds height E In]
A_4_2	[Blinds height N In, Blinds height E Out]
A_6_1	[Temp. Vent. NE Out]
A_6_2	[Temp. Vent. NE In]

TABLE 4.7: Variables according to the category code: *[A_3, A_4_1, A_4_2, A_6_1, A_6_2]*. The complete category code is in annex **??**.

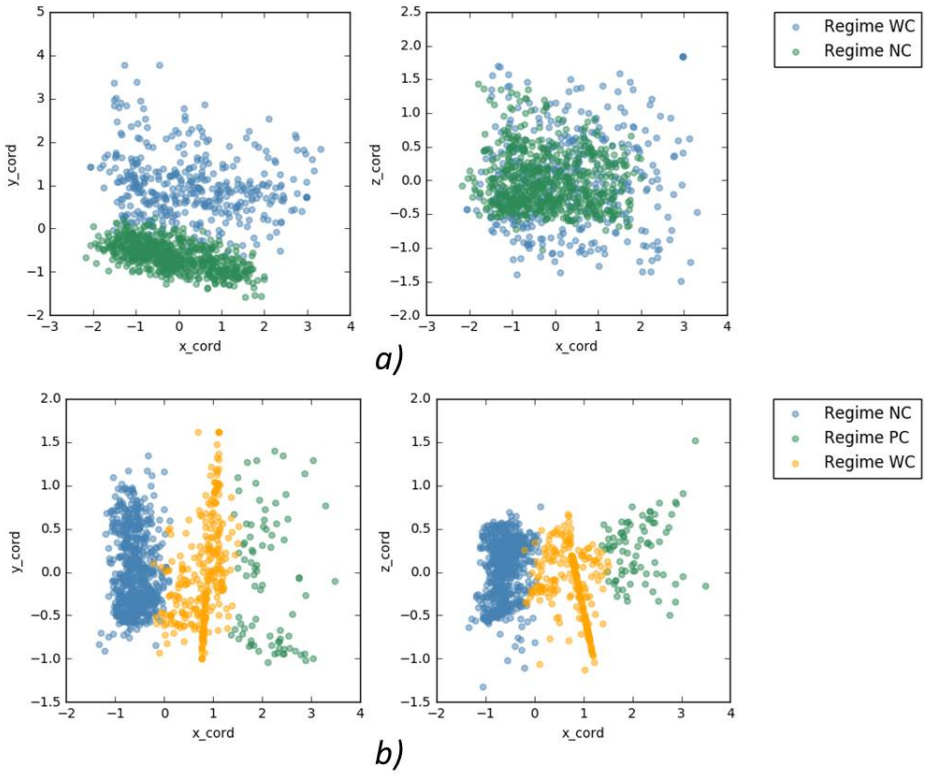


FIGURE 4.26: Dimensional reduction using PCA for visualization: a) Visualization of R_{vector} using categories *[A_1, A_2, A_3, A_4_1, A_4_2, A_5_1, A_5_2, A_6_1, A_6_2]* b) Visualization of R_{vector} using categories *[A_3, A_4_1, A_4_2, A_6_1, A_6_2]*. Colors according to the trained *GaHMM* interactional models.

5 Results

This chapter presents the results of each evaluated model (i.e. *GaHMM profile, seasonal and interactional*). The evaluation information, expossed in section **??**, is used in conjuction with the results of each model in a study case where we compare the Noth-East and South-West ventilation systems. Furthermore, one practical application is proposed for the *GaHMM* profile model. In the penultimate section, we include a comparison of our proposition (i.e. *GaHMM profile model*) against *DayFilter* approach that is considered as one of the state of art for AFDD **kim2017review**, **miller2015automated** At the end of this chapter, we include the feedback from building control system specialists of Synergy BTC AG.

5.1 Results

5.1.1 Results of the GaHMM seasonal model

As it was explained in section **??**, we can evaluate a *GaHMM seasonal* model by using the log probability of $p(\mathbb{O}|\lambda)$, average of $p(\mathbb{O}|\mathbb{S})$ and PCA visualization. Here we indicate how the clusters of *summer, winter, coldest transition and hottest transition* are distributed across years. In addition, we consider it interesting to add to the S_{vector} the *sunshine presence* variable [1]. The final model (including sunshine presence variable) with 6 features achieves an average of $p(\mathbb{O}|\mathbb{S}) = 0.992624$ which is still good, similar to the results presented in table **??**. The clusters are distributed over the whole timeline in this way: *winter* $= 26.9$, *summer* $= 21.0$, *coldest transition* $= 29.1$, *hottest transition* $= 23.0$. Each one has its own particular monthly distribution as one can see in figure **??**. For example, winter is distributed over $[Oct, Nov, Dec, Jan, Feb, Mar]$ in this manner $[2\%, 19.5\%, 22.5\%, 24.5\%, 19.5\%, 8\%, 4\%]$.

Something to note is how the spring and autumn period share some patterns in common, this is the case when the *hottest transition* appears in May or when the *coldest transition* appears in October for instance. Other interesting aspects is how the coldest and hottest transitions remain inside of the winter and summer period respectively. One can also see how these clusters are distributed in a daily fashion in Figure **??**, one observes that each cluster is distributed in an equitable fashion over **a)** working days, weekends and holidays; and **b)** the days of the week in general.

Finally, a calendar visualization is proposed to see the distribution of these clusters. Annex **??** contains a calendar visualization for **a)** S_{vector} using outdoor temperature and weather temperature; and **b)** S_{vector} using outdoor temperature, weather temperature and sunshine presence. The sunshine presence variable affects the clustering such that they become more heterogeneous and blended being in this way a more representative abstraction of the weather nature.

5.1.2 Results of the GaHMM interactional model

As it was exposed in section **??**, three clusters were found by using the best *GaHMM* interactional model. In this model, the R_{vector} represents the existing correlation between categories

[1] Finally, precipitation measures were not added because they present so many nan values that causes a bad monthly distribution.

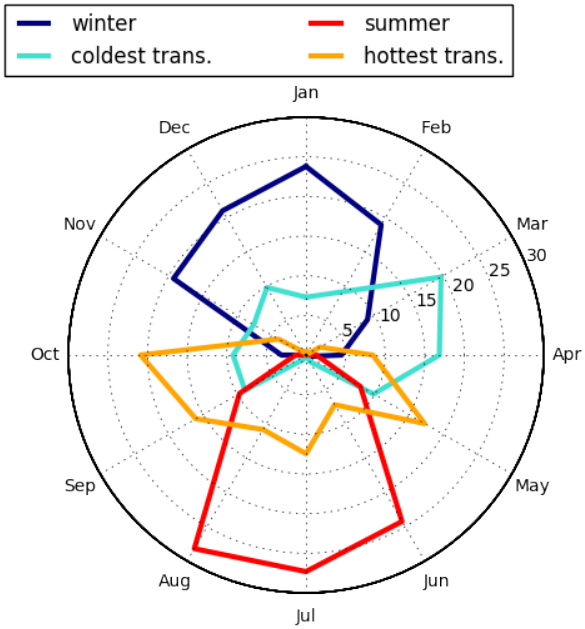


FIGURE 5.1: Monthly distribution for clusters: *summer, winter, hottest and coldest transition*. S_{vector} contains variables outdoor temperature, weather temperature and sunshine presence.

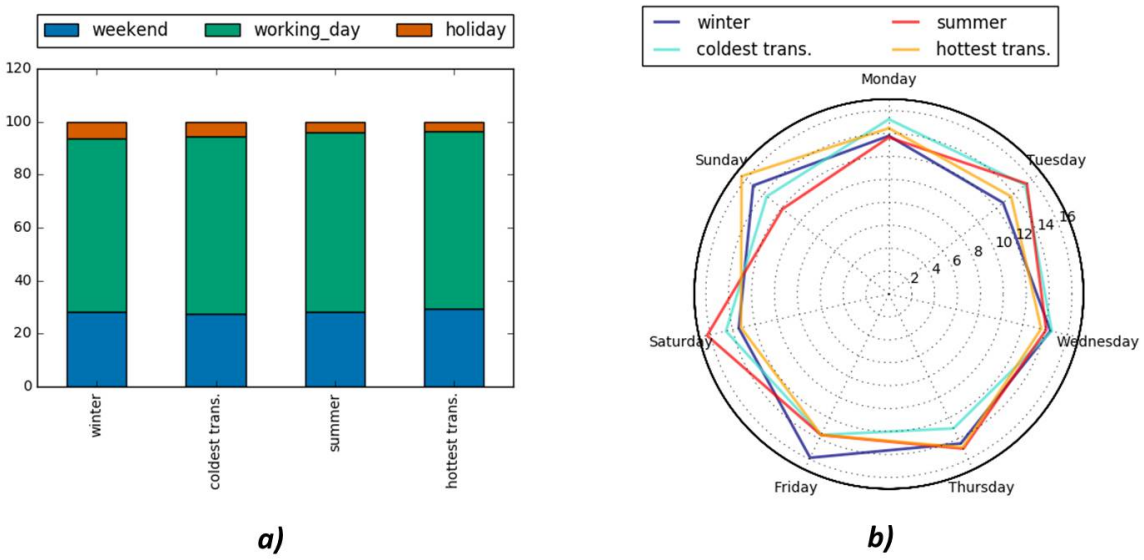


FIGURE 5.2: Monthly distribution for clusters: *summer, winter, hottest and coldest transition*. S_{vector} contains variables outdoor temperature, weather temperature and sunshine presence.

[A_3, A_4_1, A_4_2, A_6_1, A_6_2] and the CO_2 levels of the North-East part of the building. Following this idea, **Regimen of Negative Correlation** contains a collection of days where the negative correlation in the R_{vector} is relevant. **Regimen of Weak Correlation** contains a collection of days where there is a weak linear correlation in the R_{vector}, and finally, **Regimen of Positive Correlation** contains a collection of days where R_{vector} has mostly positive correlation. Each center of the cluster is represented by the mean vector and a standard deviation (see table **??**). For example, one reads that the CO_2 levels has, in average, a correlation of -0.6 with the

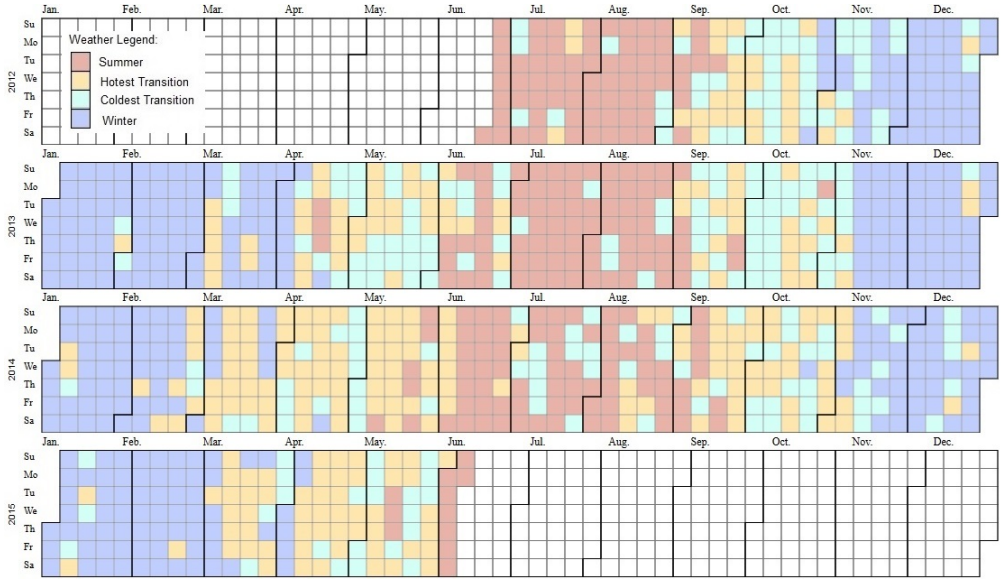


FIGURE 5.3: **a)** Daily representation of clusters: *summer, winter, coldest and hottest transition* using a S_{vector} with outdoor temperature and weather temperature.

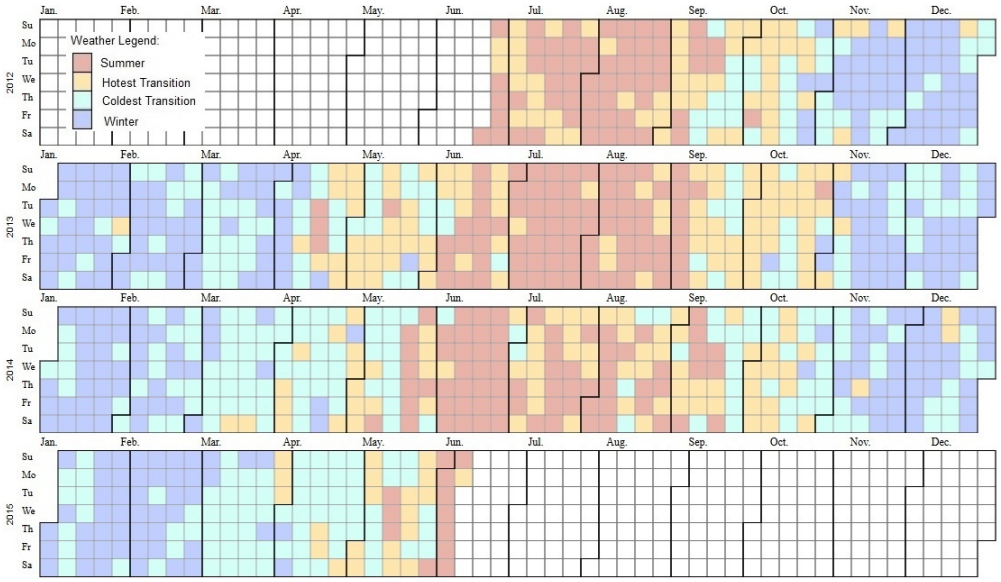


FIGURE 5.4: **b)** Daily representation of clusters: *summer, winter, coldest and hottest transition* using a S_{vector} with outdoor temperature, weather temperature and sunshine presence variables.

angles of the North exterior blinds in Regimen of Negative Correlation.

Clusters of Regimen of Negative, Weak and Positive correlation are distributed over the complete timeline (i.e. 1801 days) in the following respective manner $63.8\%, 28.2\%, 8\%$. Each cluster has a dominant component (see figure **??** and **??**). For example, the regimen of Negative correlation is formed by 631 (93.07%) working days, 26 (3.83%) weekend days and 21 (3.10%) holidays. Clearly, this cluster describes the interrelation of variables when there is occupant presence. In contrast, the regimen of Weak Correlation has a dominant component of weekend days (199 days) where generally there is absence of occupant presence. Thus one can associate a weak correlation with absence of occupants. However, there are cases where the correlation

Regimen of Negative Correlation

CO2 vs.	**Blinds angle N Out**	**Blinds angle N In**	**Blinds angle E Out**	**Blinds angle E In**	**Blinds height N Out**
mean_vector	-0.60	-0.50	-0.50	-0.60	-0.60
std	0.10	0.17	0.14	0.10	0.10
CO2 vs.	**Blinds height N In**	**Blinds height E Out**	**Blinds height E In**	**Temp. Vent. NE Out**	**Temp. Vent. NE In**
mean	-0.60	-0.10	-0.30	0.70	0.10
std	0.10	0.37	0.22	0.20	0.50

Regimen of Positive Correlation

CO2 vs.	**Blinds angle N Out**	**Blinds angle N In**	**Blinds angle E Out**	**Blinds angle E In**	**Blinds height N Out**
mean	0.50	0.50	0.50	0.50	0.50
std	0.20	0.20	0.17	0.20	0.20
CO2 vs.	**Blinds height N In**	**Blinds height E Out**	**Blinds height E In**	**Temp. Vent. NE Out**	**Temp. Vent. NE In**
mean	0.50	0.40	0.50	0.40	0.30
std	0.20	0.22	0.17	0.62	0.67

Regimen of Weak Correlation

CO2 vs.	Blinds angle N Out	Blinds angle N In	Blinds angle E Out	Blinds angle E In	Blinds height N Out
mean	-0.10	0.00	-0.20	-0.10	-0.10
std	0.00	0.00	0.00	0.00	0.00
CO2 vs.	Blinds height N In	Blinds height E Out	Blinds height E In	Temp. Vent. NE Out	Temp. Vent. NE In
mean	-0.10	0.10	-0.10	0.20	0.30
std	0	0	0	0	0

TABLE 5.1: Center vector of clusters: *[Regimen of Negative Correlation, Positive Correlation and Weak correlation]* with their correspondent standard deviation.

is weak, even if there are occupants present [2], this is the yellow area in figure **??**.

We conclude a weak correlation implies absence of occupants but at the same time is an indicative (in the case when there is actual occupant presence) of an atypical interaction of variables, that could involve cases where the blinds are not at all used during the day (totally closed for instance) or the exhaust air temperature stays static during the entire day, or other atypical situations (it could imply sensor faults). Unfortunately, we do not have a ground truth information for these nonconforming days, and therefore we cannot corroborate this hypothesis. Nevertheless when one sees the details of this cluster, one observes time periods where the correlation is weak for business days. One example, is the period of *Friday November* 8^{th}*, 2013* to *Tuesday November* 12^{th}*, 2013* that presents frozen data in all the variables of the dataset (see figure **??**). We suppose it corresponds to a building maintenance period. Finally, the Positive correlation has a dominant component of Saturdays, this correspond to days where the CO_2 levels were accumulated until Friday (CO_2 levels in the range of 800 ppm at midnight), and on Saturday the CO_2 levels diminish during the day (see profile $ID = 26$, annex **??**). All the information in this section is used in the case study in section **??**.

5.1.3 Results of the GaHMM profile model

Each variable has his own *GaHMM profile model* and their results are in the digital folder annex: *iPythonBooks/Diversity of profiles*. To exemplify the results of the *GaHMM profile* model, we propose to use the CO_2 levels of the building. For this purpose, we firstly briefly review topics related with indoor air quality (IAQ), and secondly, a proposition for spotting discord profiles is presented by using the *GaHMM profile* model in combination with the hierarchical agglomerative clustering. Finally, the results and methodology are applied in a study case in section **??**

[2]Occupant presence assumed because the CO_2 levels follows the typical profile $ID = [12, 15, 16]$ annex **??**, for instance.

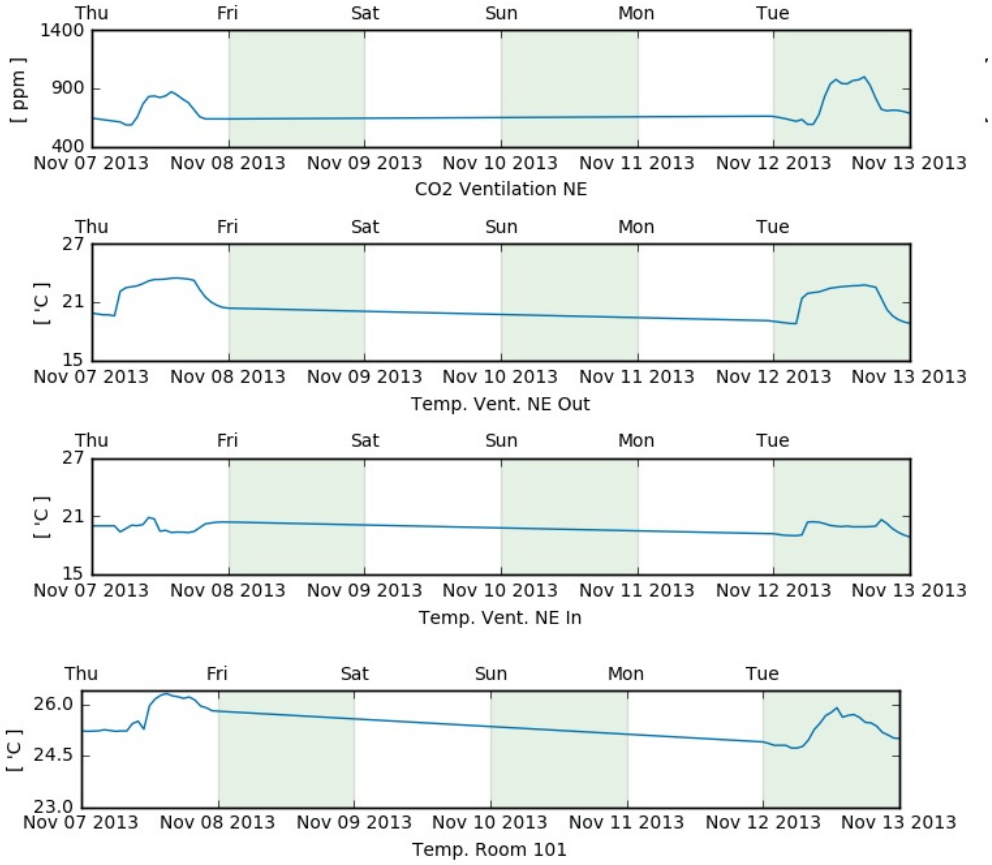
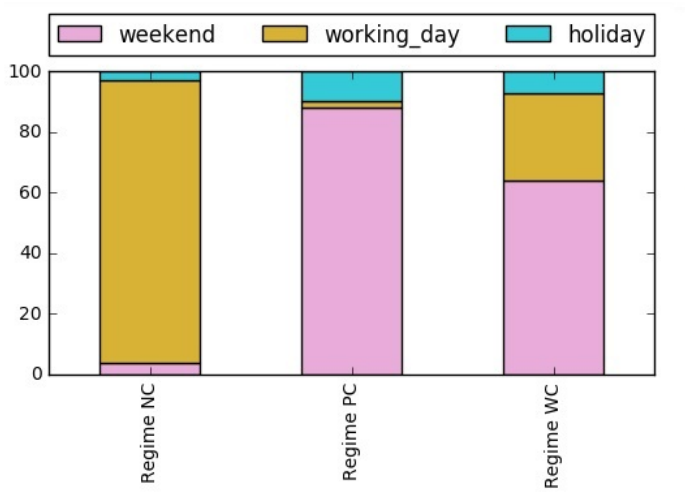


FIGURE 5.5: Discovered building maintenance period: frozen data in all the variables of the dataset.



	working_day	weekend	holiday	
Regime NC	631	26	21	
Regime PC	2	80	9	[days]
Regime WC	91	199	22	

	working_day	weekend	holiday	
Regime NC	93.07	3.83	3.10	
Regime PC	2.20	87.91	9.89	[%]
Regime WC	29.17	63.78	7.05	

FIGURE 5.6: Distribution for clusters: *Regimen of Negative, Weak and Positive correlation* over working days, weekends and holidays.

Analysis of the Carbon Dioxide CO_2 measurement in the Building

One way to evaluate the IAQ of buildings is by using CO_2 sensors in these facilities. This measurement as well as other indicators (e.g. volatile organic component (TVOC), nitrogen dioxide NO_2, etc.) have helped researches to find health problems related to poor air quality for

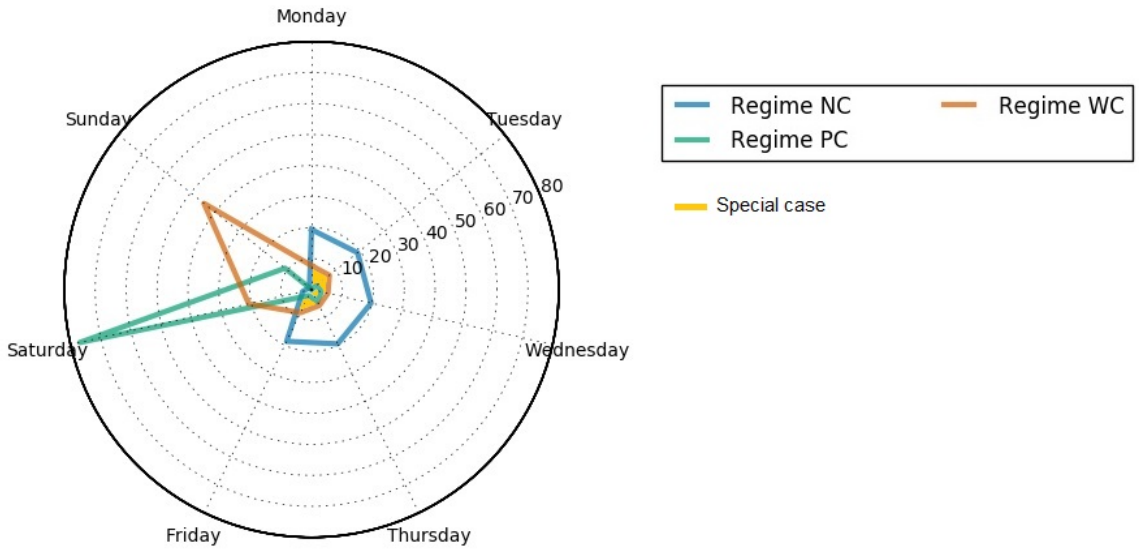


FIGURE 5.7: Distribution for clusters: *Regimen of Negative, Weak and Positive correlation* over working days, weekends and holidays.

at least the two last decades (**persily1997evaluating persily1997evaluating**)**persily1997evaluating** (**lee2000indoor lee2000indoor**)**lee2000indoor** (**erdmann2002indoor erdmann2002indoor**) **erdmann2002indoor** Carbon dioxide is not generally considered to be a health concern at the concentrations that typically occur indoors **persily1997evaluating** however new studies show that constant exposure to poor indoor environments affects health, and that some illness (e.g. allergy and asthma symptoms, and respiratory illnesses) are directly associated to the IAQ **erdmann2002indoor**). Most of the HVAC systems re-circulate the indoor air to maintain the thermal comfort and reduce cost the of operations of cooling and heating (**prill2000measure prill2000measure**)**prill2000measure** This operational strategy and the nature of CO_2 have a cumulative effect on the indoor air quality with time lags (**dong2010informationdong2010information**) **dong2010information**

Understanding the typical pattern of fluctuation of this variable during the day will provide important information to the stakeholders so that designers can create an objective air ventilation system assessment, based on the typical fluctuations of the CO_2 levels. For example, a designer might know whether the indoor contaminants depend only on the occupants or there are other sources of contamination that affect the air quality such as the emissions from building and furnishing material, the intake of outdoor contaminants **persily1997evaluating** the effects of heating/cooling operations, maintenance operations, etc. We believe that the analysis of the "typical pattern of fluctuation" of the CO_2 levels can be done by finding the motif and discord profiles.

Motif and discord profiles for the CO_2 variable: This section discuses the motif and discord profiles for CO_2 measurement of the North-East and South-West zones of the building. The selection of these profiles follows the established criteria of the technical community involved in indoor air quality evaluation and is a benchmarking between the North-East and South-West ventilation system.

Indoor air quality evaluation using Carbon Dioxide (CO_2) measurement It is widely reported by specialists that ASHRAE recommends CO_2 concentrations be below 1800 mg/m^3 (1000 ppm(v)) **persily1997evaluating**, **erdmann2002indoor**, **lee2000indoor** ASHRAE standard 62-2001 **standard2001standard** in his table 6.1 presents the minimum ventilation rates for office buildings, that is 5-15 $cfm/person$ depending on the building zone and the people outdoor air

rate [3]. The relationship between CO_2 concentrations and indoor air quality is directly affected by the rate at which people generate CO_2 and the capacity of the ventilation system to dilute indoor air pollutants **persily1997evaluating** The CO_2 rate generated by occupants depends highly on their physical activity and physical condition; if the ventilation system is not able to dilute CO2 levels slightly over 1000 ppm this is not considered as a health risk but is considered as an uncomfortable situation because of the body odors that occupants can perceive **persily1997evaluating**, **prill2000measure**

The differential between indoor and outdoor levels of $d_{CO_2} = 700ppm(v)$ (i.e. $\approx$ 1000 - 300 ppm) is a measure of acceptability with respect to body odor. We could say, if the differential between indoor and outdoor levels of CO_2 is lower than 700 ppm, 80% of building's visitors will find the odors at an acceptable level, independent of the outdoor levels of CO_2 **persily1997evaluating** Thus, a threshold of approximately $1000\ ppm$ is a limit that helps us to identify unwanted daily profiles. However, what happens with differential CO_2 lower than 700 ppm? Recent studies have indicated that even peaks of concentration below 1000 ppm are associated with an increased prevalence of certain afflictions of the mucous membranes, some respiratory problems, perceptions of stuffiness, discomfort and irritation **persily1997evaluating**, **erdmann2002indoor**

Therefore, we cannot not say that levels lower than 1000 ppm are necessarily healthy, especially if there are CO_2 emanation peaks. One can find in literature sophisticated ways of evaluating air quality, for example by using the CO_2 outdoor levels **persily1997evaluating** or estimating occupancy profiles for checking the correct ventilation rates **batterman2017review** In our case, since we only possess the indoor CO_2 levels, we only consider two types of discords that allow us to check the IAQ: *1.* profiles where the CO2 level are higher than 1000 ppm and *2.* profiles that do not follow the normal pattern of fluctuation of the CO2 variable. The latter is explained in the next section.

Benchmarking between the North-East and South-West ventilation system Here we propose a comparative analysis between the North-East and South-West ventilation system of the building using our approach, that is the *GaHMM-profile* model for the time series of measurement of CO_2 and a hierarchical agglomerative clustering. Our final goal is to find the most common daily profiles (i.e. *motifs*) and potential anomalous daily profiles (i.e. *discords*) that are present in both ventilation systems. To explain how we proceeded, we give outline of the 4 steps to follow with each step afterwards being explained in detail[4]: **a)** once the GaHMM-profile models for the CO_2 variables are trained according to what is explained in the section **??**. We select one GaHMM-profile model at a time and build a matrix of observations M using the cluster profiles of each specific GaHMM-profile model. **b)** We use hierarchical agglomerative clustering algorithms **mullner2011modern** to group our cluster profiles[5] into the observation matrix M. **c)** Once the hierarchical agglomerative clustering is done, we use the cophenetic correlation **saraccli2013comparison** to guarantee that the metrics and methods that were used for the hierarchical clustering were the correct ones and therefore the resulting dendrogram preserves the pairwise distances between the original profiles **saraccli2013comparison d)** After the correct selection of methods and metrics, we propose the selection of the discords and motifs profiles by using a dendrogram and fixing a cut-off value as a threshold.

[3] **People Outdoor Air Rate**: The outdoor airflow rate per person should be provided in the breathing zone to dilute contaminants that are emitted at a rate that is related more to population than to floor area

[4] The script code of this procedure is in files: *iPythonBooks/Diversity of profiles/.. and iPythonBooks/Cases of study/Comparison ventilation System NE vs SW.ipynb*

[5] It is important to recall that our clusters are defined by two vectors: a mean vector and a standard deviation vector of length equal to 24, each value for each hour of the day, therefore, mean vector: $P_x = \{\mu_{x_i} \forall i \in [0, 23]\}$ and standard deviation vector: $STD_x = \{\sigma_{x_i} \forall i \in [0, 23]\}$

Step a) We select a GaHMM-profile that corresponds to the variable of our interest (i.e. *V005_vent01_CO2 model* for North-East and *V022_vent02_CO2 model* for South-West). Afterwards, we compile all the cluster profiles that belong to this GaHMM-profile model (see appendix **??**). For this purpose, as is mentioned in section **??** using the *HMM* library **gahmm_manual** one can access to each profile by using his respective number of identification ID. For example: $model.means\_[0]$ returns the mean vector that correspond to profile $ID = 0$. We list all mean vectors for all clusters, and we construct a matrix M of size $N_p \times 24$ where N_p is the number of cluster profiles that belong to the GaHMM-profile model. This matrix is known as the observation matrix M.

Step b) Once the matrix M is done, we use the clustering package of SciPy [6] for performing the hierarchical agglomerative clustering. We follow the theory and indications provided by **mullner2011modern** and **saraccli2013comparison**'s work **mullner2011modern**, **saraccli2013comparison** The linkage routine [7] is applied over the observation matrix using different methods and metrics as it is suggested by **saraccli2013comparison**'s work **saraccli2013comparison**

Step c) One important aspect in hierarchical agglomerative clustering, is the faithful representation of two or more merged clusters. That is, when we merge two clusters, we would like that the resulting cluster to conserve relevant aspects of the two cluster that were merged **saraccli2013comparison** We can measure this desired effect by using the cophenetic correlation coefficient [8], the closer this measure is to 1, the better preservation of pairwise distance between the original cluster profiles we get.

We cluster the observation matrix M by using the distance metrics: *[euclidean, minkowski, cityblock, sqeuclidean]* and the linkage methods: *[average, single, complete, median, ward, weighted]* **[saraccli2013comparison]** and, we evaluate the quality of the hierarchical clustering by using the cophenetic correlation. The result of this evaluation is in table **??** where we observe that the *average linkage method* and metrics *euclidean distance, minkowski distance* have the best cophenetic correlation [9].

Step d) The resulting dendrogram of the hierarchical clustering can be obtained by using the dendrogram's plot method of the clustering package of SciPy [10]. Figure **??** shows an extract of the resulting dendrogram for the CO_2 cluster profiles for the North-East ventilation system. We observe that the cluster profiles $ID_x = [33, 1, 9, 4, 25, 17, 22, 11, 27]$ are very similar to each other (Figure **??**), however this is not the case for the cluster profile $ID_y = 29$. This fact is remarkable when we observe the similarity metric (i.e. euclidean distance using the average linkage method) between cluster $ID_y = 29$ and any profile of ID_x. Based on this similarity metric, we can conclude that clusters of ID_x do not match with ID_y and therefore they are two different classes of patterns. Looking at the differences in profile $ID_y = 29$, we observe an abnormal fluctuation, that is, that the CO_2 levels are above $800\ ppm$ at the very beginning of the day which is contrary to the normal level $\approx [400 - 650]\ ppm$. Thus, we can conclude that

[6] Hierarchical clustering package can be found on https://docs.scipy.org/doc/scipy-0.14.0/reference/cluster.hierarchy.html#module-scipy.cluster.hierarchy

[7] Linkage Method can be found on https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.cluster.hierarchy.linkage.html#scipy.cluster.hierarchy.linkage

[8] Cophenetic correlation coefficient: is a measure of how faithfully a dendrogram preserves the pairwise distances between the original unmodeled data points **saraccli2013comparison**

[9] Since euclidean distance and minkowski distance achieve the same result, in this work the euclidean distance is used due to his simplicity.

[10] Dendrogram method available on: https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.cluster.hierarchy.dendrogram.html

a) North-East ventialtion system (V005_vent01_CO2)				
linkage methods	***distance metrics***			
	euclidean	***minkowski***	***cityblock***	***sqeuclidean***
average	**0.76324**	0.76324	0.74766	0.73534
single	0.58851	0.58851	0.65695	0.50559
complete	0.55127	0.55127	0.66504	0.50986
median	0.72623	-	-	-
ward	0.65094	-	-	-
weighted	0.73530	0.73530	0.57468	0.59748

b) South-West ventilation system (V022_vent02_CO2)				
linkage methods	***distance metrics***			
	euclidean	***minkowski***	***cityblock***	***sqeuclidean***
average	**0.74394**	0.74394	0.69358	0.72524
single	0.68275	0.68275	0.70526	0.68563
complete	0.59179	0.59179	0.57658	0.55841
median	0.69560	-	-	-
ward	0.60313	-	-	-
weighted	0.60343	0.60343	0.60422	0.71199

TABLE 5.2: The cophenetic correlation values for the hierarchical clustering of: a) CO_2 cluster profiles for the North-East ventilation system b) CO_2 cluster profiles for the South-West ventilation system, using different distance metrics and linkage methods.

clusters of ID_x should be part of the motif clusters and clusters of ID_y should be part of the discord clusters.

Heuristically, we define an euclidean distance of 300 as a cut-off limit in the resulting dendrograms for the two systems (**??**, **??**), this allows to discriminate the discord and motif profiles. The cut-off value creates 11 hierarchical clusters (**??**) in the hierarchical clustering dendrogram of the North-East ventilation system (annex **??**). One can observe that the clusters are joined together in a hierarchical fashion from the closest, that is most similar (i.e. shorter distance), to the furthest apart, that is the most different (i.e. larger distance). Those clusters that join other clusters at further distances over 300 are considered as discord clusters because they differ more than the average of clusters. These discord cluster profiles (i.e. *[30, 3, 13, 18, 20, 10, 31, 29*) can be appreciated in annex **??**. The same cut-off value was applied for the hierarchical clustering dendrogram of the CO_2 clusters of the South-West ventilation system (annex **??**). We found in this case 4 hierarchical clusters (*[B1, B2, B3, B4]*) as is shown in table **??**. However, there is no clusters that join other clusters at distances greater than 300. Looking at all the clusters for the South-West system (**??**) we observe that all of them follow an "uniform" pattern and there is no big difference at first glance. Nevertheless, we decided to spot those cluster profiles that diverge slightly from the rest of the clusters in the South-West ventilation system, so for that we created a new cut-off value of 200 and found clusters **ID=23, 28, 34** (see annex **??**). These clusters represent mostly high CO_2 levels on the winter period but this fluctuation is still close to the normal. One can observe therefore that a threshold between 200 and 300 is a good value for detecting abnormal fluctuations for both ventilation systems.

We showed in this way, how our proposed approach *GaHMM-profile model* in combination with the hierarchical agglomerative clustering allow us to identify potential discord clusters and motif clusters.

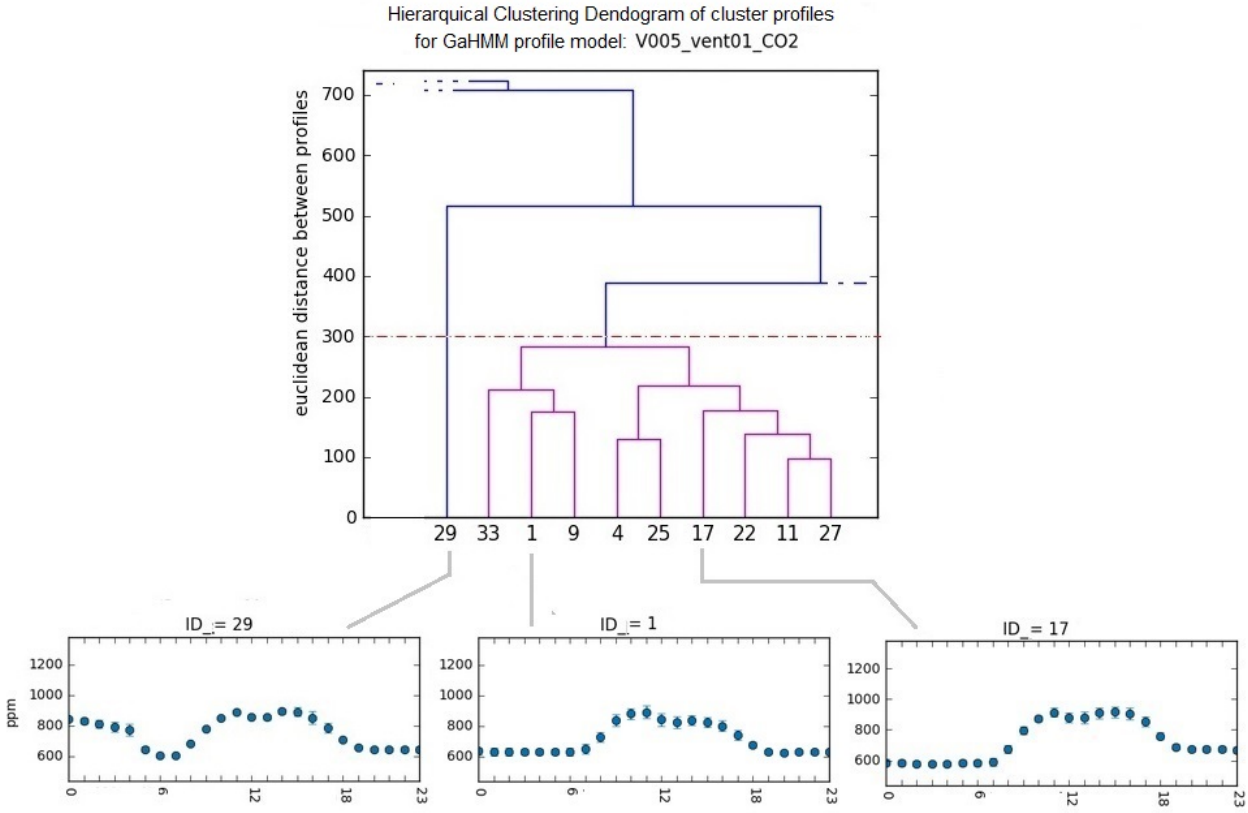


FIGURE 5.8: Extract of the resulting hierarchical clustering dendrogram for the observed matrix M using the CO_2 cluster profiles of the North-East ventilation system.

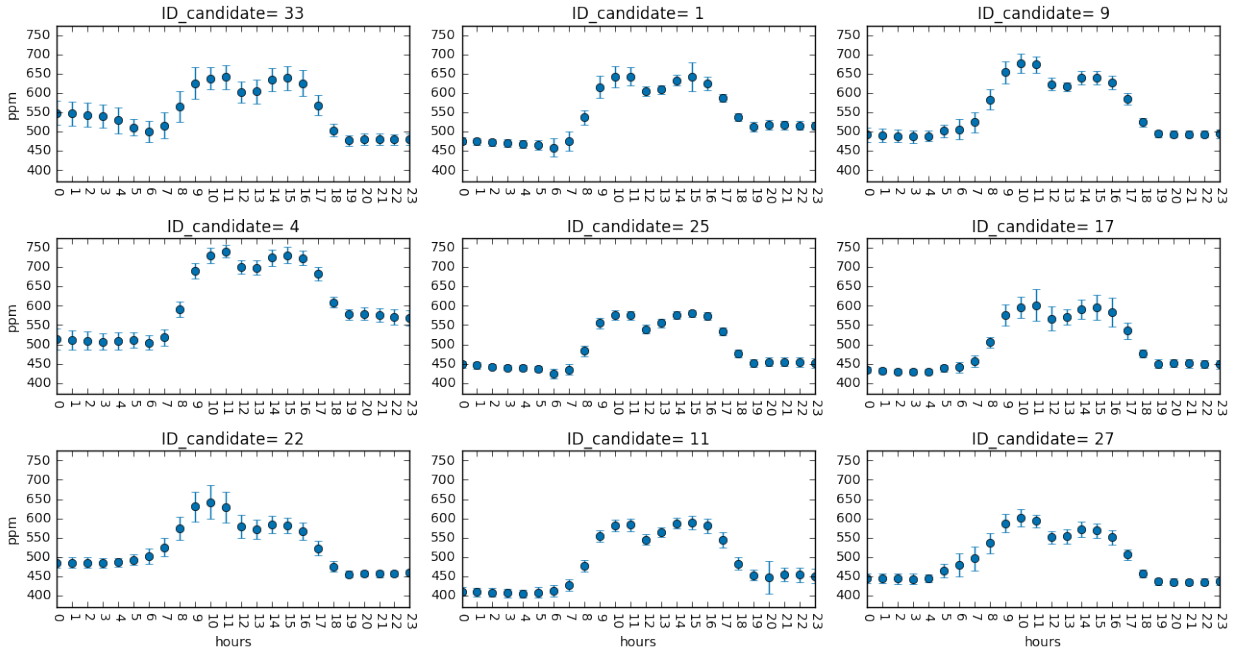


FIGURE 5.9: CO_2 cluster profiles ID_x = *[33, 1, 9, 4, 25, 17, 22, 11, 27]* for the North-East ventilation system.

a) North-East ventilation system

Observation	Hierarchical Cluster ID	V005_vent01_CO2 (CO2 levels) cluster ID
motif	**A1**	[2, 7, 21, 28]
motif	**A2**	[6, 19, 26]
motif	**A3**	[0, 5, 8, 12, 14, 15, 16, 23, 24, 32]
motif	**A4**	[1, 4, 9, 11, 17, 22, 25, 27, 33]
discord	**A5**	[30]
discord	**A6**	[3]
discord	**A7**	[13]
discord	**A8**	[18]
discord	**A9**	[20]
discord	**A10**	[10, 31]
discord	**A11**	[29]

b) South-West ventilation system

Hierarchical cluster ID	V022_vent02_CO2 (CO2 levels) Cluster ID
B1	[4, 13, 28]
B2	[2, 6, 8, 14, 18, 21, 23, 24, 29, 31]
B3	[0, 3, 10, 11, 17, 20, 25, 27, 34]
B4	[1, 5, 7, 9, 12, 15, 16, 19, 22, 26, 30, 32, 33]

TABLE 5.3: Hierarchical clustering for a) CO_2 clusters of the North-East ventilation system b) CO_2 clusters of the South-West ventilation system

Interpretation of the results As we can see on the resulting dendrograms [**??**, **??**] and the hierarchical agglomerative clustering table **??**, the tree dendrogram structure for the CO_2 clusters of the North-East ventilation system is not as homogeneous as the South-West ventilation system. This is because in the North-East ventilation system, there are some particular cluster profiles that behave very differently from the majority of the clusters. These particular clusters join the tree schema at a very latter level, and therefore, can be classified as discord profiles. We conclude that the North-East ventilation system suffered some kind of anomaly, because cluster profiles $ID = 3; 13; 18; 20; 29; 30$ do not appear frequently and they show clearly a weird behavior that is not appropriated, and furthermore, they are not present on the South-West ventilation system. For example the cluster 18 starts the day with 800 ppm which is approximately 250 ppm more than the usual value ($\approx [450 - 650]$ ppm), this CO_2 level and the additional accumulation of CO_2 coming from the occupants provoked an excessive CO_2 level of over 1300 ppm in the afternoon. Another example is the cluster 13 that presents low levels at the beginning of the day, and then, the levels increase in an unexpected way, finishing the day with the highest levels of CO_2 very close to the threshold of $1000\ ppm$. These kind of situations are not present on the South-West ventilation system where the maximum CO_2 level is not higher than 870 ppm. To verify this fact, is enough to check the daily profiles that belong to the clusters $ID = 28; 4; 13$ of the South-West ventilation system (annex **??**) since they have the highest CO_2 levels. Finally, the GaHMM-profile model allows us to spot 61 atypical days for the North-East ventilation system and 17 atypical days for the South-West ventilation system of a total of 1081 days. More details about the detected anomalies for the North-East system are exposed in the case study, in section **??**

5.1.4 Case study: North-East ventilation system

As pointed out in section **??** the North East ventilation system presents some CO_2 anomalous profiles that were clustered as discord profiles (annex **??**). In this section, we analyze in details these profiles and we show their linkage with other associated variables. To show this linkage, in a similar fashion to how we spotted motif and discord clusters for the CO_2 measurements, we use the individual *GaHMM - profile* models and their correspondent hierarchical agglomerative clustering, for spotting discord cluster profiles. The variables used to do this analysis are: *exhaust air temperature, intake air temperature, humidity of the exhausted air of the ventilation system, heating TABS comsuption (KWh) and temperature in rooms*. Each variable

has his own *GaHMM profile model* and the respective hierarchical agglomerative dendrograms. The complete list of cluster profiles of each variable are included as a digital annex in directory: */Thesis_project/iPythonBooks/Diversity of profiles*, and additionally, the hierarchical agglomerative dendrograms for each variable is in annexes **??**, **??**, **??**, **??** and **??**.

We use all our proposed models (i.e. *GaHMM seasonal, interactional and profile models*) to construct a data frame with all the labels produced by each model. Table **??** shows an example of how this data frame looks like[11]. All of this information helps to spot atypical/typical cluster profiles across seasons, interactional regimen and years. One can use different filters according to the research interests. For example, one obtains the figure **??** by using the filter "*winter, regime NC and working days*". This bar plot describes the distribution of cluster profiles using the $ID_{profile}$ of three of the analyzed variables. In the first case, the CO_2 cluster profile $ID = 32$ is the typical profile for working days in winter period, while the cluster profile $ID = 8$ is the typical profile for summer period. Note that cluster profiles that exist in winter period do not necessarily appear in summer.

					GaHMM profile models (ID_profile)				
timestamp	**weekday**	**day_type**	**interaction label**	**season label**	**V005_ vent01_CO2**	**V006_vent01 _temp_out**	**V012_vent01 _temp_in**	**V004_vent01 _hum_out**	**...**
23-Jun-12	Saturday	weekend	Reg. PC	summer	21	35	2	0	...
24-Jun-12	Sunday	weekend	Reg. WC	summer	21	1	2	12	
25-Jun-12	Monday	working_day	Reg. NC	summer	8	11	3	33	
26-Jun-12	Tuesday	working_day	Reg. NC	summer	8	7	3	33	
...									

TABLE 5.4: DataFrame includes labels of interactional, seasonal and profile *GaHMM models*. Each profile model is named using the name of the correspondent variable.

We use dendrograms: **??**, **??**, **??**, **??** and **??** to spot the discord cluster profiles (i.e. the thick blue lines). At the end of the filter process, we discover a temporal coincidence of the discord profiles between variables. This is shown in a calendar visualization in figure **??**.

Each small square represents a discord profile that was found by performing the process in section **??**. One can see a pattern of discord profiles, that is the set of blue violet and green squares, corresponding to the variables: *CO_2 levels, exhaust air temperature and intake air temperature of the ventilation system*. This pattern is a group of discords of different variables that appears all together. In contrast, variables: *humidity of the exhausted air, heating TABS comsuption (KWh) and temperature in rooms* have an occasional temporal coincidence, but this is not so evident as the later case. At least 15 atypical weeks were spotted using this approach. To exemplify one of them, we show the case that appears in period *December 03 to December 13, 2012*. Figure **??** shows the normal trend of the CO_2 levels for the ventilation system two week before the fault,and the next figure **??** shows the moment when the fault occurs. One can see that the CO_2 levels are excessive, reaching levels greater than 1000 ppm. In the bottom part, we include the sequence of ID discord clusters. It is identifiable that the fault sequence starts with clusters *13, 3* and finishes with clusters *18, 10, 26*. We observe that the same phenomena occurs with some variations on November 2012, February 2013, December 2013, and with lesser impact on 2014 and 2015. The annex **??** shows the sequence of discord clusters where one can appreciate the different patterns that occur during the faults.

Figure **??** shows on the left hand side, the normal trend of the analyzed variables, and on the right hand side, the trend of the variables in the fault period. The red line under each variable trend indicates that these daily profiles were spotted as discord cluster profiles. One can observe the temporal coincidence of discord profiles. Table **??** presents an example of the

[11] Labels *t_period_1* and *t_period_2* refers the *hottest transition* and *coldest transition* respectively.

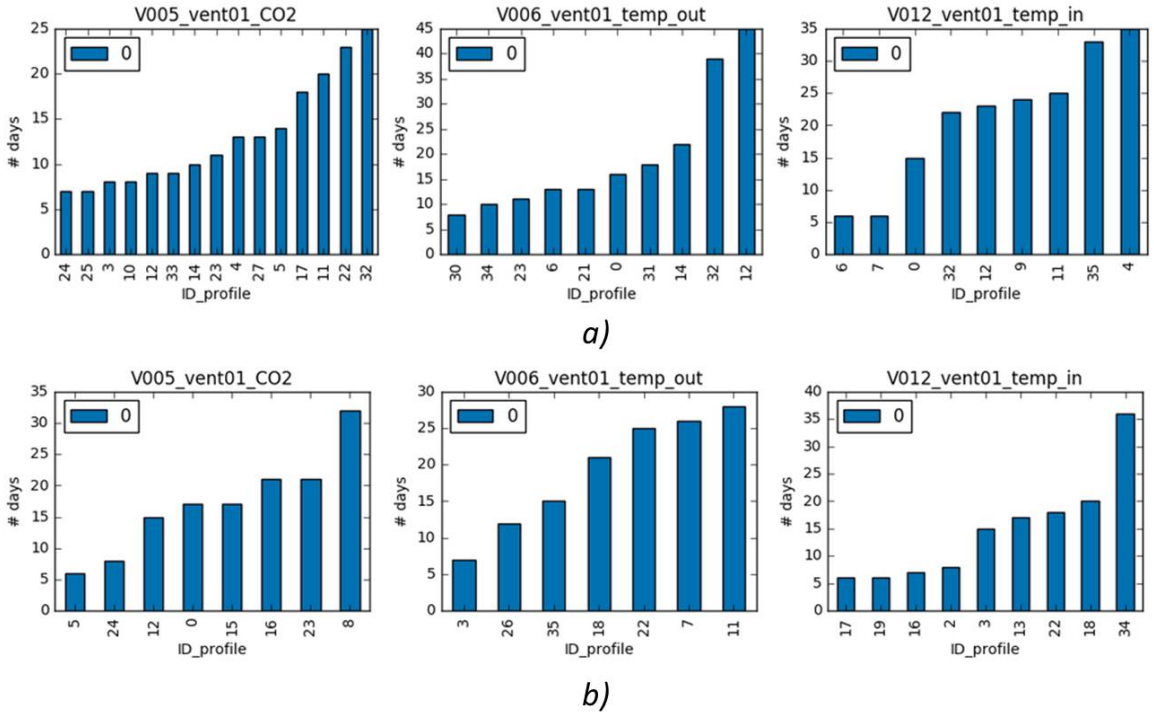


FIGURE 5.10: Distribution of cluster profiles (i.e. $ID_{profile}$) for working days: *a)* winter period, b) summer period.

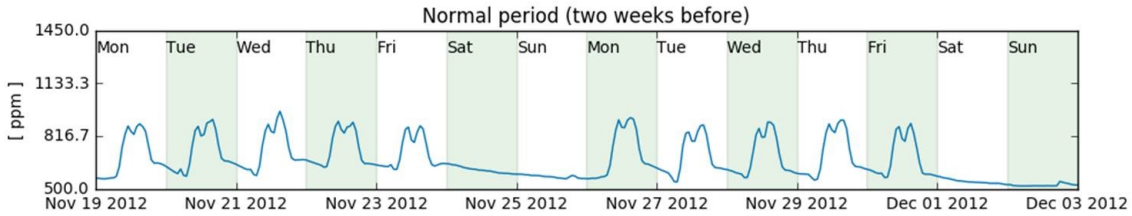


FIGURE 5.11: Levels of CO_2 of the ventilation system during a normal period.

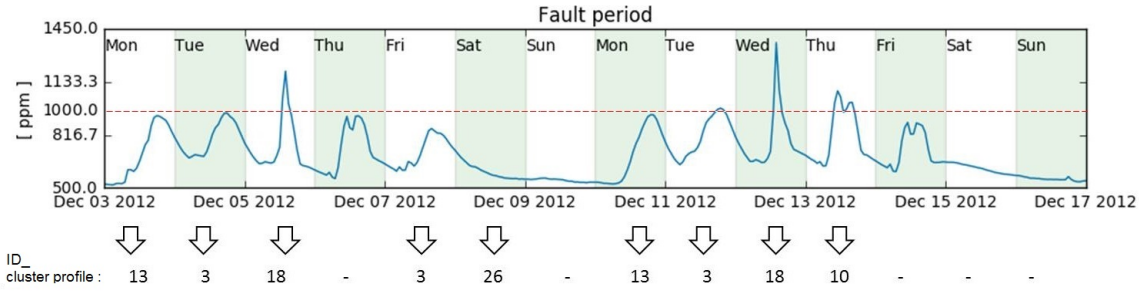


FIGURE 5.12: CO_2 levels in the North-East ventilation system during the fault period. In the bottom part, the ID of the discord clusters shows the sequence of the fault.

label data frame with information from all the models before and during the fault period. The red number in bold implies discord cluster profiles.

Finally, we observe that the fault detailed in figure **??** disappears after the discovered maintenance period (section **??**). We probed this by doing the filtering process in the label data frame

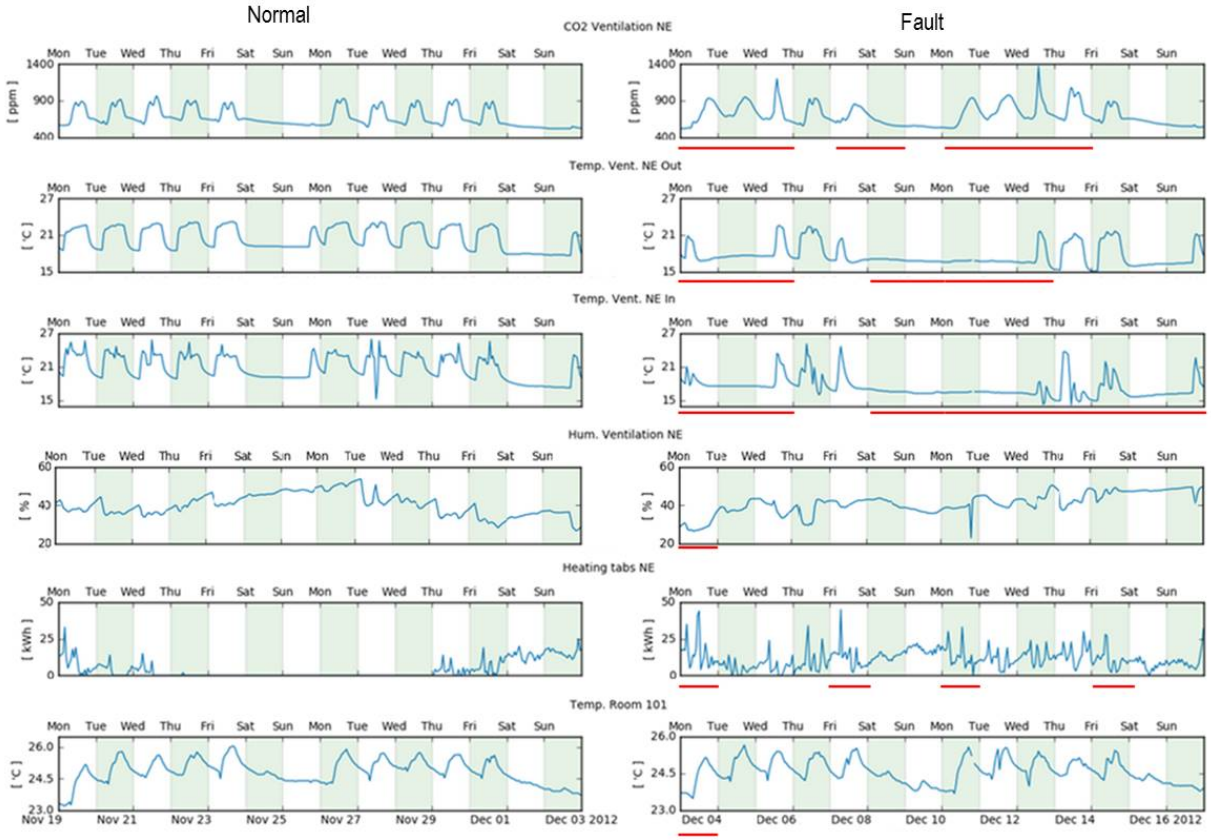


FIGURE 5.13: Trend of analyzed variables in period: December 03 to December 13, 2012.

(i.e. **??**) that the daily profiles had changed after the discovered maintenance period. We probe the last fact by performing the *Welch's t-test* [12] over different days, giving rise to the following results:

- There was a significant difference in the scores for the CO_2 levels before the maintenance period (M=568 ppm, SD=18.24) and after the maintenance period (M=451 ppm, SD=21.25) for Saturdays; t(956)= -30.85, p=0.00001.
- There was a significant difference in the scores for the CO_2 levels before the maintenance period (M=698.9 ppm, SD=81.47) and after the maintenance period (M=672.2 ppm, SD=21.71) for the typical cluster profiles of working days; t(1500)= 5.91, p=0.00001. The typical profiles before the maintenance period are cluster profiles $ID = [32; 22]$ and after the maintenance, cluster profiles $ID = [12, 7]$.

A jupyter notebook is included as a digital annex for this study case in *iPythonBooks/case study*. It includes the rest of periods where the faults appeared, we observe a similar behavior in all the detected faults.

[12] We use the package `https://docs.scipy.org/doc/scipy-0.19.0/reference/generated/scipy.stats.ttest_ind.html`, the digital annex file is: *iPythonBooks/t-test*

timestamp	*Regime*	*Season*	*CO2_ Ventilation*	*Exhaust air Temperature*	*Intake air Temperature*	*Exhaust air Humidity*	*Heating TABS (kWh)*	*Room 101 Temperature*
Mon, Nov 26, 2012	NC	t_period_2	17	24	12	23	1	30
Tue, Nov 27, 2012	NC	winter	22	30	12	19	1	0
Wed, Nov 28, 2012	NC	winter	22	6	12	30	1	7
Thu, Nov 29, 2012	WC	winter	22	31	12	15	5	13
Fri, Nov 30, 2012	WC	winter	5	31	0	34	12	13
Sat, Dec 01, 2012	PC	winter	28	5	24	15	0	3
Sun, Dec 02, 2012	NC	winter	21	20	5	34	16	9
Mon, Dec 03, 2012	**WC**	winter	**13**	**27**	**1**	**5**	**21**	**29**
Tue, Dec 04, 2012	**WC**	winter	**3**	**5**	**24**	15	5	32
Wed, Dec 05, 2012	**WC**	winter	**18**	**17**	26	15	19	13
Thu, Dec 06, 2012	**WC**	winter	17	0	27	15	24	32
Fri, Dec 07, 2012	**WC**	winter	**3**	**27**	27	30	**8**	32
Sat, Dec 08, 2012	**WC**	winter	**26**	**5**	**24**	30	18	3
Sun, Dec 09, 2012	WC	winter	28	**5**	**6**	15	14	23

TABLE 5.5: Label data frame with information from all the GaHMM models.

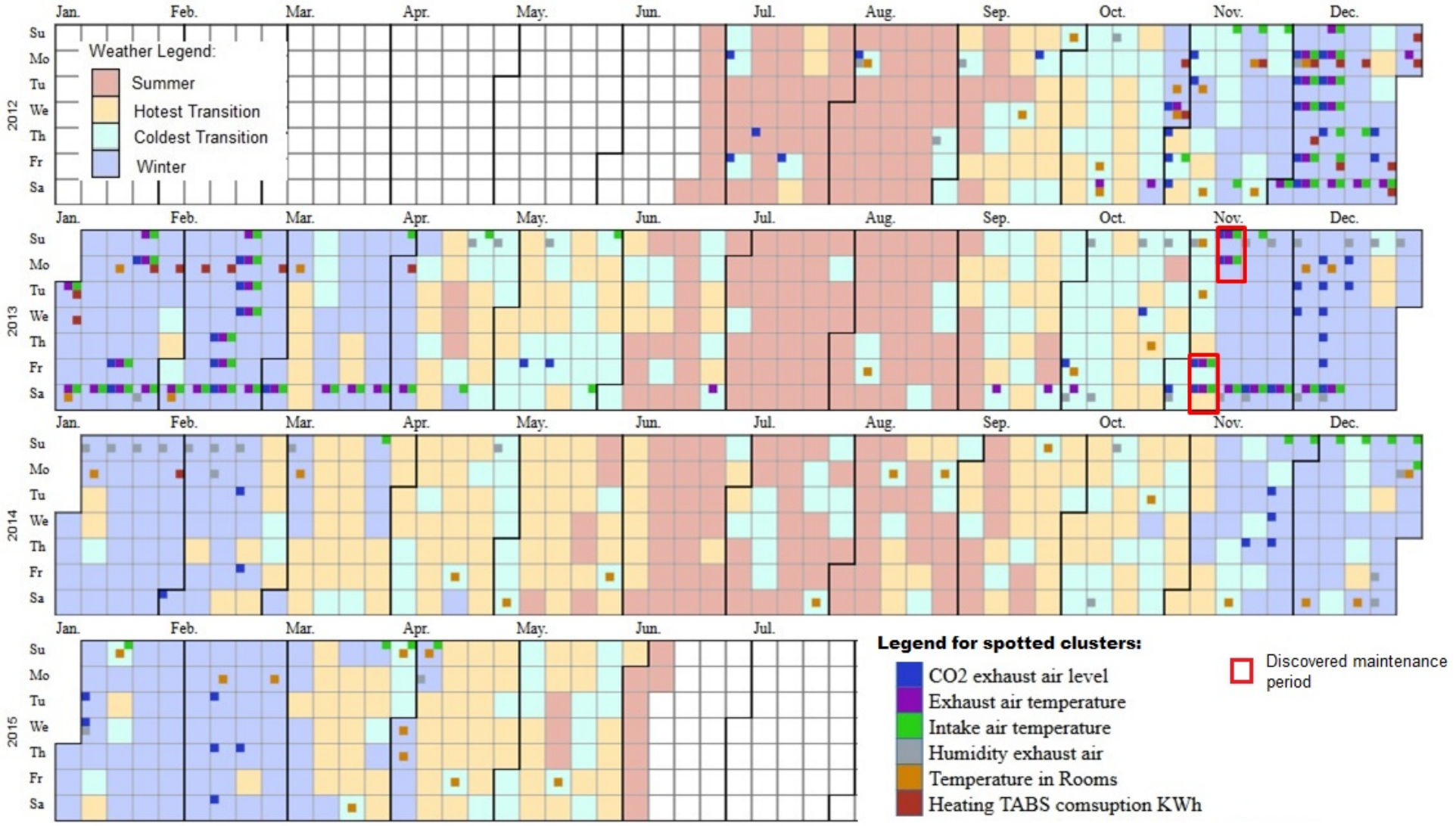


FIGURE 5.14: Case study: Discord cluster profiles of the North-East ventilation system.

5.2 Practical application of the GaHMM-profile model

Real time tracking As it was demonstrate in last section, using our approach *GaHMM-profile model* in combination with a hierarchical agglomerative clustering is an effective tool that assist stakeholders to define a threshold for detecting discord profiles in a system, and defining the motif profiles.

The motif profiles are a good guide to knowing the typical fluctuations of a certain variable, and therefore one can use this information as a reference to spot potential abnormal fluctuations in real time [13]. To exemplify this practical application, we choose the CO_2 level time series and the correspondent motif clusters (see table **??**). We propose a simple routine that checks if the current trend is inside of an expected region [14]. This region can be defined by the collection of the motif cluster profiles where the current trend fits in. Figure **??** explains this concept, the red line is the current measurement of the CO_2 in an hourly fashion. One observes in these three trend graphs how the variable evolves along the day, and how the learned profiles (i.e. motif cluster profiles) provide the shape of the expected fluctuation for the variable (i.e. the green area). One also observe how little by little the area of the expected area becomes refined until the moment, where one or two cluster profile define the shape of the current measurement of the CO_2.

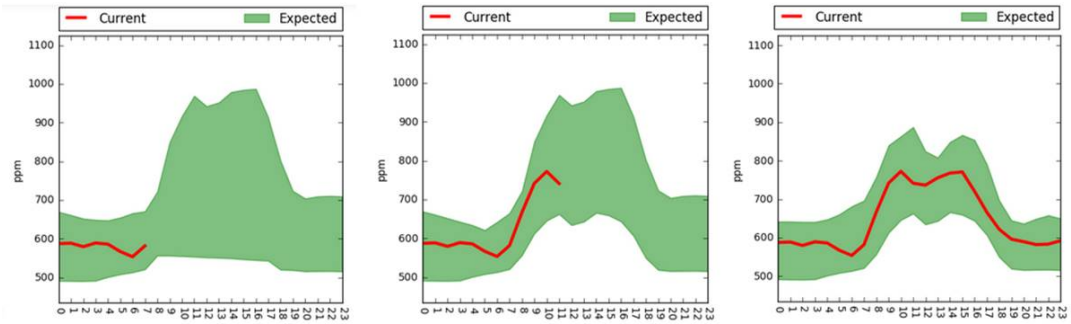


FIGURE 5.15: Tracking the current CO_2 levels in the North-East ventilation system by using the correspondent motif cluster profiles of the *GaHMM profile model: V005_vent01_CO2*.

Figure **??** shows an example where an abnormal trend is detected because it does not fit inside of the motif cluster profiles. One observes how the routine tries to fit the expected area to the current trend, but at the end, one observes a divergence of 45% (i.e. 13 values are inside of the green trend). These kinds of cases could be notified to the stakeholders for making decisions about these fluctuations, if for instance this problem is recurrent or there are suspicions that it is a serious problem.

To know whether or not the current trend fits inside of a motif cluster profile, one uses the definition of a cluster profile, that is, the mean vector and the standard deviation vector. ($P_x = \{\mu_{x_i} \forall i \in [0, 23]\}$ and $STD_x = \{\sigma_{x_i} \forall i \in [0, 23]\}$). The upper and lower bound of a cluster profile are defined as: $U_{bound} = P_x + 1.5 \cdot STD_x$ and $L_{bound} = P_x - 1.5 \cdot STD_x$ [15]. In this way, one routine checks if the current trend is inside of the intervals defined by the upper and lower bound. The collection C of all the motif profiles where the current trend fits in defines the expected area. This expected area is defined as well by an upper and lower bound, that are the maximum and minimum values among all the cluster profile that belong to C. Therefore:

[13] This is our proposition for a practical application of the *GaHMM-profile* model. It requires further research, therefore is proposed as a future work in section **??**

[14] This script is implemented as: *iPythonBooks/Application*

[15] The constant 1.5 is arbitrary, nevertheless we use 1.5 because is the usual value used in Interquartile range analysis. Other values can be used as well.

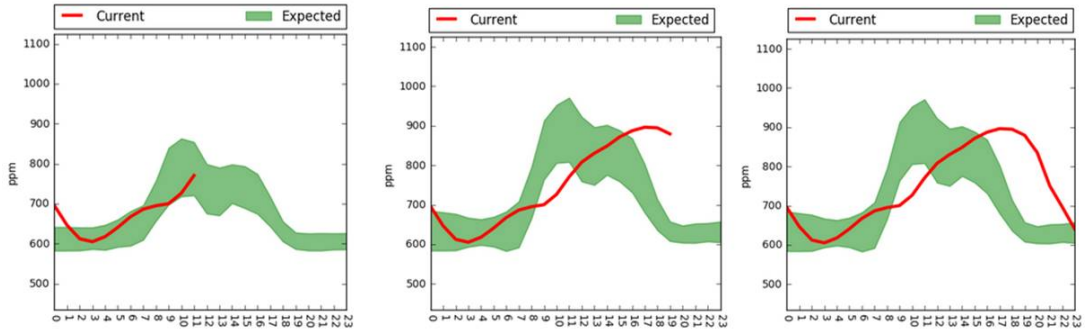


FIGURE 5.16: Tracking the current CO_2 levels in the North-East ventilation system by using the correspondent motif cluster profiles of the *GaHMM profile model: V005_vent01_CO2*.

$$\begin{aligned} \text{Expected}_{area} &= [\text{Lo}_{bound}, \text{Up}_{bound}] \\ \text{Lo}_{bound} &= \min_{i=0,j=0}^{N,T} L_{bound} \in C \\ \text{Up}_{bound} &= \max_{i=0,j=0}^{N,T} U_{bound} \in C \end{aligned} \tag{5.1}$$

Where N is the total number of cluster profiles where the current trend fit in, and T is the length of the current trend. We include a digital annex *iPythonBooks/application* the routine that applies this practical application. We think that definition of the tracking time to use for knowing whether or not the current trend follows the typical pattern of fluctuation, should be defined by a specialist. We believe that this depends on the critical degree of a specific variable. It could be that one variable needs more control than others and therefore the recommend tracking time may need to be shorter. More further studies need to be done using our proposition, but at this time, this application remains as a future work.

5.3 Comparison between DayFilter approach and GaHMM approach

DayFilter approach is presented as a pattern recognition method that uses symbolic aggregate approximation (SAX), motif and discord extraction, and clustering to detect the underlying structure of building performance data **kim2017review** The original paper **miller2015automated** uses building power measurement, but in our opinion, it can extended to any kind of measurements since it uses SAX as the core of the approach. SAX transformation is explained in section **??** for further references. In this section, we explain the procedure that we performed to compare the result of DayFilter and GaHMM approach. Our intuition tell us if there is an anomalous sequence (i.e. discord cluster) that can be spotted by a corresponding SAX word, then we can spot all the sequences that match this word and compare these profiles with the *GaHMM-profile model's* results. This is possible because the daily profile have same shape and similar magnitude for each point in the profile. Following this idea, we choose three variables that are directly associated with the occupants' comfort: (CO_2 level North-East zone, Temperature for room 101, Humidity for room 101). We applied the same process over the three variables. Only the time series of CO_2 level of the North East part of the building was chosen for illustration purposes in this section. Here the steps to follow:

a) Select discord clusters that were found with *GaHMM-profile model* for the variable of interest.

b) Using SAX transformation convert the selected discord cluster profiles in their corresponding SAX words.

c) Spot all the profiles that match with the corresponding SAX word, compare dates were both (GaHMM and SAX) have coincidence and tabulate the results.

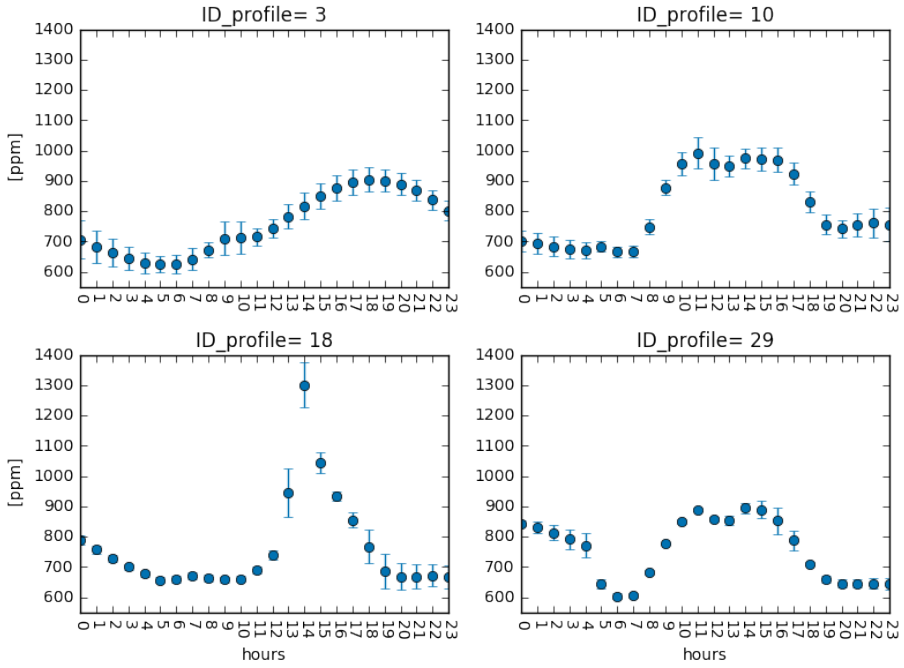


FIGURE 5.17: Example of CO_2 discord profiles for the North-East ventilation system. (4 of 11 profiles)

Step a) After the training process of the *GaHMM-profile model* for a specific variable (i.e. in this case CO_2 time series), we select the discord cluster profiles of interest[16]. The discord clusters profiles are defined by a mean vector $P_x = \{\mu_{x_i} \forall i \in [0, 23]\}$ and a standard deviation vector $STD_x = \{\sigma_{x_i} \forall i \in [0, 23]\}$ of length $N = 24$, each value for each hour of the day. Figure **??** shows examples of discord cluster profiles (*profile identifier = [3,10,18,29]*) and their corresponding values are included in annex **??**.

Step b) The selected profiles are z-score normalized using the general mean μ_T and the general deviation standard σ_T. The values $\mu_T = 649.8\ ppm$ and $\sigma_T = 97.85\ ppm$ were calculated using the entire time series where we remove the extreme points that fall outside of three standard deviations $x_i \notin [\mu - 3\sigma, \mu + 3\sigma]$. It should be noted that this must be calculated in this way to make it comparable to the DayFilter approach **miller2015automated** Additionally, we verify that the normalized data stream $Z(t)$ has an approximate 0 mean and a standard deviation of close to 1. The rest of the process of SAX transformation follows the procedure proposed by DayFilter **lin2003symbolic**, **keogh2005hot**, **lin2007experiencing**, **miller2015automated** We take the non-overlapping sub-sequence of length $N = 24$ that was previously normalized, and we divided into W equal sized segments. Then the corresponding Piecewise Aggregate Approximation is performed **lin2007experiencing** Finally, each mean of the W segments are transformed in alphabetic characters by using the vertical breakpoints

[16] To know why these profiles were chosen as discord clusters, see section **??**

$B = \beta_1, ...\beta_{a-1}$ **lin2007experiencing** The definition of the breakpoint depends on the number of symbols to use for the SAX transformation (more information in section **??**).

To make sure that the SAX transformation is correctly performed, we apply firstly this transformation over some motif clusters, this can be appreciated in Figure **??**. When we did this, we observed how the letters were well distributed for each motif profile, therefore we conclude that the SAX transformation is valid. Figure **??** shows the SAX transformation for each discord profile using parameters $W = 4$, $A = [a, b, c]$ and $B = [-0.43, 0.43]$. The corresponding SAX words for profiles 3, 10, 18, 29 are *'bbcc'*, *'bccc'*, *'cbcb'*, *'cccb'*.

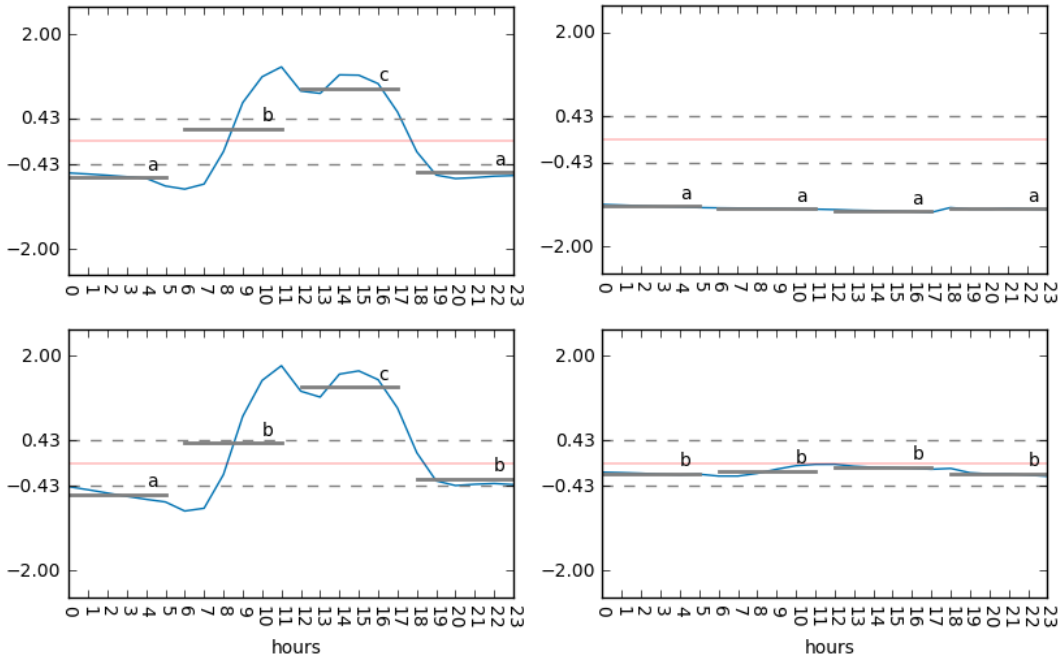


FIGURE 5.18: Example of SAX transformation for **motif clusters** profiles. (4 of 23 profiles).

Step c) To compare the dates where both approaches have coincidence, we perform SAX transformation over the entire time series changing the parameters W and A. Then each daily profile is transformed in his respective word. We list the dates of profiles that match with each SAX word and then we compare against the dates that were found by GaHMM approach. Making $W >= 4$ and $|A| >= 4$ creates so much granularity that the comparison is not fair for SAX approach. In other words, a discord cluster profile ($P_x = \{\mu_{x_i} \forall i \in [0, 23]\}$ and $STD_x = \{\sigma_{x_i} \forall i \in [0, 23]\}$) can be broken down into so many combination of letters, so that the coincidence is lower. Having so much granularity makes difficult to define the set of words that belongs to this discord cluster. On the contrary, if we set $W < 4$ and $|A| < 4$ there is a risk to spot fake discord profiles that do not actually belong to the discord cluster, but were clustered by SAX due to the space between breakpoints. Table **??** shows the results of this comparison.

We observe that both approaches have a maximum percentage of coincidence [17] when $W = 4, |A| = 5$. However, we consider more interesting the case when $W = 4, |A| = 3$ because there is a maximum number of coinciding days, but DayFilter approach spots more discord

[17] When both approaches coincide on the same date, we define percentage of coincidence as:

$$\%\ coincidence = \frac{\#\ coincidences}{\#\ GaHMM\ discords + \#\ FilterDay\ discords - \#\ coincidences} \cdot 100$$

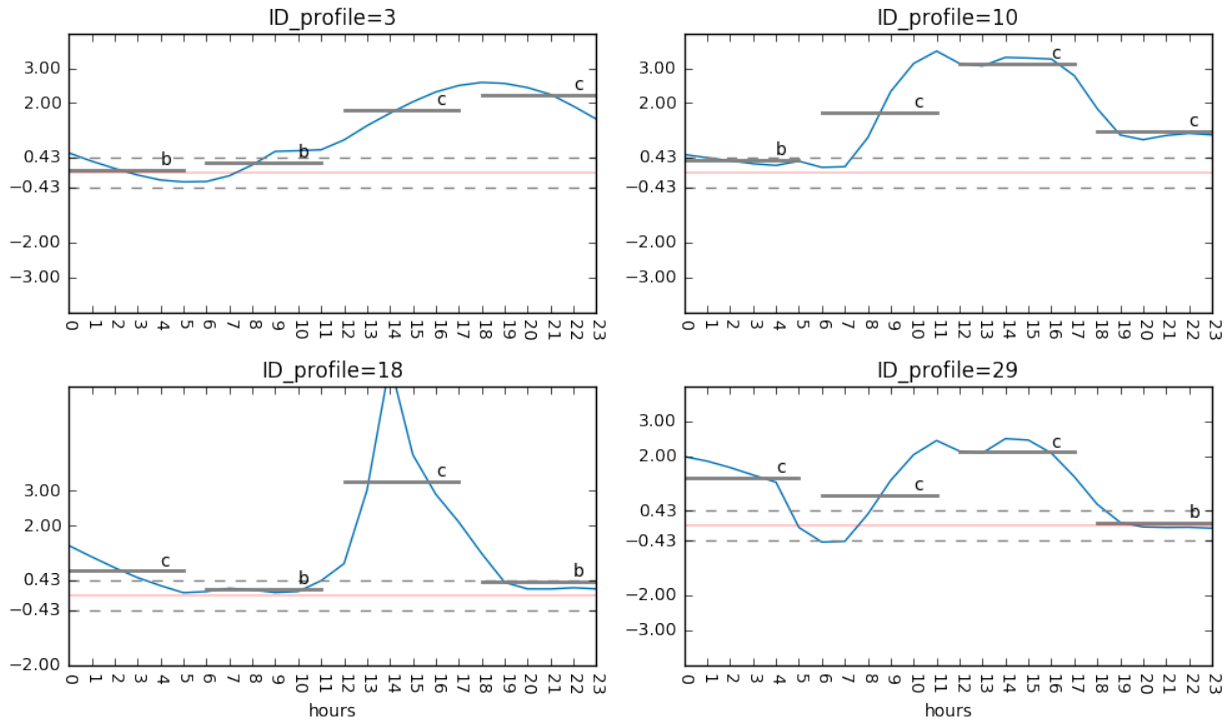


FIGURE 5.19: Example of SAX transformation for **discord cluster profiles**. (4 of 11 profiles).

profiles than *GaHMM-profile model*. Looking deeper at the difference between both approaches we found that FilterDay approach spots some fakes profiles. Figure **??** shows examples of fake profiles when we use SAX for spotting profiles corresponding to the word 'bccc' (profile 10) with $W = 4, A = \{a, b, c\}$. Observe how SAX is unable to define the changes that exists in profile 10 from 8h to 23h (i.e. Figure **??**) therefore the profile has a wide standard deviation and the $CO2$ level at hours 0-7 are lower than we expect for profile 10. We include more details about the dates when both approaches are coincident in annex **??**.

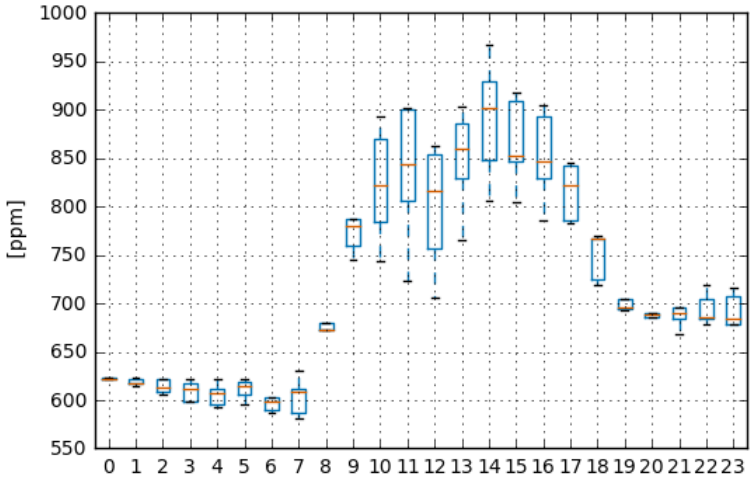


FIGURE 5.20: Example of fake profiles for cluster 10 ('bccc'), when FilterDay approach is applied with $W = 4, |A| = 3$.

	W = 4, |A| = 3	W = 4, |A| = 4	W = 4, |A| = 5	W = 4, |A| = 6	
GaHMM	61	61	61	61	days
DayFilter	311	84	29	36	days
Coincidence	49	39	25	21	days
% coincidence	15.2	36.8	38.5	27.6	%

	W = 6, |A| = 3	W = 6, |A| = 4	W = 6, |A| = 5	W = 6, |A| = 6	
GaHMM	61	61	61	61	days
DayFilter	49	33	19	16	days
Coincidence	29	25	17	15	days
% coincidence	35.8	36.2	27.0	24.2	%

TABLE 5.6: Number of days where the discord profiles were spotted by using *GaHMM-profile model* and DayFilter approach. The corresponding coincidence of both approaches is done when both have the same date.

Discussion about the experiment

In this experiment, the over/underestimation of some discord profiles for the DayFilter approach is due to two aspects: the assumption that a z-score normalized time series has a Gaussian distribution **lin2003symbolic**, **keogh2005hot**, **lin2007experiencing**, **miller2015automated** and the wide space between the breakpoints. (**li2015encyclopedia li2015encyclopedia**) **li2015encyclopedia** point out that a z-score normalization does not guarantee a Gaussian distribution, even if the mean of the normalized time series is aproximatly near to zero and the standard deviation close to one. Therefore, when SAX transformation is performed, some segments of the time series are not properly represented by the corresponding symbols. To overcome this problem, we could normalized each sample of size N as the original version of SAX suggests **keogh2005hot** However, when the latter is applied, the Piecewise Aggregate Approximation modify the original distribution of data, resulting in a shrinking standard deviation that is proportional to the number of segments that are used to define the PAA series (**butler2015sax butler2015sax**) **butler2015sax** In short, the SAX approach does not necessarily guarantee an equiprobable distribution of symbols along the time series, and therefore the underlying sub-sequences are not represented correctly giving as a result an overestimation/ spotting-lack of discord profile.

In fact, the mentioned behavior can be observed over the CO2 exhaust air time series of the building. The z-score normalization was performed over the entire time series. We can check for this time series that its mean is approximate to zero and its standard deviation close to one **miller2015automated**, **lin2007experiencing** however its distribution is not Gaussian as we observe in Figure **??**. We observe a remarkable right tail corresponding to the highest levels of CO2 in the building.

Since the underlying distribution of the time series is not a Gaussian distribution, it could be an error to use breakpoints that are defined for a Gaussian distribution. In other words, instead of using the cut-off points (i.e. breakpoints) that are specified in the original SAX approach **lin2003symbolic**, **keogh2005hot**, **lin2007experiencing** we suggest estimate the Probability Density Function (pdf) of the time series, and then, calculating the breakpoints that divide this function in similar areas. Figure **??** shows the pdf of the time series with the corresponding breakpoints.

We think that if DayFilter approach uses customized breakpoints that are fixed according to the actual distribution of the time series, then the results would improve significantly. This can be a motivation for new research. Finally, the results for variables: Temperature for room 101 and Humidity for room 101 are included in Annex **??**. We observe almost the same results for the rest of the variables.

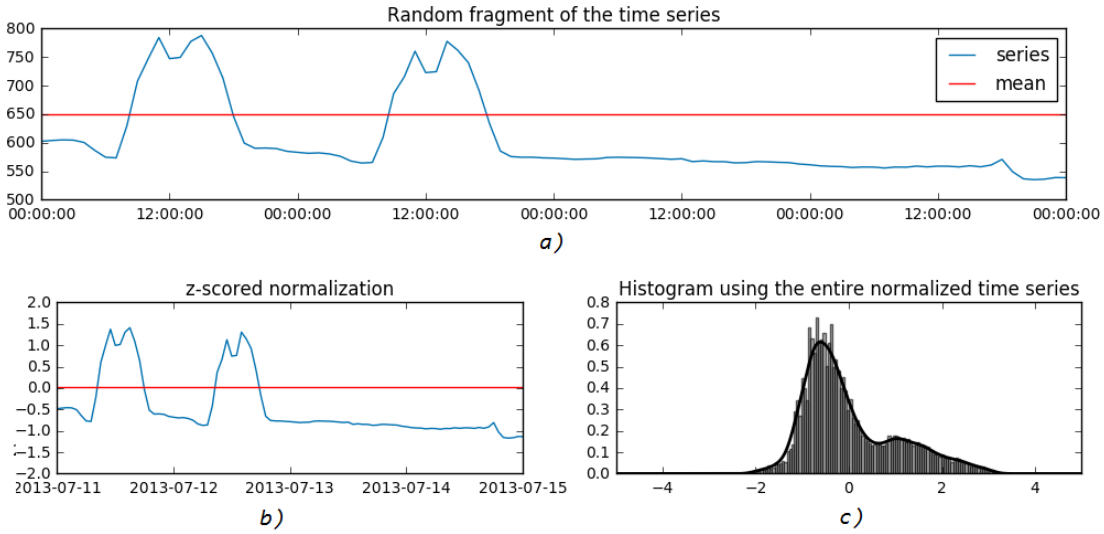


FIGURE 5.21: *a)* Extract of the time series. *b)* Its corresponding z-score normalization. *c)* Histogram of the entire z-score normalized time series.

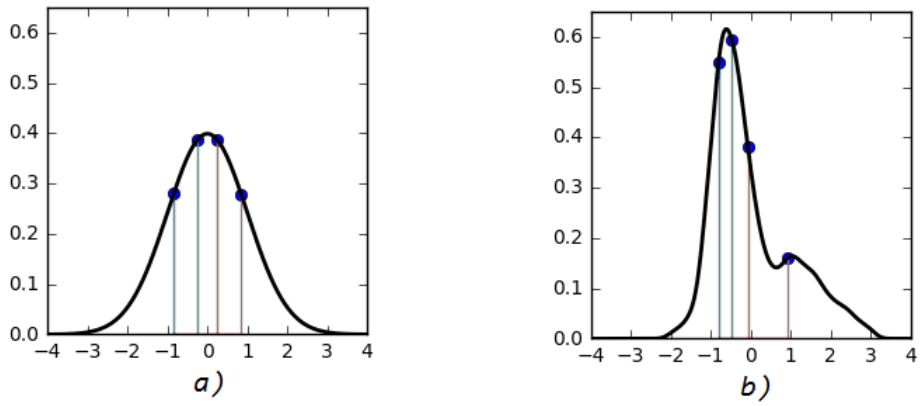


FIGURE 5.22: *a)* Gaussian distribution divided in 5 equitable areas using the corresponding breakpoints $\beta = [-0.84, -0.25, 0.25, 0.84]$ for $A = a, b, c, d, e$. *b)* Estimated pdf of the CO_2 level time series divided in 5 equitable areas using the corresponding breakpoints $\beta = [-0.81, -0.47, -0.07, 0.94]$ for $A = a, b, c, d, e$.

5.4 Comments from building specialists

The presented thesis was exposed to building control system specialists of Synergy BTC AG[18]. Several topics were discussed in reference to the proposed models. Here we list the most important comments:

- The approach is very interesting, in fact the combination of the *GaHMM profile* with the hierarchical agglomerative clustering provides in a glance the difference between motif profiles and discord profiles. Therefore, the proposed approach assist very well to designers in spotting potential anomalies in the building. There was an agreement in the sense that this approach provides a direct feedback to the stakeholders.

[18]Dimitrios Gyalistras and Carina Sagerschnig

- Regarding the practical application of atypical profiles detection that was explained in section **??**. Experts propose for a future study, the evaluation of the detection of atypical profiles using the *GaHMM profile* and HAC approach. For this purpose the ground truth of the process is needed. At the end of this evaluation, the experts expect important indicators like the false positives and false negative rates, and a metric indicator for reliability.

- Experts agree with our conclusions on the clustering quality of our proposed method on finding cluster profiles. When the experts looked at the motif and discord profiles, they asked the question of how much it is going to cost to achieve a 'perfect' performance of the building (i.e. going from discord profiles to motif profiles)? It could be that the price of this optimization task is very expensive and could be difficult to justify.

- Other topics discussed with the experts were: What if the proposed approach is used with critical variables? Variables that must be directly controlled with controller devices (bottom part of the building control). It seems that there is great potential to exploit our proposed approach in this domain.

- Money is an important aspect to consider. One possible application which the experts foresee as having a good chance of return on investment, is applying the proposed approach for selling energy to a grid by using renewable energies. Control devices could detect the ideal times to sell energy and thus benefit from this opportunity.

- Experts believe if the user awareness increases about the dynamics of building, people would be empowered to demand comfort enhancement, lower energy consumption or any other aspect of interest to the user. Our approach could assist the user in this purpose. The experts recommend that if an application is developed for this purpose, the interaction must be easy and fast (only a couple of clicks) due to the lack of interest that could be presented by the occupant after initial use.

- Since we show that the profile clusters change across seasons and years, experts recommend rerunning the training process at least every four months. This will guarantee an update on all the daily profiles. Additionally, they recommended a yearly evaluation of the patterns to see if there are abrupt changes in the patterns in the building.

During the discussion period, the experts shared their personal experiences where they pointed out that our approach might be interesting in the distant future, because currently, the building industry is more focused on practical applications, where the cost/benefit return is evident. Additionally, there is not much interest in optimizing the performance of buildings already constructed since it increases the implementation price of the building. Nevertheless, they do not discredit the use of the proposed approach in other domains where the fine control is important (i.e. building control system). For more practical purposes, our approach can be used over variables that belong to the bottom part process of the building (i.e. building control process) rather than the output variables (i.e. CO_2, room temperature, and others). In other aspects, from the point of view of the experts, the *GaHMM profile* model might be more powerful and robust if the model considered multivariate samples similar as it was done for *GaHMM seasonal and interactional* model. In this way the information of other variables (e.g. status of control devices $1/0$) makes the model more integral and robust. In short, we received a positive feedback from the experts, and their guidance is relevant for future work.

6 Conclusions and Future Work

The presented master thesis draws two types of conclusions, one related to the actual performance of the building and one related to the data mining techniques. Lastly in this chapter, propositions for future work take into consideration the feedback from Synergy BTC AG's specialists.

6.1 Conclusions

6.1.1 Building performance

The proposed learning models and the correspondent visualizations contributed to the gain of information about the interactions between occupants and the studied building. The proposed approach aims to be applicable for other multivariate building datasets as well. For our studied building, we can conclude for each of the proposed model, the following:

The proposed *GaHMM profile* model is effective to discover the daily profiles of the studied building dataset. The best trained models allows to perform the time-depending clustering of similar daily profiles. In average, from 25 to 40 cluster profiles were needed to summarize the information of the three years of time series for each variable in the dataset. Therefore more than 1200 daily profiles can be found for the whole dataset. For each particular variable of the dataset, each cluster profile was well defined by the mean vector and the corresponding standard deviation for each hour. This level of detail, in combination with the hierarchical agglomerative clustering represent a powerful tool for discovering the motif and discord cluster profiles of any variable of interest. The interviewed experts only needed a quick glance to the dendrogram to differentiate the motif and discord profiles of a selected variable. They agreed that the presented approach provides a direct feedback about the gap that exists between the desired performance and the actual performance of a building.

The proposed *GaHMM seasonal* model is an innovative aid to understand how seasons affect the daily profiles of a building. Changes of patterns of daily profiles in buildings are inevitable when seasons change across the years. The seasonal labels obtained from *GaHMM seasonal* model assists the filtering process according to the interest of the researcher. Using the seasonal labels in conjunction with the *GaHMM profile* model helped us to discover the way patterns change across seasons and years. Furthermore, using the seasonal label we observed that is possible, for instance, to identify motif and discord profiles for a selected season.

The proposed *GaHMM interactional* model was a robust model able to explain the typical correlations between variables in the studied multivariate dataset. This model helped to understand how a set of variables keep on a negative linear correlation with the CO_2 levels, when OcP is likely to occur (assuming that CO_2 levels were produced by the occupants). Furthermore, irregular situations could be spotted in periods where the negative correlation between variables does not match with the typical CO_2 profiles. These situations could be associated to sensor faults, irregular position of the sun blinds, irregular temperatures among others issues. Unfortunately, there is no ground truth for approving this hypothesis. Nevertheless, we

believe that the *GaHMM interactional* model could assist in the detection of OcP.

The potential of all the GaHMM models was tested in a case study presented in section **??** where the North-East and South-West ventilation system were compared. The label data frame assisted in the filtering process and a discovered period maintenance of the building was found by using the labels produced by *GaHMM interactional* model. Anomalous daily profiles were found in the North ventilation system before and during the maintenance period. After the maintenance period, there were no new occurrences of anomalous daily profiles. It was hence also proved that after the maintenance period, the patterns of the daily profiles were modified in significant ways for the CO_2 level variable.

6.1.2 Data mining techniques

Using a Big-Data base approach proved to be a helpful mechanism for retrieving information from several collections in different formats/structures. This was particularly advantageous during the running of the proposed algorithm, and in the saving and visualization of partial results but also throughout the entire knowledge discovery process.

By comparing SAX and GaHMM-profile model, we observed that our proposition is more precise in forming the cluster profiles of the variables. This was seen in the evaluation of the clustering quality where we observed that each of the daily profiles fits very well in the shape of the correspondent clustering profile. This is beneficial at the time of doing the discrimination between motif and discord cluster profiles. The use of the hierarchical dendrogram facilitates the definition of the manual threshold that discriminates both motif and discord profiles. We believe that this technique is very powerful and that its use could be extended to another domains.

Following to our experiences during this project, we have realized that due to the complexity of the multivariate dataset, developing information visualization mechanisms that communicate the internal dynamics of the building is a very hard task. However, we also observed that our approach is a powerful tool to summarize information of a time series. The discovered cluster profiles can be used for different purposes, and they could serve to implement new and meaningfully mechanisms for visualizing the dynamics of a building. The use of the label data frame could be a good ally in representing information about a building, since they summarize the fluctuation of the variables in cluster labels, thus the researcher can filter the information according his interests and needs. We believe that our proposed approach can be useful for powerful visualizations that assist the stakeholders get an effective grasp of the internal dynamics of a building.

6.2 Future work

Regarding knowledge discovery about building dynamics, our approach can be useful concerning information visualization systems helping to increase the knowledge of the actual performance of the building. New data visualizations can be proposed by using the label data frame exposed in section **??**. For example, we propose in figure **??**, a possible visualization where the stakeholders can navigate and find the discord profiles of interest. A discord profile can be selected by doing a click over the small fault squares; the user can then compare the discord profile against the expected profile of that day. The expected profile area can be calculated by applying our proposition that was explained in section **??**.

We consider that the feedback coming from the building expert in section **??** are important for future work in this domain, we summarize these points here:

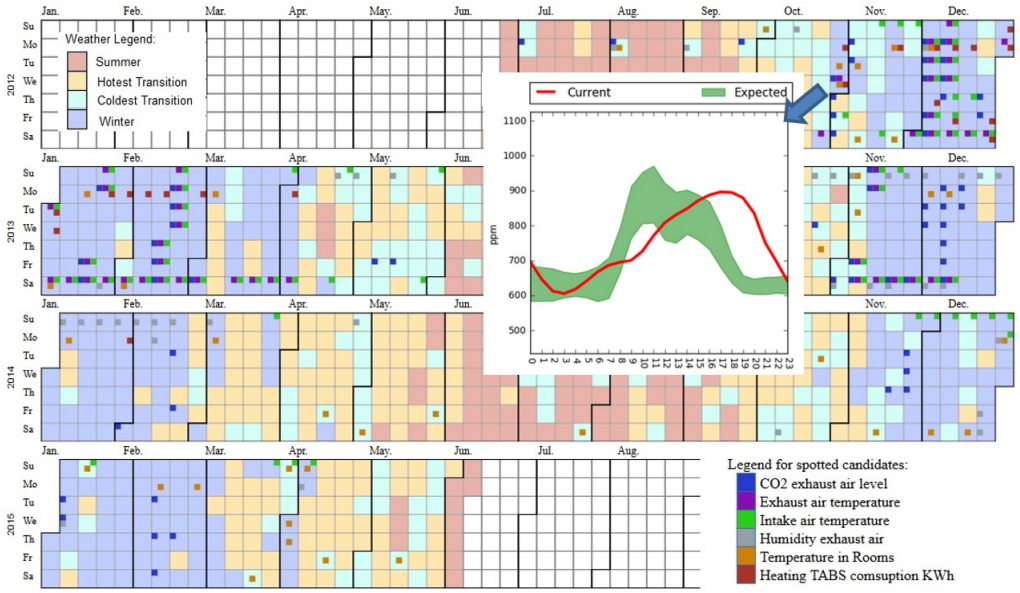


FIGURE 6.1: Possible interactive visualization

- A practical application of fault detection can be implemented and evaluated (section **??**). Experts propose our approach, that is, *GaHMM profile* and HAC approach for this purpose. Using the ground truth of the dataset, the evaluation can be done by using indicators such as the false positives and false negative rates, and a metric indicator for reliability.
- An evaluation of cost/benefits must be done. The question to solve is: How much it is going to cost to achieve a 'perfect' performance of the building (i.e. going from discord profiles to motif profiles)?.
- Apply our approach in critical variables (i.e. variables used in controller devices) can be a motivation for further studies.
- Is there a way to use our approach to discover the best moment to sell energy to the grid?
- If the user awareness increases concerning the dynamics of building, people would be empowered to demand comfort enhancement, lower energy consumption or any other aspect of interest to the user. Can our approach assist the user in this purpose?

A Building Data Set

A.1 Complete list of variable in the Building Data Set

Variable_Id	Einheit	Variable (NEU)
V005	ppm	Lueftungsanlage 1 CO2_Abluft
V022	ppm	Lueftungsanlage 2 CO2_Abluft
V037	kWh	Kaelte TABS Sued/West
V075	kWh	Kaelte TABS Nord/Ost
V034	kWh	Waerme TABS Sued/West
V074	kWh	Waerme TABS Nord/Ost
V100	%	Storen_Winkel Nord Aussen
V103	%	Storen_Winkel Nord Innen
V106	%	Storen_Winkel Ost Aussen
V109	%	Storen_Winkel Ost Innen
V112	%	Storen_Winkel Sued Aussen
V115	%	Storen_Winkel Sued Innen
V118	%	Storen_Winkel West Aussen
V121	%	Storen_Winkel West Innen
V004	% relF	Lueftungsanlage 1 Relative Feuchte Abluft
V021	% relF	Lueftungsanlage 2 Relative Feuchte Abluft
V043	% relF	Relative Feuchte Raum 1.01
V045	% relF	Relative Feuchte Raum 1.02
V076	% relF	Relative Feuchte Raum 1.03
V078	% relF	Relative Feuchte Raum 1.04
V047	% relF	Relative Feuchte Raum 2.01
V049	% relF	Relative Feuchte Raum 2.02
V080	% relF	Relative Feuchte Raum 2.03
V082	% relF	Relative Feuchte Raum 2.04
V051	% relF	Relative Feuchte Raum 3.01
V053	% relF	Relative Feuchte Raum 3.02
V084	% relF	Relative Feuchte Raum 3.03
V086	% relF	Relative Feuchte Raum 3.04

Variable_Id	Einheit	Variable (NEU)
V099	%	Storen_Heohe Nord Aussen
V102	%	Storen_Heohe Nord Innen
V105	%	Storen_Heohe Ost Aussen
V108	%	Storen_Heohe Ost Innen
V111	%	Storen_Heohe Sued Aussen
V114	%	Storen_Heohe Sued Innen
V117	%	Storen_Heohe West Aussen
V120	%	Storen_Heohe West Innen
V098	0/1	Storen_Status Nord Aussen
V101	0/1	Storen_Status Nord Innen
V104	0/1	Storen_Status Ost Aussen
V107	0/1	Storen_Status Ost Innen
V110	0/1	Storen_Status Sued Aussen
V113	0/1	Storen_Status Sued Innen
V116	0/1	Storen_Status West Aussen
V119	0/1	Storen_Status West Innen
V006	°C	Lueftungsanlage 1 Temperatur Abluft
V023	°C	Lueftungsanlage 2 Temperatur Abluft
V032	°C	Aussentemperatur
V044	°C	Temperatur Raum 1.01
V046	°C	Temperatur Raum 1.02
V077	°C	Temperatur Raum 1.03
V079	°C	Temperatur Raum 1.04
V048	°C	Temperatur Raum 2.01
V050	°C	Temperatur Raum 2.02
V081	°C	Temperatur Raum 2.03
V083	°C	Temperatur Raum 2.04
V052	°C	Temperatur Raum 3.01
V054	°C	Temperatur Raum 3.02
V085	°C	Temperatur Raum 3.03
V087	°C	Temperatur Raum 3.04
V012	°C	Lueftungsanlage 1 Temperatur Zuluft
V029	°C	Lueftungsanlage 2 Temperatur Zuluft
tre200b0	°C	Weather temperature
rre150b0	mm	Precipitation level
sre000b0	min	Sunshine Presence per hour

A.2 Metadata for the Building Data Set

tagname	alias	room	orientation	floor	category	location	units	breakout_group	alias_breakout_group
V005_vent01_CO2	CO2 Ventilation NE		NE	-	CO2		ppm	A	CO2 Ventilation NE
V022_vent02_CO2	CO2 Ventilation SW		SW	-	CO2		ppm	B	CO2 Ventilation SW
V037_tabs_cold_SW	Cooling tabs SW		SW	-	Cooling		kWh	B_1	Cooling SW
V075_tabs_cold_NO	Cooling tabs NE		NE	-	Cooling		kWh	A_1	Cooling NE
V034_tabs_warm_SW	Heating tabs SW		SW	-	Heating		kWh	B_2	Heating SW
V074_tabs_warm_NO	Heating tabs NE		NE	-	Heating		kWh	A_2	Heating NE
V100_blinds_angle_N_o	Blinds angle N Out		N	-	Blinds Angle	out	%	A_3	Blinds Angle NE
V103_blinds_angle_N_i	Blinds angle N In		N	-	Blinds Angle	in	%	A_3	Blinds Angle NE
V106_blinds_angle_O_o	Blinds angle E Out		E	-	Blinds Angle	out	%	A_3	Blinds Angle NE
V109_blinds_angle_O_i	Blinds angle E In		E	-	Blinds Angle	in	%	A_3	Blinds Angle NE
V112_blinds_angle_S_o	Blinds angle S Out		S	-	Blinds Angle	out	%	B_3	Blinds Angle SW
V115_blinds_angle_S_i	Blinds angle S In		S	-	Blinds Angle	in	%	B_3	Blinds Angle SW
V118_blinds_angle_W_o	Blinds angle W Out		W	-	Blinds Angle	out	%	B_3	Blinds Angle SW
V121_blinds_angle_W_i	Blinds angle W In		W	-	Blinds Angle	in	%	B_3	Blinds Angle SW
V099_blinds_height_N_o	Blinds height N Out		N	-	Blinds Height	out	%	A_4_1	Blinds Height N-out, E-in
V102_blinds_height_N_i	Blinds height N In		N	-	Blinds Height	in	%	A_4_2	Blinds Height N-in, E-out
V105_blinds_height_O_o	Blinds height E Out		E	-	Blinds Height	out	%	A_4_2	Blinds Height N-in, E-out
V108_blinds_height_O_i	Blinds height E In		E	-	Blinds Height	in	%	A_4_1	Blinds Height N-out, E-in
V111_blinds_height_S_o	Blinds height S Out		S	-	Blinds Height	out	%	B_4_2	Blinds Height S-out, W-out, W-in
V114_blinds_height_S_i	Blinds height S In		S	-	Blinds Height	in	%	B_4_1	Blinds Height S-in
V117_blinds_height_W_o	Blinds height W Out		W	-	Blinds Height	out	%	B_4_2	Blinds Height S-out, W-out, W-in
V120_blinds_height_W_i	Blinds height W In		W	-	Blinds Height	in	%	B_4_2	Blinds Height S-out, W-out, W-in

tagname	alias	room	orientation	floor	category	location	units	breakout_group	alias_breakout_group
V004_vent01_hum_out	Hum. Ventilation NE		NE	-	Humidity	out	%	A_5_1	Hum. Ventilation NE
V021_vent02_hum_out	Hum. Ventilation SW		SW	-	Humidity	out	%	B_5_1	Hum. Ventilation SW
V043_room101_hum	Hum. Room 101	101	N	1	Humidity	in	%	A_5_2	Hum. Room NE
V045_room102_hum	Hum. Room 102	102	E	1	Humidity	in	%	A_5_2	Hum. Room NE
V076_room103_hum	Hum. Room 103	103	S	1	Humidity	in	%	B_5_2	Hum. Room SW
V078_room104_hum	Hum. Room 104	104	W	1	Humidity	in	%	B_5_2	Hum. Room SW
V047_room201_hum	Hum. Room 201	201	N	2	Humidity	in	%	A_5_2	Hum. Room NE
V049_room202_hum	Hum. Room 202	202	E	2	Humidity	in	%	A_5_2	Hum. Room NE
V080_room203_hum	Hum. Room 203	203	S	2	Humidity	in	%	B_5_2	Hum. Room SW
V082_room204_hum	Hum. Room 204	204	W	2	Humidity	in	%	B_5_2	Hum. Room SW
V051_room301_hum	Hum. Room 301	301	N	3	Humidity	in	%	A_5_2	Hum. Room NE
V053_room302_hum	Hum. Room 302	302	E	3	Humidity	in	%	A_5_2	Hum. Room NE
V084_room303_hum	Hum. Room 303	303	S	3	Humidity	in	%	B_5_2	Hum. Room SW
V086_room304_hum	Hum. Room 304	304	W	3	Humidity	in	%	B_5_2	Hum. Room SW
V006_vent01_temp_out	Temp. Vent. NE Out		NE	-	Temperature	out	’C	A_6_1	Temp. Vent. NE Out
V023_vent02_temp_out	Temp. Vent. SW Out		SW	-	Temperature	out	’C	B_6_1	Temp. Vent. SW Out
V012_vent01_temp_in	Temp. Vent. NE In		NE	-	Temperature	in	’C	A_6_2	Temp. Vent. NE In
V029_vent02_temp_in	Temp. Vent. SW In		SW	-	Temperature	in	’C	B_6_2	Temp. Vent. SW In
V032_outdoor_temp	Outdoor Temperature		N/A	-	Temperature	out	’C	C_1	Outdoor Temperature
V044_room101_temp	Temp. Room 101	101	N	1	Temperature	in	’C	A_6_3	Temperature Room NE
V046_room102_temp	Temp. Room 102	102	E	1	Temperature	in	’C	A_6_3	Temperature Room NE

tagname	alias	room	orientation	floor	category	location	units	breakout_group	alias_breakout_group
V077_room103_temp	Temp. Room 103	103	S	1	Temperature	in	′C	B_6_3	Temperature Room SW
V079_room104_temp	Temp. Room 104	104	W	1	Temperature	in	′C	B_6_3	Temperature Room SW
V048_room201_temp	Temp. Room 201	201	N	2	Temperature	in	′C	A_6_3	Temperature Room NE
V050_room202_temp	Temp. Room 202	202	E	2	Temperature	in	′C	A_6_3	Temperature Room NE
V081_room203_temp	Temp. Room 203	203	S	2	Temperature	in	′C	B_6_3	Temperature Room SW
V083_room204_temp	Temp. Room 204	204	W	2	Temperature	in	′C	B_6_3	Temperature Room SW
V052_room301_temp	Temp. Room 301	301	N	3	Temperature	in	′C	A_6_3	Temperature Room NE
V054_room302_temp	Temp. Room 302	302	E	3	Temperature	in	′C	A_6_3	Temperature Room NE
V085_room303_temp	Temp. Room 303	303	S	3	Temperature	in	′C	B_6_3	Temperature Room SW
V087_room304_temp	Temp. Room 304	304	W	3	Temperature	in	′C	B_6_3	Temperature Room SW
sre000b0	Sunshine		N/A	-	Weather	out	min	C_2	Sunshine
rre150b0	Precipitation		N/A	-	Weather	out	mm	C_3	Precipitation
tre200b0	Weather Temperature		N/A	-	Weather	out	′C	C_4	Weather Temperature
V098	St. blind N Out		N	-	Status blind		I/O		
V101	St. blind N In		N	-	Status blind		I/O		
V104	St. blind E Out		E	-	Status blind		I/O		
V107	St. blind E In		E	-	Status blind		I/O		
V110	St. blind S Out		S	-	Status blind		I/O		
V113	St. blind S In		S	-	Status blind		I/O		
V116	St. blind W Out		W	-	Status blind		I/O		
V119	St. blind W In		W	-	Status blind		I/O		

Feature	**Definition**
max	Maximun value in the sample
min	Minimun value in the sample
mean	Mean value of the sample
75%	Percentil 75% of the sample
50%	Percentil 50% of the sample
25%	Percentil 25% of the sample
std	Standar deviation of the sample
dev_u	It calculates the deviation of the scores in the sample against an arbitrary value u
*r_factor	ripple factor for a sample with respect to his mean (i.e. u= mean).
r_factor_u	ripple factor for a sample with respect to an arbitrary value u
r_factor_st	ripple factor for a sample with respect to an arbitrary value u, in hours [0h to 6h]
r_factor_ed	ripple factor for a sample with respect to an arbitrary value u, in hours [18h to 23h]
max_st	Maximun value in hours [0h to 6h]
max_ed	Maximun value in hours [18h to 23h]
min_st	Minimun value in hours [0h to 6h]
min_ed	Minimun value in hours [18h to 23h]
min_me	Minimun value in hours [7h to 18h]
(max-min)*std	It calculate the expression: (max-min)*std

*r_factor can be multiplied by the mean, or any other feature in order to combine information.
His definition is in section **??**

TABLE A.1: List of features used for the GaHMM seasonal model

B Results

B.1 Cluster profiles

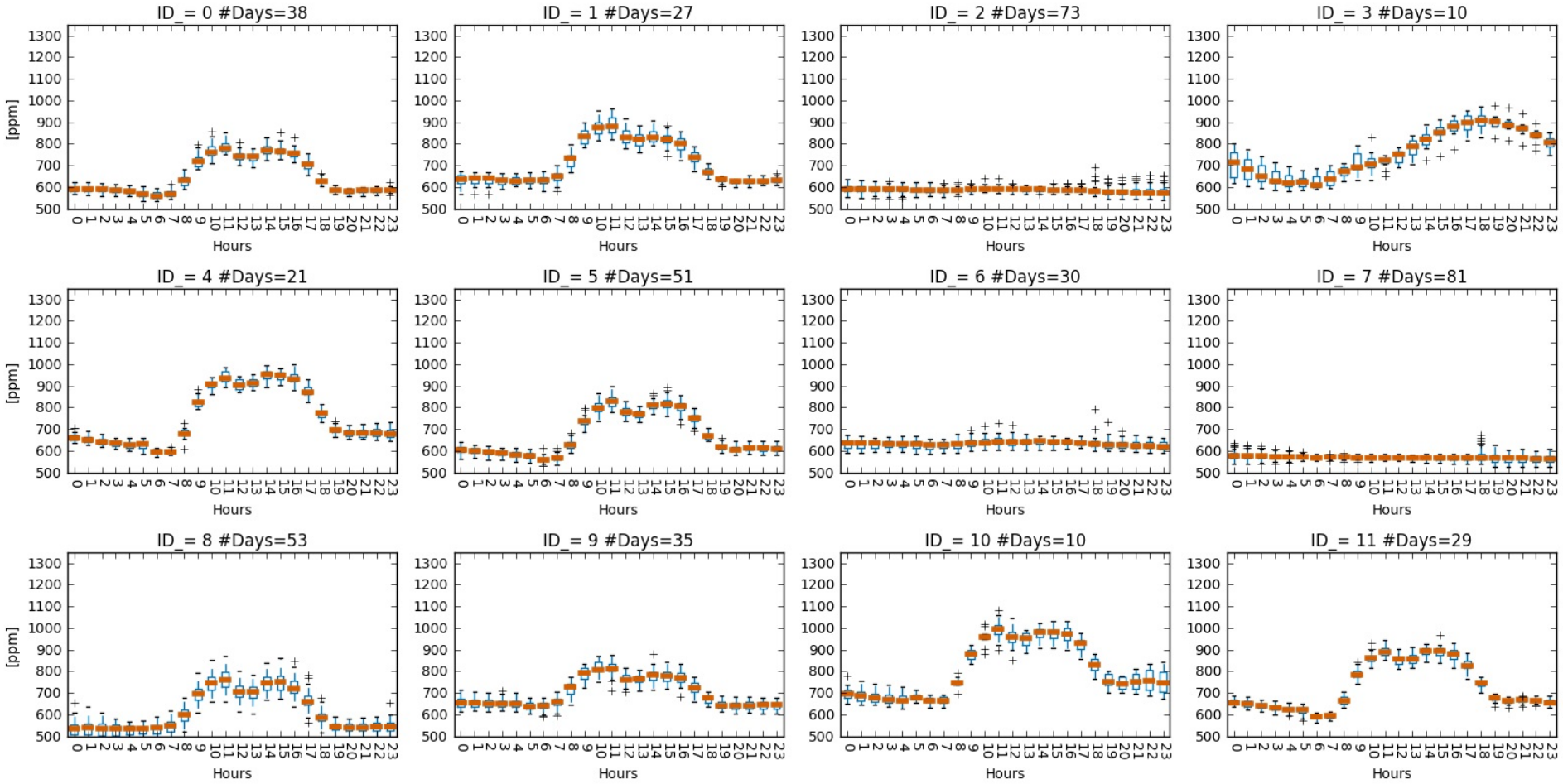


FIGURE B.1: CO_2 cluster profiles for the North-East ventilation system of the building. (1/3)

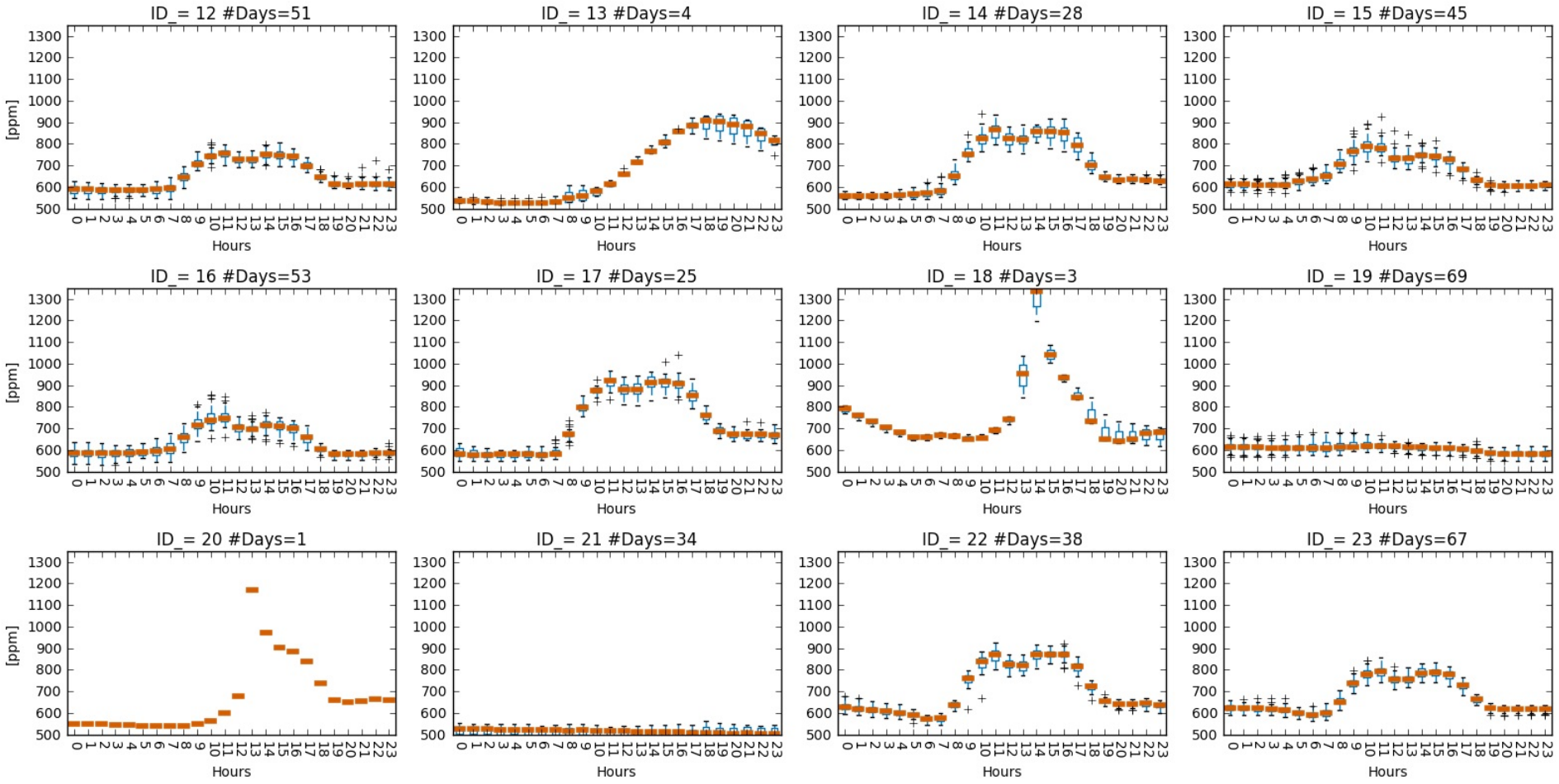


FIGURE B.2: CO_2 cluster profiles for the North-East ventilation system of the building. (2/3)

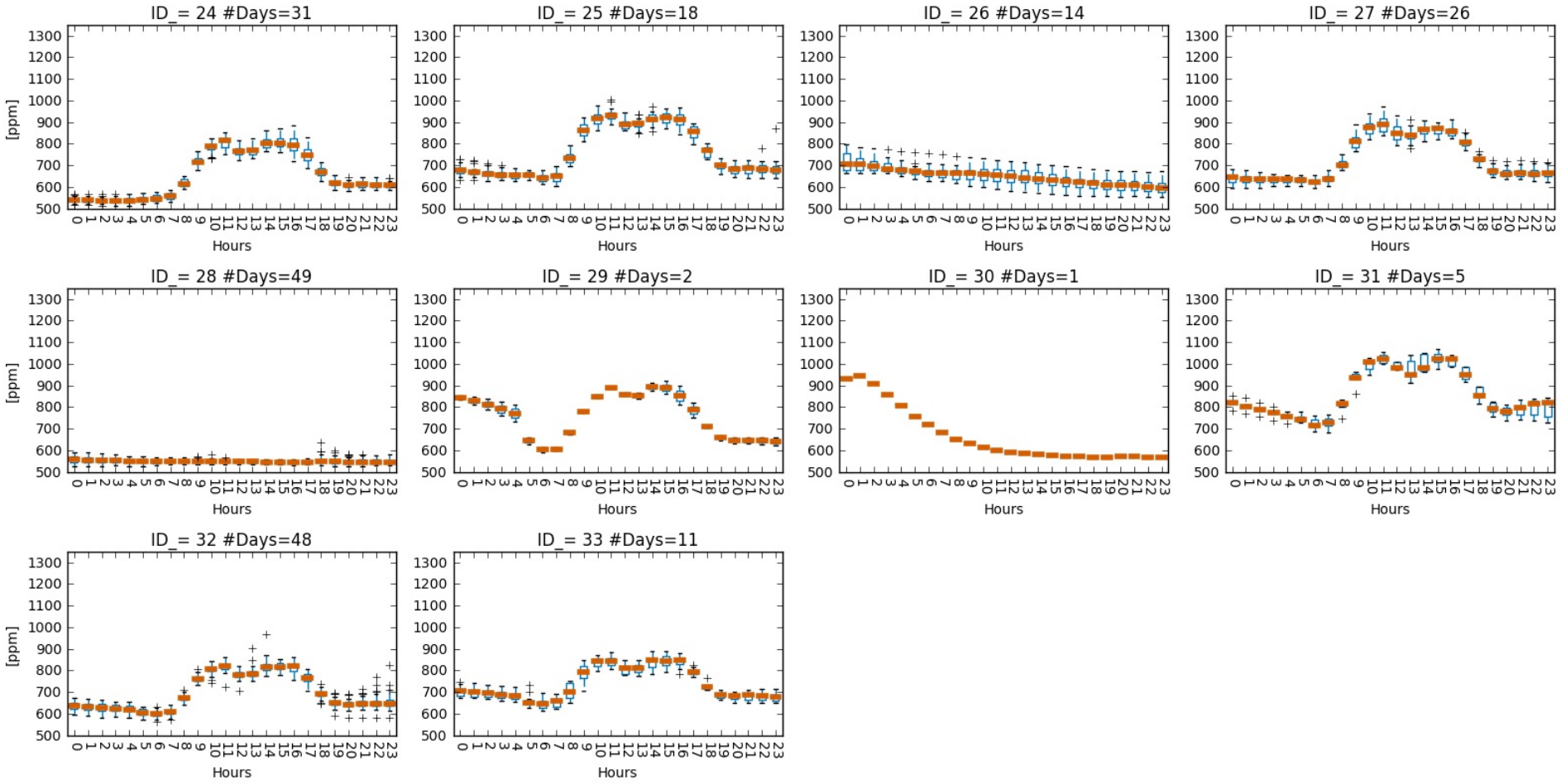


FIGURE B.3: CO_2 cluster profiles for the North-East ventilation system of the building. (3/3)

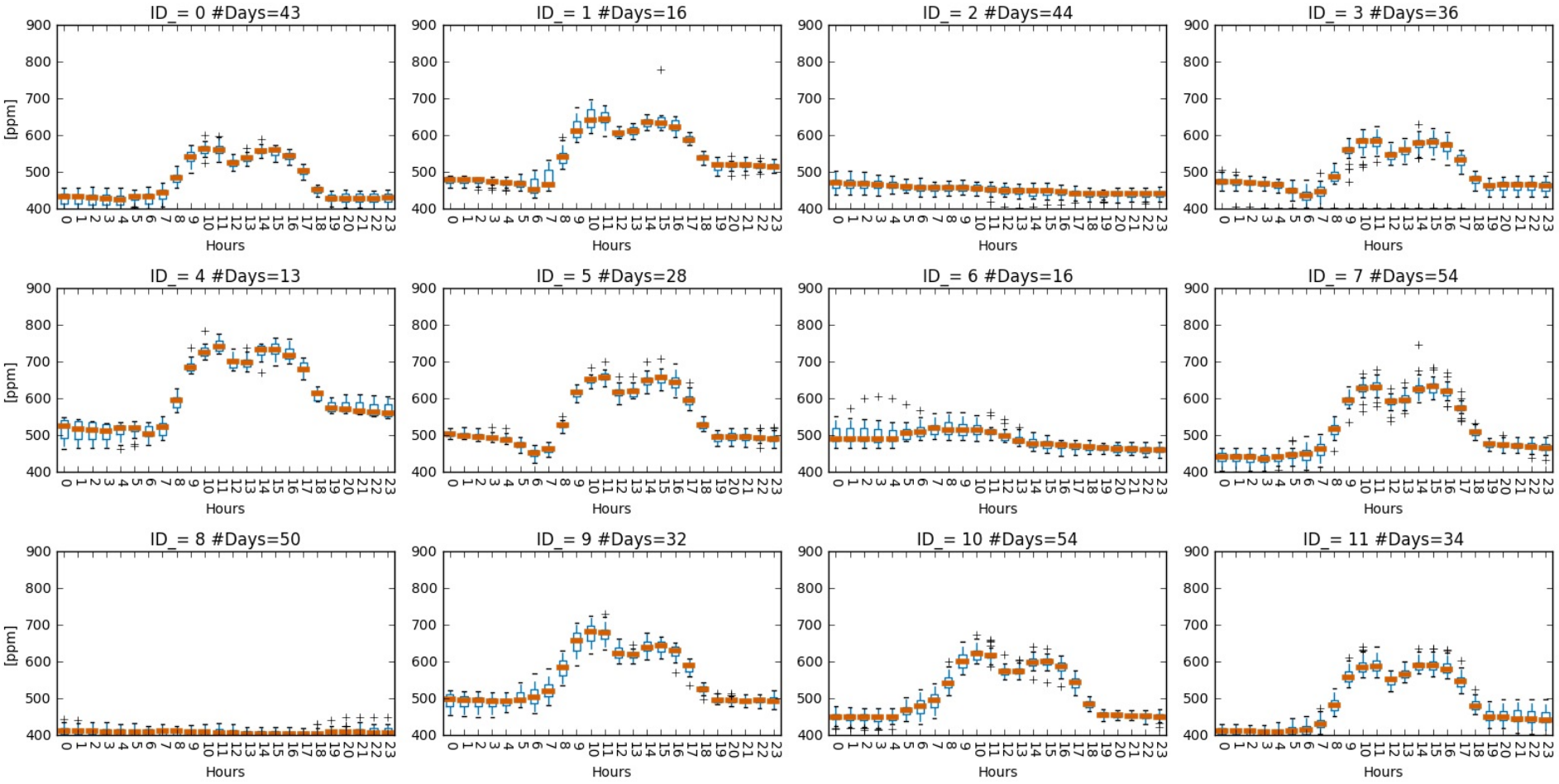


FIGURE B.4: CO_2 cluster profiles for the South-West ventilation system of the building. (1/3)

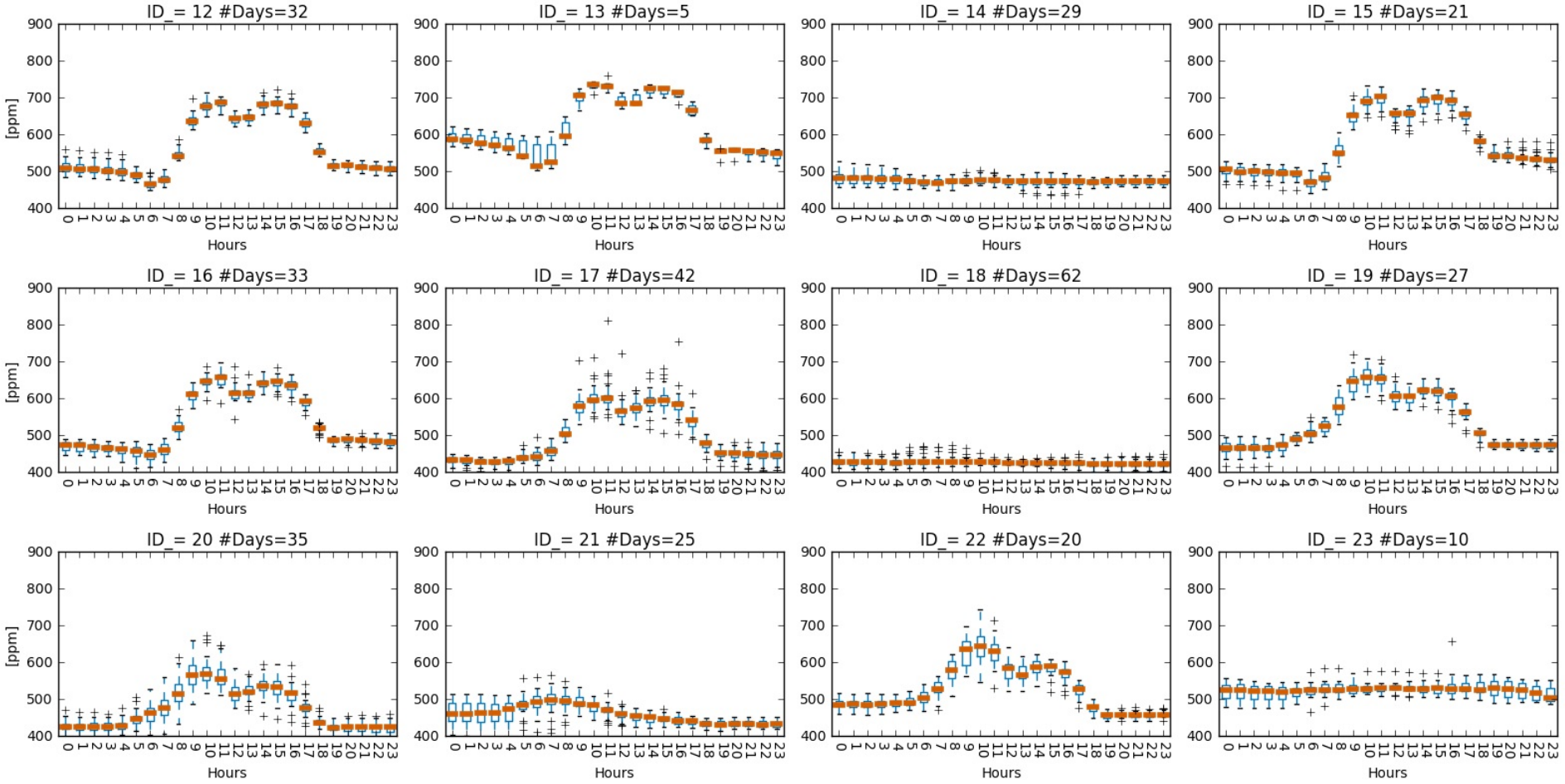


FIGURE B.5: CO_2 cluster profiles for the South-West ventilation system of the building. (2/3)

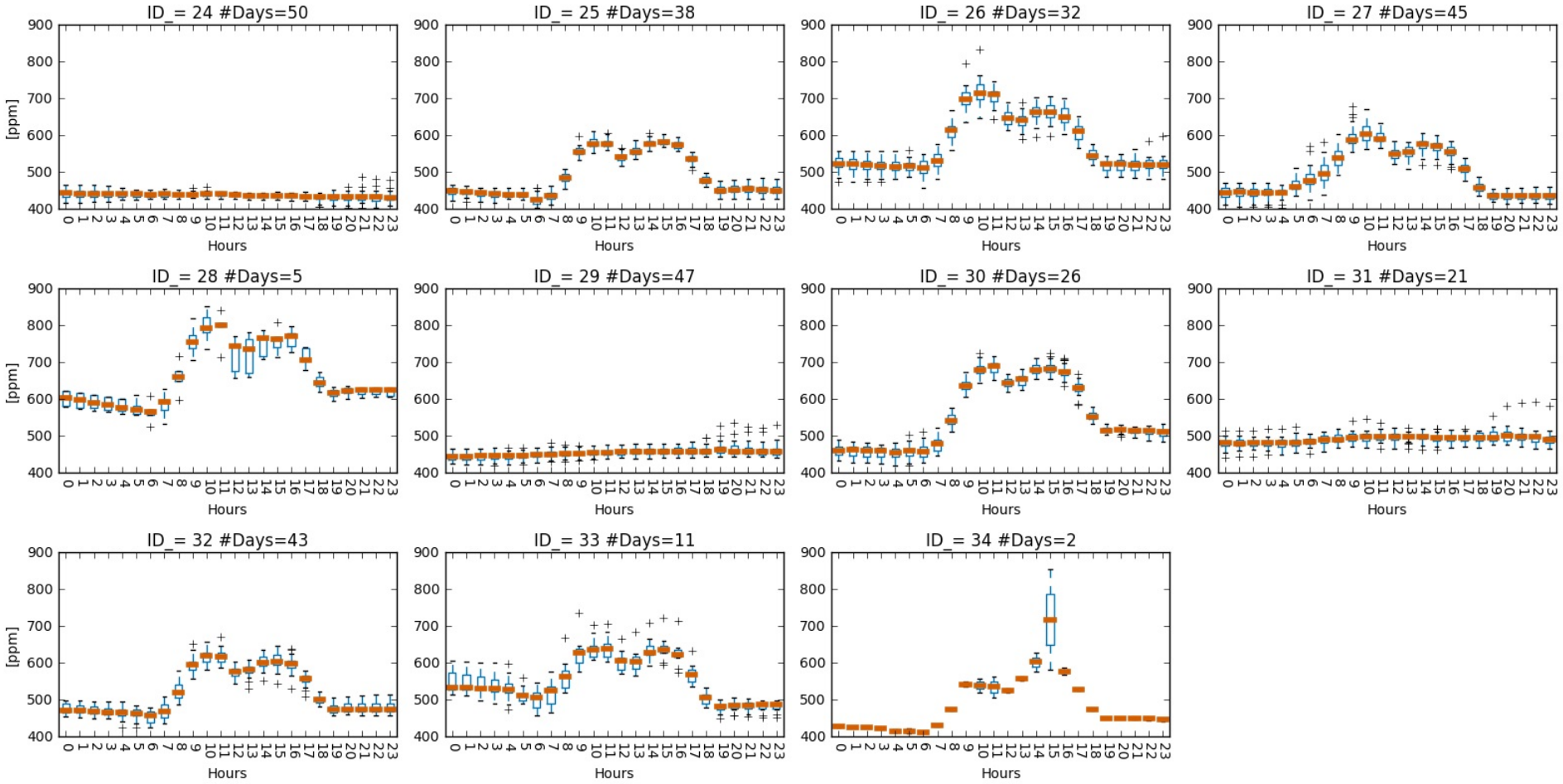


FIGURE B.6: CO_2 cluster profiles for the South-West ventilation system of the building. (3/3)

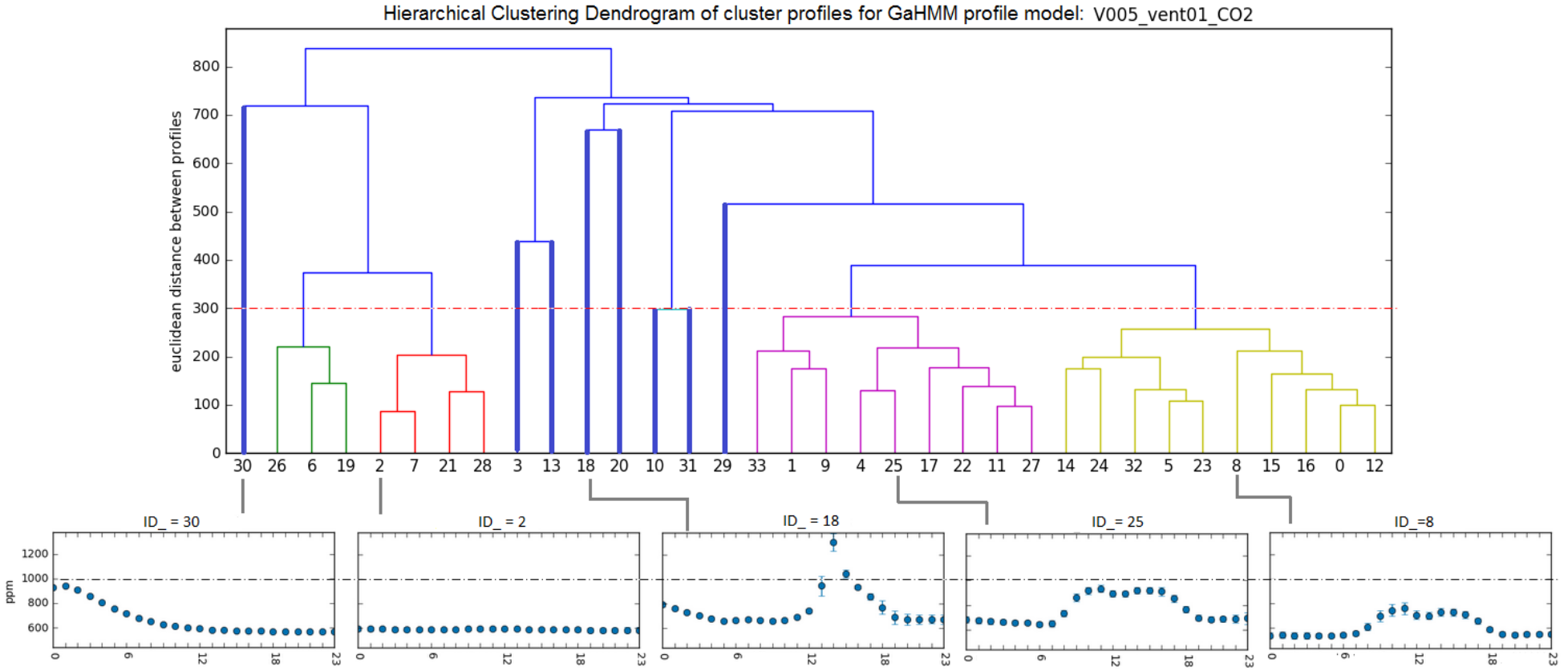


FIGURE B.7: Hierarchical clustering dendrogram for the CO_2 cluster profiles of the North-East part of the building.

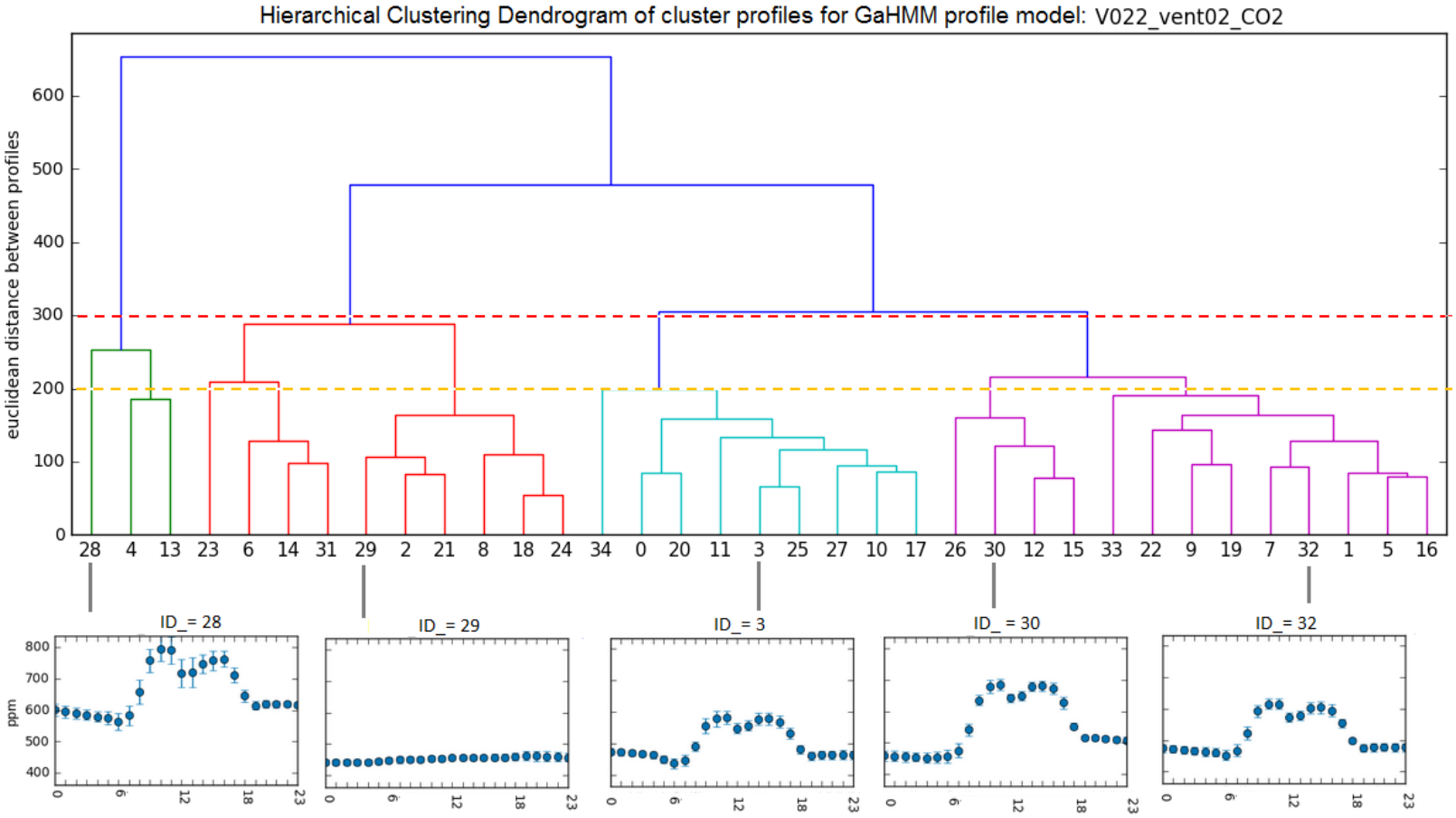


FIGURE B.8: Hierarchical clustering dendrogram for the CO_2 cluster profiles of the South-West part of the building.

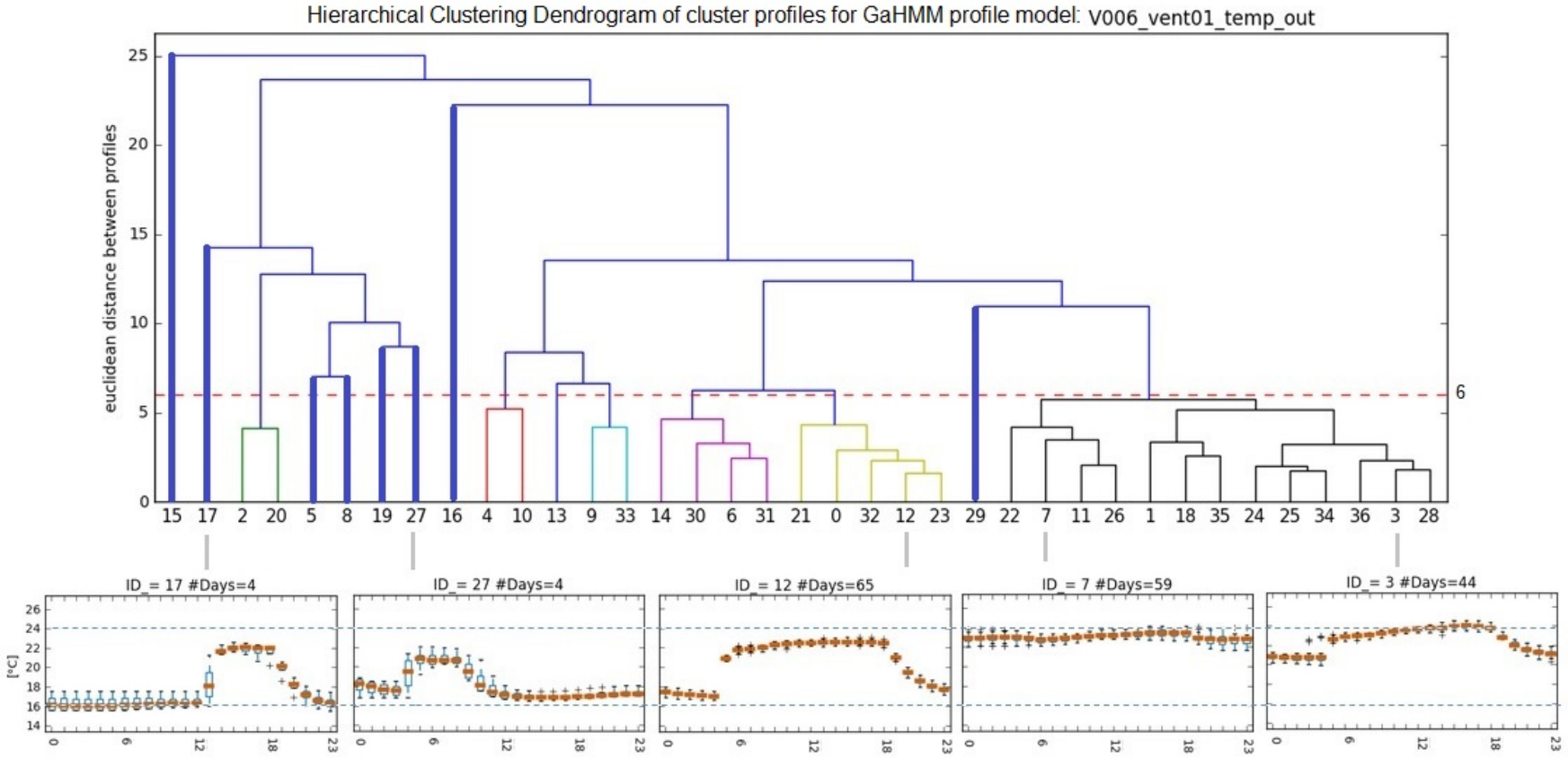


FIGURE B.9: Hierarchical clustering dendrogram for *exhausted air temperature - cluster profiles* of the North-East ventilation system.

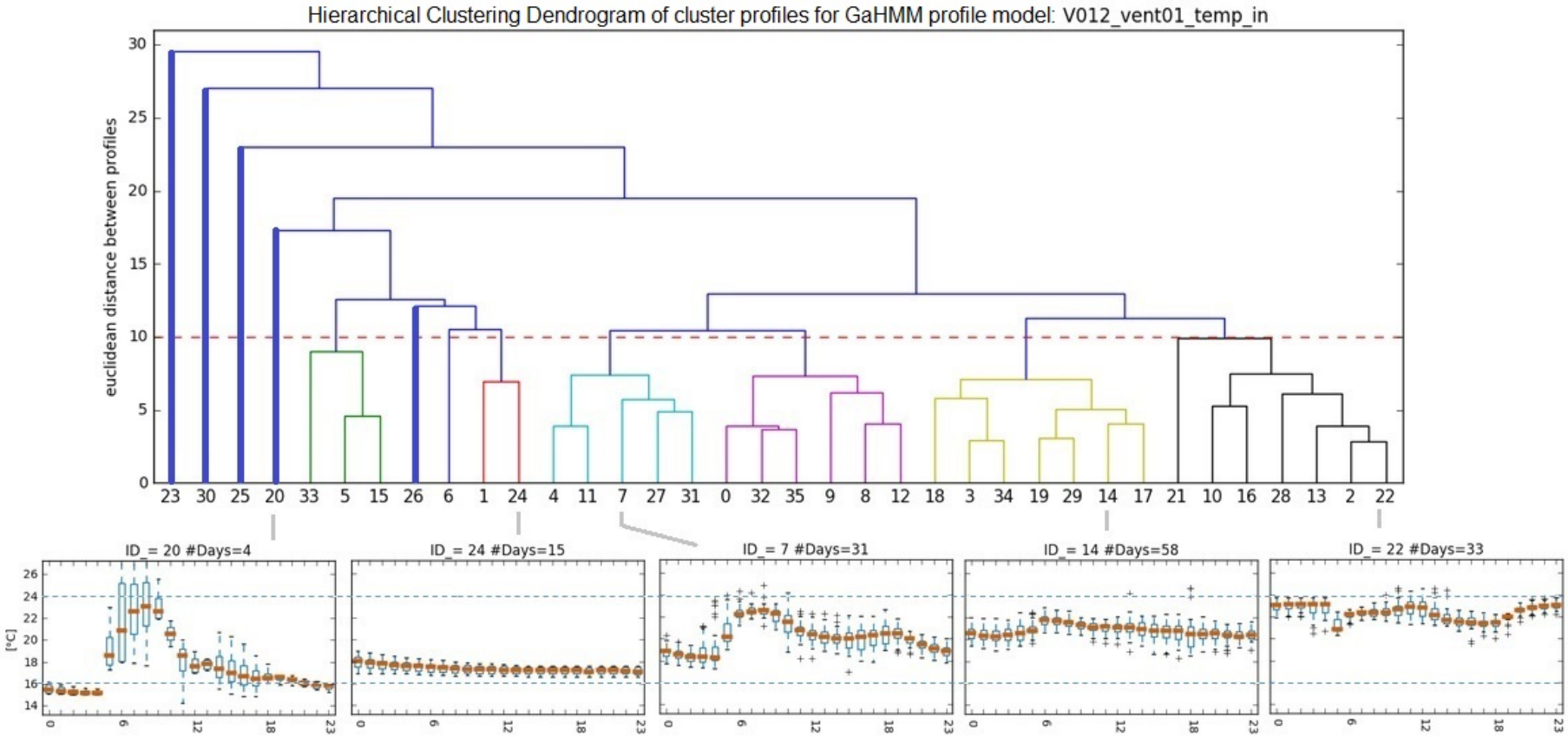


FIGURE B.10: Hierarchical clustering dendrogram for *intake air temperature - cluster profiles* of the North-East ventilation system.

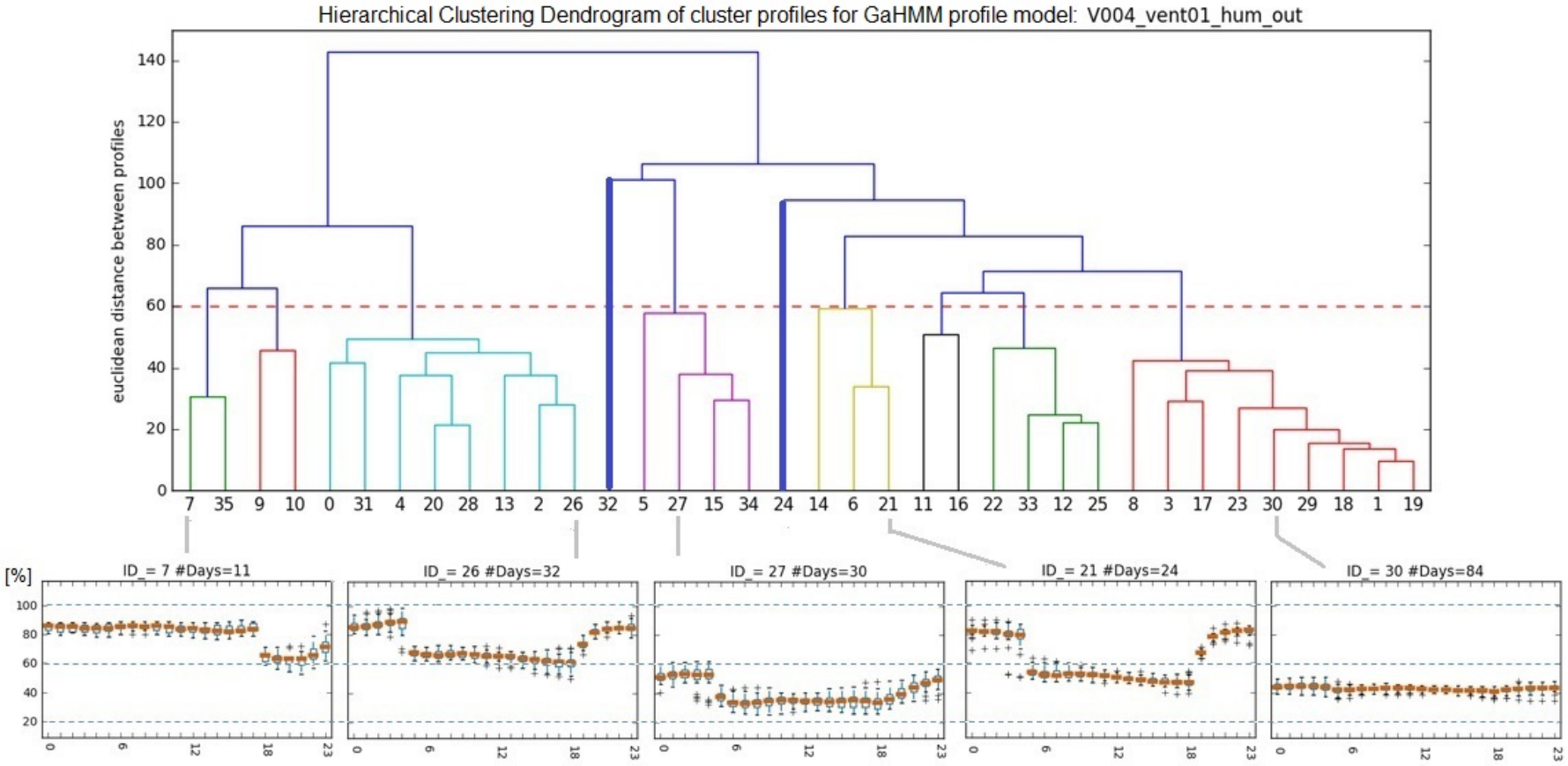


FIGURE B.11: Hierarchical clustering dendrogram for *exhausted air humidity - cluster profiles* of the North-East ventilation system.

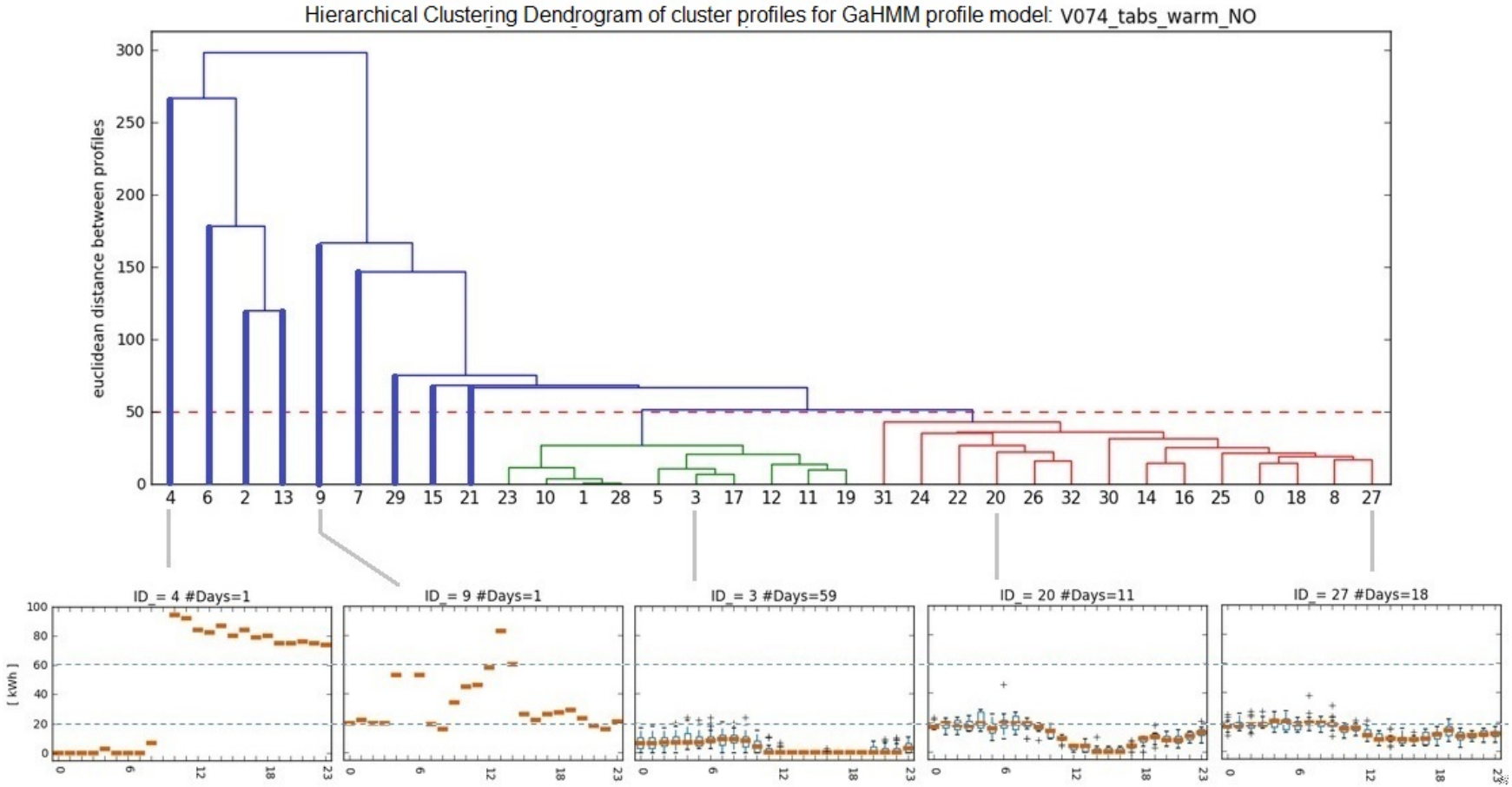


FIGURE B.12: Hierarchical clustering dendrogram for *heating consumption energy - cluster profiles* of the North-East part of the building.

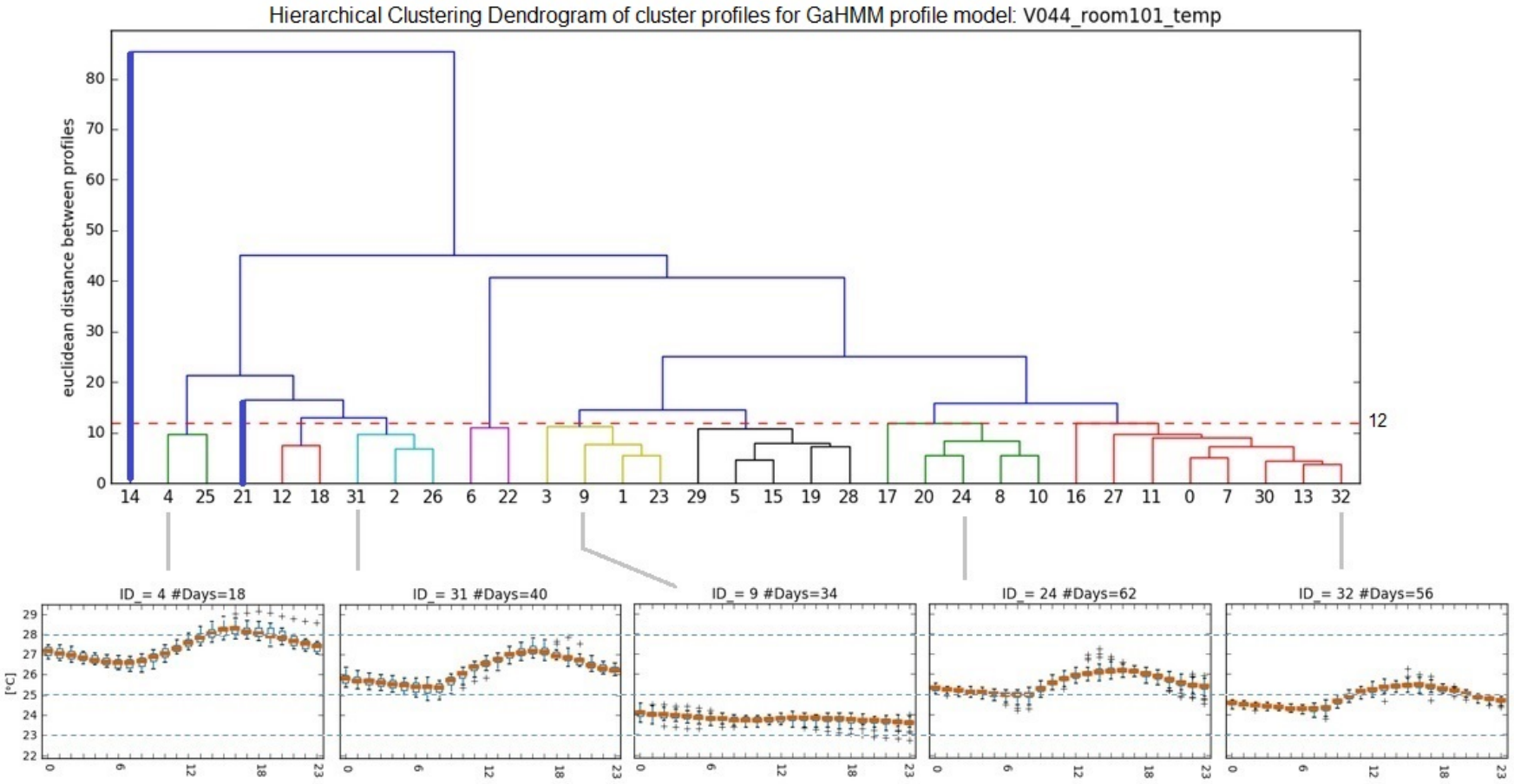


FIGURE B.13: Hierarchical clustering dendrogram for *room temperature - cluster profiles* of room 101 that belongs to the North-East part of the building.

Profile code		00:00	01:00	02:00	03:00	04:00	05:00	06:00	07:00	08:00	09:00	10:00	11:00
3	**mean**	705.9	682.5	662.2	643.8	629.7	624.4	625.4	641.6	672.0	710.1	711.9	714.6
	std	63.1	53.2	44.8	37.8	34.0	26.6	31.5	36.9	24.6	54.3	53.9	28.0
10	**mean**	701.2	692.3	683.9	675.4	670.6	682.6	665.2	667.1	748.5	878.1	956.8	991.6
	std	35.6	34.4	32.4	30.4	26.4	17.2	16.3	18.8	24.0	24.4	39.5	52.1
18	**mean**	790.1	758.1	727.7	699.7	677.3	657.4	660.5	670.0	665.2	657.8	660.8	691.1
	std	15.9	13.3	12.6	10.7	10.2	8.8	10.9	11.0	11.4	10.1	9.1	11.4
29	**mean**	843.8	831.1	813.1	792.2	771.8	644.8	603.8	605.8	681.9	776.9	849.2	888.9
	std	9.6	18.2	24.3	31.8	38.8	15.9	10.9	8.2	7.2	1.4	7.9	6.1
13	**mean**	537	536	534.3	531.2	532	532	533	537	559	566	580	615
	std	10.6	11	11.4	11.88	9.27	9.86	12.3	13.5	32.6	29.9	17	11.1
26	**mean**	719.1	712.1	703.6	694.6	685.5	676.0	670.0	667.0	663.9	661.0	658.5	655.3
	std	45.9	37.9	33.1	30.7	30.0	30.6	31.3	31.3	32.9	34.6	37.9	40.0
20	**mean**	551	549	549	545.1	544	542	542	541	542	549	564	600
	std	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
30	**mean**	930.8	942.8	908.5	859.1	805.4	758.0	718.0	681.0	652.7	629.8	613.9	601.6
	std	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
31	**mean**	819	806	788.4	772.2	756	748	722	727	803	928	996	1025
	std	21.3	22.8	21.58	20.24	18.9	17.7	25.6	28.7	30.9	35.9	30.2	20.7
33	**mean**	702.7	699.8	695.1	689.6	683.3	660.3	645.8	650.1	705.2	791.3	835.3	842.7
	std	23.3	22.2	21.2	20.1	18.9	28.6	26.0	22.4	35.3	40.2	24.9	23.8
4	**mean**	664	654	644.7	634.8	625	627	596	597	681	826	905	939
	std	19.5	17.4	15.65	13.52	13.5	19.6	11.1	10.7	23.3	22.7	20.7	27.5

Profile code		12:00	13:00	14:00	15:00	16:00	17:00	18:00	19:00	20:00	21:00	22:00	23:00
3	**mean**	742.2	782.2	817.1	849.3	876.9	895.0	903.6	900.8	889.0	869.8	836.9	800.4
	std	31.3	40.4	44.3	43.1	43.1	42.9	40.1	36.9	36.0	34.8	32.9	32.5
10	**mean**	956.8	949.9	973.9	972.0	969.4	922.7	830.1	756.2	742.9	754.7	760.6	756.5
	std	51.6	34.5	33.8	37.8	40.1	36.2	35.4	32.8	28.4	36.9	49.0	55.8
18	**mean**	739.1	945.1	1301.5	1044.4	933.7	855.7	767.3	686.4	668.0	667.9	671.3	668.1
	std	15.8	79.0	73.7	33.7	14.2	23.6	54.9	56.2	44.6	39.6	36.3	37.7
29	**mean**	859.1	854.1	894.2	890.1	853.6	787.3	709.7	659.2	646.2	644.8	645.3	642.9
	std	8.1	13.9	17.4	29.6	43.6	33.8	6.0	11.3	13.6	13.8	16.5	18.8
13	**mean**	664	720	768.9	811.3	859	886	893	889	880	865	836	804
	std	13.5	12.8	13.13	21.64	7.64	26.6	42	48.8	51.3	48.7	41.5	35
26	**mean**	650.6	646.4	639.0	631.9	625.5	619.7	615.7	613.0	611.0	609.9	606.2	603.4
	std	40.9	40.3	36.5	35.1	34.2	34.1	33.8	35.6	35.7	36.5	37.6	38.6
20	**mean**	676	1167	973.4	902.5	883	839	737	661	648	656	662	658
	std	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
30	**mean**	592.2	585.0	580.8	577.1	574.0	573.1	568.5	566.5	570.5	570.2	567.9	566.5
	std	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
31	**mean**	988	971	1001	1024	1013	953	862	794	777	788	798	796
	std	15.9	47.7	37.59	30.83	21.2	28.4	29.3	26.3	25.2	31.5	40.3	46
33	**mean**	805.2	810.6	838.3	842.9	840.0	794.6	725.5	686.4	680.5	680.9	679.3	677.2
	std	21.8	22.6	32.8	29.0	25.6	15.7	14.2	15.1	15.8	17.2	17.8	19.1
4	**mean**	907	914	950.3	947.2	939	876	773	698	683	684	685	681
	std	25.3	19.3	24.66	22.82	29.1	27	22.1	19.5	18.7	19.6	21.8	24.1

TABLE B.1: Discord clusters profile of the CO2 level variable that belongs to the North-East zone of the building.

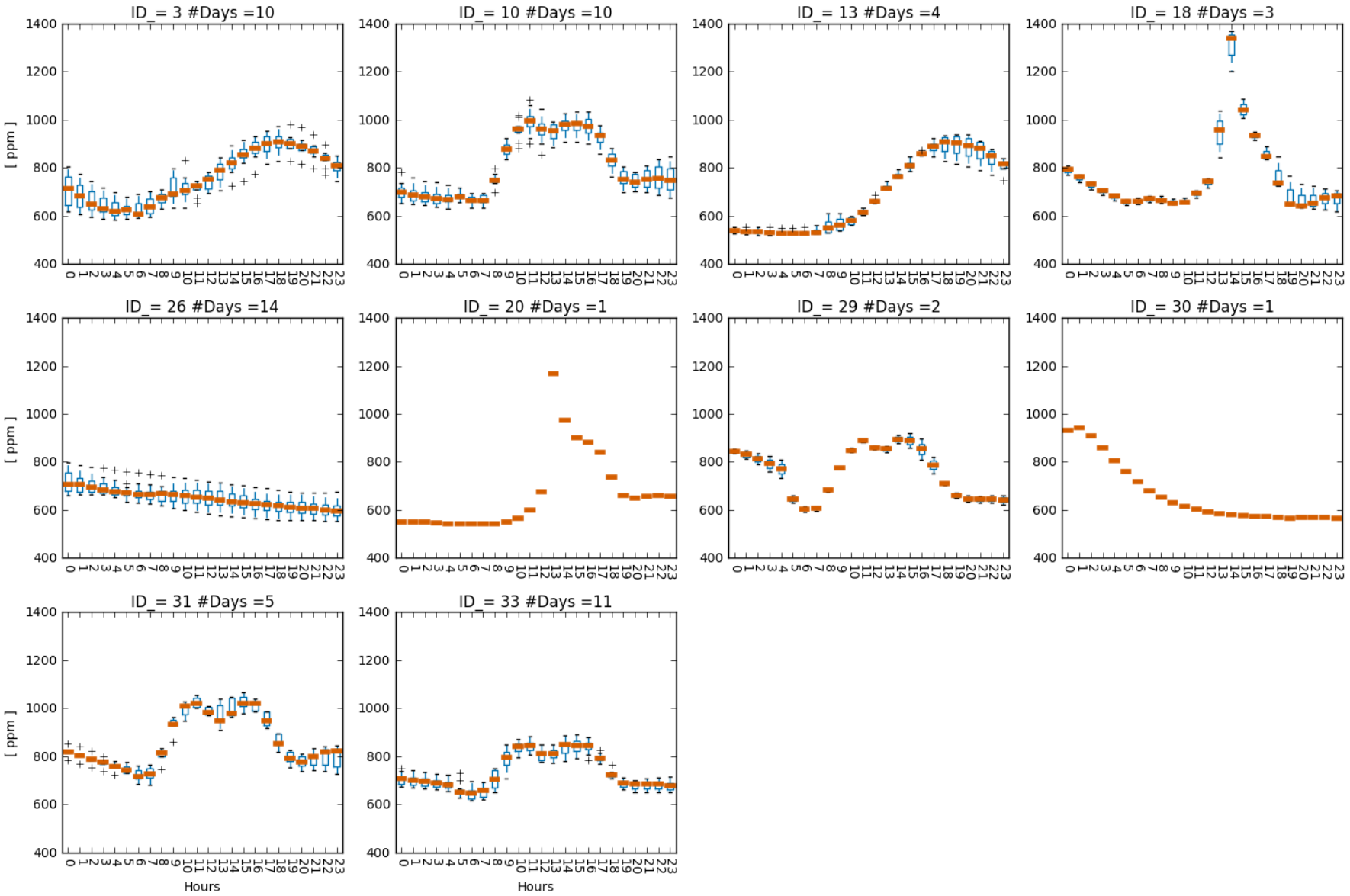


FIGURE B.14: CO_2 discord clusters of the North-East ventilation system of the building.

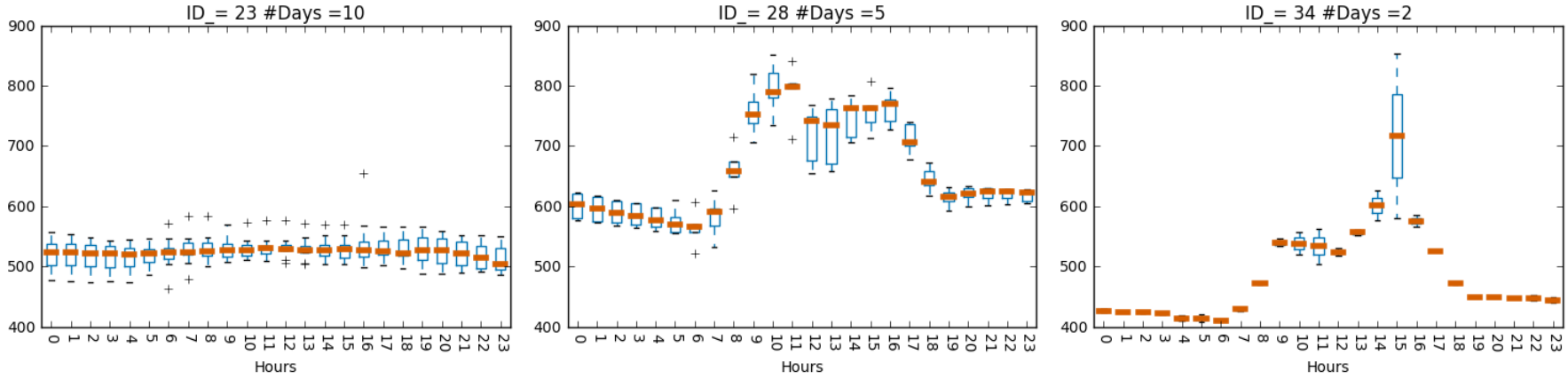


FIGURE B.15: CO_2 discord clusters of the South-West ventilation system of the building.

CO2 level North East (V005_vent01_CO2)

	W = 4, |A| = 3	W = 4, |A| = 4	W = 4, |A| = 5	W = 4, |A| = 6	
GaHMM	61	61	61	61	days
DayFilter	311	84	29	36	days
Coincidence	49	39	25	21	days
% coincidence	15.2	36.8	38.5	27.6	%

	W = 6, |A| = 3	W = 6, |A| = 4	W = 6, |A| = 5	W = 6, |A| = 6	
GaHMM	61	61	61	61	days
DayFilter	49	33	19	16	days
Coincidence	29	25	17	15	days
% coincidence	35.8	36.2	27.0	24.2	%

Temperature room 101 (V044_room101_temp)

	W = 4, |A| = 3	W = 4, |A| = 4	W = 4, |A| = 5	W = 4, |A| = 6	
GaHMM	114	114	114	114	days
DayFilter	319	145	120	99	days
Coincidence	114	87	85	68	days
% coincidence	35.7	50.6	57.0	46.9	%

	W = 6, |A| = 3	W = 6, |A| = 4	W = 6, |A| = 5	W = 6, |A| = 6	
GaHMM	114	114	114	114	days
DayFilter	209	138	109	95	days
Coincidence	102	85	74	70	days
% coincidence	46.2	50.9	49.7	50.4	%

Humidity room 101 (V043_room101_hum)

	W = 4, |A| = 3	W = 4, |A| = 4	W = 4, |A| = 5	W = 4, |A| = 6	
GaHMM	74	74	74	74	days
DayFilter	562	357	259	215	days
Coincidence	74	73	72	72	days
% coincidence	13.2	20.4	27.6	33.2	%

	W = 6, |A| = 3	W = 6, |A| = 4	W = 6, |A| = 5	W = 6, |A| = 6	
GaHMM	74	74	74	74	days
DayFilter	523	328	238	204	days
Coincidence	73	73	72	72	days
% coincidence	13.9	18.2	23.1	25.9	%

TABLE B.2: Number of days where discords clusters were spotted by using *GaHMM-profile model* and DayFilter approach. The corresponding coincidence of both approaches is done when both have the same date.

B.1.1 Case Study

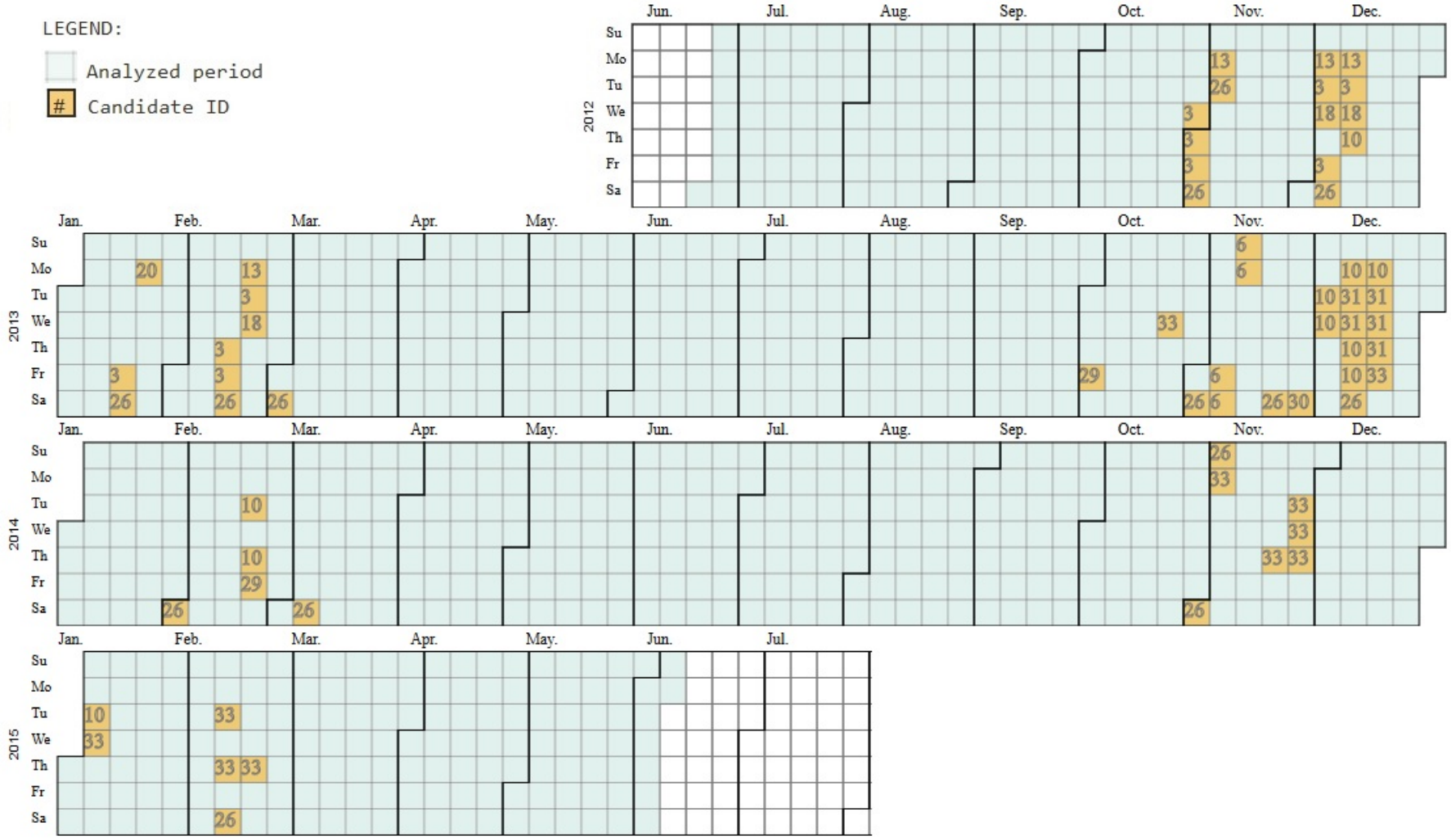


FIGURE B.16: Sequence of CO_2 discord clusters of the North-East ventilation system. Cluster 26 appears in mostly all the fault periods.

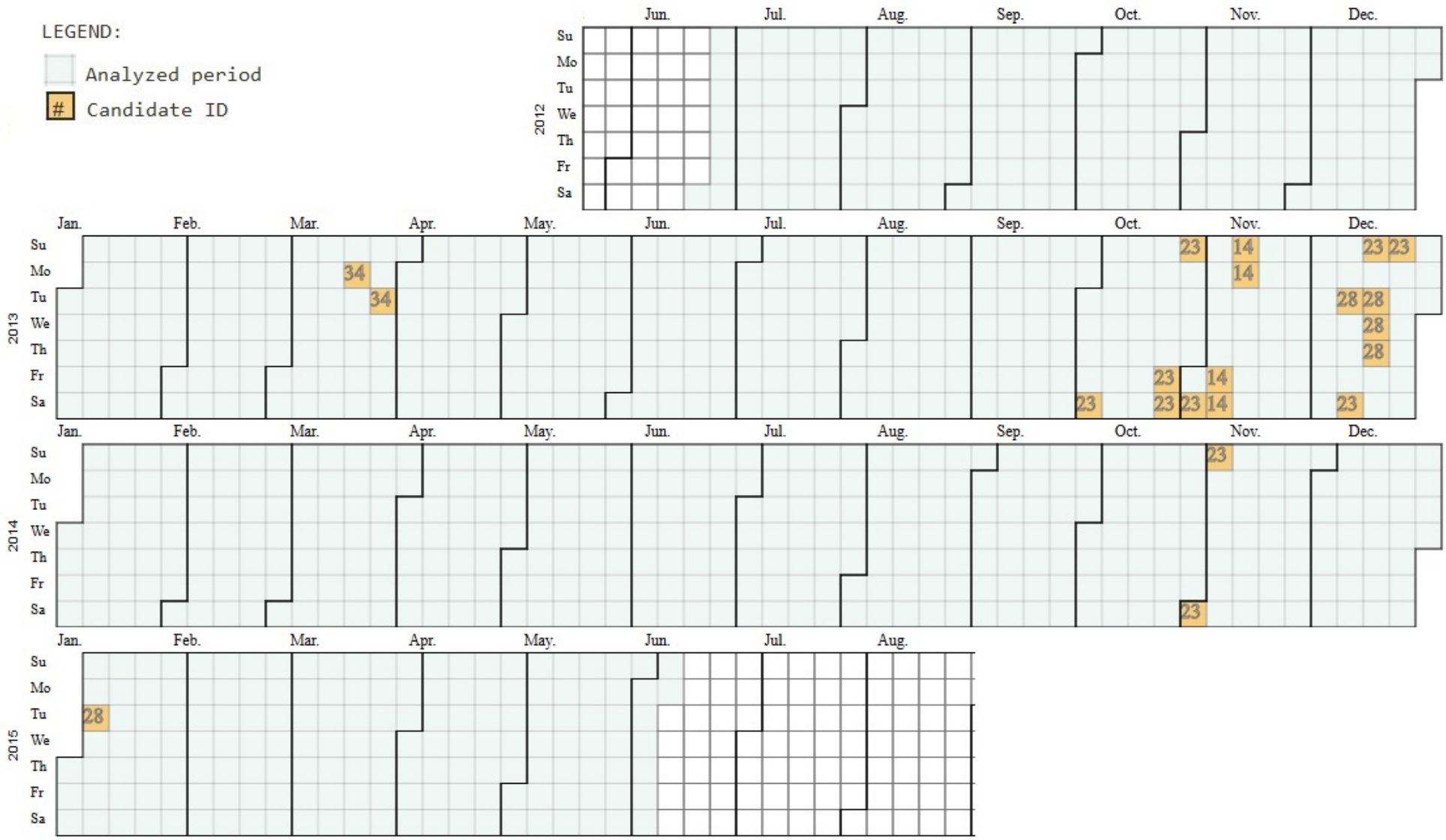


FIGURE B.17: Sequence of CO_2 discord clusters of the South-West ventilation system.

B.1.2 Coincidence between FilterDay and GaHMM-profile model

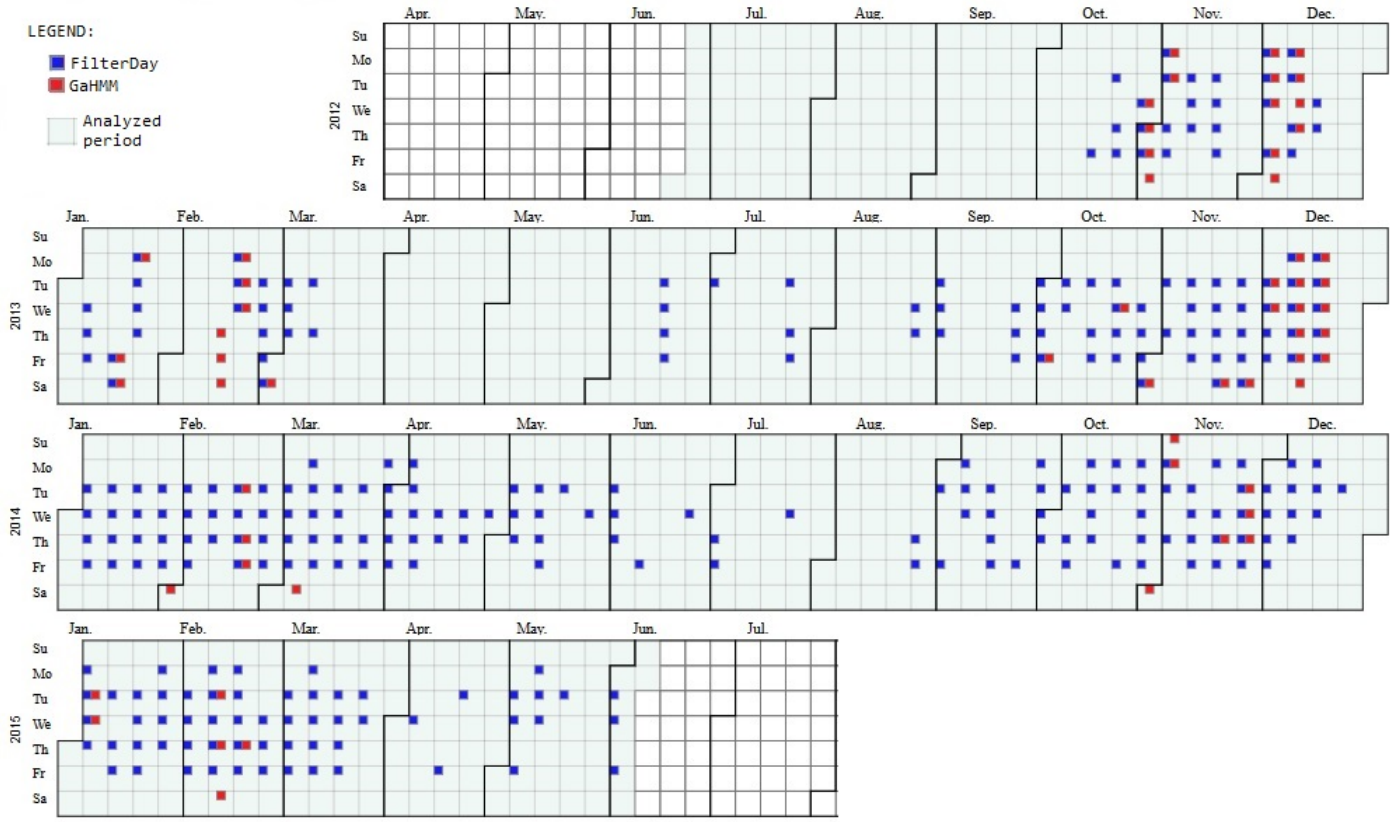


FIGURE B.18: Coincidence FilterDay vs. GaHMM-profile when $W = 4, |A| = 3$. A small square indicates a CO_2 daily profile spotted as discord cluster for the North-East ventilation system.

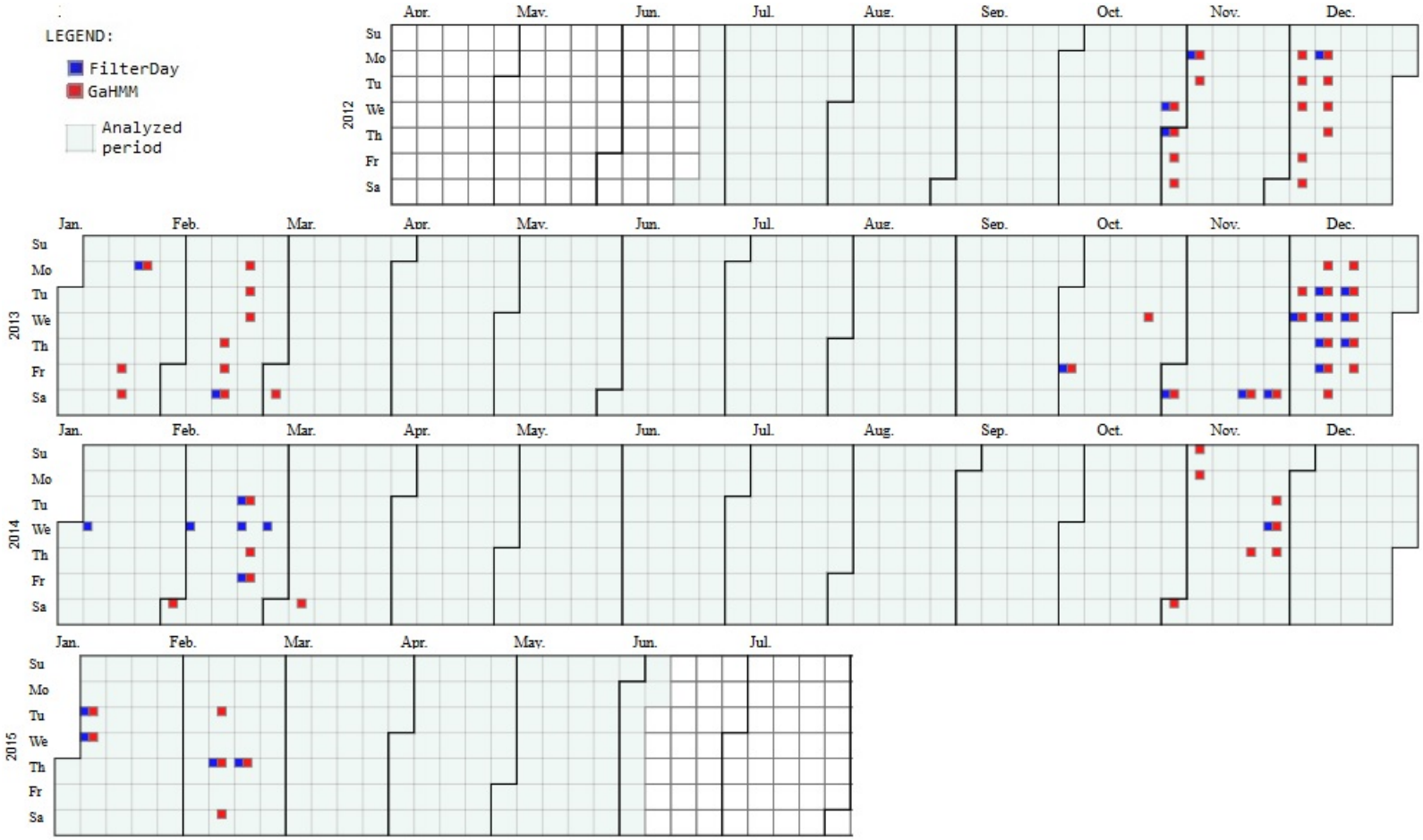


FIGURE B.19: Coincidence FilterDay vs. GaHMM-profile when $W = 4, |A| = 5$. A small square indicates a CO_2 daily profile spotted as discord cluster for the North-East ventilation system.

ID cluster	Date (yyyy-mm-dd)
3	2012-10-31, 2012-11-01, 2012-11-02, 2012-12-04, 2012-12-07, 2012-12-11, 2013-01-18, 2013-02-14, 2013-02-15, 2013-02-19
10	2012-12-13, 2013-12-03, 2013-12-04, 2013-12-09, 2013-12-12, 2013-12-13, 2013-12-16, 2014-02-18, 2014-02-20, 2015-01-06
13	2012-11-05, 2012-12-03, 2012-12-10, 2013-02-18
18	2012-12-05, 2012-12-12, 2013-02-20
26	2012-11-03, 2012-11-06, 2012-12-08, 2013-01-19, 2013-02-16, 2013-03-02, 2013-11-02, 2013-11-23, 2013-12-14, 2014-02-01, 2014-03-08, 2014-11-01, 2014-11-02, 2015-02-14
20	2013-01-21
29	2013-10-04, 2014-02-21
30	2013-11-30
31	2013-12-10, 2013-12-11, 2013-12-17, 2013-12-18, 2013-12-19
33	2013-10-23, 2013-12-20, 2014-11-03, 2014-11-20, 2014-11-25, 2014-11-26, 2014-11-27, 2015-01-07, 2015-02-10, 2015-02-12, 2015-02-19

TABLE B.3: Dates when the CO_2 discord clusters of the North-East ventilation system were spotted by using the *GaHMM-profile model* approach.