TÉCNICO LISBOA

Occupancy Prediction from Electricity Consumption Data in Smart Homes

Davide Fialho Pereira

Thesis to obtain the Master of Science Degree in

Energy Engineering and Management

Supervisors: Prof. Rui Manuel Gameiro de Castro
Eng. Pedro Magalhães Adão

Examination Committee

Chairperson: Prof. Duarte de Mesquita e Sousa
Supervisor: Prof. Rui Manuel Gameiro de Castro
Members of the Committee: Dr. Sílvio Miguel Fragoso Rodrigues
Eng. Pedro Manuel Neves Geirinhas Rocha

June 2017

Acknowledgments

I would first like to thank EDP for providing me the opportunity for developing this thesis in the company. It was a pleasure to contribute for the interests of the company. I wish to acknowledge my two supervisors, Prof. Rui Castro (from Instituto Superior Técnico) and Pedro Adão (from EDP) for their support and guidance during the period of this work. Without them, this work would not be possible.

I would also to thank Prof. Alexandre Bernardino, for the availability in clarifying many doubts that I had with regard to specific topics in the statistics field.

In this work, we collected the necessary data from five households. A special thanks to these five participants who made themselves available in helping me during a period of about five months.

Last but not least, I would like to thank my family and my friends for the support and encouragement that they provided to me during this work, especially in the more difficult periods.

Abstract

The problem of occupancy detection and prediction plays a major role in the smart home era, since can provide benefits with regard to comfort, safety and energy savings to electricity consumers. Many authors have already explored occupancy monitoring and prediction systems, however, very few approached the occupancy detection and prediction by using smart meter data. In this work, we showed that an occupancy detection accuracy up to 92% can be achieved by using solely electricity consumption data.

Also, we address the problem of generalizing a classification model, i.e., we analyze the possibility of using a single classification model to monitor occupancy in multiple households. We found that an occupancy detection accuracy of up to 82% was possible by using a generic classification model.

Regarding occupancy prediction, we showed that it is possible to predict occupancy in multiple households, with an accuracy of up to 78%, by using solely electricity consumption data. This result is not fantastic but we believe that better result can be obtained by: 1) exploring more features from the dataset 2) using a more sophisticated feature selection method and 3) using hybrid prediction approaches, since they combine schedule-based with context-aware approaches, e.g. combine a prediction timetable with information from mobile phones (GPS and wireless networks).

For both occupancy monitoring and prediction, we consider that households with low level of occupancy can benefit more from these systems.

Keywords: Occupancy prediction, occupancy detection, electricity consumption, smart meter, opportunistic sensing

Resumo

Monitorizar e prever a ocupação representa um grande interesse na era das casas inteligentes, pois permite que casas residenciais se beneficiem de mais conforto, segurança e eficiência energética. Muitos autores já exploraram sistemas de monitorização e previsão de ocupação, contudo, muito poucos abordaram a monitorização e previsão de ocupação com base em dados provenientes de contadores inteligentes. Neste trabalho, mostrámos que é possível obter uma acurácia de classificação superior a 92% é possível, utilizando apenas dados de consumo de eletricidade.

Neste trabalho, também abordamos o problema de generalizar modelos de classificação, isto é, analisamos a possibilidade de utilizar um único modelo de classificação para monitorizar ocupação em várias casas residenciais. Verificámos que uma acurácia de deteção de ocupação superior a 82% é possível de obter usando um modelo genérico.

Relativamente à previsão de ocupação, mostrámos que é possível prever ocupação em várias casas residenciais, com uma acurácia superior a 78%, utilizando apenas dados de consumo de eletricidade. Este resultado não é fantástico, mas acreditamos que melhores resultados podem ser obtidos através de: 1) explorar mais características/variáveis do conjunto de dados 2) utilizando um método de seleção de variáveis mais sofisticado e 3) usando abordagens hibridas de previsão, visto que combinam abordagens baseadas em cronograma com abordagens baseadas no contexto, como por exemplo, podem combinar tabelas de probabilidades de ocupação com informação derivada de telemóvel (GPS e de redes sem fio).

Palavras-chave: Previsão de ocupação, deteção de ocupação, consumo de eletricidade, contadores inteligentes, deteção oportunista

Contents

List of tables

List of figures

List of Acronyms

ANN	Artificial Neural Networks
AUC	Area Under the Curve
C	soft margin constant
CPU	Central Processing Unit
CRISP-DM	CRoss Industry Standard Process for Data Mining
ECO	Electricity Consumption and Occupancy
EDP	Energias de Portugal
fn	False negative
FNR	False Negative Rate
fp	False
FPR	False Positive Rate
GMM	Gaussian Mixture Models
GPS	Global Positioning System
GT	Ground Truth
HMM	Hidden Markov Models
HVAC	Heating, Ventilation and Air Conditioning
IoT	Internet of Things
KDD	Knowledge Discovery in Databases
KNN	K-Nearest Neighbor
MCC	Mathews Correlation Coefficient
NIOM	Non-Intrusive Occupancy Monitoring
ODBC	Open Database Connectivity
PCA	Principal Component Analysis
PH	PreHeat
PIR	Passive InfraRed
PLC	Power Line Communication
PPS	Presence Probabilities Simplified
re:dy	remote energy dynamics
RFID	Radio-frequency identification
ROC	Receiver Operating Characteristic
SDP	Service Delivery Platform

SEMMA	Sample, Explore, Modify, Model, and Assess
SFS	Sequential Forward Selection
SVM	Support Vector Machines
THR	Threshold
tn	True negative
TNR	True Negative Rate
tp	True positive
TPR	True Positive Rate

1 Introduction

1.1 Motivation for occupancy detection and prediction

In the era of smart home, achieving a higher energy efficiency, comfort and safety at home are three important factors that households are interested in. With the advance in the statistics field and with the appearance of new fields such as the Internet of Things (IoT) and machine learning, energy utilities have now the possibility of providing these benefits to the customers.

One way of providing these benefits to the customers is by controlling automatically their electrical appliances depending if the house is occupied or not. According to [1], Heating, ventilation and cooling (HVAC) represents the largest source of residential energy consumption in the U.S., Canada and U.K. The same study refers that 20-30% of this energy could be saved by simply turning off the HVAC systems when residents are sleeping or away. However, the author states that these savings have been difficult to realize since households typically do not manually adjust the thermostat several times a day. Also, smart thermostats, such as the NEST thermostat, attempts to solve this problem by automatically programming itself based on occupancy patterns that it learns by a built-in motion sensor [2]. However, the relatively high cost represents a major disadvantage of the NEST thermostat.

According to [3], Non-Intrusive Occupancy Monitoring (NIOM) is possible by using smart meters and allows utilities to determine: 1) how much a programmable thermostat would benefit in each home and 2) to suggest an optimal customized thermostat schedule. In this work, we do not focus on analyzing the savings potential of smart thermostats, but on investigating how viable is it to monitor and predict occupancy by using solely smart meter data.

In addition to a higher energy efficiency, occupancy monitoring also provides more safety to the consumers. For example, if a high electricity consumption is verified in periods that are not supposed, occupancy monitoring systems can be used as an intruder's detector, by sending alarms in real time to the smartphones of the occupants. If we analyze occupancy at the room-level, occupancy monitoring systems can also be used for health monitoring applications. For example, by verifying the occupancy of the rooms (e.g. dispenser or kitchen), the algorithm can infer if a sick people missed his medication in the supposed time or if missed any meal during the day, and remind the caretaker. In this work, we monitor and predict occupancy at a binary level and not at the room-level, i.e., our objective is to detect and predict if a certain household is occupied or not.

Smart meters are already deployed in millions of households worldwide, representing an opportunity for occupancy monitoring systems without any additional cost, in contrast to conventional approaches, such as motion sensors.

The problem of occupancy detection trough smart meter data has not been yet widely investigated in the literature. Relevant works in this area have only explored the viability of detecting occupancy trough smart meter data individually for each household [4], [5], [6]. However, to the best of our knowledge, no

single work analyzed the possibility of using a single and generic classification model to detect occupancy in multiple households. For electricity consumers, this represents a major interest, since they would benefit from occupancy monitoring applications (such as home automation) without the need of monitoring the real occupancy through direct systems (e.g. motion sensors).

Occupancy prediction isn't also a mature topic in the literature. Many works exist but no one, to the best of our knowledge, investigated the possibility of predicting occupancy by using solely the smart meter data.

1.2 Objectives

The present work proposes to investigate the viability of detecting and predicting occupancy by using solely the electricity consumption data, obtained from smart meters. More specifically, we define the following research questions: 1) How accurate can occupancy be monitored through electricity consumption data? 2) Is it possible to use a single classification model to monitor occupancy in multiple households? In which conditions? 3) Is it possible to predict occupancy by using solely electricity consumption data?

To this end, we installed the EDP re:dy service in 5 households, in Portugal, in order to collect both electricity consumption and occupancy data, which is necessary to perform our experiments. This service represents an energy management system for the residential sector, provided by EDP (Energias de Portugal), the largest energy operator in Portugal. EDP re:dy allows customers to visualize their electricity consumption data (both aggregated and device-level) and also provide energy management functions, such as remote control and automation of their equipment's.

1.3 Overview of the work

In terms of occupancy detection, this represents a problem that can be solved through supervised classification algorithms. These algorithms are trained by using input and output information. In this case, the input information represents the electricity consumption data and the output represents the occupancy data. After having a classification model, we use solely the electricity consumption data (input) and we compare the occupancy data generated by the model (output) with the real occupancy data (ground truth occupancy). Then, classification performance metrics are defined to measure the occupancy detection performance. By analyzing similar works, we decided to use the neural network, support vector machines and random forest models as our methods for occupancy detecting/monitoring.

For occupancy prediction, we first investigated the different type of approaches that exist. We found out that context aware, schedule-based and hybrid approaches can be used to predict occupancy. However, we decided to perform our occupancy prediction experiments by using a schedule-based approach, since it is compatible with our type of data. More specifically, we used the Presence Probabilities

Simplified (PPS) algorithm, since it had the best results in a similar study [6], and also because it generates a 7-day timetable with the probabilities of presence, which can be further used for programming a smart thermostat (similar as NEST thermostat do).

1.4 Organization of the document

This work is organized as follows:

- Chapter 2: we start by investigating the existent occupancy monitoring methods, which can be direct (e.g. motion sensors) or indirect (e.g. from smart meter data). Then, we focus on exploring occupancy monitoring approaches through electricity consumption data (using smart meters). In this section, we investigate the relation between the electricity consumption and occupancy and the best classification algorithms used in similar works. Regarding occupancy prediction, we start by exploring the different approaches existent. Then, we focus on explaining the prediction algorithms that are suitable to be applied in this work.
- Chapter 3: we start by presenting the architecture that we have adopted to answer our research questions, from a data mining perspective. Then, we explain each of our three classification algorithms chosen (neural networks, support vector machines and random forest) and also the metrics used for evaluating the classification performance. Finally, we explain the chosen prediction algorithm for predicting occupancy (Presence Probabilities Simplified) and also the metrics used in evaluating the prediction performance.
- Chapter 4: in this chapter, we explain how we have prepared our data before performing the experiments. We start by explaining the data collection process and its measurement infrastructure. Then, we explain the data cleaning and pre-processing stage. Finally, we describe the features that we extracted from the collected data and the method that we used in their scaling.
- Chapter 5: we perform the experiments that we consider necessary to answer our research questions. We start by analyzing the viability of detecting occupancy in households by using solely the electricity consumption data. Then, we analyze the possibility of using a single and generic classification model to detect occupancy in multiple households. In the second part, we investigated the possibility of predicting household's occupancy by using a 7-day prediction timetable.
- Chapter 6: we present the conclusions of our work and recommendations for further work.

2 Related work

In this chapter, relevant works with regard to occupancy monitoring and prediction are reviewed. We start by explaining conventional methods to monitor occupancy, which can be direct (intrusive) or indirect (non-intrusive), and the advantages and disadvantages of each. Then, we focus on reviewing works that monitor/detect occupancy based on the electricity consumption data. In this section, we also investigate the typical features extracted from electricity consumption data to monitor occupancy, the methods used to reduce the high dimensionality of the classification problems and the different types of classifiers used by other authors.

Finally, we investigate state of the art approaches to predict occupancy, focusing on the ones with higher potential to be applied in this work.

2.1 Occupancy monitoring

In literature, many ways of detecting and monitoring occupancy have been already explored. These techniques can be divided into direct and indirect methods. Direct approaches require physical devices to be installed in the buildings while indirect approaches use existent information to detect occupancy.

2.1.1 Direct methods

In direct methods, occupancy is monitored by using different types of sensors, such as motion, door, acoustic, camera, contact and CO2 sensors [3]. The most common direct methods to detect occupancy are based on Passive Infrared (PIR) and ultrasonic technologies, microwave and sound sensors, video cameras and Radio-frequency identification (RFID). There isn't a good solution for every application and the number and type of sensors typically depends on the trade-off between desired occupancy detection accuracy, the overall cost and the complexity of the system [5]. For instance, PIR sensors (motion sensor) are frequently used to control the switching of lighting systems by detecting human body movement. A relevant drawback of motion sensors is that they require movement to detect occupancy and if the occupant is inactive for a while, this may lead to uncomfortable situations, e.g., by switching off the lighting system. For this reason, motion sensors are typically used in conjunction with other sensors, such as audible sound sensors, to improve the occupancy detection efficiency. However, sound sensors cannot distinguish between human and non-human noises and may detect false occupancy [7].

Regarding video cameras and RFID systems they are mostly used to detect occupancy for security purposes. These techniques are typically avoided in building control systems applications since they suffer from privacy concerns [7].

In [8], PIR sensors were used in conjunction with magnetic reed switch door sensors (Synergy Node), as illustrated in Figure 1, to detect occupancy in individual offices. The occupancy information was then used to control a HVAC (Heating, ventilation and air conditioning) system and the authors shown potential energy savings between 10% and 15%. The reed switch senses if the door is open or closed while PIR sensors detects movement. With this combination it is possible to detect not only human movement but also to detect if any user entered/exited the house.

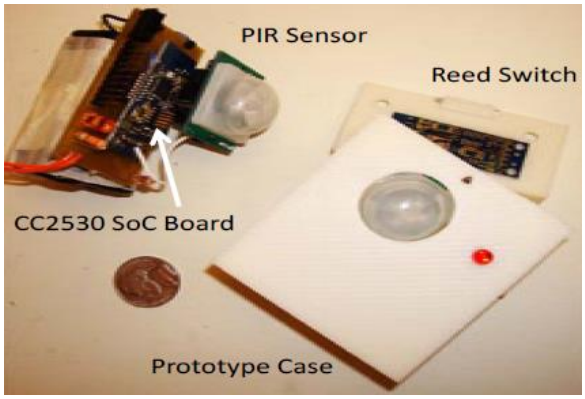


Figure 1: Synergy node. PIR sensor and magnetic reed switch used to detect occupancy [8].

From Figure 2 it is possible to observe that the combination of the two sensors provide higher occupancy detection accuracy than the situation with PIR sensors only.

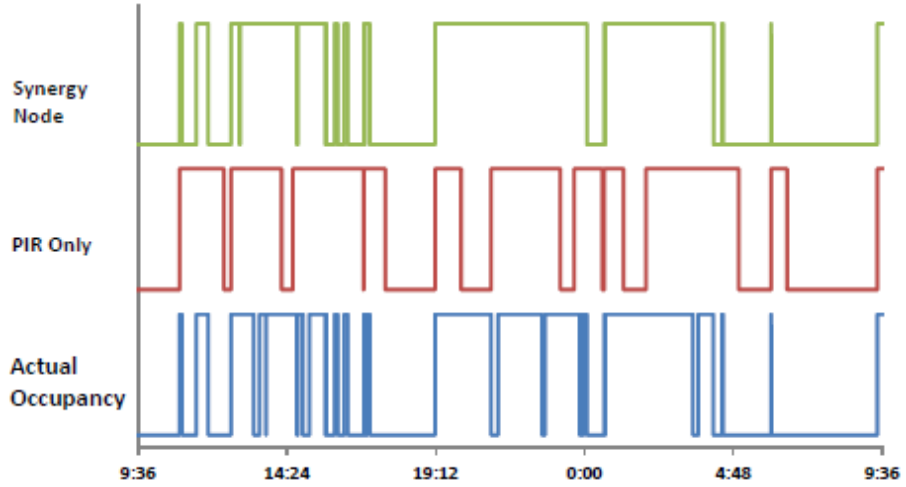


Figure 2: Comparison between occupancy detection from the Synergy Node (PIR sensor and magnetic reed switch) and PIR only with respect to the actual occupancy [8].

Despite the combination of multiple sensors providing a higher occupancy detection accuracy, the higher cost and complexity of the system often do not compensate. For this reason, in the residential sector, smart thermostats often include only one PIR sensor [4]. This reduces the accuracy of the occupancy sensing and, as a result, erroneous control decisions may be made. According to [2], users of smart thermostats based on motion sensors often turn off its automatic functions due to the discomfort caused by incorrect decisions of the thermostat.

Direct approaches to detect occupancy suffer from many other drawbacks. It is necessary to purchase sensors, install, calibrate, power and maintain them. Battery-powered sensors are often used to avoid the need of power cables, however, the cost of maintain the battery have to be considered and the sensor performance may be affected by the battery discharge. Faulty installations and lack of maintenance also degrades the performance of the sensors [4], [7]. Motion sensors also have to be carefully placed and calibrated in order to avoid detecting undesired movements, e.g., from pets or from events outside the windows [3].

These limitations motivate researchers to look for indirect approaches to detect occupancy, since they do not suffer from these problems.

2.1.2 Indirect methods

Indirect methods to detect occupancy are cheaper and less intrusive than direct methods and uses contextual information sources, such as wired and wireless network traffic and online sources (e.g. calendars and chat applications). These techniques monitor occupancy by correlating one or more user activities with occupancy [3].

In [9] it was analyzed, through a pilot work, the effect of using the data of Global Positioning System (GPS) from mobile phones of occupants to control a traditional thermostat. This GPS data was used to estimate the arrival time of the occupants and this information was used to notify the thermostat when it should start to pre-heat so that the house achieve the comfortable temperature when the user arrives. This study showed that 7% of savings can be obtained by controlling thermostats with GPS information and refers that a potential for 50% of savings is possible for U.S. households that do not change their thermostats temperature settings when the house is unoccupied. However, the major drawback of this approach is that estimating the arrival time of occupants may be prone to errors, since it depends on the unpredictable road and traffic conditions. The loss of connectivity or the drain of battery of the mobile phone also affects the performance of the system.

Another indirect way to monitor occupancy is thought the electricity consumption data obtained by smart electric meters, as already addressed by many authors [3], [4] and [5]. For instance, in [5] the authors achieved an occupancy detection accuracy of up to 94% using features/variables extracted by the electric consumption data, such as the mean power and the standard deviation of the mean power.

It is also possible to detect occupancy from electrical events (e.g. detecting which appliance have been turned on or off), however, this approach requires extensive, specialized, and high-calibrated equipment to record and analyze the high-frequency data (more than 2kHz). On the other side, smart meters can also be used to detect occupancy, since they can provide power consumption data in every few seconds or every few minutes both to the utility and to the customers.

According to [3], by 2011, about 493 utilities in the United States had installed more than 37 million of smart meters in the country. In 2015, approximately 45 million of smart meters were installed in Finland,

Italy and Sweden. In Germany, the installation of smart meters is now mandatory for all new and renovated buildings and the European Commission estimates an amount close to 200 million of smart meters by 2020 in its Member States [6]. Thus, because smart meters are already widely deployed and their installation, use and maintenance do not represent an additional cost to the residents, they represent an opportunity for both utilities and households to benefit [5]. For households, this information could be used to detect and predict occupancy for systems automation applications. For utilities, statistical models could be constructed over this data to predict future electricity consumption or to model daily routines to improve the energy supply management of energy providers (e.g. load shaping functions) [4], [10].

This works focuses on detecting and predicting household's occupancy solely through the electricity consumption data obtained by smart meters.

2.2 Occupancy and electricity consumption data

In general, building occupancy detection isn't yet a mature and efficient process. As shown before, the most common used techniques are based on sensors, which represents an extra cost for the residents and are prone to errors. Detecting buildings occupancy from electric consumption data is even a less mature process and there isn't an extensive research in this field. Existing works focuses mostly on the residential and commercial sector [11] with the purpose of controlling HVAC and lighting systems, which represents the systems with higher potential for savings [2].

In [12], three homes were instrumented with smart meters and data was collected every second during two months. They told to the participants to register which appliances they have used at what time. It was demonstrated that smart meters have the potential to provide relevant information about the households, such as: how many people are in the home, sleeping routines and eating routines. The major limitation of this study is that these conclusions were obtained by visual inspection of the electric load curves.

In [3], motivated by the need for more efficient energetic systems and dissatisfied with the drawbacks of direct methods to monitor occupancy (e.g. sensors), the authors decided to analyze the viability of using smart meters to perform a non-intrusive occupancy monitoring. They instrumented two homes with smart meters and collected power consumption data in a summer week. Ground truth information (real occupancy) was collected through the interaction of the occupants with electrical loads (that imply occupancy) and via smartphones (through GPS). Three statistical metrics were extracted from the data: average power, standard deviation of the power and the power range. Then, a simple threshold-based method was applied to infer occupancy from the aggregate electricity consumption. The maximum value of each metric at night was used as the threshold value to classify the occupancy state for the next day. Whenever one or more metrics are above its threshold during daytime, the binary classifier assumes the house as occupied.

Despite the simplicity of this study, the authors concluded that the algorithm performed well. However, this approach may not be feasible in households that have appliances with patterns of high consumptions at night (e.g., electric water heater) since the threshold would be higher and lead to a worst classification. Furthermore, authors refer that machine learning algorithms would perform better than the threshold-based method.

2.2.1 Datasets containing both electricity consumption and occupancy (ECO) data

As shown before, to perform a quantitative analysis of the possibility to detect occupancy from electricity consumption data it is important to have an extensive dataset containing both electricity consumption and occupancy (ECO) data.

Due to the lack of these datasets, authors of [4] decided to perform an extensive ECO data collection in 5 households in Switzerland during 8 months. In addition to the aggregate electricity consumption data, this dataset also contains data from PIR sensors and smart plugs. Ground truth occupancy data was obtained through a tablet computer. This was a preliminary study that used standard classification techniques to evaluate the occupancy detection accuracy with information provided only by smart meters. In this work, 10 features were extracted from the dataset and 4 classification models were used. The study showed that occupancy detection accuracies over 80% are feasible in most of the scenarios.

The same author performed an improvement of this preliminary work [5] by presenting a more detailed analysis of supervised machine learning methods to detect occupancy from electricity consumption data. Instead of 10, 35 features were extracted from the dataset and 7 classification models were used. Also, two methods were used to reduce the higher dimensionality of the feature set. An occupancy detection accuracy up to 94% was obtained, however, the author states showed that there isn't an ideal feature set that performs always well over all households.

2.2.2 Features deriving from the electric load curve

The choice of the right features plays and important role for the performance of any classifier. For this reason, it is important to understand which features are relevant and may have a high correlation with occupancy.

According to [4], when a household is occupied, its electricity consumption is likely to have a higher power mean and standard deviation. This conclusion is easily understandable since when occupants are the home, typically, they interact with electrical devices such as TV's, lights and kitchen appliances.

The authors analyzed, in one household, the relative frequencies of various power consumptions over one day and divided into two graphs (when occupants are at home and when they are away). It is

possible to observe that when the house is unoccupied, the electricity consumption is centered around 100W with a low variance. When occupants are at home, it is possible to observe the higher power consumption and standard deviation, as can be seen in Figure 3.

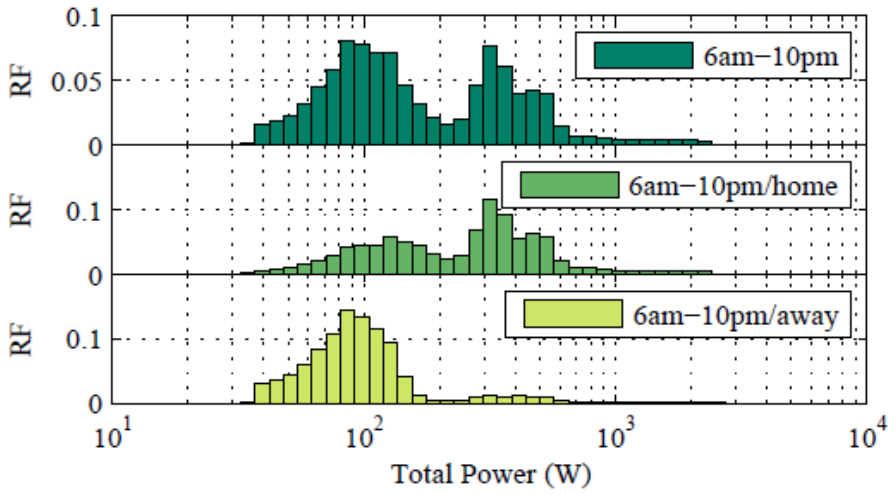


Figure 3: Relative frequencies of the various power consumption, divided into presence and absence periods, respectively [4].

In addition to the power mean and standard deviation, authors included a third feature, the sum of absolute differences. This feature was added in order to provide another measure of power variability and is calculated by summing the absolute differences between adjacent power measurements.

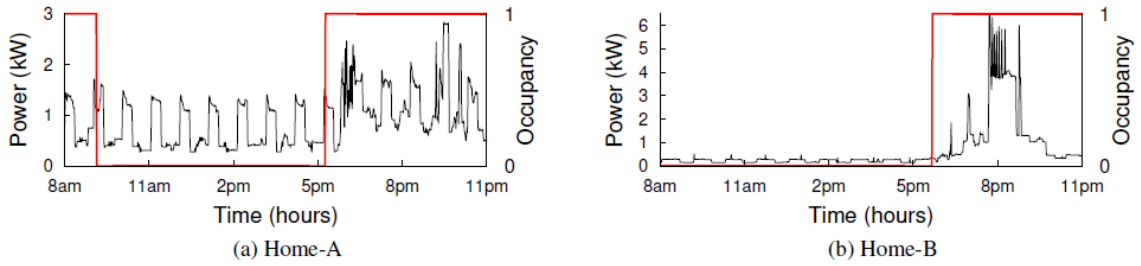


Figure 4: Average power consumption (black) and binary occupancy (red) values for two homes [3].

In [3], it was overlaid the average power consumption every minute (black line) with binary occupancy of the house (red line). Occupancy vector is a binary value where 1 indicates that at least one person is in the house and 0 indicate that the house is unoccupied. This illustration is shown in Figure 4. From this figure, the authors observed that an occupied home has a higher average power, higher variability in the power consumption and a large absolute range in power consumption. Therefore, they considered three features in their threshold-based method: average power, standard deviation and power range, as similarly with [13].

From Figure 4 it is also possible to observe that different households may have different patterns of electricity consumption and thus stronger or weaker correlations with occupancy. For instance, Home-

A has multiple air conditioning units and high-power background loads than Home-B. Consequently, Home-A has a weaker correlation between electricity consumption and occupancy.

In [5], 35 features were grouped into three different characteristics that may be correlated with occupancy, which are: absolute value of power, power variability and temporal dependence of occupancy. These features are described in Figure 5.

#	Feature names	Description
f_1, f_2, f_3	min_1, min_2, min_3	Minimum of the samples for phase 1, 2 and 3
f_4	min_{123}	Minimum of the samples for the sum of phase 1, 2 and 3
f_5, f_6, f_7	max_1, max_2, max_3	Maximum of the samples for phase 1, 2 and 3
f_8	max_{123}	Maximum of the samples for the sum of phase 1, 2 and 3
f_9, f_{10}, f_{11}	$mean_1, mean_2, mean_3$	Arithmetic average of the samples for phase 1, 2 and 3
f_{12}	$mean_{123}$	Arithmetic average of the samples for the sum of phase 1, 2 and 3
f_{13}, f_{14}, f_{15}	std_1, std_2, std_3	Standard deviation of the samples for phase 1, 2 and 3
f_{16}	std_{123}	Standard deviation of the samples for the sum of phase 1, 2 and 3
f_{17}, f_{18}, f_{19}	sad_1, sad_2, sad_3	Sum of absolute differences of the samples for phase 1, 2 and 3
f_{20}	sad_{123}	Sum of absolute differences of the samples for the sum of phase 1, 2 and 3
f_{21}, f_{22}, f_{23}	$corl_1, corl_2, corl_3$	Autocorrelation at lag 1 computed over the samples for phase 1, 2 and 3
f_{24}	$corl_{123}$	Autocorrelation at lag 1 computed over the samples for the sum of phase 1, 2 and 3
f_{25}, f_{26}, f_{27}	$onoff_1, onoff_2, onoff_3$	Number of detected on/off events for phase 1, 2 and 3
f_{28}	$onoff_{123}$	Number of detected on/off events for the sum of phase 1, 2 and 3
f_{29}, f_{30}, f_{31}	$range_1, range_2, range_3$	Range of the samples for phase 1, 2 and 3
f_{32}	$range_{123}$	Range of the samples for the sum of phase 1, 2 and 3
f_{33}	p_{prob}	Empirical probability of the slot to be occupied
f_{34}	p_{fixed}	1 (occupied) from 9 a.m. to 5 p.m., 0 (unoccupied) otherwise
f_{35}	p_{time}	Slot number (*i.e.* 1 – 65)

Figure 5: Features computed in [5] for occupancy monitoring. All features were computed over 15-minute intervals.

Absolute value of the power consumption

min, *max* and *mean* features were extracted to measure the absolute value of power consumption and represent, respectively, the minimum, maximum and average power consumption in each slot (15 minutes' interval).

Variability of the power consumption

Because a high power consumption variability may indicate human presence, due to their interactions with appliances, the following features were extracted to detect this variability: *std* (standard deviation of the power consumption), *sad* (sum of absolute differences), *cor1* (autocorrelation at lag one), *range* (difference between the maximum and minimum power in one interval), and *onoff*. The feature *onoff* contains on/off events that occur when an appliance is switched on or off.

Temporal dependence of occupancy

The same authors defend that building occupancy is also dependent upon the current time of the day. This correlation is easily understandable since typically each occupant has activities with specific and constant schedules (e.g., some occupants may spend the day in the work or school). Thus, authors considered three features to model the temporal aspects of occupancy, which are: p_{prob}, p_{fixed}, p_{time}. p_{prob} represents the empirical probability of one slot (of 15 minutes) to be occupied. p_{fixed} was defined as "dummy" prior probability that assumes that the house is unoccupied between 9 a.m. and 5 p.m. on weekdays and to be always occupied during the weekends. p_{time} is defined as the number of the time slot in the day, introducing a notion of time to the classifier. In this work, the classification analysis was

performed between 6:00h. and 20:15h p.m., thus, p_{time} represents the 65 intervals of 15 minutes between the considered period.

2.2.3 Device-level electricity consumption

According to [6], the consumption of electrical devices is a better indicator of occupancy than the aggregated electricity consumption. This conclusion can be inferred intuitively since there are appliances that are only activated by the occupants and, when consuming electricity, indicate with high precision that someone is in the house. These appliances are also considered as interactive loads since requires physical interaction to be activated and some examples are: TV's, electric stove, kettle, toaster, coffee machine, etc.

On the other hand, there are appliances that have a weak correlation with occupancy, such as: electric water heaters, heat pumps, washing machines and fridges. These appliances have specific consumption patterns which are highly independent with the human presence in the house. For instance, an electric water heater typically consumes energy every few minutes/hours to keep the water temperature above a certain value, as similarly with the fridge. These appliances can be considered as background loads and have a low correlation with occupancy.

In [3], the authors analyzed the correlation between the appliances consumption and occupancy for one household. More specifically, they analyzed the correlation between background loads and interactive loads in order to prove that interactive loads consumption is a good indicator of occupancy, as shown in Figure 6. The red circles in Figure 6-b represent the physical interaction of occupants with electrical loads.

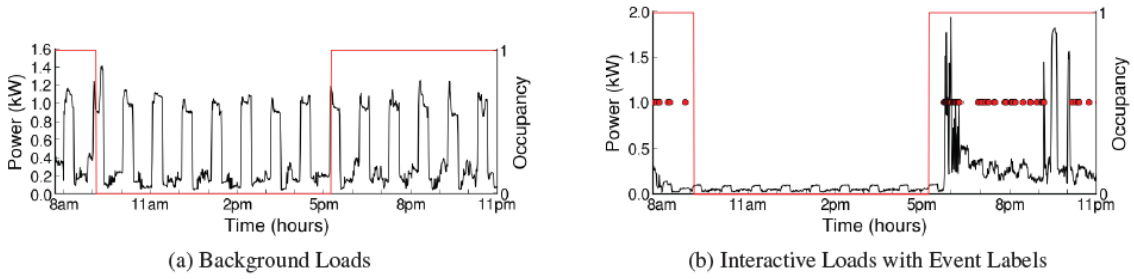


Figure 6: Power usage from background loads (a) and interactive loads (b) from one home [3]. In (b) it is included the event label, which corresponds to the periods in which occupants interacted with electrical loads.

Also, one appliance may be a good indicator of occupancy in one household and a bad indicator in other. For instance, if a washing machine is activated by humans it would have stronger correlation with occupancy rather than if it would be activated by an automatic scheduler.

2.2.4 Reducing the high dimensionality of the feature set

Most of the existent work in this field extract few features and uses them all in the classification process [4], [3], [7] and [13]. However, when dealing with high dimensionality data, some classifiers may not perform well. Thus, it is important to use a technique to reduce the high dimensionality of the data. In [5] it was analyzed two methods with the objective of selecting the most describing features of the data: sequential forward selection (SFS) and principal component analysis (PCA).

Sequential forward selection (SFS) is a direct and nonparametric method to determine the best subset of *d* measurements out of a set of *D* total measurements. In the first iteration it is found the best feature that maximizes the performance of the algorithm. At each iteration it is included one more feature that maximizes the performance. The process stops until all the features have been selected or a number of *d* features have been reached [14].

In [5], 35 features were used in the classification process, as shown in the Figure 5. By using SFS, authors tested the features that best describe occupancy by counting the number of times that a specific feature has been chosen, as part of the feature subset for a certain household and classifier. In Figure 7, it is shown that, for household 2, the *onoff* feature is the most frequently chosen in both summer and winter. Thus, this is the best describing feature of occupancy.

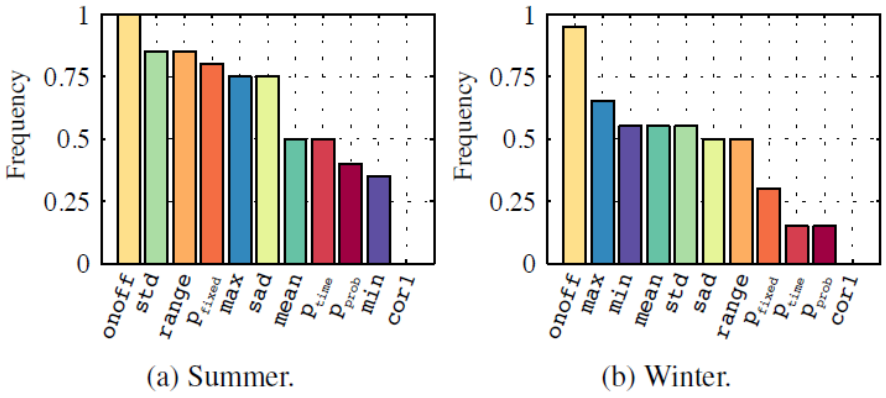


Figure 7: Frequency of the features chosen by Sequential forward selection (SFS) algorithm for a certain household and classification method [5].

An important conclusion of the study is that no single feature is chosen consistently over all households, due to the high correlation between individual features. For instance, the range feature represents the absolute difference between max and min features. The features *onoff* counts the number of switching's on/off of the devices in an interval while the feature *sad* is a measure of power variability, making these features highly correlated between them.

Regarding principal component analysis (PCA), it represents an orthogonal transformation that transforms a set of features possible correlated into a set of linear combination (uncorrelated variables called principal components). Principal components became ordered from the most correlated with occupancy to the less and usually, the first few components account for most of the variance in the input

data. In this study, the author restricted the number of components to the first *L* components that account for more than 95% of the variance of the input data [5]. Authors obtained a better performance by using PCA rather than SFS and they refer that the high redundancy (or correlation) between some features make difficult for SFS to choose the best subset of features.

Another way to select the optimal feature set is through a brute-force evolution of all possible combinations, however, this process complexity grows exponentially with the number of features [5].

2.2.5 Non-intrusive load monitoring

As explained before, the device-level electricity consumption represents a good indicator of occupancy, specially of the interactive loads. Thus, the importance of measuring this consumption has been growing in the literature and many works has already focused on this field.

There are two ways to measure this consumption. The first way consists on using physical devices, smart plugs, that measures the device consumption. The second way is through a non-intrusive load monitoring (NILM) algorithm.

NILM is an algorithm that aims to disaggregate the aggregate electricity consumption to obtain the electricity consumption of the appliances.

One of the first studies in this area is presented in [15]. In this work, the author identified the step changes in the electricity consumption when a certain appliance is switched on/off. Then, he compared this step changes with the ones previously recorded in a signature database. The author concluded that it is possible to detect when appliances are being switched on or off.

In [16], Gupta *et al.* showed that NILM allows to identify and classify most consumer electronic and fluorescent lighting devices correctly with a mean accuracy of 93,82%, however, special hardware is required to measure the electricity at multiple kilohertz, making it an unfeasible process due to the higher complexity and costs [3].

According to [17], the frequency of the data is the most important factor to identify correctly those appliances. The author states that, hourly data typically identifies three end-user's categories, such as: loads that correlate with outdoor temperature, continues loads and loads that depend on the time (e.g. pool pumps and outdoor lighting). Data frequency from one minute to one second (1 Hertz) allows to identify 8 appliances types (e.g. Refrigerator, Heaters, Washers and dryers), data in multiple kHz of frequency identify between 20 to 40 appliance types. Finally, data in the MHz frequency range has the potential to identify close to 100 distinct appliances, such as different types of lights.

2.2.6 Classification algorithms

The inference of occupancy through the electricity consumption data represents a supervised classification problem. Supervised classification is a machine learning technique typically used for pattern recognition. A supervised machine learning algorithm is an algorithm that required label data to learn the patterns and recognize. In this work, the label data represents the ground truth occupancy.

Classification can be divided into two groups: binary (when the output has two classes, 1 or 0) and multi-class (when the output has more than two classes). In the literature, most of the related works focuses on monitoring/predicting occupancy as a binary classification (the output is occupied or not occupied) [3], [4], [5], [13], and different models have been used.

In [5], the following models were used: support vector machine (SVM), K-nearest neighbor (KNN), Gaussian mixture models (GMM), hidden Markov models (HMM) and thresholding (THR) for monitoring occupancy in five households.

According to [7], KNN and SVM are two state of the art algorithms and SVM is an efficient classification algorithm widely used for pattern recognition for two reasons: has a high ability to generate nonlinear decision boundaries (through kernel methods) and gives a large margin boundary in the classifier. In the same study, HMM was chosen in order to analyze the temporal dependence of occupancy. Most of the models require a parameter fitting in order to find the optimal parameters that maximizes the classification result.

The process of fitting the model parameters is performed in the training phase. In this phase, typically the whole dataset in divided in two parts: *training set* and testing set. The *training set* is used to perform cross-validations with the aim of finding the optimal feature set, optimal model parameters and also to avoid overfitting, and the testing set is used to evaluate the model [5].

Other models have been used in similar works with good results, such as neural networks [18] and random forest [13]. A review of classification techniques was performed in [19].

2.3 Occupancy prediction algorithms

2.3.1 Existent approaches

As it was already explained, predicting human presence can provide many benefits to electricity consumers. Many works focus on predicting occupancy for HVAC controlling in households, such as thermostats, due to the higher savings potential.

Occupancy prediction algorithms predict occupancy in two possible ways: binary level and occupancy level. Predicting the occupancy level consists on predicting how many occupants are present in a building and is typically done in office building. Predicting binary occupancy consists on predicting whenever a building is occupied or not, and represents the most commonly used approach in residential systems [6].

There are three types of binary occupancy prediction algorithms, which are: schedule-based, context-aware and hybrid algorithms. Schedule-based approaches predict occupancy by using solely the historical occupancy data of a building. Context-aware approaches use the information about the current position, activity and environmental factors (e.g. current traffic conditions) to predict the arrival time of each occupant. Hybrid approaches predict occupancy by combining schedule-based and context-aware algorithms.

The present work focus on predicting occupancy using schedule-based approaches, since we do not have data about the current context or activity of each occupant.

2.3.2 Schedule-based approaches

Schedule-based approaches can be divided in two categories. The first detect routines in the historical occupancy schedules and the second assumes that routines can be explained by daily or weekly timetable (i.e., depends with the day of the week and the time of the day). In [6], several state-of-art schedule-based algorithms were analyzed and the study concluded that, for their occupancy dataset, the Presence Probabilities (PP), Presence Probabilities Simplified (PPS) and PreHeat (PH) algorithms provided the best results.

2.3.2.1 PreHeat

In [20] it is presented the PreHeat (PH) algorithm, which is a schedule-based approach that predicts the future occupancy by analyzing the occupants' routines and finding the most similar historical patterns. In this study, five houses (3 in U.S. and 2 in U.K.) were used to analyze the benefit of controlling automatically home heating systems. Authors concluded that PreHeat algorithm allows to obtain a higher home heating efficiency (between 8% and 12% of savings in gas usage) while removing the necessity for users to program their thermostats.

In this study, occupancy was detected by using RFID tags and motion sensors. RFID tags were placed on the house keys of each household while motion sensors were placed in rooms to detect human movement. Occupancy was sensed and PreHeat algorithm was used to predict the arrival time of the occupants. This information is then used to inform the home heating system the ideal time to start heating.

Space occupancy was defined as a binary vector, which is 1 if the house is occupied and 0 if it is unoccupied. As the day progress, the algorithm maintains a partial occupancy vector from the midnight up to the current time and predicts the future occupancy by finding similar days in the past. To find similar days, authors used the Hamming distance between the current partial day and the corresponding parts of past days. The Hamming distance counts the number of unequal corresponding binary elements between the current partial day and the past days (previous five days in this case). Then, occupancy probability is predicted for the remainder of the day by simply computing the average of the past binary occupancies of the most similar five days.

Figure 8 illustrates the PreHeat algorithm. Each vertical bar represents each day occupancy vector, divided in 15-minute intervals.

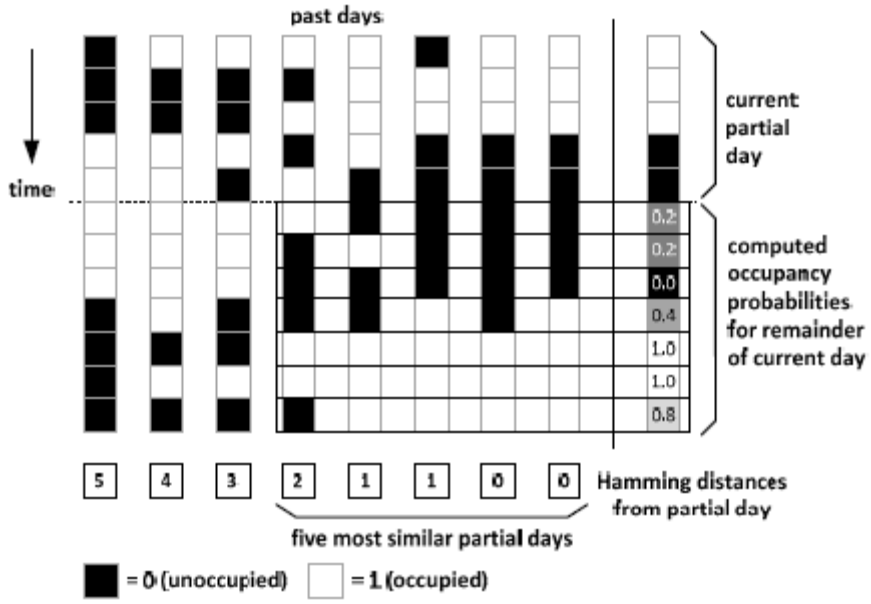


Figure 8: PreHeat algorithm. Each vertical bar represents one-day occupancy data divided by 15-minutes intervals [20].

Because PreHeat computes occupancy probabilities it is necessary to define a threshold value to predict the class occupied or unoccupied. A threshold of 0.5 was chosen in this study [20] and also in a similar work [6], [20] since typically it represents a good tradeoff between the true positives (correctly predicting occupancy) and false positives (predicting the house as occupied when it is unoccupied).

If occupancy prediction is used to control home heating, predicting an occupied house as unoccupied may cause discomfort, while predicting an unoccupied house as occupied may reduce the energy efficiency. The tradeoff between false positives and true negatives, for the 90 minutes into the future and using PreHeat, was analyzed in [20]. Figure 9 shows the Receiver Operating Characteristic (ROC) curves for the 5 households (US stands for households in United States while UK for households in United Kingdom). Each line contains one circle that represents the point where threshold is equal to 0.5.

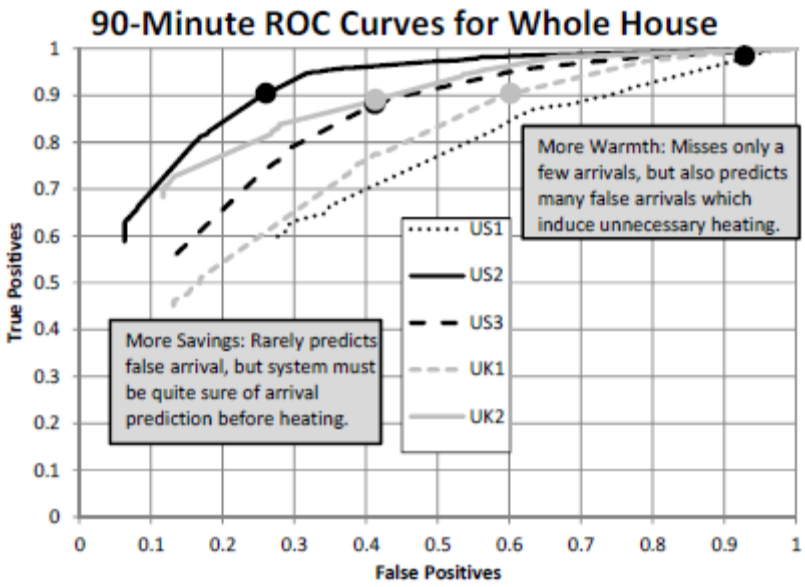


Figure 9: ROC curves for the 5 households analyzed in [20], showing that it is possible to adjust the prediction errors by varying the threshold.

The ideal point of a ROC curve would be the upper left corner where the true positive rate is 1 and the false positive rate is 0, however, this point may not exist in real problems. Thus, it is important to define the threshold according to the end-user objective. In home heating applications, if the user is more concerned about energy savings than comfort, the system should be very confident in predicting occupancy before start heating (a higher threshold should be chosen to minimize the false positive rate). On the other side, If the user is more concerned about comfort rather than energy savings, it is more important to have a high true positive rate rather than a low false negative rate and a lower threshold should be defined.

2.3.2.2 Presence Probabilities

Another schedule-based approach to predict occupancy is the Presence probabilities (PP) algorithm, presented in [21]. While PreHeat algorithm uses the current and historical occupancy data to detect routines and predict future occupancy, PP algorithm uses only historical occupancy data to build a 7-day timetable of occupancy.

In the same study, occupancy was detected in 11 households by using GPS data. Each participant carried a GPS device and the house is considered as occupied if any residence is less than 100 meters away from the house. By using the past occupancy data, PP algorithm computes the probability of the house being unoccupied, p_{away}, during any 30-minute period of a day of the week. Thus, p_{away} is a vector with 336 elements (7 days per week and 48 slots per day) and its values are smoothed using the values of previous and subsequent slot values. In order to adjust the values of p_{away} for the weekdays,

the vector $p_{genericweekday}$ was included and represents a generic weekday, i.e., the average values of p_{away} during weekdays. By using a regularization factor, λ_{wd}, this vector allows to obtain more or less occupancy variability on weekdays.

The probabilistic schedule vector is then computed as the sum of the elements of p_{week} and the relevant elements of $p_{genericweekday}$.

Figure 10 shows the 7-day timetable with the probability of the house being unoccupied between 7AM and 6PM. From this figure it is possible to observe that weekdays have similar probabilities of occupancy and that the house is typically more occupied during the weekend days.

Time of Day	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
7:00 AM	0.000	1.000	0.812	0.427	0.170	0.962	0.000
7:30 AM	0.000	1.000	0.934	0.583	0.461	0.964	0.000
8:00 AM	0.000	1.000	0.999	0.649	0.565	0.875	0.000
8:30 AM	0.000	0.833	0.294	0.797	0.587	0.875	0.000
9:00 AM	0.000	0.857	0.091	0.560	0.379	0.810	0.182
9:30 AM	0.000	0.857	0.200	0.546	0.090	0.714	0.200
10:00 AM	0.149	0.993	0.443	0.429	0.000	0.514	0.200
10:30 AM	0.376	1.000	0.833	0.637	0.341	0.571	0.011
11:00 AM	0.600	1.000	0.833	0.804	0.571	0.571	0.101
11:30 AM	0.567	1.000	0.714	0.625	0.400	0.574	0.189
12:00 PM	0.383	1.000	0.714	0.581	0.400	0.541	0.368
12:30 PM	0.400	1.000	0.714	0.714	0.703	0.400	0.375
1:00 PM	0.388	1.000	0.714	0.714	0.750	0.500	0.348
1:30 PM	0.376	1.000	0.714	0.714	0.750	0.352	0.287
2:00 PM	0.400	0.985	0.714	0.667	0.750	0.310	0.345
2:30 PM	0.721	1.000	0.714	0.667	0.714	0.315	0.143
3:00 PM	0.750	0.897	0.667	0.729	0.714	0.250	0.208
3:30 PM	0.600	0.500	0.650	0.712	0.559	0.328	0.427
4:00 PM	0.600	0.600	0.571	0.440	0.498	0.250	0.375
4:30 PM	0.600	0.368	0.709	0.336	0.429	0.151	0.148
5:00 PM	0.600	0.200	0.612	0.251	0.519	0.142	0.125
5:30 PM	0.595	0.314	0.429	0.375	0.506	0.125	0.143
6:00 PM	0.333	0.500	0.510	0.599	0.571	0.125	0.000

Figure 10: 7-day table containing the probability of a house being unoccupied in any time of the day and any day of the week, computed by Presence Probabilities (PP) algorithm [21].

As similar to the Presence Probabilities (PP) algorithm, the Presence Probabilities Simplified (PPS) algorithm predicts occupancy by using historical occupancy data. The difference is that it doesn't consider the smoothing and the generic weekday schedule as PP.

To compute the prediction accuracy, it is necessary to define a threshold for the occupancy prediction probabilities. As shown before in the PreHeat algorithm, the decision of the threshold is a tradeoff between the true positive and the false positive rate. In the examples showed, the authors consider a threshold of 0.5, meaning that the cost of a false positive is the same as the cost of a false negative. In

this work [21], instead of predicting presence, authors are predicting the state *away* and the choice of the threshold was made in order to obtain an equal error rate (equal false positive and false negative rate).

In Figure 11 it is shown the ROC curve of one home using the PP algorithm. The diagonal line intercepts the ROC curve at the equal error rate.

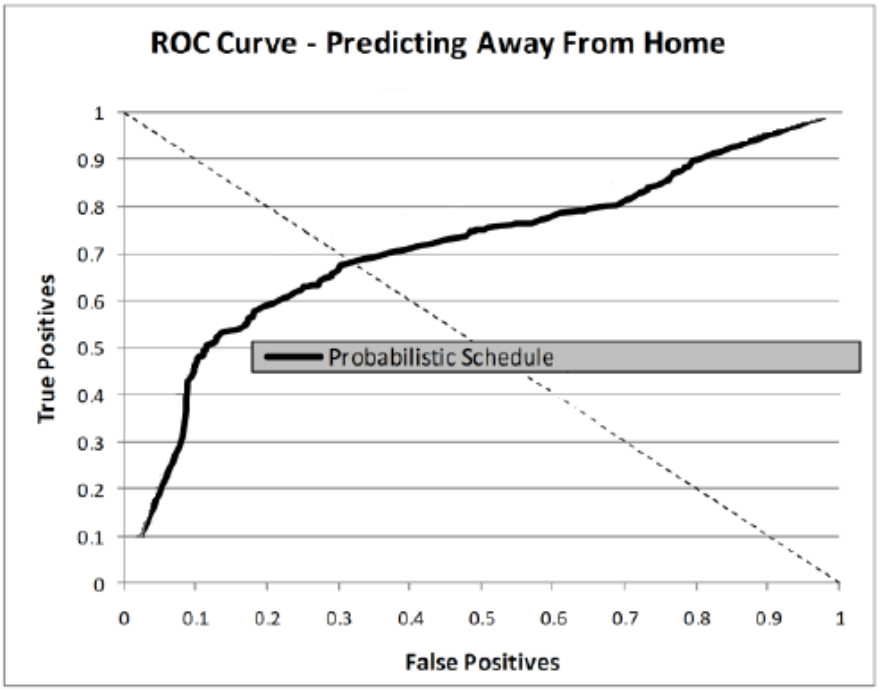


Figure 11: ROC curve for predicting away from home. The diagonal line represents the equal error rate points.

To analyze the performance of the PP algorithm, Figure 12 shows a table with the confusion matrix at the equal error point. It is possible to see that, when the algorithm infers that the occupants are at home, 64% of the times it predicts correctly. When it infers that the occupants are away, 65% of the times if predicts correctly.

		Inferred	
		home	away
Actual from GPS	home	64%	36%
	away	35%	65%

Figure 12: Confusion matrix containing the probability accuracies for the PP algorithm.

In [6], a similar work was performed and a median prediction accuracy of 85% were obtained for PP and PPS algorithms and 80% for the PreHeat algorithm. According to the same study, schedule-based algorithms for occupancy prediction are limited to the accuracy of 90%, since it relies only on the past occupancy data. To obtain a higher accuracy, the author refers that combining these algorithms with

context-aware approaches could push the accuracy above the 90% limit, by providing information about the current context or activity of each occupant.

In this work, we use three supervised machine learning algorithms for detecting/classifying occupancy: neural networks, support vector machines and random forest. To predict occupancy, we use the Presence Probabilities Simplified algorithm. These techniques are explained in chapter 3 in more detail.

3 Methodology

In this Chapter, we explain how the research questions are approached. The main objective of this work consist on analyzing the viability of monitoring and predicting occupancy in houses based on electricity consumption data. To this end, we divide our work into two parts. In this first part, we use machine learning algorithms to classify historical occupancy from electricity consumption data. In the second part, we use schedule-based approaches to predict occupancy from historical occupancy data.

3.1 Proposed architecture

To answer our research questions, we follow a data mining approach. Data mining is a relatively recent field of computer science and consists on discovering patterns from large and unstructured datasets, involving the areas of pattern recognition, machine learning, artificial intelligence, cloud architecture and data visualization [22]. Many methodologies exist to solve data mining problems, such as: CRISP-DM (CRoss Industry Standard Process for Data Mining), SEMMA (Sample, Explore, Modify, Model, and Assess) and KDD (Knowledge Discovery in Databases). According to a 2014 poll, presented in [23], CRISP-DM is the most popular methodology for data mining projects.

CRISP-DM was chosen as the methodology for our work since it focuses on delivering real value for business and answering to their needs. This methodology involves six steps, as shown in Figure 13, and each step is described below.

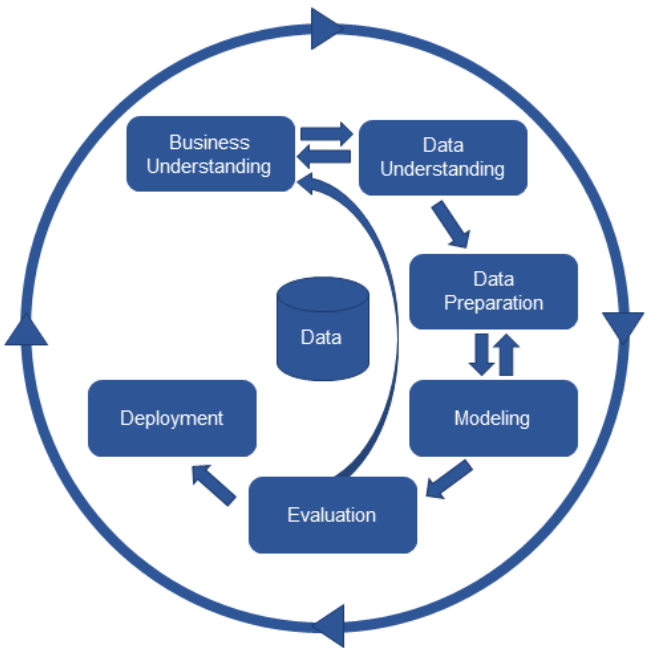


Figure 13: Six phases of the CRISP-DM methodology.

1. Business Understanding

The main goal of this work, from a business perspective, is to create value for electricity consumers from available opportunistic data. This work was done in conjunction with EDP-Comercial, which is a Portuguese company in the retailer electricity sector, and we shall use the EDP re:dy service in our experiments. EDP re:dy is an energy management service for the residential sector that provides electricity consumption data, control of appliances and smart energy functions to the customers. The data generated in this service can be used to create value for both electricity consumers and producers.

As shown in chapter 2, there is a correlation between electricity consumption and the occupancy state of a household (occupied or not). However, this is a recent field and it isn't clear yet in which situation it might be viable to detect and predict occupancy based on electricity consumption data.

An example of valuable application of electricity consumption data is to predict occupancy. From a business perspective, predicting occupancy opens the door to many useful applications that provides multiple benefits for the customers, such as: more safety, more comfort, more energy efficiency. Examples of these applications are smart thermostats and Heating, Ventilation and Air Conditioning (HVAC) systems. If we could predict household's occupancy with high accuracy, then we could program their thermostats to work only when the house is occupied, thus, increasing the energy efficiency and occupants comfort.

To respond to the business questions, it is necessary to define a plan from a data mining perspective. We start by acquiring data from five EDP re:dy customers. After validating the data, we choose the classification and prediction algorithms to be use in our experiments. Then, we evaluate our models based on common performance criteria for this type of problems. Finally, we extract useful conclusions from our experiments and also provide recommendations for further work.

2. Data Understanding

To collect the electricity consumption data, we installed the EDP re:dy service in five households in Portugal, during approximately 5 months (December 2016 to April 2017). In addition to electricity consumption data, we extracted the occupancy state of each household (ground truth occupancy), since it is necessary to perform supervised classification. The data has a sample frequency of 15 minutes and was extracted via queries from a database, using the open source R software. We extracted aggregate electricity consumption data through smart meters and we used smart plugs to collect the ground truth occupancy data, as explained in chapter 4. In this phase, we also verified the quality of the data, by observing its mean, maximum and also through a box plot observation.

3. Data Preparation

In this phase, we start by cleaning erroneous data, including missing values. Then, we remove night hours from our analysis (we considered the hours between 7:00h to 23:00h), since we do not focus on analyzing the occupancy during the sleeping period. We decided to use the same amount of data in every household to have a fair comparison of the results among the five participants. Thus, we delimited the number of records used in our experiments based on the minimum amount of samples verified in the five participants.

Then, we use expert knowledge to extract relevant features from the electricity consumption data, which are used in our classification models. The final step consists on scaling the features using the *standardization* process, which a common method used in these type of problems [7], [24].

4. Modeling

To detect occupancy from electricity data, we used three classification algorithms: neural networks, support vector machines and random forest (based on decision trees). We selected this three models since they are commonly used in similar problems and have proven to provide good results [18], [5], [13]. They are very different from each other. For example, neural networks may require more data to provide good results and takes more time to run while support vector machines runs faster and may provide better results with a smaller amount of data (comparing to neural network). Random forest is an ensemble algorithm that is based on decision trees to perform the classification. Typically, ensemble methods provide good results since they classify according to the contribution of multiple classifiers/decision trees, avoiding the overfitting. For each classifier, we repeat a 10-fold cross-validation ten times over our training data in order to obtain the best feature combination and model parameters.

In order to predict occupancy, we use the Presence Probabilities Simplified (PPS) algorithm, which is a schedule-based approach for prediction. This algorithm was chosen since it only uses historical occupancy information and is compatible with our business objective. This algorithm creates a 7-day timetable that contains the probability of a households being occupied in every 15 minute's interval. By defining a threshold, these probabilities are converted into two classes (occupied or not) and this prediction table can be used to program a smart thermostat, for example. A more detailed explanation of the PPS algorithm is presented in section 3.4.1.

In our experiments, we start by dividing the historical data (electricity consumption and occupancy data) of each household into 3 parts, as shown in Figure 14:

1. **Training set**: 40% of the data is used to train the classification models through cross-validations.

2. **Classification set**: 40% of the data is used to test the classification models and to measure the accuracy of the occupancy detection. Accuracy is defined as balanced performance metric for classification problems, and is described further. By using the results from the occupancy

classification, we create a probabilistic weekly timetable, containing the probability of a house being occupied in a certain time and weekday.

3. **Future set**: 20% of the data is used only to test the prediction accuracy using the probabilistic timetable, previously computed, and to measure its occupancy prediction accuracy. In reality, this dataset partition corresponds to historical data but we consider as future data in order to simulate a real problem.

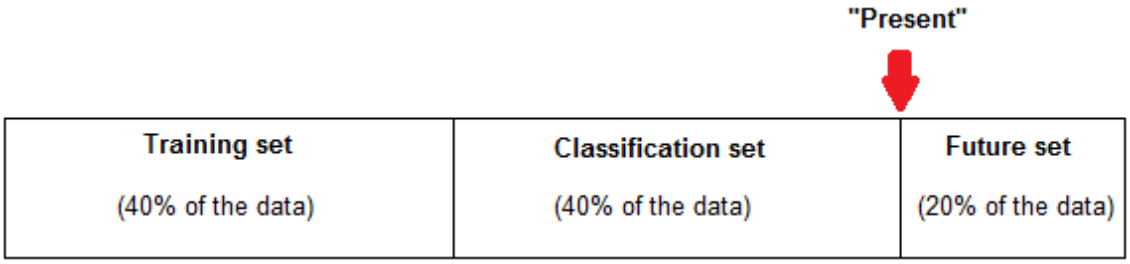


Figure 14: Partition of the historical electricity consumption and occupancy data into 3 subsets.

In the *training set*, cross validations are performed to avoid specific allocation of data (overfitting) and to obtain the optimal feature set and model parameters.

We perform a 10-fold cross validation, i.e., we divided the *training set* into ten parts with the same size and 10 iterations are done. At each iteration, 9 parts were used to train the model and one part to valid/test it, until every fold is tested. Each iteration generates a classification error metric, *E*, which, in our case, represents the accuracy, and the performance of the classifier is the average of the 10 iteration's results, as shown in Figure 15. This process is repeated 10 times in order to obtain a higher stability in the results

Figure 15: 10-fold cross-validation, used to calibrate our model parameters and to choose the optimal feature set [25].

For each classifier, we then repeat the process of cross-validation using different feature combinations and model parameters in order to maximize the accuracy. This part is very important since the wrong choice of features or model parameters can severely reduce the performance of a classifier [7].

After performing the cross-validations, we use the best feature combination and model parameters to test/evaluate the classifier on the *classification set* (40% of the initial dataset).

5. Evaluation

We use accuracy as the main evaluation criteria for our classification and prediction models, which indicates the percentage of correct classification or predictions. Other metrics were also used as a complementary measure, as described further in this in chapter.

We evaluate the possibility of predicting the occupancy of a household based solely on its electricity consumption by training a model and testing it in the same household. However, the main business challenge is to use a generic model that predicts with high accuracy the occupancy of any household. To this end, we trained a model in a single household and tested in multiple households.

6. Deployment

This phase includes the value and knowledge extracted from our data mining project and is represented in the experiments section. This includes the discussion of relevant questions and also recommendations for further work.

This data mining project were performed in a laptop with 2.40 GHz CPU and 16GB of RAM, running on windows 7 and all the statistical processes were done by using the open-source R software.

3.2 Machine learning algorithms

As shown before, machine learning is a relevant method of data mining. There are two types of machine learning methods: supervised learning and unsupervised learning [22].

Supervised Learning

In supervised learning, it is used a labeled dataset, which is a dataset containing both input and output data, used to train a model. The labeled data allows the model to compare and adjust its parameters so that the performance is maximum. In Figure 16, it is illustrated the working principle of supervised learning.

The input matrix is represented by (x_1, x_2…, x_n), where each element represents a vector, with many records, for a given feature of *n* features. The output or target variables are denoted by (y_1, y_2…, y_n). In

our case we only have one output variable and its value can be 1 (when a house is occupied) or 0 (when a house is unoccupied).

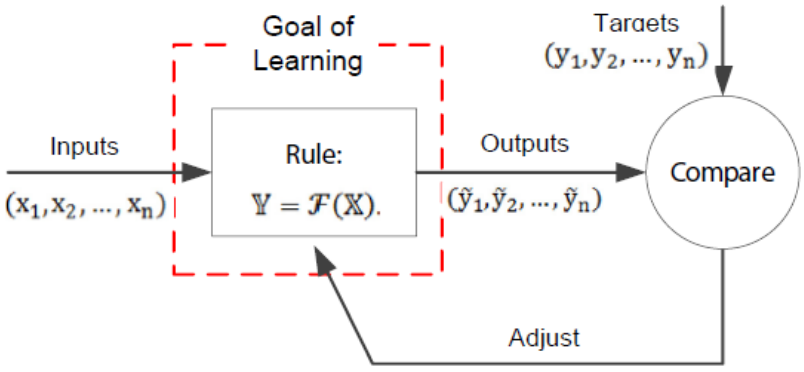


Figure 16: Mechanism of supervised machine learning algorithms [22].

The goal of supervised learning is to learn a general rule (F(X)) that maps the inputs X to outputs Y.

Unsupervised learning

Unsupervised learning represents the methods that do not use labeled data to train the goal function (F(X)). The goal of these methods is to discover hidden patterns in the input data X by using its features (without using the target data to compare and adjust the model), as can be seen in Figure 17.

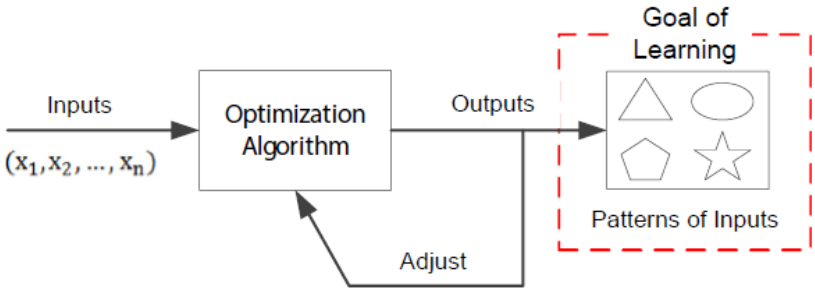


Figure 17: Mechanism of unsupervised machine learning algorithms [22].

Approach used

For occupancy detection from the electricity consumption data, it is necessary to use a supervised classification method since unsupervised learning, alone, cannot solve this problem. For this reason, we needed to collected both electricity consumption and occupancy data and to choose our classification algorithms: neural networks, support vector machines and random forest (based on decision trees).

Our main objective consists on predicting occupancy from electricity consumption data and can be divided into two parts: 1) have a classification model that detects, with high accuracy, the occupancy of

any household by using solely the electricity consumption data; 2) have a prediction model that predicts occupancy based on the detected occupancy data.

Another objective consists on analyzing the viability of using a generic model to detect and predict occupancy in multiple households using only the respective electricity consumption data. Because households are very different from each other in many aspects (occupancy rate, number of occupants, occupant's routines, interaction of households with electric devices, etc.), we do not believe that a single classification model detects occupancy with good results in every household. For instance, if we train a classification model in a household that consumes very little electricity when occupied (low correlation between power usage and occupancy) and apply it in a household that consumes a lot of electricity (high correlation between power usage and occupancy), we do not expect to obtain good results.

From our intuition and from expert knowledge, combining supervised learning with unsupervised learning may improve the desired results, comparing to an approach that uses solely classification and prediction models. Unsupervised approaches, such as clustering analysis, can be used to group households with similar characteristics/patterns of occupant presence. Then, for each group of households it would be attributed different classification and prediction algorithms.

Because we only have five participants in our analysis, we didn't perform a cluster analysis. Instead, we investigated which household characteristics would be relevant in an unsupervised learning analysis.

In the next sections we explain the theory behind each classification algorithm chosen.

3.2.1 Neural networks

Artificial neural network (ANN) is an artificial intelligence technique that is inspired in the human brain to solve problems. It is often used to solve both supervised and unsupervised classification problems (e.g. pattern recognition) and have been applied in many applications, such as: bankruptcy prediction, fault detection, speech recognition and product inspection [26].

The ANN structure is composed by three types of nodes: input, hidden and output nodes, as shown in Figure 18. The input nodes represent the nodes that receives the input information and emit signals to other nodes. The output nodes receive information from the network nodes and sends it as output to the environment. The hidden nodes are between the input and output nodes and do not interact directly with the external sources, i.e., do not receive information from the external environment and also do not outputs information to the environment [27].

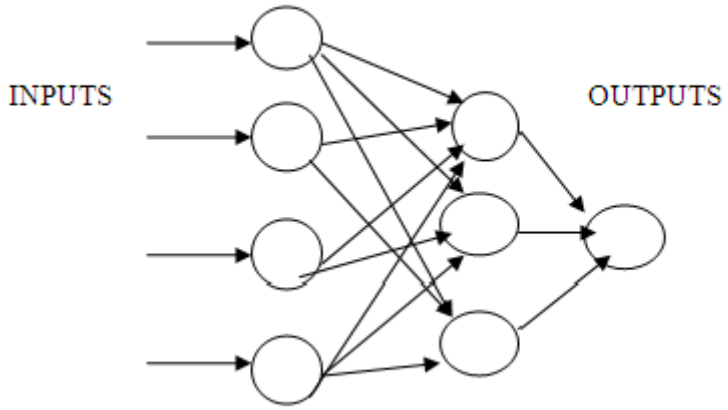


Figure 18: Illustrative structure of a neural network [28].

Every node in the structure is connected through a weight (*w*). Each connection multiplies the output of a node by the respective connection weight and sends the value to the input of other node. In Figure 19 it can be seen how the input value of a node is computed, according to the equation (3.1). w_p and w_q represent the connection weight between the node *i* and node *k*, and between the node *j* and node *k*, respectively [29].

$$input_k = (output_i \times w_p) + (output_j \times w_q) \qquad (3.1)$$

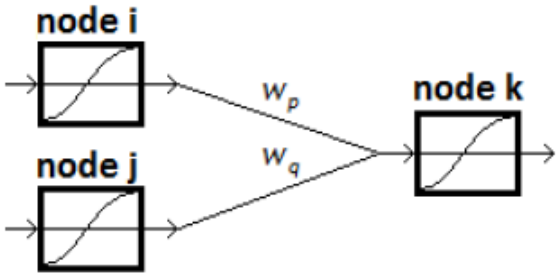


Figure 19: Relation between neural network nodes.

The output of the node *k*, a, can be calculated by applying an activation function to the sum between its inputs (multiplied by their respectively weights w) and a constant b. Equation *(3.2)* represents the output for a generic ANN unit.

$$a = f(\sum wp + b) \qquad (3.2)$$

The output of an ANN for a dataset with *N* records of data can be calculated by applying the equation *(3.2)* over the entire number of hidden layers of the neural network (K), as shown in equation *(3.3)*. w_i and $\beta_i$represents the weigh vector between the hidden layer and the input and output layer, respectively.

$$\Psi_K(p_j) = \sum_{i=1}^{K} \beta_i f(w_i p_j + b_i), j = 1 \dots N \qquad (3.3)$$

Activation Function

The activation function is responsible for calculating the output of a node given inputs from one or more nodes. In an artificial neural network, the activation function can be linear or sigmoidal (non-linear). Nonlinear activation function gives to the network nonlinear capabilities and the most frequently sigmoid functions used are the standard logistic function and the hyperbolic tangent. In this work we used the standard logistic function, which outputs a value between 0 and 1, representing the probability of a household being occupied [30].

The equation for the standard logistic function is given by equation *(3.4)*, where $f(x)$ represents the output of a node and x represents the input.

$$f(x) = \frac{1}{(1 + e^{-x})} \qquad (3.4)$$

The graphical representation of the standard logistic function is shown in Figure 20.

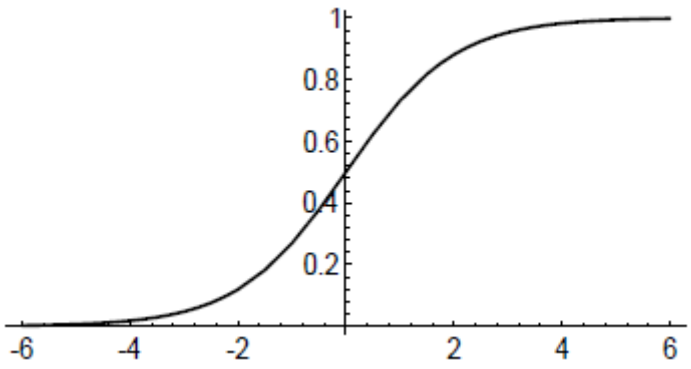


Figure 20: The standard logistic function.

Back Propagation

To train our neural network model, we used the backpropagation, which is a common method used for training neural networks. This process consists on adjusting the weights between the connections and can be divided into two steps:

1. Initially, random weights are given to the connections between each node.
2. The output for the random weights is compared to the real output and the weights are adjusted in order to minimize the error/difference.

This process is repeated until the algorithm converges or the maximum number of iteration is reached.

Typically, artificial neural networks are trained by minimizing an error function. Many error functions can be used, however, for binary classification problems, the cross-entropy error function is an appropriate choice. In this work, we used the cross-entropy error function, which is given by equation *(3.5)* [29].

$$E = \sum_{i=1}^{N} (a_i' \log(a_i)) + (1 - a_i' \log(1 - a_i)) \qquad (3.5)$$

In binary classification problem, a_i' represents the true outcome (1 or 0) for the record *i* in a dataset with *N* records, and a_i represents the predicted class by the neural network (1 or 0).

3.2.2 Support vector machines

Support vector machines (SVM) is an efficient supervised machine learning algorithm which is widely used for pattern recognition and classification problems such as face recognition, gene extraction and speaker identification. It can be used to perform both linear and non-linear classification through kernel methods and performs well even with less training data samples (comparing to neural networks) [24]. SVM also have a good ability to deal with high dimensionality data [31].

First, SVM models the examples of the *training set* as points in the space. Then, classifies them by constructing a hyperplane, or decision boundary, that divides the classes with the widest possible margin.

In this work, three types of SVM models were used: linear kernel, radial (or Gaussian) kernel and polynomial kernel.

3.2.2.1 Linear kernel

If the data is completely separable, SVM constructs a hyperplane by maximizing the margin between the two classes. The bigger is the margin, the best separated are the classes and less prone to overfitting is the model. Figure 21 shows an example of a linear SVM classifier.

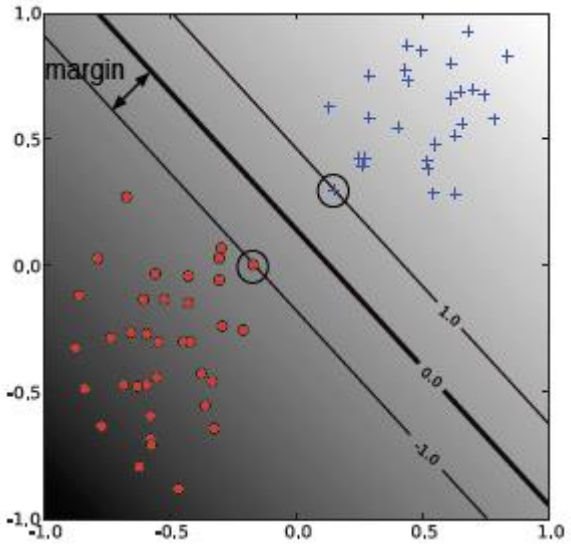


Figure 21: Example a two-class linear SVM classifier.

For a linear classifier and a two class learning problem, each data sample is labeled with the value 1 or 0, representing the two possible outcomes. In our case, the class 1 indicate that a household is occupied while the class 0 indicates that it is unoccupied.

We define xi as the i^{th} vector in a dataset $\{((xi, yi)\}_{i=1}^{n}$ with *n* samples, where y_i represents the label associated with xi. An important concept in linear classifiers, is the dot product (or scalar product) between two vectors, defined as $w^T x = \sum_i w_i x_i$.

Equation *(3.6)* describes the equation of the hyperplane, which comprises the set of points x where $w^T x = 0$ (considering $b = 0$). $f(x)$ represents a discriminant function, which can be 1 or 0, indicating the side of the hyperplane where a point is. The term w represents the weigh vector and b represents the bias term, which translates the hyperplane away from the origin.

$$\{x: f(x) = w^T x + b = 0\} \qquad (3.6)$$

SVM can be seen as an optimization problem that maximizes the margin of the hyperplane and minimize the misclassification [24]. The equation for this optimization problem is given by equation *(3.7)*, subject to the constrain *(3.8)*. This constrain ensures that each data sample has the maximum margin.

$$Min \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\varepsilon_i \qquad (3.7)$$

$$s.t. \quad y_i\,(w^T x_i + b) \geq 1 - \varepsilon_i, \quad \varepsilon_i \geq 0 \qquad (3.8)$$

The term ε_i is the margin error and represents slack variables. This term is included to allow some data samples to be in the margin ($0 \leq \varepsilon_i \leq 1$) or misclassified ($\varepsilon_i \geq 1$), since in the practice, data is often not linearly separable. The parameter *C* represents the soft-margin constant and is used to set the relative importance of maximizing the margin and minimizing the amount of slack, i.e., controls the relevance of misclassifications.

From Figure 22 it is possible to visualize the effects of the soft-margin constant on the hyperplane (large black line). For a small value of *C*, less importance is given to the misclassifications and so the margin is larger (distance between the two thick lines). Red circles represent the negative samples (one class) while the blue crosses represent the positive examples (the other class) [31].

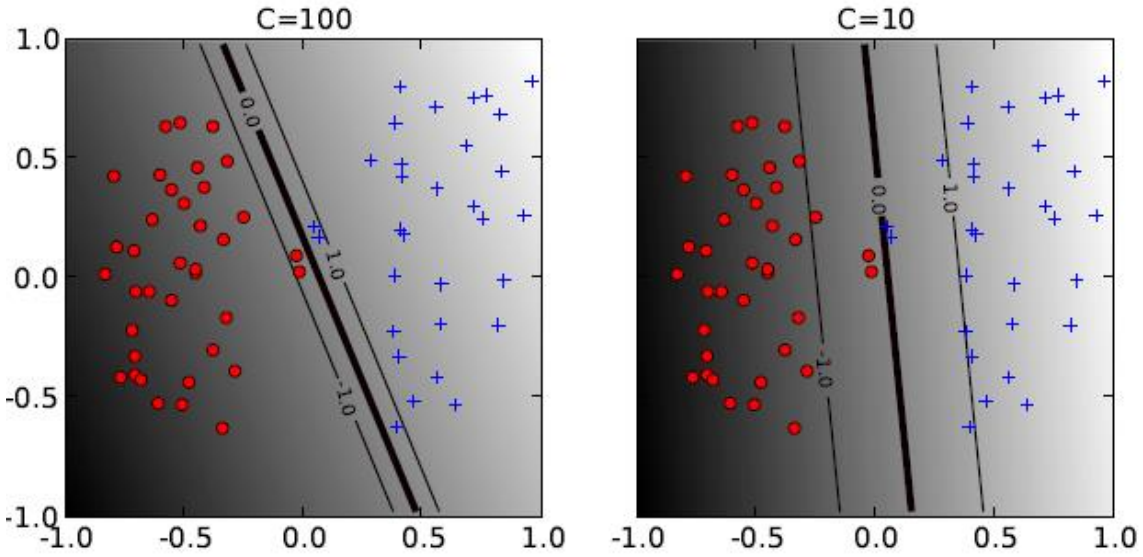


Figure 22: Effects of varying the soft-margin constant (C) on the decision boundary of a SVM classifier linear kernel.

3.2.2.2 Radial kernel

Classification problems with linearly separable data are rare in practice. Fortunately, SVM allows to perform a non-linear classification by applying kernel functions that map the data into a higher dimensional feature space, so that it is possible to perform a linear separation (kernel trick).

The radial (or Gaussian) kernel function is given by equation *(3.9)*, where x and x' are vectors in the input space [31].

$$k(x, x') = \exp(-\gamma \|x - x'\|^2) \qquad (3.9)$$

The parameter γ (gamma) represents the inverse-width of the Gaussian kernel and adjusts the bias-variance tradeoff. This tradeoff consists on minimizing two source of errors that reduce the ability of supervised learning models to generalize beyond the *training set*. The error due the bias consists on the difference between the expected prediction from our model and the correct value and occur due to erroneous assumption in the learning algorithm. The error due to variance represents the error due to small fluctuations in the *training set*.

In Figure 23 it is illustrated how a radial (or Gaussian) kernel function maps 2D data into a 3D space. As it can be seen, it is impossible to separate linearly the two classes in data with two dimensions, while it is possible in a three dimensional space.

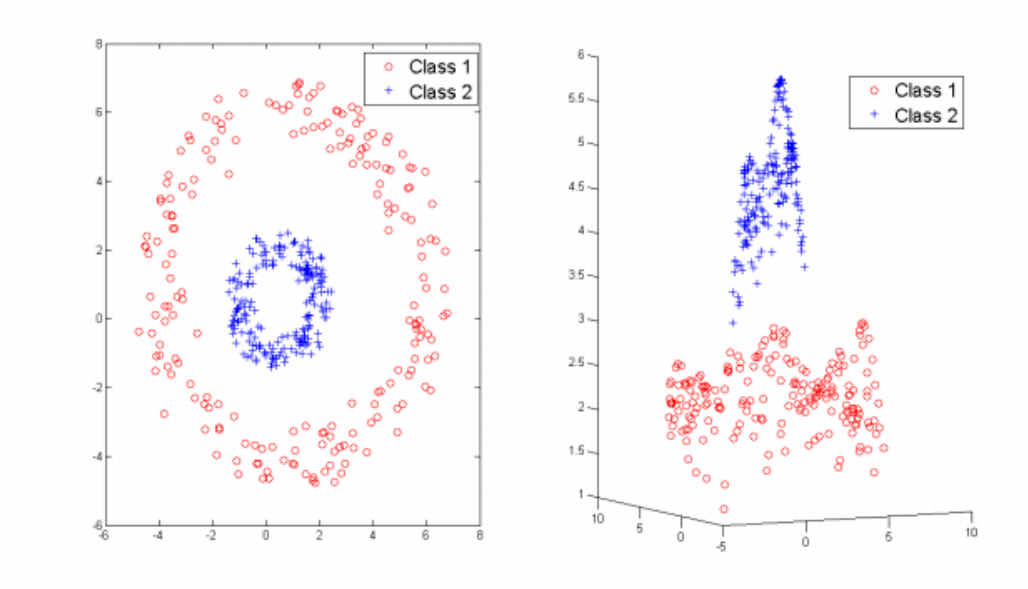


Figure 23: Mapping of 2D data to 3D data using Gaussian kernel, in order to perform a linear separation between two classes [32].

3.2.2.3 Polynomial kernel

The SVM with polynomial kernel uses equation *(3.10)* to map the data in a feature space over polynomials of the original variables. The parameter d represents the degree of the polynomial and controls the flexibility of the classifier.

$$K(x_i, x_j) = (\gamma x^T x' + 1)^d \qquad (3.10)$$

Figure 24 illustrates a comparison between the linear and polynomial kernel. It can be see that a higher degree polynomial kernel function allows to obtain a more flexible decision boundary. However, high degree polynomial kernel models are more complex and more prone to overfitting [31].

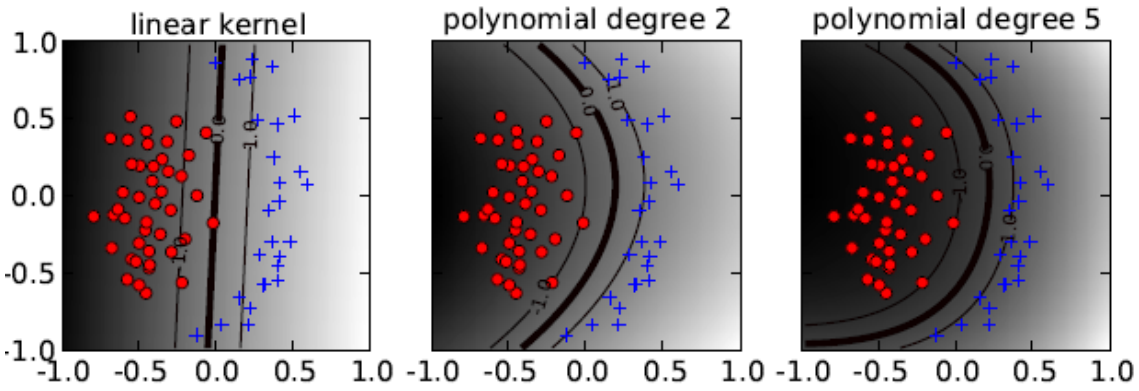


Figure 24: Comparison between a SVM classifier using a linear kernel and polynomial kernel of degree 2 and 5.

3.2.3 Random forest

Random forest is an ensemble machine learning method that that is typically used for classification and regression tasks. This ensemble learning method combines multiple uncorrelated decision trees and the predictions are obtained by combining the results from each decision tree. The combination of multiple classifiers reduces the model overfitting, thus increasing the classification accuracy.

A common technique used in ensemble algorithms is called Bootstrap aggregating or Bagging and is shown in Figure 25. The technique is used to re-sample the data and to generate different training sets for each classifier with replacement of the data.

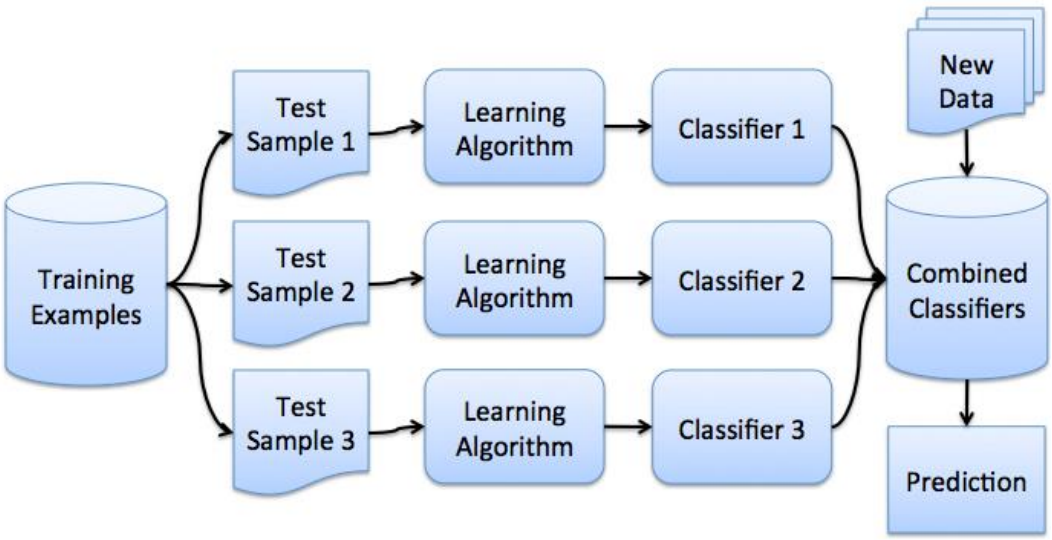


Figure 25: Phases of ensemble learning approaches to solve classification problems [23].

If the original dataset contains *N* number of records and we want to do our analysis with *J* classifiers (trees), then *J* independent training sets are generated of size *N* from the original dataset by sampling with replacement [33].

A single decision performs classification by recursively partitioning the data and is composed by three types of nodes: root node (*at1*), internal node (*at2*, *at3* and *at4*) and leaf nodes (yes and no outcomes), as shown in Figure 26. The root and internal nodes represent a test on an attribute/feature and the branch represents the outcome of that test. The tree is partitioned until no more tests are required for the algorithm to perform the classification. The tree ends with the leaf (or terminal) node that contains the classification outcome [34].

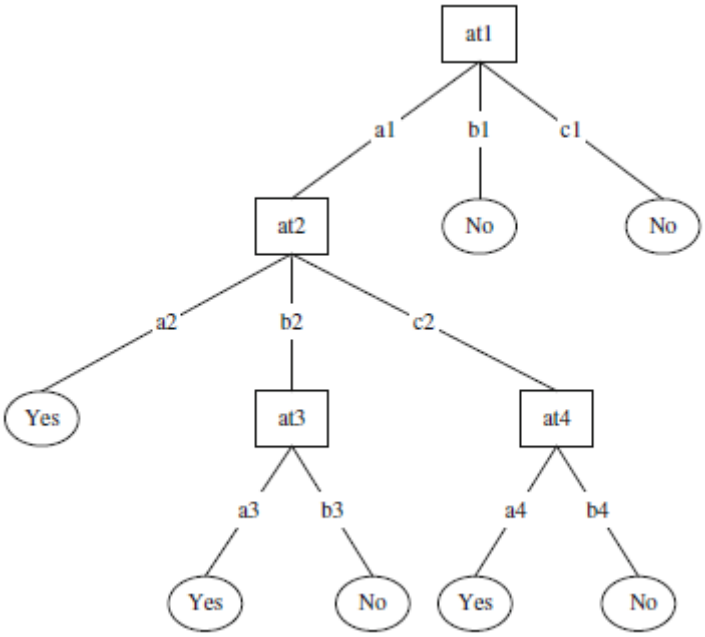


Figure 26: Decision tree algorithm structure [35].

In a simple decision tree algorithm, the root and internal nodes are ordered by decreasing order of importance, where the root node contains the feature that best describes the outcome. The prediction function of a simple decision tree is shown in equation *(3.11)*, where K represents the number of features/attributes, c_{full} represents the value at the root of the node and $contrib(x,k)$ represents the contribution of the *k-th* feature in the feature vector x [36].

$$f(x) = c_{full} + \sum_{k=1}^{K} contrib(x,k) \qquad (3.11)$$

Decision trees split the data at a node according to splitting measures. Many types of splitting measures exist, such as Entropy and Gini value as described in [35], and both of these measures aim to split the data so that partition purity is maximized (a partition is pure when all of its elements belongs to the same class).

The prediction function of the random forest model is the average of the predictions of its trees and is given by equation *(3.12)*, where J represents the number of trees in the forest and $f_j(x)$ represents the prediction of the *j-th* tree.

$$F(x) = \frac{1}{J}\sum_{j=1}^{J} f_j(x) \qquad (3.12)$$

By replacing the equation *(3.11)* in equation *(3.12)*, the prediction function of a random forest is simply computed by summing the average of the bias terms with the average contribution of each feature, as shown in equation *(3.13)* [36].

$$F(x) = \frac{1}{J}\sum_{j=1}^{J} c_{j_{full}} + \sum_{k=1}^{K}\left(\frac{1}{J}\sum_{j=1}^{J} contrib_j(x,k)\right) \qquad (3.13)$$

Each tree of the forest provides one classification results (vote) and the final outcome of the random forest model corresponds to the class that have most of the votes. The choice of the number of trees is important and depends on the tradeoff between the desired accuracy and the computation complexity. For small datasets, 50 trees may be sufficient while 500 or more trees may be necessary for large datasets [37], [38].

Each root and internal node of the random forest represents the best features chosen from a subset of random *mtry* features. The parameter *mtry* is the number of input features available for splitting at each tree node and is typically defined as shown in equation *(3.14)*, for classification tasks [37], [38].

$$mtry = \sqrt{number\ of\ features} \qquad (3.14)$$

3.3 Classification evaluation

Binary occupancy classification represents a two-class classification problem since a household can be occupied or unoccupied in each interval of 15 minutes. There are four different possible outcomes for the process of occupancy classification, which can be seen in the confusion matrix, presented in Figure 27.

		Actual class (ground truth)		
		p (*occupied*)	n (*unoccupied*)	Total
Predicted class	p' (*occupied*)	True Positive	False Positive	$tp+fp$
	n' (*unoccupied*)	False Negative	True Negative	$fn+tn$
	Total	$tp+fn$	$fp+tn$	N

Figure 27: Confusion matrix of a binary classifier.

A *true positive (tp)* occur when occupancy is correctly classified while a *true negative (tn)* refers to an unoccupied moment correctly classified. A *false negative (fn)* occurs when an occupied interval is falsely labeled as unoccupied while a *false positive (fp)* is when an unoccupied interval is falsely classified as occupied.

Many evaluation metrics can be used to evaluate a classifier. In this work we used the metrics accuracy and Mathews Correlation Coefficient (MCC) as our main evaluation criteria, since they complement with

each other and were used in similar works [4], [5]. The false negative rate (FNR) and false positive rate (FPR) are also used in order to have more detailed information about errors

3.3.1 Accuracy

Accuracy is a simple metric to evaluate a classifier and can be computed by dividing the number of correct classifications by the total number of classifications, as shown in equation *(3.15)*.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \qquad (3.15)$$

In order to have a baseline for comparing the obtained classification accuracy by each algorithm for a certain household, we use the Prior classifier, which assumes that the house is always occupied or unoccupied (if the house is typically occupied or unoccupied, respectively, in most of the times). For example, if a house is more than 50% of the time occupied, then Prior assumes the house to be always occupied, otherwise, assumes the house to be always unoccupied.

As already shown in chapter 2, the type of classification error has different consequences for the occupants. For instance, if occupancy classification is used to control home heating, classifying an occupied house as unoccupied (fn) may cause discomfort to the occupants, since the heating system would be turned off and the temperature lowered. Classifying unoccupied periods as occupied (fp) may reduce the energy efficiency of the heating system, since the heating system would be working without any occupant in the house.

For this reason, before choosing the evaluation criteria of a classifier that, for example, controls a home energy system, it is important to first define the cost of a false positive and a false negative error. In our work, we assumed the same cost for *fn* and *fp* errors. In a situation with different error costs, other metrics for evaluate binary occupancy should be used, such as: *Precision*, *Recall* and *F1-score* [13].

3.3.2 Matthews correlation coefficient

Because accuracy describes only partially the performance of a classifier, especially for data with unbalanced classes, the Matthews Correlation Coefficient is considered as a complementary evaluation metric. The value of MCC varies between -1 and 1. A value of -1 represents that no single instance was correctly classified. A value of 1 represents a perfect classification and a value of 0 indicates that the classification isn't better than a random guess. The MCC of a classifier is calculated as shown in equation *(3.16)* [4].

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}} \quad (3.16)$$

3.3.3 False positive and false negative rate

As already referred above, classifying an unoccupied house as occupied, in a 15-minute interval, represents a *false positive* and may reduce the energy efficiency of home heating systems due to unnecessary heating. The false positive rate (FPR) is a measure of these occurrences and is computed by dividing the number of false positives by the number of total unoccupied slots (equation *(3.17)*).

$$FPR = \frac{fp}{fp + tn} \quad (3.17)$$

A low false positive rate indicates that the classification algorithm has a good performance on detecting absence.

On the other side, labelling an occupied house as unoccupied, in a 15-minute interval, represents a *false negative* and may cause discomfort to the occupants due to the temperature lowering (in a thermostat application). The frequency of this type of errors can be measured with the false negative rate (FNR), which is calculated by dividing the false negatives by the total of occupied slots, as shown in equation *(3.18)*.

$$FNR = \frac{fn}{fn + tp} \quad (3.18)$$

A low false negative rate indicates that the classification algorithm has a good performance on detecting presence.

3.3.4 True positive and true negative rate

The true positive rate (TPR) is defined as the percentage of positive instances that are correctly classified and is given by equation *(3.19)*. It is also known as sensitivity or recall.

$$TPR = \frac{tp}{tp + fn} \quad (3.19)$$

The true negative rate (TNR), or specificity, is the proportion of negatives that are correctly identified, and is calculated by equation *(3.20)*.

$$TNR = \frac{tn}{fp + tn} \qquad (3.20)$$

3.4 Occupancy Prediction

As referred in chapter 2, many algorithms exist to predict occupancy. Some of them use current information (context-aware approaches), such as current electricity consumption or GPS location, others use only historical information and others use mixture between these two types (hybrid approaches). Our work focuses on using historical occupancy data to predict the future.

To this end, we used a schedule-based approaches that simply computes the occupancy probability in every interval of 15 minutes in every day of the week.

3.4.1 Presence Probabilities Simplified

We choose the Presence Probabilities Simplified (PPS) algorithm to create an occupancy probabilistic timetable, which is then tested in the *future set*. The probability of a house being occupied in a given weekday and time period, $p_{occ}(w,t)$, is computed by dividing the number of occupied periods by the total number of existent periods in the respective time and weekday of the *classification set*. For example, if our *classification set* has 3 weeks of data, and if in this 3 weeks the house was always classified as occupied in the Mondays between the interval of 20:00h-20:15h, then the probability of presence in this period is 1 (3/3). The probabilistic timetable was then computed by applying the equation *(3.21)* over each interval of the week.

$$p_{occ}(w,t) = \frac{number\ of\ occupied\ records(w,t)}{number\ of\ records(w,t)} \qquad (3.21)$$

Our work focuses on classifying and predicting occupancy between the 7:00h – 23:00h. Thus, our probabilistic timetable has 64 periods of 15 minutes per day and 448 periods in total. Table 1 shows the structure of our probabilistic timetable. Weekdays are numerated between 1 (Sunday) to 7 (Saturday) and time between 1 (7:00h-7:15h) to 64 (22:45h-23:00h).

Table 1: Probabilistic timetable created through the Presence Probabilistic Simplified (PPS) algorithm. This table contains, for each hour of the day (between 7:00h and 23:00h) and for each day of the week, the probability of presence in a certain household.

time	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
7:00h-7:15h	$p_{occ}(1,1)$	…	…	…	…	…	$p_{occ}(7,1)$
…	…	…	…	…	…	…	…
22:45h-23:00h	$p_{occ}(1,64)$	…	…	…	…	…	$p_{occ}(7,64)$

3.4.2 Prediction evaluation

To evaluate the prediction performance of the probabilistic timetable, it is first necessary to convert the probabilistic timetable into a binary timetable. This is done by choosing a cut-off value as our decision rule (or threshold). To this end, we first analyze the effect of choosing different cut-off values through the Receiver Operating Characteristic (ROC) curve. Then, we use the accuracy to evaluate the prediction performance.

3.4.2.1 Receiver Operating Characteristic curve analysis

The prediction timetable constructed through the Presence Probabilities Simplified (PPS) algorithm, when applied in the *future set*, statistically divides the data into two populations, one representing the periods when the house is occupied (label 1), and the other representing the periods when a house is not occupied (label 0) as function of the cut-off value of our decision rule, c.

In Figure 28, each line represents the model-generated probabilities distributions for the two classes. The closer are the two distribution curves, the more confuse is the model when performing the prediction. The best models represent the ones that complete separate the two classes so that the overlap between the two curves is minimum.

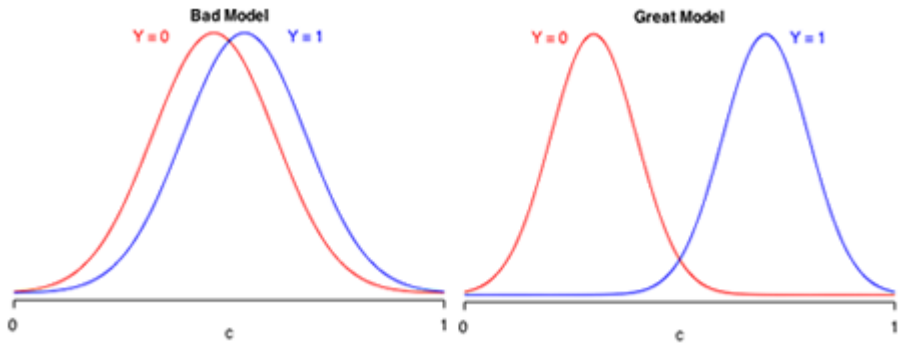


Figure 28: Probability distribution curves generated by a bad and a great model for each class.

To obtain a binary output from the prediction model, it is necessary to define a threshold as decision rule. From Figure 29, it is possible to observe the effects of varying the threshold, c, on the results. A lower threshold value increases the true positive rate. In other words, more occupied periods are correctly predicted. However, it also increases the false positive rate, i.e., more unoccupied periods are falsely predicted as occupied.

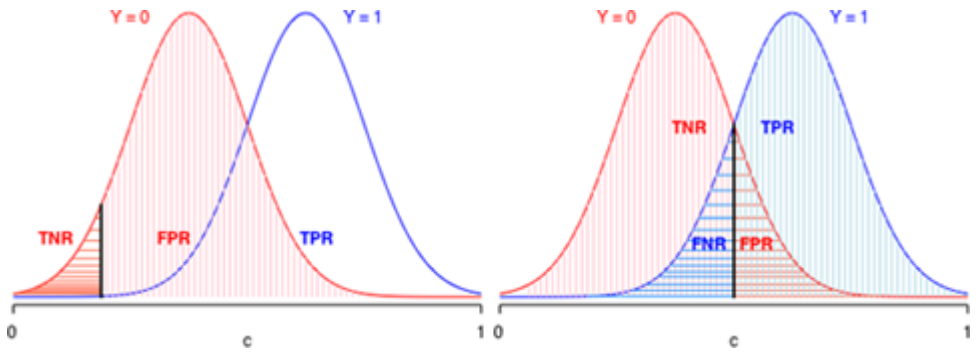


Figure 29: Effect of varying the cut-off value (c) in the prediction results.

As already referred in chapter 2, before defining a threshold, it is important to know the goal of our prediction. In our work, because we considered an equal cost of a false positive and a false negative, we decided to use a threshold equal to 0.5 for our decision rule.

The Receiver Operating Characteristic (ROC) curve is used to show how changing the threshold in the decision rule affects the TPR and FPR. For the same value of FPR, the higher is the TPR, the better. The ideal model would have a ROC curve passing through the point where TPR is 1 and FPR is 0. In Figure 30, it is possible to observe three ROC curves, for a bad, a good and a great prediction model. The dashed diagonal line represents a ROC curve for a random choice scenario, e.g., assuming that a household is always occupied, and the brown diagonal line represents the equal error rate line, where the false positive rate is equal to the false negative rate.

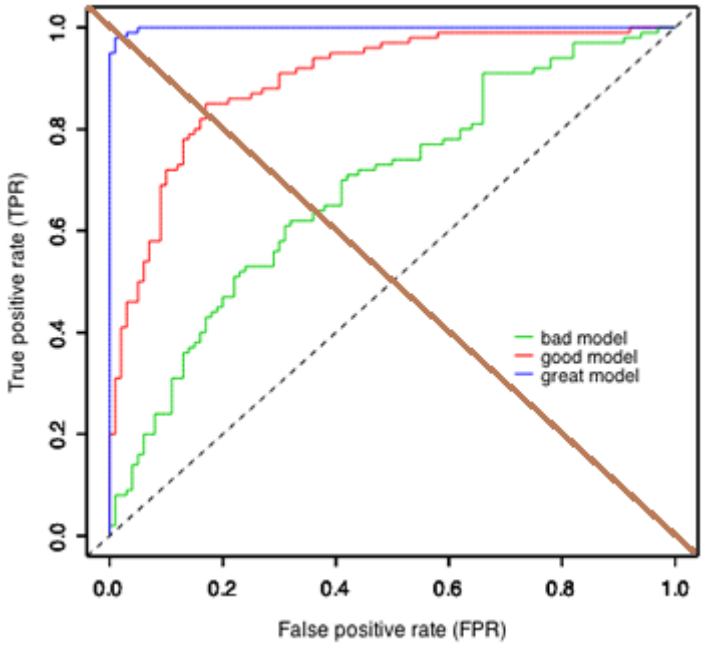


Figure 30: ROC curves for a bad, a good and a great classification model [39].

If occupancy prediction were performed with the goal of controlling a thermostat and if the customer prefers more comfort rather than energy efficiency, then we should define a threshold so that a high TPR is obtained, even if we misclassify some unoccupied periods as occupied. If we want to provide to the customer the possibility of choosing, at any time, its preference between energy savings or comfort, then the model should be training in order to maximize Area Under the Curve (AUC). AUC represents the area under the ROC curve and the higher is the value, the better is the model, in general, for every threshold.

3.4.2.2 Prediction accuracy

After defining a threshold equal to 0.5, we consider that, whenever the probability of occupancy, in a certain time period, is equal or higher than 0.5, the house is occupied in the respective period (and is unoccupied otherwise). Therefore, the probabilistic timetable is converted to a binary timetable.

To evaluate the prediction performance of our algorithm, we use the metric accuracy, given by equation *(3.22)*, that we recall here for convenience.

$$Prediction\ Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \qquad (3.22)$$

4 Data preparation

In this chapter, it is explained the process of collecting, pre-processing and extracting value from the data. Every step of this process is essential in order to obtain useful and valid data that is further used to monitor household's occupancy based on the chosen machine learning algorithms.

We start by describing the data collection process and its measurement infrastructure (infrastructure installed to collect the data), including a brief description about the characteristics of the chosen participants. Then, we explain the procedures that are used to guarantee the quality of the data. Finally, it is explained the features extracted and their scaling method.

4.1 Selection of households and data collection

To collect the necessary data for our study we installed a smart energy management system in five households. To obtain more information about each participant and to verify if they meet all the requirements for our analysis, we asked for each household to fill a questionnaire.

The questionnaire contained multiple questions such as: the number of occupants, their age and occupation; the characteristics of the house occupancy during the weekdays and the weekend days; pet ownership and type of heating. We also asked which appliances are switch operated, i.e., require human interaction to turn on and if they have any equipment scheduled to work automatically.

Not having any pet and any scheduled appliance are the mandatory requisites because it would affect our analysis (since some appliances could be turned on when a household is not occupied). Table 2 contains a summary of the gathered information. Household 1 have 8 occupants while household 5 have only one. Household 4 and 5 have seems to have similar occupancy profiles during the week, since their occupants are full-time workers. However, household 4 seems to have more variability in the occupancy during the week, since one occupant have irregular work schedules. Households 1 considers its occupancy profile as very occupied during the all days of the week.

Table 2: Information collected of each household from the questionnaire.

Household	Num. of occupants	Weekdays' occupancy	Weekend days' occupancy	Type of heating
1	8	Very occupied	Very occupied	Electric and natural gas
2	2	Unoccupied: 9h-11h; 14h-16h: Very occupied Variable profile	Very occupied Variable profile	Electricity (Water and space)
3	3	Unoccupied: 11h-15:30h	Very occupied	Electricity (Water)
4	2	Unoccupied: 8:30h-19:00h Variable	Unoccupied: 12h-16h Variable profile	Electricity (space), Natural gas (Water)
5	1	Unoccupied: 9:30h-19:30h	Very occupied Variable profile	Electricity (space) Natural gas (Water)

Some occupants of households 2,4 and 5 may spend one or two weekends away from the house per month. Thus, a more variable occupancy profile is associated to these households. Household 1 and 3 have similar occupancy patterns during the weekends.

Using the information provided by the participants in the questionnaire, we divided the appliances of each household into two categories: Switch operated and other appliances, as shown in Table 3. We consider that switch-operated devices represent the appliances that, when consuming electricity, indicate that the house is occupied.

Table 3: Description of the appliances of each participant. Switch-operated appliances represent appliances that, when consuming electricity, indicate that the house is occupied.

Household	Switch-operated appliances	Other appliances
1	2 TV's	Fridge, Freezer, Clothes dryer, 3 space heater
2	Kitchen appliances (Microwave, kettle, toaster, electric plate)	Fridge, electric water heater
3	-	living room appliances (2 TV's and PC), Fridge, electric water heater
4	Kitchen appliances (coffee machine, kettle and toaster), Space heater	TV, Fridge
5	Kitchen appliances (coffee machine, kettle and toaster), Microwave	TV, Fridge, Dishwasher machine

Some appliances are switch operated but are categorized as other appliances, since the households mentioned that those appliances may be active while the house is not occupied (e.g. they leave the house for a short time period without turning off the TV).

From the measurement infrastructure it was possible to obtain both electricity consumption and occupancy data. The measurement infrastructure corresponds to interconnection between the EDP re:dy service devices, such as the smart meter, smart plugs and the gateway. To collect the ground truth occupancy data, we gave each household a 7W LED lamp and instructed them to keep the light on whenever the house is occupied and turn it off whenever it's not occupied. Later, a simple condition was applied to convert this information into a binary value, which is 1 when the lamp is consuming (house is occupied) and 0 when it's not consuming (house is unoccupied). Other authors gave a tablet to each participant in order to register this information [4], [7], however, we considered that our approach was a reasonable solution for the problem.

We also advised our participants to register the time periods that erroneous ground truth data was registered. This information is later used to correct these faults.

4.2 Measurement infrastructure

To obtain the aggregate energy consumption (kWh) and maximum power (kW) in every 15 minutes, we replaced the typical electricity meter by a smart meter, which was also used for billing purposes. The smart meter is connected to a Power Line Communication (PLC) modem and sends the information with the aggregate energy (kWh) and power (kW) consumption to the modem. The modem modulates this digital information and transmits it to the electrical wiring through PLC.

Then, a management gateway captures this signal, through PLC, and demodulate it to obtain the desired data. Figure 31 illustrate the measurement infrastructure used to collect our data.

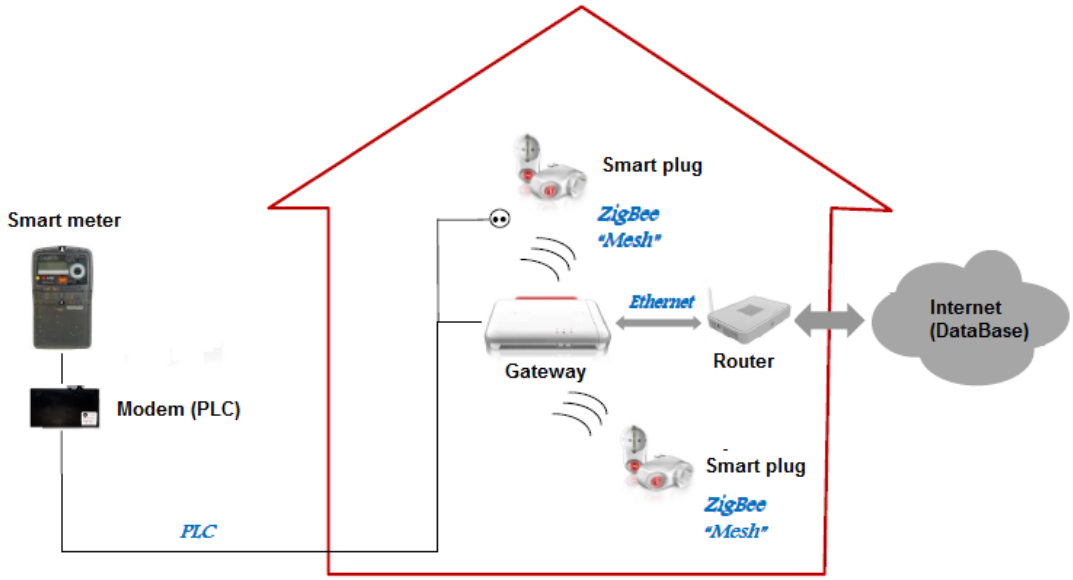


Figure 31: Measurement infrastructure installed in every participant to collect the data.

The consumption of the LED lamp was obtained by using a smart plug. Smart plugs measure the energy consumption (kWh) every 15 minutes of one appliance or group of appliances and sends this information to the gateway via ZigBee, which is a low-cost wireless communication protocol used for low-power applications (e.g. sensors). This ZigBee connection has a maximum transmission distance of 20 meters to the gateway, however, because it works in a mesh topology, higher distances can be obtained by using smart plugs in intermediate distances.

Thus, the gateway receives data from the smart meter trough PLC and from the smart plugs through the ZigBee protocol and structures the data to the desired format. The gateway is connected to a router through an Ethernet connection, which sends the data to a Service Delivery Platform (SDP) via Internet. Then, the SDP stores the date in a database.

To extract this information, it was used the open source software R. We set up a ODBC (Open Database Connectivity) connection, which is a database connector, to connect to the computer to the database. This was done by using the public available R package “RODBC” and performing queries to extract the necessary data for our analysis.

Table 4 contains the characteristics of the collected data and the respective sensor. The maximum power is a vector containing the maximum power consumption (kW) observed in a 15-minutes interval. The aggregate energy consumption represents the total energy consumption of the house (kWh), at every 15-minutes interval. As mentioned above, we use the LED lamp consumption, in every 15 minutes, to obtain the ground truth occupancy data.

Table 4: Description of the extract data from the measurement infrastructure and the respective sensor.

Extracted raw data	Units	Interval (min)	Sensor
Maximum power	kW	15	Smart meter
Aggregate energy consumption	kWh	15	Smart meter
LED lamp (ground truth occupancy)	KWh	15	Smart plug

An example of the aggregate energy consumption data of household 1 is shown in Figure 32.

Date	Consumption_kWh
2017-03-07 09:45:00	0.101
2017-03-07 10:00:00	0.158
2017-03-07 10:15:00	0.726
2017-03-07 10:30:00	0.724
2017-03-07 10:45:00	0.731
2017-03-07 11:00:00	0.715
2017-03-07 11:15:00	0.713
2017-03-07 11:30:00	0.743
2017-03-07 11:45:00	0.794
2017-03-07 12:00:00	0.724
2017-03-07 12:15:00	0.094
2017-03-07 12:30:00	0.284

Figure 32: Example of the aggregate energy consumption of household 1 (kWh).

4.3 Data cleaning and pre-processing

For many reasons, the raw data extracted from the measurement infrastructure may contain missing and erroneous data. Common causes of these problems are the bad communication between the sensors and a weak internet signal. Erroneous ground truth information may be obtained since occasionally some occupants forget to correctly register the occupancy sate of the house (e.g. the house is occupied and the “occupancy lamp” is off or vice-versa).

In order to overcome the problem of missing data, we simply removed missing values from our dataset, as similar to other works [4], [22]. Regarding the erroneous ground truth occupancy data, each household registered manually the periods in which the occupancy state was wrongly registered. Then, we used this information to correct the respective faults.

We limited our analysis between 7:00h and 23:00h because our work does not intend to analyze the occupancy during the sleeping period. Related works also do not analyze the nigh period since there is

a low correlation between the electricity consumption at night and the occupancy state [4], and a combination with other methods may be necessary (e.g. passive infrared sensor sensors).

Before continuing our analysis, we validate our data by using statistical methods such as box plot and mean to guarantee that the data is valid. Box plot is a graphical way of analyzing groups of numerical data according to their quartiles

Table 5 contains relevant statistics concerning the aggregate energy consumption and maximum power of each household, relative to intervals of 15 minutes. During the analysis period (about five months), three of the five households have exceeded their contracted power. This is only possible for very short time periods, otherwise the cut-out intervenes preventing a power overload. All the values in the table except the contracted power are referred to the entire dataset of the respective household and to the considered period (between 7:00h and 23:00h).

Table 5: Statistic metrics with regard to the data obtained by the smart meters, for each household. This table was used to guarantee the quality of the data (e.g. detecting possible outliers or other errors in the data).

Household	Aggregate energy consumption (kWh)		Power (kW)		Percentage of occupancy (%)
	Mean (15 min.)	Max (15 min.)	Contracted	Max (15 min.)	
1	0.698	2.809	17.25	12.139	89.77
2	0.087	0.816	3.45	4.582	68.19
3	0.122	0.968	3.45	5.194	62.22
4	0.042	0.666	3.45	4.68	60.03
5	0.042	0.666	6.9	3.916	29.58

As it was expected, household 1 is the most occupied since it has 8 occupants with different occupancies. On the other side, household 5 is the less occupied between 7:00h and 23:00h, since the only occupant works as full-time during the day. Households 2,3 and 4 have a similar occupancy percentage.

After the data clean and pre-processing, we obtain a different amount of days of data for each household, during the months of December, January, February March and April of 2017, as shown in the Table 6:

Table 6: Number of days available for each household after data clean and pre-processing.

Household	Number of Days
1	83
2	97
3	95
4	99
5	87

For our experiments, it is important to have the same amount of samples for each household. For this reason, we limited the number of samples in the evaluation to 83 days for every household, which corresponds to the minimum number of days obtained.

Figure 33 shows the electricity consumption (black line) and occupancy state (red line), in intervals of 15 minutes, on a typical weekday of the household 5. This household have one occupant that works between 10:00h and 19:30h during the weekdays (Monday to Friday).

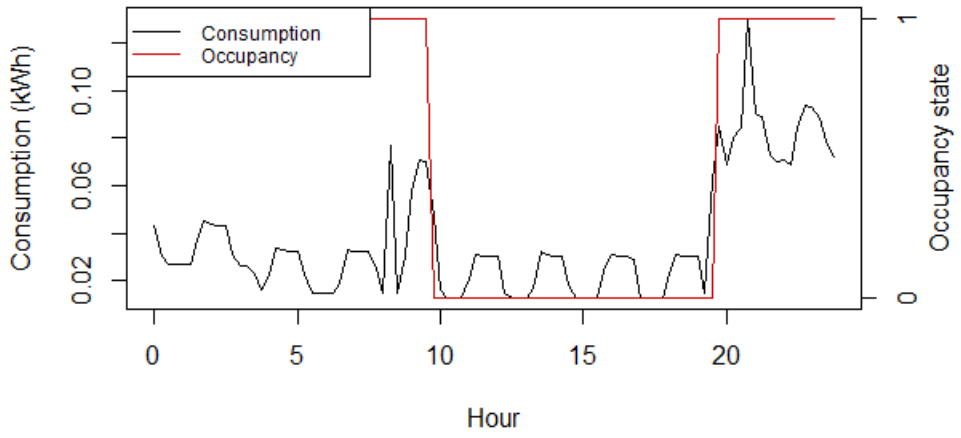


Figure 33: Electricity consumption and occupancy profile, every 15 minutes, of a typical weekday of household 5.

It can be observed that, when household 5 is occupied, there is a higher energy consumption with irregular patterns due to the random interaction of the occupant with the electric appliances (TV, lights, computer, kitchen appliances, etc.). When the house is not occupied, the energy consumption still has some fluctuations but with a regular pattern, since they are caused by appliances such as the fridge and the freezer.

4.4 Feature extraction and description

In order to extract relevant features to predict the occupancy of a household it is first important to understand the relation between power consumption and occupancy.

According to [4] and [3] when a house is occupied, typically, its average power consumption is higher with a higher standard deviation and absolute range. In general, a high power consumption variability may indicate human activity due to the interaction with electric appliances (e.g. kettle, TV, lights, etc.) as shown in Figure 6.

From section 4.2 we saw that there are appliances that, when consuming, are a clear indicator of human presence and would provide an added value for our classification analysis. There are some ways to detect these devices energy consumption. One way consists on installing smart plugs on each appliance

to obtain their individual consumptions. The drawback of this approach is that each smart plug has a cost for the household. Other way could be using non-intrusive load monitoring (NILM) algorithms to estimate the appliance electricity consumption from the aggregate electricity consumption data, as already addressed by many authors [5], [3], [40]. These algorithms identify the activation states of the appliances through the energy "signature", which is unique for every appliance, and associate to the respective installed equipment in the house. However, because these algorithms require extensive training period and the results may vary with the number of appliances in the house [4], they will not be considered in this work.

This work focuses on predicting occupancy using features that do not depend on devices consumption, namely the average power and the maximum power observed in every 15 minutes.

The first feature that was calculated was the average power in every 15 minutes (*p_mean*). To do this, we start by subtracting the aggregate energy consumption (kWh) by the "occupancy lamp" consumption in order to remove the influence of this artificial consumption, which was only used to provide the ground truth occupancy data. Then we simply converted the energy consumption (kWh) to power units (kW) by dividing by 0.25, obtaining the feature *p_mean*.

There are two situations that reduce the correlation between the power consumption and occupancy. The first is when occupants are at home and are not using any electrical device. The second is when the house is unoccupied and high consumptions occur (e.g. the consumption of an electric water heater to maintain the temperature above a certain value) [3].

In order to capture the variability of the power consumption, *p_mean_sd* and *p_mean_sad* features are calculated. *p_mean_sd* represents the standard deviation of the mean power and measures the distance to the mean of the data. A high standard deviation means that relevant changes in the power consumption occurred, which is likely to be caused by human interaction. The general formula of the standard deviation is given by equation *(4.1)*.

$$standard\ deviation = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \qquad (4.1)$$

p_mean_sd represents a vector of elements, with the same length as the length of the vector *p_mean*, where the *i-th* element is computed by applying the standard deviation in three intervals: *i-th*$_{-1}$, *i-th, i-th*$_{+1}$. Thus, *N* is equal to three and the formula, for the *i-th* element, became as shown in equation *(4.2)*.

$$p\_mean\_sd_i = \sqrt{\frac{1}{3}[(p\_mean_{i-1} - \mu)^2 + (p\_mean_i - \mu)^2 + (p\_mean_{i+1} - \mu)^2]} \qquad (4.2)$$

Where μ represents the mean of three elements, given by equation *(4.3)*.

$$\mu = \frac{p\_mean_{i-1} + p\_mean_i + p\_mean_{i+1}}{3} \qquad (4.3)$$

The feature *p_mean_sad* is another measure of the power variability. It consists on the sum of absolute differences of the mean power and is calculated by summing the absolute differences between the adjacent power measurements. For the *i*-th element of *p_mean*, the vector *p_mean_sad* is computed according to equation *(4.4)*.

$$p\_mean\_sad_i = |p\_mean_i - p\_mean_{i-1}| + |p\_mean_i - p\_mean_{i+1}| \qquad (4.4)$$

Because occupancy is also dependent on the current time of the day [5], the feature *time* was considered. This feature represents the time slot in the day and its value varies between 1-64 since one day have 64 periods of 15 minutes (after removing the night hours), where 1 represents the 7:00h-7:15h interval and 64 represents the 22:45h-23:00h interval.

Table 7 describes the features extracted that are available for our analysis.

Table 7: Features extracted and used for occupancy classification.

Feature	Description
p_mean	Mean power excluding the “occupancy lamp” consumption
p_mean_sd	Standard deviation of the mean power
p_mean_sad	Sum of absolute differences of the mean power
p_max	Maximum power verified in every 15 minutes
time	Time slot number (1-64)
workday	1 if it is a working day (Monday to Friday); 0 otherwise
before_workday	1 if the day precedes a working day (Sunday to Thursday); 0 otherwise
night_time	1 if it is dark outside (between sunset and sunrise); 0 otherwise

We also included three features for our own experiment: *workday*, *before_workday* and *night_time*. Because every household have at least one worker occupant, we decided to analyze if having characteristics about the day would improve the accuracy of occupancy prediction. The feature night time were included in order to test if separating the periods of the day with and without sunlight would improve the prediction accuracy. It is expected that at night the occupants consume more energy due to lightings.

4.5 Feature scaling

After pre-processing and extracting relevant features from the raw data we obtained a set of heterogeneous features with different characteristics, such as different units, scale and range. If one feature varies more than others the classification process would take much more time and worst performance results would be obtained. By applying feature scaling, all the features became to the same scale and consequently they have a similar contribution to the classification algorithm [7].

Many techniques exist to scale the data and since there isn't a best choice for all applications, the type of the data and its application have to be analyzed. Common methods are min-max normalization and standardization. Min-max normalization typically rescales the data between the range [0,1] keeping its distribution and can be calculated by simply dividing each value of the feature vector by its maximum value or by subtracting each value by the minimum value of the vector and dividing by the range between the maximum and the minimum values [41]. The equation for min-max normalization is given by equation *(4.5)*:

$$X'_1 = \frac{X_1 - \min(X_1)}{\max(X_1) - \min(X_1)} \qquad (4.5)$$

where X'_1 is the new vector after min-max normalization and X_1 is the initial feature vector.

The standardization method consists in scaling the features to a common range, i.e., ensure that all dimensions of the dataset have zero mean and standard deviation equal to one. This is done by calculating the mean and standard deviation for every feature and, for every feature, subtract each element by the respective mean and divide by the standard deviation. This equation is given by equation *(4.6)*:

$$X'_1 = \frac{X_1 - \mu(X_1)}{\sigma(X_1)} \qquad (4.6)$$

Where X'_1 is the new vector after the standardization process, $\mu(X_1)$ and $\sigma(X_1)$ are, respectively, the statistical mean and standard deviation of the initial feature [42].

Min-max normalization depends on the maximum and minimum values of the vector while standardization depends on the mean and the standard deviation, meaning that both techniques are highly sensitive to outliers. This reinforces the need for a good data pre-processing and validation as already explained.

Because standardization destroys the scarcity of the data, it is not appropriated to be used when the data is sparse. However, this is not the case, since most of the features used are continuous values. In applications where the features are continuous values measured at different scales and have a different

range, standardization method is well suited, since brings all features to a common range and removes distortions due to data heterogeneity [31]. For this reason, the standardization method was applied in this work, similarly to related works [7], [24].

5 Experiments, Results and Discussion

In this chapter we perform the necessary experiments to answer our research questions. We also include a discussion of the obtained results.

We start by analyzing the electricity consumption profile of each household, in order to discover possible relations between the characteristics of the households and the suitability of the classification models.

Then, we analyze the performance of detecting occupancy through electricity consumption data. In this part, we also investigate the possibility of generalizing our classification models, i.e., the possibility of using a single and generic model do detect occupancy in multiple households, with a good classification performance.

Finally, we investigate the possibility of predicting occupancy in multiple households based solely on historical electricity consumption data.

5.1 Analysis of the electricity consumption load profile

In Figure 34 and Figure 35, it is illustrated the hourly electricity consumption curves on a typical week day and on a typical weekend day, respectively. These curves contain normalized data, so that it possible to compare each household's electricity consumption profile.

Many useful information can be inferred by simply observing these curves. Regarding the electricity consumption curves for a typical week day, at a first glance, we can observe that household 1 have an electricity consumption more stable during the day, when comparing to households 2 and 3. We can also infer that household 1 may have appliances that consume a relatively high amount of energy during the night period, such as electric heating.

In general, each household presents two or three energy consumption peaks during a typical weekday. In the morning, these peaks may be caused by the activation of electrical appliances, such coffee machine, electric water heaters, kettle, toaster. After the morning peak, most of the occupants go to the work and less electric energy is consumed. Around 19:00h, the majority of the occupants arrive to home and more electric energy is consumed, due to the interaction with electric devices, e.g. on the preparation of the dinner, watching TV, etc.

Household 2 contains the higher peak during the morning period hours. In practice, this peak occurs since the electricity intensive appliances are turned on in this period, such as: electric water heater (for showers), kettle and toaster (for the preparation of the breakfast).

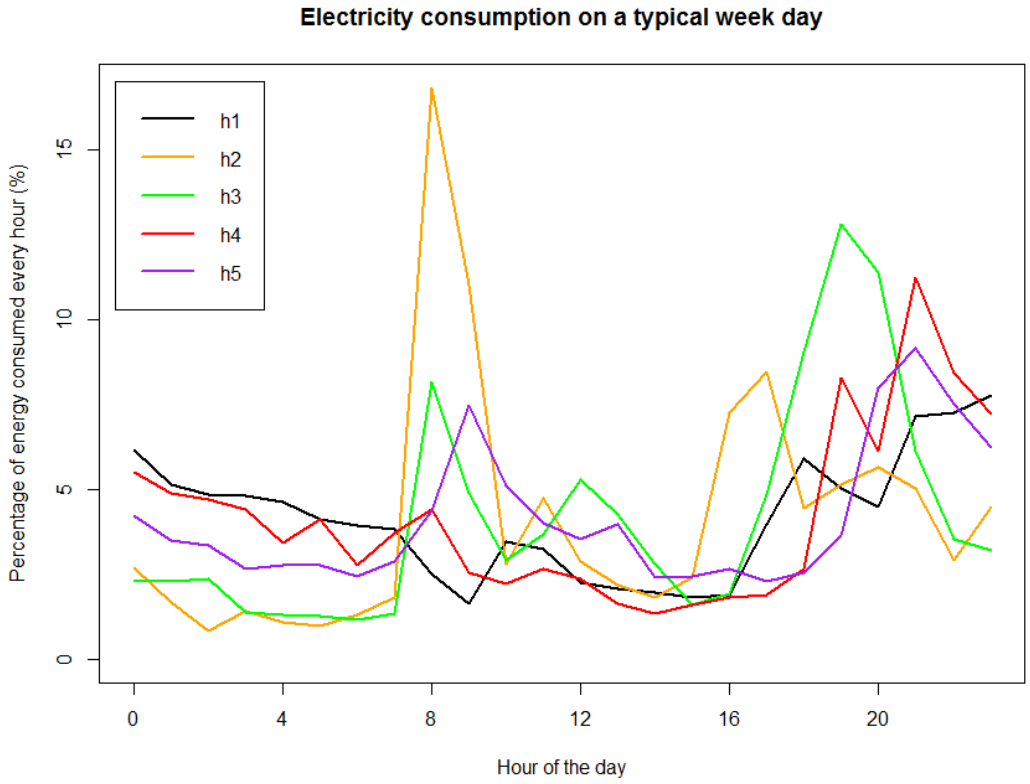


Figure 34: Hourly normalized electricity consumption curves of our five participants in a typical weekday.

By observing Figure 35, we can observe that, in weekend days, the electricity consumption of our five participants is steadier during the day. Also, the morning peaks are lower and tend to occur in a later period. This fact can be easily understandable, since most of the occupants of our five participants, do not work during the weekend. Thus, they tend to wake up later and consequently, interact later with the electrical loads.

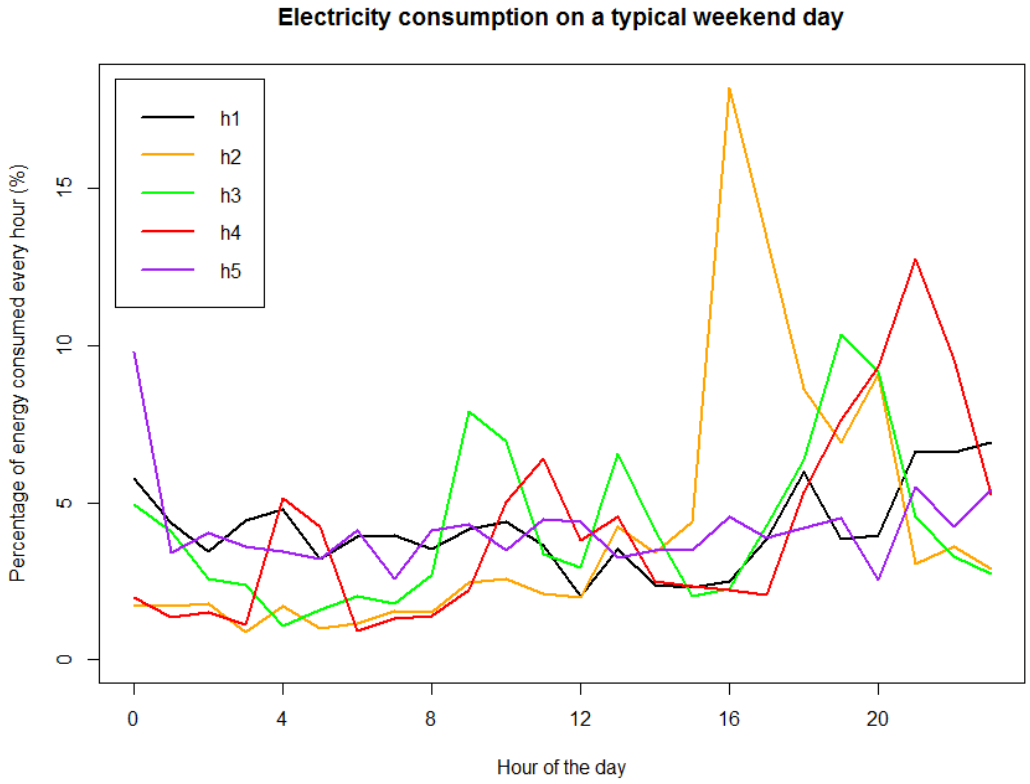


Figure 35: Hourly normalized electricity consumption curves of our five participants in a typical weekend day.

5.2 Occupancy detection from electricity consumption data

To analyze the viability of detecting occupancy from the electricity consumption data, we use the three classification algorithms explained in chapter 3 (neural network, support vector machines and random forest). In our classification analysis, we start by presenting the feature combinations that we considered relevant for our work. Then, we divide our experiments in two parts: *self-test* and *other's-test*.

In the *self-test*, we analyze the performance of detecting occupancy in each household, by using their respective classification models. In the *other's-test*, we investigate the possibility of using a single and generic classification model to detect occupancy in multiple households.

5.2.1 Feature selection

As seen in chapter 2, occupancy may be correlated with the power consumption, variability of the power consumption and also with the time. In Table 7 it was used the expert knowledge to define which features

best describe occupancy. Table 8 presents five different feature combinations that we considered relevant and propose for our analysis.

The feature *p_mean* represents the average power in every 15 minute-interval. The features *p_mean_sd* and *p_mean_sad* represents, respectively, the standard deviation and the sum of absolute differences of the average power (*p_mean*). *p_max* refers to the maximum power observed in every 15 minute-interval. The feature *time* contain the time slot in the day (varies between 1-64, representing the periods between 7:00h-23:00h). The features *workday* and *before_workday* indicate, respectively, if the day of the referred interval is a workday (between Monday and Friday) or if precedes a workday (between Sunday and Thursday). The last feature is *night_time*, which indicates if it is dark outside the house (between the sunset and the sunrise).

Table 8: Proposed feature sets for our first experiment. All features are computed over a 15-minute interval and are described in Table 7.

Feature set	Features
1	p_mean, p_mean_sad, p_mean_sd, p_max
2	p_mean, p_mean_sad, p_mean_sd, p_max, time
3	p_mean, p_mean_sad, p_mean_sd, p_max, time, workday
4	p_mean, p_mean_sad, p_mean_sd, p_max, time, workday, before_workday
5	p_mean, p_mean_sad, p_mean_sd, p_max, time, workday, before_workday, night_time

5.2.2 Detecting household's occupancy (*self-test*)

To detect occupancy separately for each household (*self-test*), we start by dividing their datasets into three parts (*training set*, *classification set* and *future set*), as shown in Figure 36. Then, we train our classification algorithms in the *training set*, using the feature combinations above described, and test the occupancy detection accuracy in the *classification set*.

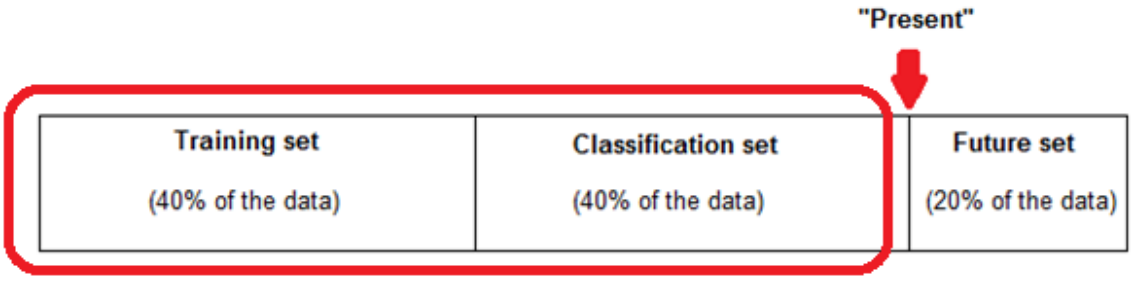


Figure 36: Data partitioning, highlighting the datasets used in the occupancy classification experiments.

In the *self-test* experiment, we first try to obtain the highest classification accuracy possible, for each household (optimization by household). Then, in order to simplify our next experiments, we repeat the process by using equal classification parameters for every model and every household.

In order to have a baseline for comparing and evaluating our classification algorithms, we introduce the Prior as a classifier that assumes that the household is always occupied or unoccupied (if the household is most of the times occupied or unoccupied, respectively). Table 9 contains the results obtained by applying the Prior classifier in the *classification set*.

Table 9: Classification accuracies obtained in the classification set by the Prior classifier, for each household. For each household, the Prior classifier simply assumes that the house is always occupied or unoccupied, according to the Prior class.

Household	Prior accuracy (%) - *classification set*	Prior class
1	90.14	Occupied
2	69.31	Occupied
3	75.22	Occupied
4	50.59	Occupied
5	66.82	Unoccupied

We can observe that in all households with exception to household 5, the Prior assumes that the house is always occupied.

5.2.2.1 Optimization by household

In this experiment, we describe, for each model and household, the model parameters and the feature set that provide the highest accuracy in the *training set*, and the classification accuracy when applying the model in the *classification set*.

For each classification algorithm, we start by defining several combinations of parameters to test. These combinations result from the possible combinations that can be made by changing the feature set from 1 to 5 and by changing the parameters of the respective algorithm, within values defined by us. Then, for each combination, we test the classification accuracy obtained in the *classification set* through cross-validations. The classification accuracy of each household represents the average of 100 runs. Finally, we register combination that provides the higher accuracy.

Neural network

In the training phase of the neural network algorithm, the number of hidden neurons was set to 2 and 3. By also changing the feature set, ten different combinations are possible to make. Table 10 presents the training results for household 5 ordered by decreasing order of accuracy.

Table 10: Results of the training phase of the neural network model in household 5. AUC represents the area under the Receiver Operating Characteristic (ROC) curve.

combination	Number of hidden neurons	Feature set	AUC - *training set*	Accuracy (%) - *training set*
1	**3**	**4**	**0.9746**	**92.96**
2	2	5	0.9663	92.62
3	2	4	0.9745	92.60
4	2	3	0.9772	92.50
5	2	2	0.9765	92.33
6	3	2	0.9654	92.06
7	3	3	0.9699	91.08
8	3	5	0.9551	91.08
9	2	1	0.8794	89.14
10	3	1	0.8338	87.32

We can observe that, by using neural network model with 3 hidden neurons and using the feature set 4, an accuracy of 92.96% was obtained in the *training set* of household 5.

For each client, Table 11 shows the optimal configuration (number of hidden neurons and feature set) during the training phase and the obtained accuracy by applying the optimal configuration in the *classification set*, for each household. We can verify that no single combination provided the best results in all households. We can also observe that all households, except household 1 with a small difference, performed better than the Prior classifier.

Table 11: Optimal number of hidden neurons and feature set, for each household, and the respective accuracy when applying the model in the classification set.

Household	Optimal number of hidden neurons	Optimal feature set	Accuracy (%) – *classification set*	Prior accuracy (%)
1	2	4	89.72	90.14
2	2	2	85.41	69.31
3	3	3	77.90	75.22
4	3	4	82.87	50.59
5	3	4	92.68	66.82

Classification summary and conclusion

For the three considered SVM models, the parameter cost was set to 0.1,1 and 10. For the SVM model with radial and polynomial kernel, the parameter gamma was to set 0.1 and 1. In the training of the random forest model, the number of trees was set to: 5,10,50,100,250,500,750 and 1000. Due to the excess of information, we do not present here the training results for the remaining households. In Table

12, we present a summary of the overall accuracy obtained (average of the results in the five household), by each classification algorithm. This table also indicates the respective tables that contain more detailed information with regard to the training of each model.

Table 12: Summary of the classification models training and their respective overall accuracy when tested in the classification set. The overall accuracy represents the average of the classification accuracy in the five households.

Model	Nº of combinations tested during the training phase	Overall accuracy (%) – *classification set*	Table containing the training results
Neural network	10	85.72	Table 11
Linear	15	81.55	Appendix A (Table_apx A-1)
Radial	30	85.97	Appendix A (Table_apx A-2)
Polynomial	30	83.61	Appendix A (Table_apx A-3)
Random forest	40	88.47	Appendix A (Table_apx A-4)

We can observe that the neural network, SVM with radial kernel and the random forest models performed the higher overall classification accuracy in the five households.

In Table 13 we show the classification accuracy and the MCC value obtained in each household by the different models. The values in bold represent the highest accuracy obtained in each household.

Table 13: Resume of the classification performance obtained (accuracy (%) and MCC) by the three considered models, for each household.

Household	neural network		SVM		random forest		Prior accuracy (%)
	Accuracy (%)	MCC	Accuracy (%)	MCC	Accuracy (%)	MCC	
1	89.72	0.05	**90.15**	0	89.96	0.11	90.15
2	85.41	0.65	87.05	0.69	**90.43**	0.77	69.31
3	77.90	0.38	75.97	0.32	**80.85**	0.44	75.22
4	82.87	0.49	85.31	0.71	**88.22**	0.78	50.59
5	92.68	0.83	92.12	0.82	**92.91**	0.84	66.82

It was possible to obtain higher classification accuracy than the Prior classifier in household 2,3,4 and 5. In household 1, the same accuracy as the Prior was obtained (90.15%). Despite of household 1 having a high classification accuracy, the high Prior accuracy indicates that the house is typically occupied (90.15% of the time). In this case, the obtained accuracy does not represent an improvement when comparing to a random guess (e.g. assuming that the household is always occupied would provide the same value). In terms of the Matthews Correlation Coefficient (MCC) values obtained, it can be seen

that the random forest model provided the highest performance in all households, confirming the results from the accuracy analysis (expect for household 1). Despite of the lower classification accuracy obtained in household 1, the MCC value of 0.11 obtained by the random forest model indicates that some of its unoccupied periods were correctly classified.

Table 14 and Table 15 contains, respectively, the false positive rate and false negative rate obtained in each household by the three models. We can verify that the lower FPR and FNR are typically obtained by the random forest classifier. Values in parenthesis represent the average misclassified minutes per day, according to the error type (false positive and false negative). We can observe that household 5 has the lower FPR (3.86%), which indicates that 25 minutes of absence, in average and per day, are wrongly classified as occupied.

Table 14: False positive rate (%) obtained for each household and model and the respective misclassified minutes, in average and per day, in the optimization by household experiment.

Household	neural network	SVM	random forest
1	97.15 (92 min)	100 (95 min)	**96.67 (91 min)**
2	31.65 (93 min)	26.61 (78 min)	**20.80 (61 min)**
3	51.7 (123 min)	56.25 (134 min)	**53.22 (127 min)**
4	53.60 (128 min)	**22.41 (106 min)**	22.51 (107 min)
5	3.93 (25 min)	4.14 (27 min)	**3.86 (25 min)**

Regarding the false negative rate, the lower value was obtained in household 4 (1.3%). This low value indicates that only 6 minutes of presence are wrongly classified as absence.

Table 15 False negative rate (%) obtained for each household and model and the respective misclassified minutes, in average and per day, in the optimization by household experiment.

Household	neural network	SVM	random forest
1	0.8 (7 min)	**0**	0.57 (5 min)
2	7.04 (47 min)	6.91 (46 min)	**4.6 (31 min)**
3	12.29 (89 min)	13.41 (97 min)	**7.92 (57 min)**
4	5.11 (37 min)	7.14 (35 min)	**1.3 (6 min)**
5	14.14 (45 min)	15.42 (49 min)	**13.58 (43 min)**

5.2.2.2 Assuming equal model parameters and feature set

We continue our next experiments with the neural network, SVM and random forest models. Regarding the SVM model, we use only the SVM model with the linear kernel in the next experiments, since it provided the best overall result, comparing to the other types of kernel.

To simplify our next experiments, we use only a single feature set and equal model parameters, for each algorithm.

Finding the optimal model parameters

To find the optimal model parameters for a certain algorithm, we choose the most frequent parameters verified in the top five results of the training phase, in the five households. Table 16 shows the top five results of the SVM model (with radial kernel) training in household 1.

Table 16: Best 5 results obtained in the SVM model (with radial kernel) training process in household 1.

Combination	Cost	Gamma	Feature set	Accuracy (%) - training phase
1	**1**	**1**	**4**	**94.61**
2	1	1	5	94.57
3	10	1	4	94.56
4	10	1	5	94.50
5	10	1	3	94.43

In the SVM model, a cost equal to 10 and a gamma equal to 1 were chosen 72% of the times, in all households. For the neural network model, 2 hidden neurons were chosen in 60% of the times. For the random forest model, the number of trees equal to 100 was chosen most frequently.

Table 17: Summary of the most frequent chosen model parameters, for each algorithm.

Algorithm	kernel	Number of hidden neurons	Number of trees	Cost	Gamma
neural network	-	2	-	-	-
SVM	radial	-	-	10	1
random forest	-	-	100	-	-

Finding the optimal feature set

Similarly, to choose the best feature set, we simply identify the feature combination that was the more often chosen within the top five results of the training phase. Figure 37 shows the relative frequencies of the various feature combinations.

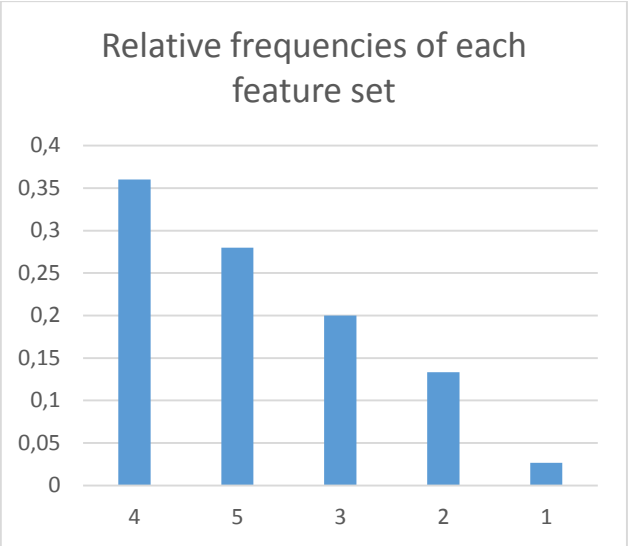


Figure 37: Relative frequencies of the choice of each feature set.

We can observe that the feature set 4 was chosen most frequently. Therefore, we continue our analysis considering this feature combination. Feature set 4 contains 7 features: *p_mean*, *p_mean_sad*, *p_mean_sd*, *p_max, time*, *workday* and *before_workday*. However, because we intend to generalize our models, i.e., to train a model using data from one household and apply it in a different household, we remove the feature *time* from the feature set 4, since this feature is highly dependent on the household (contains the occupancy patterns for a specific household according to the hour of the day). Thus, we obtain the feature set 6, defined as shown in Table 18.

Table 18: Feature set 6, used for the other's-test experiment.

Feature set	Features
6	p_mean, p_mean_sad, p_mean_sd, p_max, workday, before_workday

To analyze the effect of using equal conditions in all households, Figure 38 illustrates, for each household, a comparison between highest classification accuracy obtained in this experiment with the highest accuracy obtained in the *optimization by household* experiment. In Appendix B (Table_apx B-1) it is shown a more detailed comparison between these two experiments.

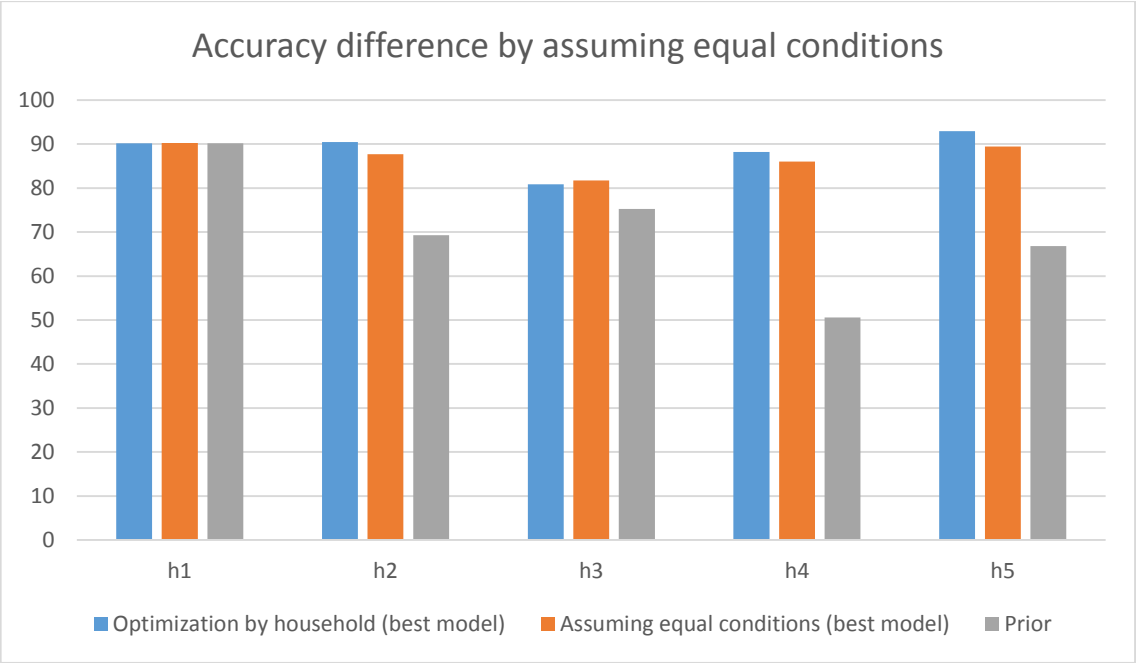


Figure 38: Comparison between the classification accuracies obtained in the optimization by household experiment and assuming equal conditions for the models.

As it was expected, we can verify that, by assuming fixed conditions for all households, a lower classification accuracy is obtained, in general. However, there are still some exceptions (see Appendix B (Table_apx B-1)), where the accuracy obtained by using this new configuration is higher than the accuracy obtained in the *optimization by household* experiment. In fact, in the *optimization by household* experiment, we could improve our results by using a different feature selection method, such as sequential forward selection (SFS) or principal component analysis (PCA). However, our main interest consists on analyzing the possibility of generalizing the classification models in other households. Therefore, we considered our approach reasonable for our objectives and we proceed our experiments using the previously mentioned model parameters and the feature set 6.

5.2.3 Detecting household's occupancy in other households (*other's-test*)

To analyze the possibility of using a single and a generic classification model to detect occupancy in multiple households, we test, for each household, the classification models of the remaining households. For example, for testing in household 1, we first train the three algorithms in the *training set* of households 2,3,4 and 5 (separately). Then, we test the classification accuracy by applying these algorithms in the *classification set* of household 1.

Testing on household 1

Household 1 contains a Prior accuracy of 90.15%. Any model providing a classification accuracy below this value, or a MCC value equal or below to zero, is not useful to detect occupancy in household 1.

Table 19 contains the classification performance in household 1. Despite of the high classification accuracies obtained (around 90% - similar to the Prior accuracy), the low MCC values indicate that the models provide almost no value, when comparing to a random choice. Thus, we can already conclude that, in household 1, no single classification model from other household performed well.

Table 19: Classification performance on household 1, by applying classification algorithms trained in the remaining households.

Train Household	neural network accuracy (%)	neural network MCC	SVM accuracy (%)	SVM MCC	random forest accuracy (%)	random forest MCC
2	**90.15**	0	89.82	0.13	89.86	0.05
3	**89.91**	0.07	88.5	0.01	85.73	0.02
4	**90.15**	0	**90.15**	0	**90.15**	0
5	**90.15**	0	90.05	0.06	89.96	0.01

***Other's-test* summary**

To summarize the results in this experiment, Table 20 shows, for each household, the highest accuracy obtained by testing models trained in the remaining households. In the previous experiments (*self-test*), the random forest model provided the best overall results. In this experiment, the best results were obtained by the neural network model. This indicates that the random forest model may be better if we want to maximize the accuracy in a given household (*self-test*) and that the neural network model may be better if our goal is to use a single model to detect occupancy in multiple households.

Table 20: Best accuracy results obtained in the other's-test experiment.

Testing household	Best model(s)	Accuracy (%)	MCC	Prior accuracy (%)	Table
1	ANN model of household 2 and 5; All models of household 4	90.15	0	90.15	Table 19
2	ANN model of household 3	82.64	0.57	69.31	Appendix C (Table_apx C-1)
3	ANN model of household 4	80.10	0.39	75.22	Appendix C (Table_apx C-2)

4	ANN model of household 3	74.85	0.51	50.59	Appendix C (Table_apx C-3)
5	ANN model of household 3	72.27	0.49	66.82	Appendix C (Table_apx C-4)

The neural network model trained in household 3 provided the highest accuracy in the remaining households, except for household 1, with a small difference. Household 3 performed the best by using the neural network model from household 4.

In households 2,3,4 and 5, the accuracies obtained are higher than the respective Prior accuracy and the lower value of MCC is 0.39, which clearly indicates that it is possible to use a single generic model to detect occupancy in multiple households. The best accuracy improvement, comparing to the Prior, was achieved in household 4 (48% improvement), which is in agreement with the highest value of MCC observed (0.51).

For a more detailed analysis, we present, in Table 21, the false positive and false negative rates for each household and for each model correspondent in Table 20. The values in parentheses represent the number of minutes per day, in average, that are misclassified for the respective error type (false positive or false negative). For household 2, the table can be interpreted in the following manner: the neural network model of household 3 applied in the household 2, would misclassify in average and per day, 146 minutes of unoccupied periods as occupied (false positives) and 20 minutes of occupied periods as unoccupied (false negatives).

Table 21: False positive rate and false negative rate (%) for best results of the other's-test experiment.

Household	FPR (%)	FNR (%)
1	100 (95 min)	0
2	49.70 (146 min)	3.41 (20 min)
3	64.58 (154 min)	5.2 (37 min)
4	39.03 (185 min)	11.60 (56 min)
5	34.90 (224 min)	13.30 (43 min)

In households 1,2 and 3, a higher FPR is verified while a higher FNR is observed in households 4 and 5. In household 1, a FPR equal to 100% and a FNR equal to 0% indicate that the model assumed that the house was always occupied. In order words, all unoccupied periods were misclassified as occupied (100% false positive rate) and no single occupied period was correctly classified (false negative rate equal to 0%).

These results shows that false positive errors are the most frequent errors verified in all households by using a classification model that was trained in other household. It means that the model has more difficult in detecting the unoccupied periods.

To summarize the results obtained in the occupancy detection experiments, Figure 39 illustrates the highest classification accuracy obtained in each of the three previous experiments:

1) Optimization by household (*self-test*): for each household, are chosen the optimal model parameters and feature set;
2) Optimization with assumed parameters (*self-test*): the same model parameters and feature set is applied in every household;
3) Using other household's models (*other's-test*): for each household, are applied the models of the remaining households and is selected the best model.

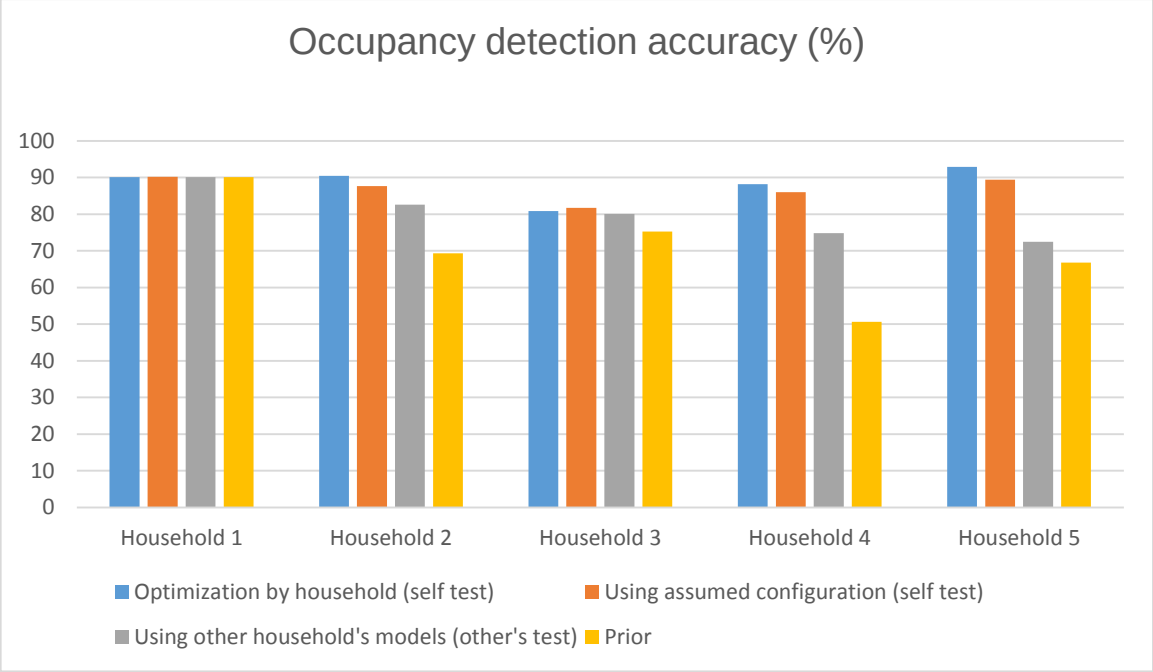


Figure 39: Comparison between the classification results obtained in the self-test and in the other's-test.

5.3 Occupancy prediction

To analyze the possibility of predicting occupancy using solely electricity consumption data, we first use a classification algorithm to generate occupancy data from electricity consumption data. Then, we use the Presence Probabilities Simplified (PPS) to construct a prediction timetable based on the occupancy data that was generated by the classification algorithm. The classification algorithm is trained in the *training set* and is applied in the *classification set*. The PPS algorithm uses occupancy data with regard to the *classification set* and the prediction accuracy is calculated by applying the prediction timetable in the *future set*, as shown in Figure 40.

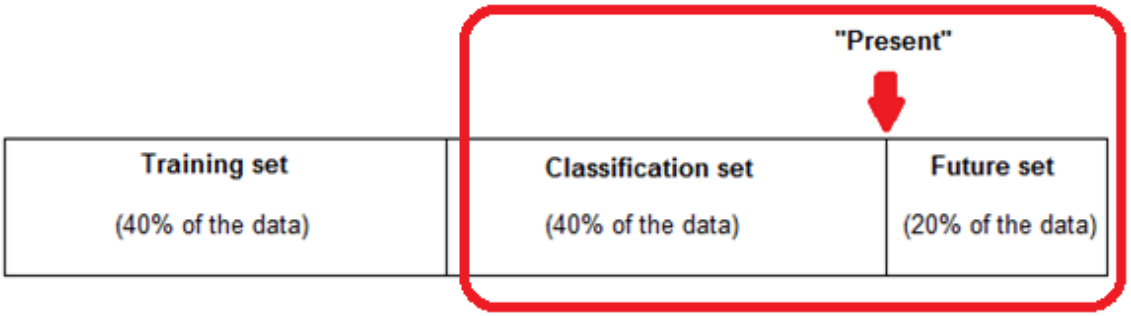


Figure 40: Data partitioning, highlighting the datasets used for occupancy prediction experiments.

Again, we divide our occupancy prediction experiments in two parts: *self-test* and *other's-test*. In the *self-test* we analyze the occupancy prediction separately in each household. In the *other's test*, we investigate the possibility of predicting occupancy in multiple households, using solely their electricity consumption data.

5.3.1 Using occupancy data generated by a classification model (type 1) and ground truth occupancy (type 2) (*self-test*)

To analyze the prediction accuracy separately in each household (*self-test*), we start by creating two types of prediction timetables. Then, we test the prediction accuracy of each type of timetable on the *future set*.

The first type of prediction tables (type 1) is computed by using the occupancy data that was generated by the best classification algorithm in the previous experiment (random forest for household 1,2,4 and 5 and the neural network model for household 3). The second type of prediction table (type 2) represent the timetable that is constructed by using the ground truth occupancy data and is used as a baseline for our analysis. Obviously, we expect to obtain a higher prediction accuracy by using the prediction timetables type 2, since they do not incur on the error due to the classification process.

Figure 41 illustrates the processes that we perform in order to construct and apply the prediction timetables type 1 and 2, for each household.

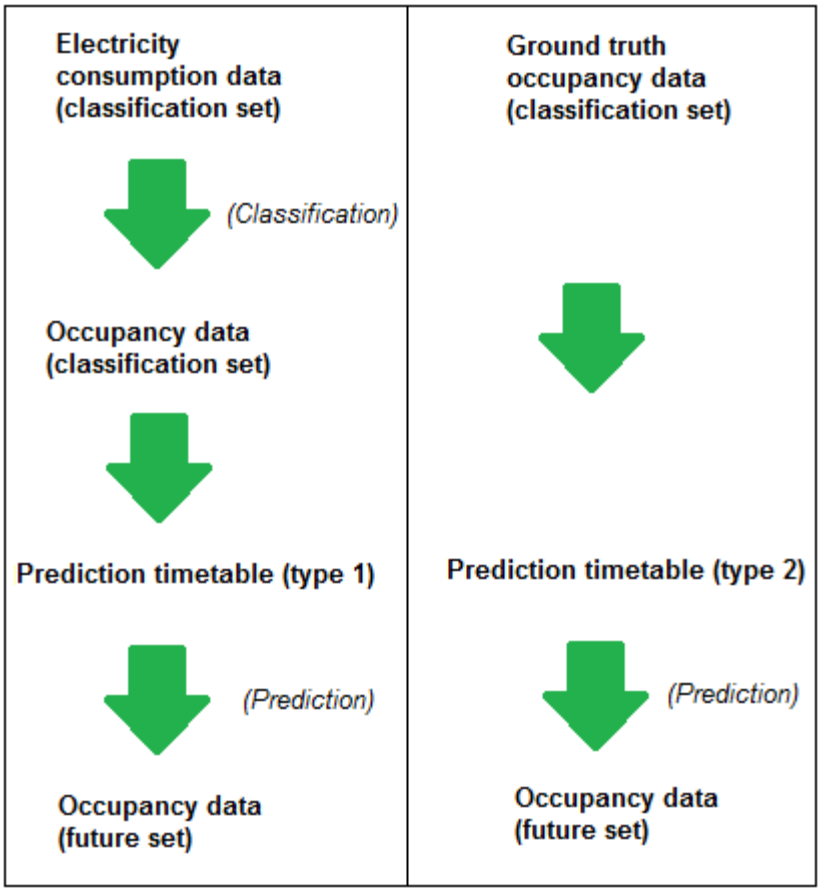


Figure 41: Schematic representation of the construction and application of the two types of prediction timetables.

Examples of these tables can be seen in the Appendix D. For household 2, in Appendix D, Table_apx D-1 and Table_apx D-2 illustrates the prediction timetables type 1 and type 2, respectively. From table Table_apx D-3 of the same appendix, it is possible to observe, for each interval of the day and day of the week, the number of respective periods existent in the *classification set*, that were used for constructing this prediction timetable.

***Self-test* prediction summary**

Table 22 contains the results obtained, for each household, by applying the two types of prediction tables. In households 2, 4 and 5, for both prediction tables type 1 and 2, the prediction accuracy was higher than the Prior accuracy and positive MCC values were obtained. This result indicate that our prediction algorithm allowed to predict occupancy in these households, but provided no value for households 1 and 3.

Table 22: Occupancy prediction results obtained by applying the prediction timetables type 1 and 2 on the future set of each household.

Household	PP table – type 1		PP table – type 2		Prior accuracy (%)
	Accuracy (%)	MCC	Accuracy (%)	MCC	
1	**88.74**	0	88.18	0	**88.74**
2	78.42	0.37	**80.02**	0.44	74.11
3	83.11	0	76.83	-0.03	**85.27**
4	59.85	0.24	**66.89**	0.34	53.66
5	76.27	0.39	**80.77**	0.52	70.54

A more detailed analysis can be seen in the Appendix D (Table_apx D-4). In Appendix D (Table_apx D-4), it is shown the false positive and the false negative rates for the occupancy prediction through the prediction timetable type 1. Also, Appendix D (Figure_apx D-1, Figure_apx D-2 and Figure_apx D-3) illustrate, for households 2, 4 and 5, their respective Receiver Operating Characteristic (ROC) curves.

We proceed to our next experiment with households 2, 4 and 5, since their prediction timetables type 1 provided a higher prediction accuracy than the Prior classifier.

5.3.2 Using occupancy data generated by a classification model (type 1) from other household (*other's-test*)

Our final experiment consists on investigating the possibility of predicting occupancy in multiple households using solely their electricity consumption data. More specifically, our objective is to analyze the viability of predicting occupancy by using the prediction timetable type 1, but now, constructed with the best classification models from other households (*other's-test*).

For example, to predict occupancy in household 2, we first apply the neural network model of household 3 in household 2 to obtain occupancy data from electricity consumption data (classification). Then, we construct a prediction timetable type 1 by using this occupancy data generated by the classification algorithm.

In section 5.2.3 (*other's-test* experiment), we verified that the neural network model of household 3 provided the highest occupancy detection accuracy in the remaining households. Thus, the prediction timetable type 1 of households 2, 4 and 5 are constructed by using occupancy data generated by this model.

Other's-test prediction summary

After constructing the prediction timetables for household 2, 4 and 5, we test their occupancy prediction performance in the *future set*. Table 23 summarizes the results obtained in this experiment.

Table 23: Occupancy prediction accuracy obtained by using the prediction timetable type 1 (constructed by using occupancy data generated by the neural network model of household 3) and the respective MCC value.

Household	Prediction accuracy (%)	MCC	Prior accuracy (%) – *future set*
2	75.42	0.20	74.11
4	58.91	0.24	53.66
5	61.73	0.31	70.54

Despite of the not so high prediction accuracies, we can observe that a prediction accuracy higher than the Prior was obtained in households 2 and 4 and that all of the three households presented a positive MCC value, which indicates that the prediction algorithm performed better than a random choice in all households.

A more detailed analysis of each error type can be seen in Table 24. We can observe that the false positive rate (false occupancy detection) is the most common error verified in all households, which indicates that the algorithm fails mostly in predicting the unoccupied periods. Household 2 provided the lowest false negative rate. In terms of misclassified minutes, the prediction algorithm misclassifies 215 minutes per day (in average) as occupied and only misclassifies 21 minutes as unoccupied.

Table 24: False positive and false negative rates obtained in households 2, 4 and 5, by using the prediction timetable constructed with occupancy data generated by the neural network model of household 3.

Household	FPR (%)	FNR (%)	Prior accuracy (%)
2	86.59 (215 min)	3 (21 min)	74.11
4	62.41 (322 min)	16.40 (73 min)	53.66
5	45.74 (310 min)	20.4 (58 min)	70.54

In this work, we do not intend to analyze the cost of each type of error, however, if our prediction was intended to control automatically a thermostat, for example, our algorithms would provide more comfort than energy savings to the users (considering a threshold of 0.5).

If the user prefers energy savings rather than comfort, than we can increase the threshold of the decision rule in order to decrease the false positive rate, according to the Receiver Operating Characteristic (ROC) curve.

In Figure 42, Figure 43 and Figure 44 it is shown, for households 2, 4 and 5, the ROC curves correspondent to their prediction. The blue line represents the curve with regard to the prediction timetable type 2 (using the ground truth occupancy data) and the green line represents the curve with

respect to the prediction timetable type 1 (*other's-test*). Each curve contains a red point, which corresponds to the point where the threshold is equal to 0.5.

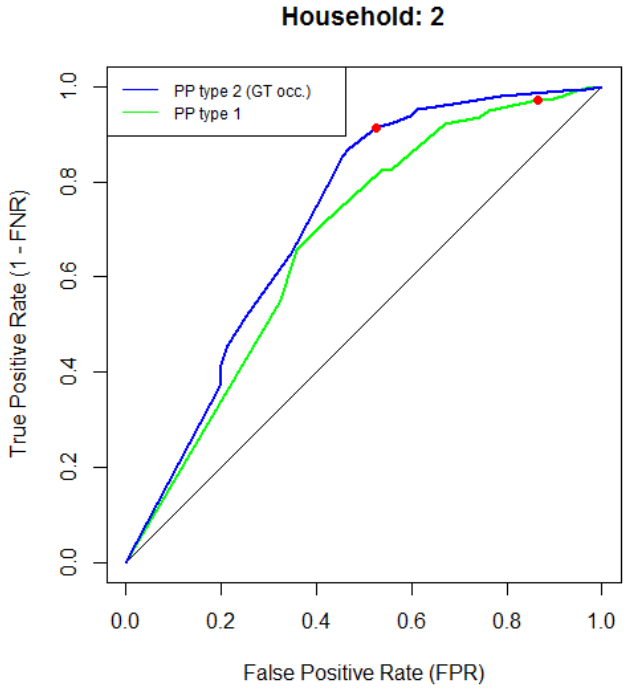


Figure 42: ROC curves for occupancy prediction in household 2. The green line corresponds to the prediction timetable constructed with data generated by the neural network model of household 3.

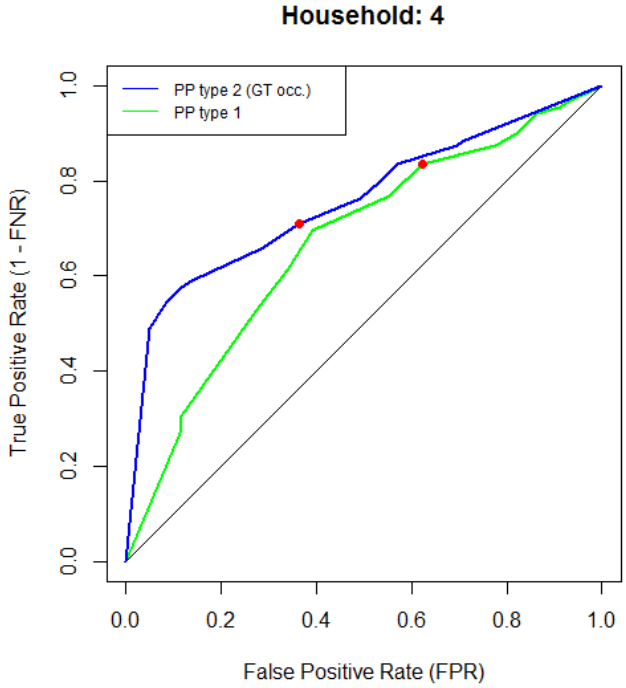


Figure 43: ROC curves for occupancy prediction in household 4. The green line corresponds to the prediction timetable constructed with data generated by the neural network model of household 3.

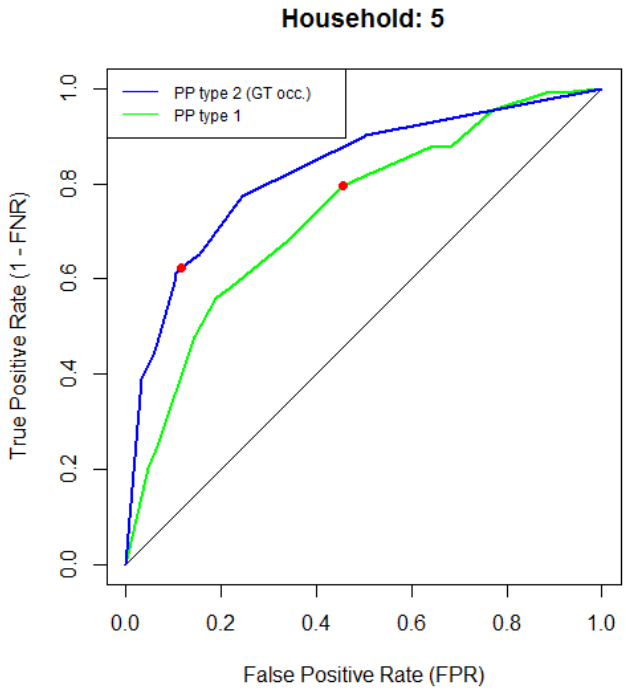


Figure 44: ROC curves for occupancy prediction in household 5. The green line corresponds to the prediction timetable constructed with data generated by the neural network model of household 3.

From the curves above illustrated, we can observe that: 1) all prediction timetables provided a higher value than a random choice and 2) the prediction timetables type 2 provided a better prediction performance than the prediction timetables type 1, for every threshold chosen (as expected).

If we compare these ROC curves with the respective ROC curves of the previous experiment, shown in Appendix D (Figure_apx D-1, Figure_apx D-2 and Figure_apx D-3), we can observe that, by using a generic classification model, a higher false positive rate is obtained in the three households.

To summarize the results obtained in the occupancy prediction experiment, Figure 45 illustrates the occupancy prediction accuracy obtained in households 2, 4 and 5, in the two previous experiments:

1) Prediction by household/*self-test* (type 1): for each household, we apply the respective best classification model to generate occupancy data. Then we construct the prediction timetable using this generated occupancy data and test the prediction accuracy.
2) Prediction by household/*self-test* (type 2): for each household, we construct the prediction timetable based on the ground truth occupancy data. Then, we test the prediction accuracy.
3) Prediction in other households/*other's-test* (type 1): we use the best overall classifier (neural network model of household 3) to detect occupancy in households 2, 4 and 5. Then, we construct the prediction timetable using this generated occupancy data and test the prediction accuracy.

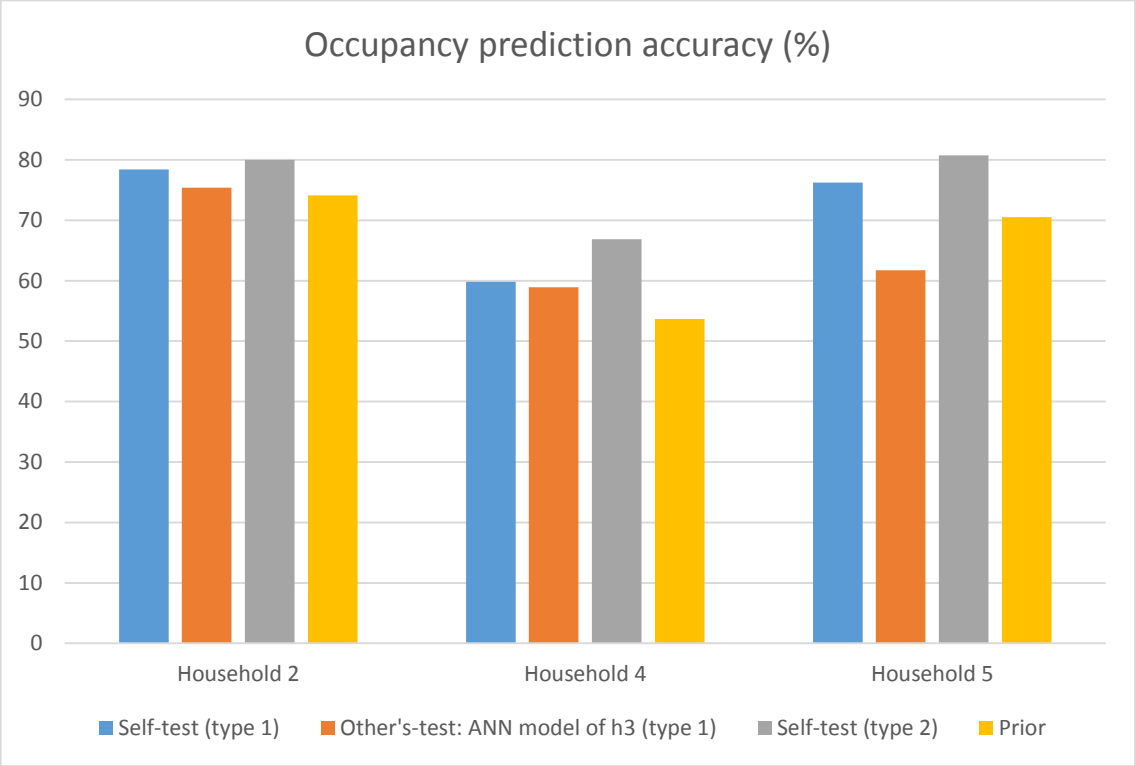


Figure 45: Comparison between the occupancy prediction accuracy (%) obtained by using the prediction timetable type 1 constructed by using occupancy data generated from the respective household's models and the neural network model of household 3.

6 Conclusion

Occupancy monitoring and prediction systems can provide multiple benefits with regard to safety, comfort and energy savings. In this work, we analyzed the viability of detecting and predicting household's occupancy through solely the electricity consumption data.

To perform our experiments, we first installed the EDP re:dy service in five households, in order to collect the necessary data for our experiments. Electricity consumption and occupancy data was extracted, with a granularity of 15 minutes, during a period of approximately 5 months.

In this chapter, we present our conclusions concerning the defined research questions in the introduction: 1) How accurate can occupancy be monitored through electricity consumption data? 2) Is it possible to use a single classification model to monitor occupancy in multiple households? In which conditions? 3) Is it possible to predict occupancy by using solely electricity consumption data?

6.1 Occupancy detection from electricity consumption data

In section 5.2.2, we analyzed how accurate can occupancy be monitored through electricity consumption data. To do this, we first extracted 8 features (most of them related with the electricity consumption). Then, 5 different feature combinations were used to train three classification algorithms (neural network, support vector machines and random forest), for each household. We observed that the three classification models allowed to obtain a classification accuracy of up to 92% in households with low occupancy level. In [6], a classification accuracy of up to 94% was obtained by using the support vector machines algorithm. However, the authors analyzed 35 features and used the principal component analysis as feature selection method.

For households with high levels of occupancy (e.g. more than 75%), we verified that the obtained accuracy does not represent a relevant improvement when comparing to a random choice (e.g. assuming that the household is always occupied). We consider that in households with a high occupancy level, more periods of low power consumption and power variability may exist when the household is occupied. Thus, a lower correlation between occupancy and the electricity consumption may exist, which reduces the occupancy detection accuracy.

To analyze the viability of using a single classification model to monitor occupancy in multiple households, we tested, for each household, the classification models trained in the remaining household, as shown in section 5.2.3. We observed that a single classification model can be used to monitor occupancy in 4 out of 5 participants. The only exception is for the household with the higher level of occupancy (more than 90%), which provided a classification accuracy equal to the Prior method. Excluding this household, a classification accuracy of up to 82% was obtained in this experiment. This result is not fabulous, but definitely indicates that it is possible to generalize classification algorithms and motivates researchers to investigate in more detail this area.

To understand the conditions in which a classification model would generalize in multiple households, a study with a higher number of participants would be necessary. However, the observed results may indicate that a classification model may be suitable to households with similar characteristics, i.e., households with a similar level of occupancy, number of occupants, occupancy patterns, type of heating and similar behaviors with regard to electric devices.

6.2 Occupancy prediction from electricity consumption data

To predict occupancy through the electricity consumption data, we first used a classification model to obtain occupancy data from electricity consumption data. Then, we used the Presence Probabilities Simplified (PPS) algorithm to create a prediction timetable from the generated occupancy data.

As mentioned in chapter 2, schedule-based approaches are limited to a prediction accuracy of 90%, since they rely only on historical occupancy data. In our work, we expect to obtain lower values, since we are using occupancy data that is generated by a classification algorithm (incur on the classification error).

By using the respective classification model in each household, we observed that 3 out of 5 households presented a higher prediction accuracy than the Prior method, and that an accuracy of up to 78% was possible to obtain. In households with high occupancy level (e.g. more than 85%), the prediction accuracy was lower than the Prior method, as expected.

In the second experiment, we used a single classification model to generate occupancy data. Then, we constructed prediction timetables based on this data. We observed that 2 out of 3 households provided a higher prediction accuracy than a Prior classifier and that a classification accuracy of up to 75% was possible to obtain. This result is not magnificent and may not be sufficient to justify the use of our approach to predict occupancy, however, it indicates that it is possible to predict occupancy in multiple household by using a single classification model.

To summarize, we consider that two conditions are fundamental in order to make it viable to predict occupancy by using solely electricity consumption data and a schedule-based approach: 1) It is necessary to have a classification model that generates occupancy data with a high accuracy (very similar to the ground truth occupancy) and 2) Households should have similar patterns of occupancy during each weekday and hour of the day.

6.3 Future work

To increase the occupancy monitoring accuracy obtained through smart meter data, we consider that households could be grouped (e.g. through a clustering method) according to their similarities. Then, a different classification model could be used to monitor occupancy in each group of households. Also, different models could be used according to the season, since the electric consumption behavior of the occupants tends to be different in the summer and in the winter.

Another possibility to increase the occupancy monitoring accuracy is by combining smart meter data with data from mobile phones. For example, in addition to the electricity consumption data, obtained from smart meters, it could be also used data from GPS systems and wireless network traffic. In times where occupants are at home but no electrical devices are consuming, the GPS system could correctly identify presence.

This improvement in the occupancy monitoring accuracy by using mobile phones would allow to obtain a higher occupancy prediction accuracy. Furthermore, occupancy prediction accuracy could be even

more improved by combining schedule-based approaches with context-aware approaches (hybrid approaches).

Context-aware approaches use information about the current position, activity and environmental factors. In a hybrid approach, for example, the context aware component could improve the prediction timetable performance (schedule component) by analyzing the current traffic conditions and predicting the arrival time of each occupant.

7 Bibliography

[1] J. Lu, T. Sookoor, V. Srinivasan, G. Gao e B. Holben, “The Smart Thermostat: Using Occupancy Sensors,” em *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, Zürich, 2010.

[2] R. Yang and M. W. Newman, "Learning from a learning thermostat: lessons for intelligent systems for the home," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 2013.

[3] D. Chen, S. Barker, A. Subbaswamy, D. Irwin and P. Shenoy, "Non-Intrusive Occupancy Monitoring using Smart Meters," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, 2013.

[4] W. Kleiminger and C. Beckel, "Occupancy Detection from Electricity Consumption Data," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, 2013.

[5] K. Wilhelm, B. Christian and S. Silvia, "Household Occupancy Monitoring Using Electricity Meters," *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing,* pp. 975-986, 2015.

[6] K. WILHELM, "Occupancy Sensing and Prediction for Automated Energy Savings," 2015.

[7] A. Akbar, M. Nati and F. Carrez, "Contextual occupancy detection for smart office by pattern recognition of electricity consumption data," in *2015 IEEE International Conference on Communications (ICC)*, 2015.

[8] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei and T. Weng, "Occupancy-driven energy management for smart building automation," in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, 2010.

[9] M. Gupta, S. Intille and K. Larson, "Adding GPS-Control to Traditional Thermostats: An Exploration of Potential Energy Savings and Design Challenges," *International Conference on Pervasive Computing,* vol. 5538, pp. 95-114, 2009.

[10] J. M. Abreu, F. C. Pereira and P. Ferrão, "Using pattern recognition to identify habitual behavior in residential electricity," *Energy and Buildings,* vol. 49, p. 479–487, 2012.

[11] T. A. Nguyen and M. Aiello, "Energy intelligent buildings based on user activity: A survey," *Energy and Buildings,* vol. 56, p. 244–257, 2013.

[12] H. Souri, A. Dhraief, S. Tlili, K. Drira and A. Belghith, "Smart Metering Privacy-preserving Techniques in a Nutshell," *Procedia Computer Science,* vol. 32, pp. 1087-1094, 2014.

[13] L. Yang, K. Ting and M. Srivastava, "Inferring occupancy from opportunistically available sensor data," in *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2014.

[14] A. W. Whitney, "A Direct Method of Nonparametric Measurement Selection," *IEEE Transactions on Computers,* vol. 20, pp. 1100-1103, 1971.

[15] G. W. HART, "Nonintrusive Appliance Load Monitoring," in *PROCEEDINGS OF THE IEEE*, 1992.

[16] S. Gupta, M. S. Reynolds and S. N. Patel, "ElectriSense: Single-point Sensing Using EMI for Electrical Event Detection and Classification in the Home," in *12th ACM International Conference on Ubiquitous Computing*, 2010.

[17] K. C. ARMEL, A. GUPTA, G. SHRIMALI and A. ALBERT, "Is disaggregation the holy grail of energy efficiency? The case of electricity," *Energy Policy,* vol. 52, p. 213–234, 2013.

[18] M. C. Mozer, L. Vidmar and R. H. Dodier, "The Neurothermostat: Predictive Optimal Control of Residential Heating Systems," *Advances in neural information processing systems,* 1997.

[19] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," in *Proceedings of the 2007 conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, 2007.

[20] J. Scott, A. J. Brush, J. Krumm, B. Meyers, M. Hazas, S. Hodges and N. Villar, "PreHeat: Controlling Home Heating Using Occupancy Prediction," in *Proceedings of the 13th international conference on Ubiquitous computing*, 2011.

[21] J. Krumm and A. J. Bernheim Brush, "Learning Time-Based Presence Probabilities," in *Pervasive Computing*, 2011.

[22] X. Liang, T. Hong e G. Q. Shen, “Occupancy data analytics and prediction: A case study,” em *Building and Environment*, Elsevier, 2016, pp. 179-192.

[23] kdnuggets, "CRISP-DM, still the top methodology for analytics, data mining, or data science projects," [Online]. Available: http://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html. [Accessed 25 03 2017].

[24] A. Fleury, M. Vacher and N. Noury, "SVM-Based Multimodal Classification of Activities of Daily Living in Health Smart Homes: Sensors, Algorithms, and First Experimental Results," in *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*, IEEE, 2010, pp. 274 - 283.

[25] S. Raschka, "Machine Learning FAQ," [Online]. Available: https://sebastianraschka.com/faq/docs/evaluate-a-model.html. [Accessed 04 04 2017].

[26] M. Swain, S. K. Dash, S. Dash and A. Mohapatra, "An Approach for IRIS Plant Classification Using Neural Network," *International Journal on Soft Computing,* vol. 3, 2012.

[27] T. Durieux, "Exploring the use of artificial neural network based subgrid scale models in a variational multiscale formulation," 2015.

[28] S. K e S. Sasithra, “REVIEW ON CLASSIFICATION BASED ON ARTIFICIAL NEURAL NETWORKS,” *International Journal of Ambient Systems and Applications (IJASA),* vol. 2, 2014.

[29] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification," *Journal of Biomedical Informatics,* p. 352–359, 2003.

[30] Y. LeCun, L. Bottou, G. Orr and K.-R. Müller, "Effiicient BackProp," in *Neural Networks: Tricks of the Trade*, Springer-Verlag, 1998, pp. 9-50.

[31] A. Ben-Hur and J. Weston, "A User's Guide to Support Vector Machines," in *Methods in molecular biology*, 2010, pp. 223-239.

[32] stackexchange, "Use Gaussian RBF kernel for mapping of 2D data to 3D," [Online]. Available: http://stats.stackexchange.com/questions/63881/use-gaussian-rbf-kernel-for-mapping-of-2d-data-to-3d. [Accessed 06 04 2017].

[33] E. Goel and E. Abhilasha, "Random Forest: A Review," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 7, no. 1, pp. 251-257, 2007.

[34] Anuradha and G. Gupta, "A Self Explanatory Review Of Decision Tree," in *IEEE International Conference on Recent Advances and Innovations in Engineering*, 2014.

[35] S. B. Kotsiantis, "Decision trees: a recent overview," *Artificial Intelligence Review,* vol. 39, no. 4, p. 261–283, 2011.

[36] "Diving into data," 19 10 2014. [Online]. Available: http://blog.datadive.net/interpreting-random-forests/. [Accessed 17 5 2017].

[37] G. Biau, "A Random Forest Guided Tour," *TEST,* vol. 25, no. 2, p. 197–227, 2016.

[38] ArcToolbox, "Fit Random Forest Model," [Online]. Available: http://code.env.duke.edu/projects/mget/export/HEAD/MGET/Trunk/PythonPackage/dist/TracOnlineDocumentation/Documentation/ArcGISReference/RandomForestModel.FitToArcGISTable.html. [Accessed 27 03 2017].

[39] J. Weiss, "Lecture 22—Wednesday, November 10, 2010," [Online]. Available: https://www.unc.edu/courses/2010fall/ecol/563/001/docs/lectures/lecture22.htm. [Accessed 02 04 2017].

[40] J. Kelly and W. Knottenbelt, "Neural NILM: Deep Neural Networks," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, 2015.

[41] A. Jain, K. Nandakumar and A. Ross, "Score normalization in multimodal biometric systems," in *Pattern Recognition*, Elsevier, 2005, p. 2270–2285.

[42] I. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Elsevier, 2005.

Appendix A Results of the classification algorithms training

Table_apx A-1: Classification results of the SVM model with linear kernel, for each household.

Household	Optimal cost	Optimal feature set	Accuracy (%)	Prior accuracy (%)
1	0.1	1	90.15	90.15
2	10	5	75.6	69.31
3	10	5	70.34	75.22
4	10	5	83.53	50.59
5	0.1	5	88.13	66.82

Table_apx A-2: Classification results of the SVM model with radial kernel, for each household.

Household	Optimal cost	Optimal gamma	Optimal feature set	Accuracy (%)	Prior accuracy (%)
1	1	1	4	89.63	90.15
2	10	1	2	87.05	69.31
3	1	1	5	75.97	75.22
4	10	1	3	85.08	50.59
5	1	0.1	4	92.12	66.82

Table_apx A-3: Classification results of the SVM model with polynomial kernel, for each household.

Household	Optimal cost	Optimal gamma	Optimal feature set	Accuracy (%)	Prior accuracy (%)
1	0.1	1	5	89.58	90.15
2	0.1	1	5	78.88	69.31
3	0.1	1	5	74.28	75.22
4	0.1	1	4	85.31	50.59
5	0.1	1	5	90.00	66.82

Table_apx A-4: Classification results of the random forest model, for each household.

Household	Optimal trees	Optimal feature set	Accuracy (%)	Prior accuracy (%)
1	250	5	89.96	90.15
2	100	2	90.43	69.31
3	1000	5	80.85	75.22
4	500	4	88.22	50.59
5	50	4	92.91	66.82

Appendix B Classification results by assuming equal model parameters and feature set

Table_apx B-1: Classification accuracy (%) obtained by using the assumed parameters and the feature set 6. Values in parenthesis represent the accuracy in the optimization by household experiment

Household	neural network	SVM	random forest	Prior accuracy (%)
1	89.82 (89.72)	89.86 (90.15)	**90.24** (89.96)	90.15
2	82.97 (85.41)	82.73 (87.05)	**87.66** (90.43)	69.31
3	**81.75** (77.90)	73.53 (75.97)	78.18 (80.85)	75.22
4	82.73 (82.87)	81.65 (85.31)	**86.02** (88.22)	50.59
5	85.92 (92.68)	84.89 (92.12)	**89.44** (92.91)	66.82

Appendix C Classification results by using other household's classification models

Table_apx C-1: Classification results by testing the other household's models in household 2.

Train Household	neural network accuracy (%)	neural network MCC	SVM accuracy (%)	SVM MCC	random forest accuracy (%)	random forest MCC
1	67.90	0.34	**69.45**	0.12	67.76	0.18
3	**82.64**	0.57	77.15	0.43	77.48	0.46
4	**80.62**	0.52	77.10	0.43	79.45	0.48
5	65.18	0.35	67.71	0.11	**74.00**	0.43

Table_apx C-2: Classification results by testing the other household's models in household 3.

Train Household	neural network accuracy (%)	neural network MCC	SVM accuracy (%)	SVM MCC	random forest accuracy (%)	random forest MCC
1	64.57	0.17	**72.78**	0.12	66.31	0.25
2	72.69	0.15	**75.97**	0.18	73.72	0
4	**80.10**	0.39	74.28	0.16	76.72	0.21
5	**69.87**	0.25	67.57	0.25	68.28	0.11

Table_apx C-3: Classification results by testing the other household's models in household 4.

Train Household	neural network accuracy (%)	neural network MCC	SVM accuracy (%)	SVM MCC	random forest accuracy (%)	random forest MCC
1	64.29	0.33	55.33	0.13	**70.20**	0.42
2	54.34	0.20	**72.45**	0.48	55.75	0.17
3	**74.85**	0.51	73.96	0.5	72.69	0.46
5	**72.60**	0.54	68.14	0.4	70.62	0.41

Table_apx C-4: Classification results by testing the other household's models in household 5.

Train Household	neural network accuracy (%)	neural network MCC	SVM accuracy (%)	SVM MCC	random forest accuracy (%)	random forest MCC
1	68.89	0.35	47.35	0.21	**71.98**	0.38
2	33.18	0	33.41	0	**38.95**	0.14
3	**72.27**	0.49	50.54	0.28	44.63	0.15
4	**53.40**	0.32	52.28	0.3	32.85	-0.08

Appendix D Occupancy prediction timetables and ROC curves

Table_apx D-1: Prediction timetable type 1 of household 4, constructed by using occupancy data generated by the random forest model. wday represents the weekday, which vary between 1 (Sunday) to 7 (Saturday). Time indicate the time of the day, where 1 corresponds to 1 corresponds to the interval of 7:00h-7:15h and 64 the interval of 22:45h-23:00h.

time	wday_1	wday_2	wday_3	wday_4	wday_5	wday_6	wday_7
1	1	0.8	0.8	0.75	0.333	0.833	1
2	0.8	0.6	1	0.5	1	0.667	0.8
3	0.8	0.8	1	1	1	0.833	0.6
4	0.4	0.8	1	1	1	1	0.8
5	0.6	0.8	1	1	1	1	1
6	0.8	0.8	1	1	0.75	1	0.6
7	0.75	0.8	1	1	1	1	0.4
8	0.75	1	1	1	0.75	1	0.6
9	1	1	1	0.8	0.75	1	0.8
10	1	0.8	1	0.8	0.5	0.667	1
11	1	0.6	0.6	0.8	0.5	0.5	1
12	0.8	0.6	0.2	0.6	0.8	0.667	1
13	0.8	0.5	0.4	0.6	0.6	0.333	1
14	0.8	0.5	0.2	0	0.6	0.5	1
15	0.8	0.5	0.2	0.333	0.8	0.333	0.8
16	0.8	0.75	0.2	0.667	0.8	0.333	0.75
17	0.8	0.75	0	0.333	0.6	0.333	0.75
18	0.5	0.75	0	0.333	0.6	0.333	0.75
19	0.5	0.667	0.2	0.333	0.6	0.167	0.6
20	1	0	0	0	0.6	0	0.4
21	0.75	0	0.2	0	0.4	0	0.2
22	0.75	0.25	0.2	0	0.8	0.167	0.4
23	0.75	0.2	0.4	0	0.6	0	0.2
24	0.75	0	0.2	0	0.8	0	0.4
25	0.5	0	0.2	0	0.4	0	0.6
26	0.75	0.2	0.2	0	0.6	0.2	0.6
27	0.8	0.4	0.2	0	0.4	0	0.8
28	0.6	0	0	0	0.4	0	0.4
29	1	0.2	0.2	0	0.2	0	0.4
30	0.8	0.2	0.2	0	0.4	0	0.4

31	0.6	0.2	0.2	0	0.2	0	0.6
32	0.6	0	0.2	0	0.2	0	0.4
33	0.667	0	0.2	0	0.4	0	0.6
34	0.667	0.2	0	0	0.6	0	1
35	0.667	0.2	0.2	0	0.4	0	0.8
36	0.667	0.2	0.2	0.25	0.6	0	0.6
37	0.5	0.2	0	0	0.4	0	0.6
38	0.667	0	0.2	0	0.4	0.2	0.6
39	0.5	0.2	0.2	0.25	0.6	0.2	0.6
40	0.5	0	0.4	0.25	0.8	0.25	0.6
41	0.833	0.25	0.2	0	0.6	0.25	0.6
42	0.833	0.5	0.25	0	0.8	0.25	0.6
43	0.833	0.25	0.25	0	0.6	0.5	0.6
44	0.833	0.2	0.25	0.25	0.8	0.5	1
45	0.667	0.2	0.2	0.25	1	0.6	0.8
46	0.667	0.167	0.6	0.25	1	0.6	0.8
47	0.833	0.2	0.8	0.25	1	0.6	0.6
48	0.8	0.2	1	0.5	1	0.8	0.6
49	0.8	0.5	1	0.75	1	1	0.6
50	0.8	0.5	1	0.75	1	1	0.8
51	1	0.667	1	0.75	1	1	0.6
52	1	1	1	1	1	1	0.75
53	1	1	1	1	1	1	0.75
54	1	1	1	1	1	0.8	1
55	1	1	1	1	1	0.8	0.8
56	1	1	1	1	1	0.8	0.8
57	1	1	1	1	1	0.6	0.4
58	1	1	1	1	1	0.667	0.4
59	1	1	1	1	1	0.667	0.6
60	1	1	1	1	1	0.667	0.6
61	1	1	1	1	1	0.667	0.8
62	1	1	1	1	1	0.667	0.6
63	1	1	1	1	1	0.667	0.8
64	1	1	1	1	1	0.833	0.8

Table_apx D-2: Prediction timetable type 2 of household 4, constructed by using ground truth occupancy data.

time	wday_1	wday_2	wday_3	wday_4	wday_5	wday_6	wday_7
1	0.6	0.8	1	1	1	1	1
2	0.6	0.8	1	1	1	1	1
3	0.6	0.8	1	1	1	1	1
4	0.6	0.8	1	1	1	1	1
5	0.6	0.8	1	1	1	1	1
6	0.6	0.8	1	1	1	1	1
7	0.5	0.8	1	1	0.75	1	1
8	0.5	0.8	1	0.8	0.75	1	1
9	0.5	0.8	1	0.8	0.75	1	1
10	0.6	0.8	1	0.8	0.75	0.5	1
11	0.6	0.6	0.4	0.6	0.75	0.5	1
12	0.6	0.6	0.4	0.6	0.6	0.5	1
13	0.6	0.5	0.4	0.6	0.6	0.333	0.8
14	0.6	0.5	0.2	0.333	0.6	0.333	0.6
15	0.6	0.5	0	0.333	0.6	0.333	0.4
16	0.6	0.5	0	0.333	0.6	0.333	0.5
17	0.4	0.5	0	0.333	0.6	0.167	0.5
18	0.25	0.5	0	0.333	0.6	0.167	0.25
19	0.25	0.333	0	0	0.6	0.167	0
20	0.25	0	0	0	0.6	0.167	0
21	0.25	0	0	0	0.4	0	0
22	0.5	0	0	0	0.4	0	0
23	0.5	0	0	0	0.4	0	0
24	0.5	0	0	0	0.4	0	0
25	0.5	0	0	0	0.2	0	0
26	0.5	0	0	0	0.2	0	0.2
27	0.6	0	0	0	0.2	0	0
28	0.6	0	0	0	0.2	0	0.2
29	0.6	0	0	0	0.2	0	0.2
30	0.6	0	0	0	0.2	0	0.2
31	0.4	0	0	0	0.2	0	0.2
32	0.4	0	0	0	0.4	0	0.2
33	0.333	0	0	0	0.4	0	0.2
34	0.333	0	0	0	0.4	0	0.2
35	0.333	0	0	0	0.4	0	0.2
36	0.333	0	0	0	0.4	0	0.2
37	0.333	0	0	0	0.4	0	0.2

38	0.333	0	0	0	0.4	0	0.2
39	0.333	0	0	0	0.6	0.2	0.2
40	0.333	0.2	0	0	0.6	0.25	0.2
41	0.5	0.25	0.2	0	0.6	0.25	0.4
42	0.5	0.25	0	0	0.6	0.25	0.4
43	0.5	0.25	0	0	0.6	0.25	0.4
44	0.5	0.2	0	0	0.6	0.5	0.4
45	0.5	0.2	0.2	0	0.6	0.4	0.4
46	0.5	0.167	0.2	0.25	1	0.6	0.4
47	0.667	0.2	0.4	0.25	1	0.6	0.4
48	0.8	0.2	0.6	0.25	1	0.6	0.4
49	0.8	0.25	0.8	0.75	1	1	0.4
50	0.8	0.5	0.8	0.75	1	1	0.4
51	1	0.667	0.8	0.75	1	1	0.4
52	1	0.75	0.8	0.75	1	1	0.5
53	1	1	1	1	1	0.8	0.5
54	1	1	1	1	1	0.8	0.5
55	1	1	1	1	1	0.8	0.4
56	1	1	1	1	1	0.6	0.4
57	1	1	1	1	1	0.6	0.2
58	1	1	1	1	1	0.5	0.2
59	1	1	1	1	1	0.667	0.2
60	1	1	1	1	1	0.667	0.2
61	1	1	1	1	1	0.667	0.4
62	1	1	1	1	1	0.667	0.4
63	1	1	1	1	1	0.667	0.4
64	1	1	1	1	1	0.667	0.4

Table_apx D-3: Number existent periods, in the classification set of household 4, that were used to construct prediction timetables.

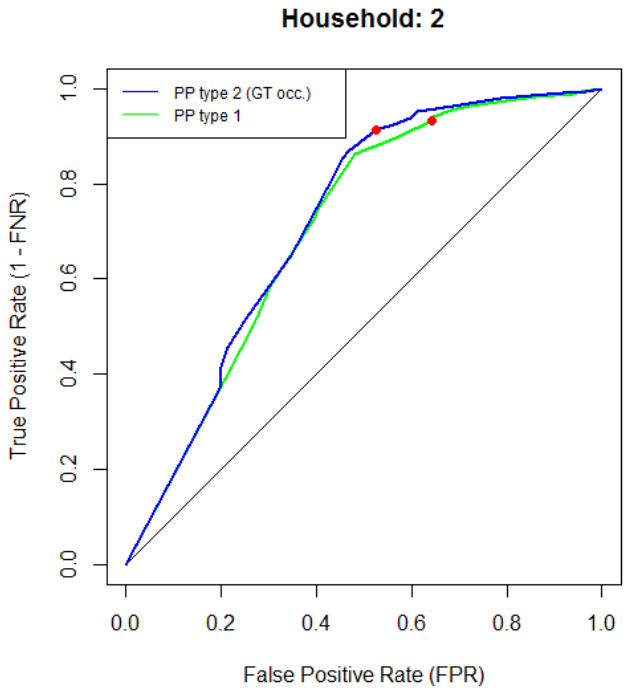
time	wday_1	wday_2	wday_3	wday_4	wday_5	wday_6	wday_7
1	5	5	5	4	3	6	5
2	5	5	5	4	3	6	5
3	5	5	5	4	4	6	5
4	5	5	5	4	4	6	5
5	5	5	5	5	4	6	5
6	5	5	5	5	4	6	5
7	4	5	5	5	4	6	5

8	4	5	5	5	4	6	5
9	4	5	5	5	4	6	5
10	5	5	5	5	4	6	5
11	5	5	5	5	4	6	5
12	5	5	5	5	5	6	5
13	5	4	5	5	5	6	5
14	5	4	5	3	5	6	5
15	5	4	5	3	5	6	5
16	5	4	5	3	5	6	4
17	5	4	5	3	5	6	4
18	4	4	5	3	5	6	4
19	4	3	5	3	5	6	5
20	4	2	5	3	5	6	5
21	4	2	5	3	5	6	5
22	4	4	5	3	5	6	5
23	4	5	5	3	5	6	5
24	4	5	5	3	5	6	5
25	4	5	5	3	5	6	5
26	4	5	5	3	5	5	5
27	5	5	5	3	5	4	5
28	5	5	5	3	5	4	5
29	5	5	5	3	5	5	5
30	5	5	5	3	5	5	5
31	5	5	5	3	5	5	5
32	5	5	5	3	5	4	5
33	6	5	5	3	5	4	5
34	6	5	5	4	5	4	5
35	6	5	5	4	5	5	5
36	6	5	5	4	5	5	5
37	6	5	5	4	5	5	5
38	6	5	5	4	5	5	5
39	6	5	5	4	5	5	5
40	6	5	5	4	5	4	5
41	6	4	5	4	5	4	5
42	6	4	4	4	5	4	5
43	6	4	4	4	5	4	5
44	6	5	4	4	5	4	5
45	6	5	5	4	5	5	5
46	6	6	5	4	5	5	5

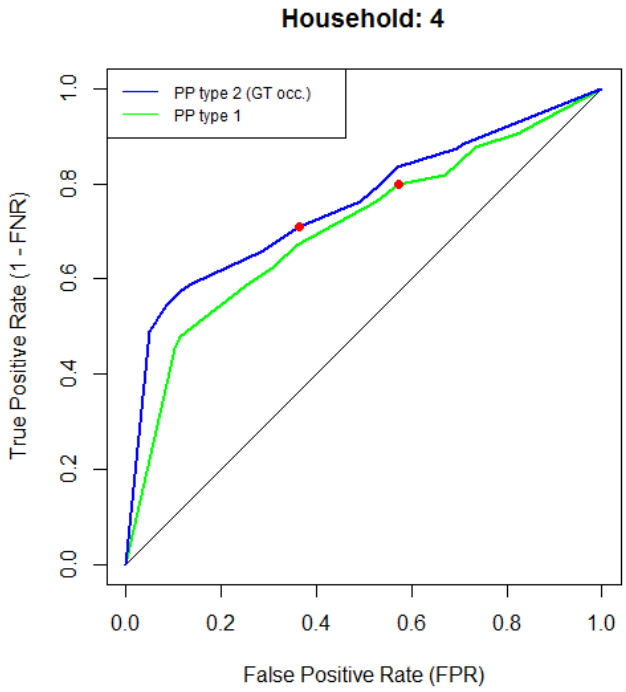
47	6	5	5	4	5	5	5
48	5	5	5	4	5	5	5
49	5	4	5	4	5	4	5
50	5	4	5	4	5	4	5
51	5	3	5	4	5	4	5
52	5	4	5	4	5	5	4
53	5	4	4	4	5	5	4
54	5	4	4	4	5	5	4
55	5	4	4	4	5	5	5
56	5	5	4	3	5	5	5
57	5	5	4	3	5	5	5
58	5	5	4	3	5	6	5
59	5	4	4	4	5	6	5
60	5	5	3	4	5	6	5
61	6	5	3	4	5	6	5
62	6	6	4	4	5	6	5
63	6	6	5	4	5	6	5
64	6	6	5	4	5	6	5

Table_apx D-4: False positive and false negative rates for the prediction timetables type 1, constructed by using the occupancy data generated by the best classification algorithm, for each household. Values in parentheses represent the number of misclassified minutes per day in average.

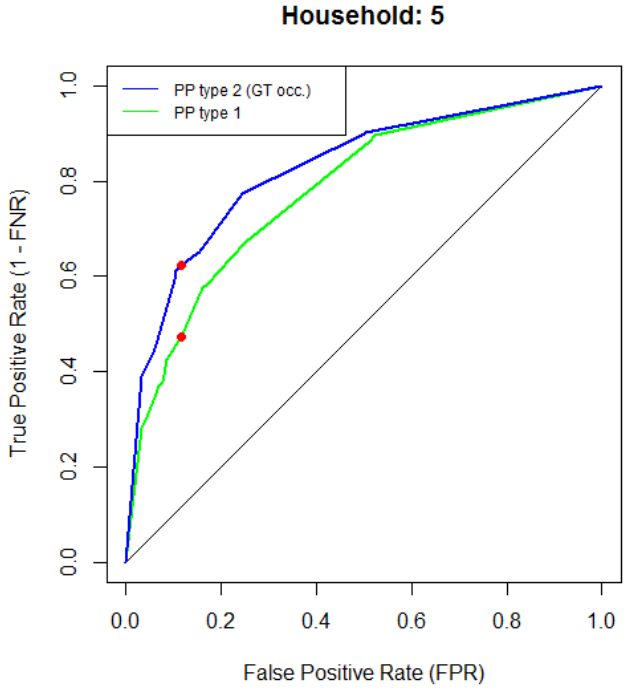
Household	Best classifier	FPR (%)	FNR (%)	Prior accuracy (%)
1	random forest	100 (108 min)	0	88.74
2	random forest	64.13 (159 min)	6.71 (48 min)	74.11
3	neural network	96.82 (137 min)	3.08 (25 min)	85.27
4	random forest	57.34 (295 min)	20.24 (90 min)	53.66
5	random forest	11.70 (79 min)	52.54 (149 min)	70.54



Figure_apx D-1: ROC curves for occupancy prediction in household 2. The green line corresponds to the prediction timetable constructed with occupancy data generated by the random forest model of household 2. The red point corresponds to the point where the threshold is equal to 0.5.



Figure_apx D-2: ROC curves for occupancy prediction in household 4. The green line corresponds to the prediction timetable constructed with occupancy data generated by the random forest model of household 4. The red point corresponds to the point where the threshold is equal to 0.5.



Figure_apx D-3: ROC curves for occupancy prediction in household 5. The green line corresponds to the prediction timetable constructed with occupancy data generated by the random forest model of household 5. The red point corresponds to the point where the threshold is equal to 0.5.