POLITECNICO DI TORINO

Master course in
ICT for Smart Societies

UNIVERSITÉ DE LIMOGES

Master International
Auton'Hom-e

Master Thesis

Room occupancy evaluation through a single PIR-based motion detector and Machine Learning

Supervisors
prof. Monica Visintin
prof. Laurent Billonnet

Candidate
Simone Longobardi

Academic Year 2018/2019

Abstract

This master thesis aims at creating a smart Pyroelectric Infrared Radiation based motion detector able to evaluate the room occupancy of an office.

Since from the output signal of the motion detector it is not possible to directly evaluate how many people are present in the room, a counter of entrances and exits was used. In order to transform a motion detector into an entrance and exit detector, classical machine learning was exploited.

Due to the limited time available for the thesis, the PIR-based motion detector could only be mounted in one office and be analyzed during the spring season. For these reasons, it was not possible to record enough signal samples to generalize the results to other offices or seasons. Thus, a procedure was established to automatically detect, a posteriori, some entrances and exits in any possible room the detector will be mounted.

In this way, after the installation, the smart motion detector is able to recognize some entrances and exits and use their related signals to train two one-class Support Vector Machines. One in charge of detecting in almost real-time all the entrances and the other all the exits.

Before training the two machines, the best possible PIR-based motion detector was searched for. Moreover, a long time was spent to define rules to extract the correct portion of the signal to be used in the training phase. Initially, the signal was divided into portions lasting a few seconds and then it was divided according to its peaks.

Another important step in the thesis was the feature extraction and selection. Since the machines only understand numbers, the signal had to be described by numerical properties and, since the less the features the easier the classification becomes, these properties had to be selected.

Finally, the smart PIR-based motion detector was tested on a real-life application and the results are discussed in the conclusions.

Acknowledgements

I would like to thank the Legrand company, in particular the Innovation & Systems department, for the trust in me. Without accepting my application all of this would not have been possible. Special thanks to my tutor Pascal Doré who helped me in every choice made during the thesis, being always available.

Moreover, I would like to thank Jean-François Gobeau for teaching me the fundamentals of PIR technology and for the interest shown in this thesis.

Huge thanks to professors Monica Visintin, Laurent Billonnet, Dan Istrate, and Jerome Boudy who took time out of their busy lives to periodically listen to me and propose suggestions to make progress in the thesis.

The experience in a different country would have been definitely harder without good friends, thanks to Tom Grand, Milan Thomas Manoj, Lorenzo Trolese and the A.S. Legrand Limoges, for helping me to have fun after hours spent in front of the computer.

Last but not least, I am really grateful to my family and girlfriend, for encouraging me even from a different country. I truly felt loved by you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The PIR-based motion detector is a device whose purpose is to detect movement. It is made of a Pyroelectric Infrared Radiation (PIR) sensor which generates a temporary voltage when exposed to a change in infrared radiation. The detection is referred to as passive because, unlike in active sensors, it is achieved without radiating energy.

PIR-based motion detectors are commonly used in security alarms and automatic lighting applications. PIR technology has been on the market for more than 40 years [1] and, thanks to its low price, low power consumption and privacy respect, it is still the most commonly used method to detect human presence. However, this technology is not the best solution to detect human presence because it is prone to false triggering. For example, a pet moving around the house or quick environment variations such as light or temperature variations would erroneously trigger the detector [2]. In addition to this, consider a not-empty room where nobody is moving in the field of view of the detector, then the room would erroneously be considered empty by the detector.

The improvement of PIR-based motion detectors has been, and currently is, a popular research topic for universities and electronic manufacturers.

Legrand, the global specialist in electrical and digital building infrastructures, is working on a new detector with better accuracy than the one currently available on the market. In order to achieve its goal, Legrand decided to test the possibility of applying Machine Learning methods to the detector. Methods that can automatically detect patterns in data, and then use them to predict future data, or to make decisions under uncertainty [6].

In particular, the aim of the company (and of this thesis) is to create a smart PIR-based motion detector able to recognize room occupancy with a maximum delay of 5 seconds and with an accuracy higher than 90% when applied in offices.

1.1 Motivation

There are many reasons behind the development of a smart PIR-based motion detector.

The first reason is energy savings, a study conducted for the U.S. Department of Energy [3] shows that is possible to save around 20% of energy consumption thanks to advanced occupancy sensors and around 6% using common occupancy sensors. The difference between the two type of sensors is that the former count the number of occupants in a room, while the latter measure whether occupants are present or not. Obviously, counting people does not impact the lighting energy savings, but it strongly impacts the heating, ventilating and air-conditioning (HVAC) energy savings.

The second reason is space management, which is the management of an organization's physical space inventory. In fact, reliable occupancy information would allow to use spaces more efficiently and reduce costs. For example, the organization could re-allocate underutilized space or evaluate the possibility for an expansion based on up-to-date data. Effective space management is a game changer for companies whose employees do not work everyday in the same office but moves around subsidiaries. In fact, their offices could be made available for other visiting employees. Moreover, also cleaning expenses could be cut by cleaning toilets after a certain number of usages and conference rooms after a certain number of people have been inside them.

Finally, there are other reasons which are related to how PIR-based motion detectors, currently on the market, recognize room occupancy. In general, in case some motion is detected, the room is occupied, while in case no motion is detected during a certain amount of time, the room is empty. The quantity of motion is related to the output voltage of the sensor, and the threshold must be set, together with the timer, during the installation of the detector. It goes without saying that the choice of these parameters strongly affects the performance of the detector.

In order to avoid false triggering and ease the job of the installer, Legrand would like to find Machine Learning techniques able to remove the aforementioned parameters. It is important to notice that, in common detectors, the timer is set around 10 minutes to avoid false triggering caused by a static person. Reducing this delay in the output would not only allow to have real-time occupancy, but also improve the users acceptance because they would see the light turning off while leaving the room.

1.2 Thesis structure

Chapter 2 provides a brief description of the company, the PIR-based motion detector and Machine Learning. Moreover, the results of the articles found during the literature review are mentioned.

Chapter 3 describes the materials provided by the company together with useful knowledge for the solution set-up, like the "automatic rule".

In chapter 4, the first of the two PIR-based motion detectors used in the thesis is presented. The idea was to train, in a semi-supervised approach, two machines with signals related to "first entrance" and "last exit" respectively, in order to predict similar events once the smart detector is placed. It was decided to begin the analysis of the features to be extracted by comparing "last exits" with "other exits" and "false exits" samples. These three classes were chosen because their samples are extractable by the "automatic rule", however, it has to be noticed that recorded videos were used to classify each sample.

Fifty-five features, related to the peaks of different functions applied to the two outputs of the detector, were extracted. Then supervised classification was performed to check if, with the selected features, it was possible to distinguish the three classes. Since the results showed a difficulty in distinguish between "last exits" and "other exits", it was decided to change the initial idea of predicting whether the room was empty or not, and to realize a counter able to track the number of people in the room. This time, in order to study the features, exits were compared to entrances. The conclusion was to consider a different motion detector.

In chapter 5 the second motion detector is presented, together with the analysis of the signals related to entrances and exits. Based on the analysis, it was decided to change the set of features by using the *tsfresh* Python package. After feature selection, entrances and exits were correctly classified. However, when semi-supervised classification was performed, the machine trained with "last exits" was not able to correctly distinguish between exits and not-exits.

To address the problem, in chapter 6, a mask was applied to the motion detector. Moreover, due to the fact that close in time samples were assigned to the same event, a different method of sample extraction was searched. After having analyzed specific motions, it was concluded that it is not possible to detect every entrance and exit by training a machine with only "first entrances" and another one with only "last exits". Anyway the research went further, samples were extracted, features were selected and semi-supervised classification was performed on a test set of an hour and half.

Conclusions and possible improvements are discussed in chapter 7.

Chapter 2

Basic concepts

2.1 Legrand

Legrand was founded in 1904 at Limoges by Frédéric Legrand after the acquisition of a porcelain factory. The company then started to produce electrical wiring devices using porcelain as insulating material. The capacity for adaptation to the market led Legrand to become the global specialist in electrical and digital building infrastructures.

Legrand is considered a pioneer in international development thanks to several acquisitions, among which BTicino in 1989. Nowadays the company is present in nearly 90 countries with a workforce of more than 38,000 employees. Their offering is vast and ranges between the commercial, residential and industrial markets. The company sells products to manage electricity in buildings like switches, sockets, cables, protection devices but also data infrastructures, home automation products and many others.

With a strong vision for the future, Legrand recently launched the Eliot program to answer the needs of consumers and installers related to connected objects. The mission of the company is to create simple and innovative electrical and digital systems aiming to improve comfort, security and quality of life inside buildings.

Internally Legrand is divided into several Strategic Business Units according to the product developed:

- User interfaces
- Installation components
- Cable management
- Digital infrastructure
- Building systems
- Energy distribution
- UPS

These SBUs report directly to the Executive Committee but are not on their own. In fact, there are other SBUs which are transversal and collaborate with all of them. An example is the unit inside which this master thesis was pursued, the Innovation and Systems unit, which collaborates with other units to bring technological innovation on the market.

2.2 Pyroelectric Infrared Radiation sensors

The pyroelectricity is the ability of certain materials to generate an electric signal when they are heated or cooled. PIR sensors exploit this property to detect moving bodies, the change in temperature is provided by infrared radiations. Infrared is the portion of electromagnetic spectrum (represented in figure 2.1) with wavelengths ranging between 750 nm and 1 mm.

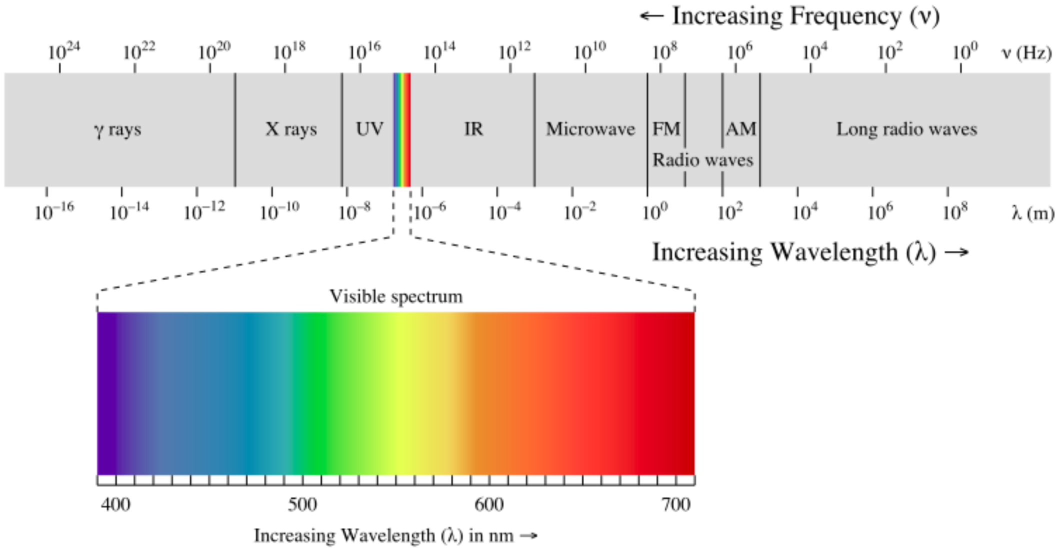

FIGURE 2.1: The Electromagnetic Spectrum.

According to Planck's radiation law, every object at a temperature T not equal to 0 K emits radiation. Moreover, according to Stephan Boltzmann law, the emitted power per unit area at all wavelengths by a black body is related to the absolute temperature as

$$W_b = \sigma T^4 \tag{2.1}$$

where: W_b is the emitted power per unit area, σ is a constant equal to $5.67 \times 10^{-8}\ Wm^{-2}K^{-4}$, and T is the object temperature expressed in K.

A black body is an ideal emitter and absorber of radiations. Real bodies emit less energy than a black body at the same temperature, the proportion is called emissivity:

$$Emissivity = \frac{\text{Radiant emittance of an object}}{\text{Radiant emittance of a black body at same temperature}} \tag{2.2}$$

Wien's displacement Law states that objects of different temperature emit spectra that peak at different wavelengths (as shown in figure 2.2). The relationship between the temperature of the black body and its peak dominant wavelength is:

$$\lambda_{max} = b/T \tag{2.3}$$

Where T is the absolute temperature in kelvin and b is a constant equal to $2.898 \times 10^{-3} mK$

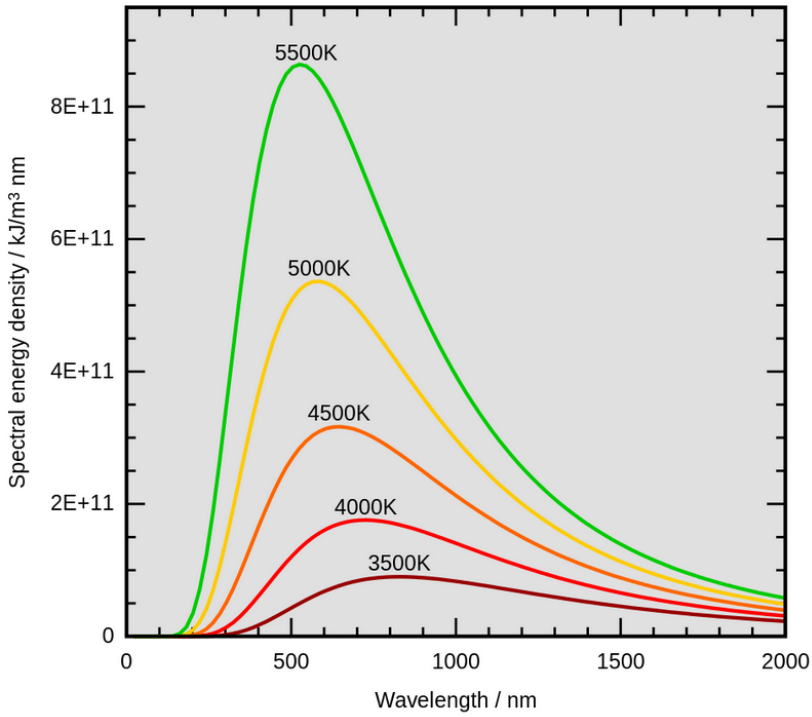

FIGURE 2.2: Wien's displacement Law.

With the aforementioned physical laws in mind it is possible to understand how infrared detection works.

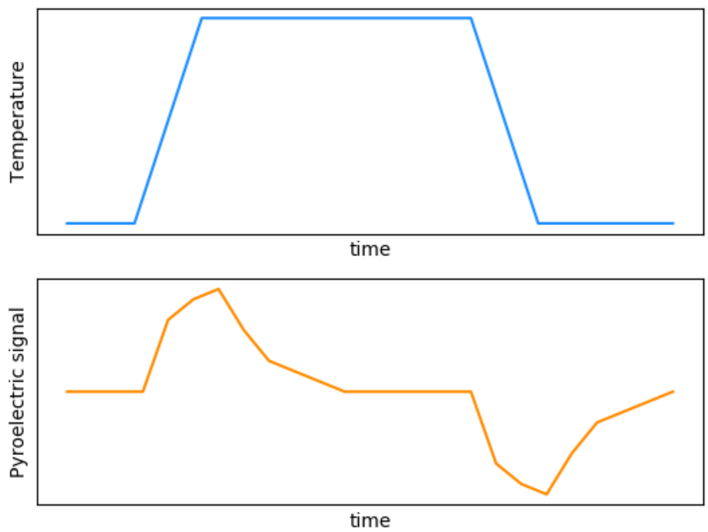

FIGURE 2.3: PIR signal behaviour according to the room temperature.

As previously said, a pyroelectric component generates an electric signal when exposed to a change in temperature (provided by infrared radiations). As shown in figure 2.3, there is a raising peak in correspondence to a raise in temperature, and a falling peak in correspondence to a drop in temperature. Moreover, the peak is not symmetric, in fact, the time constants of the raising and the falling edge are different. The first one is a thermal time constant, while the second is an electrical time constant.

A sensor made of one pyroelectric component is able to detect a moving body because when the body enters the visible region it generates a difference in temperature due to the emitted infrared radiations. However, false detections may occur because of a raise in temperature in radiators or walls exposed to the sun. For this reason, PIR sensors are made of two pyroelectric components with opposite polarization, which allows to cancel signals generated from the same source at the same time.

The configuration of the two components together with the theoretical output, in presence of a moving body, is shown in figure 2.4.

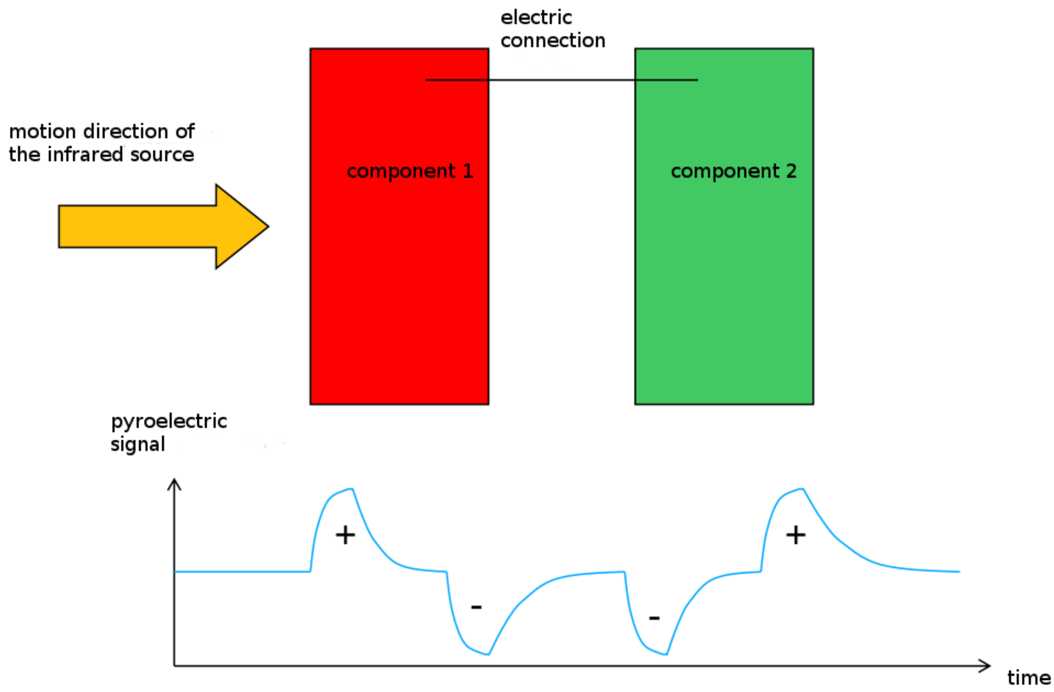

FIGURE 2.4: Signal generated by two inversely polarized components. Courtesy of CSTB.

The theoretical signal requires a very narrow infrared source. The human body emits infrared radiations in a wide region, resulting in a real signal not symmetrical at all. A sample signal is shown in figure 2.5.

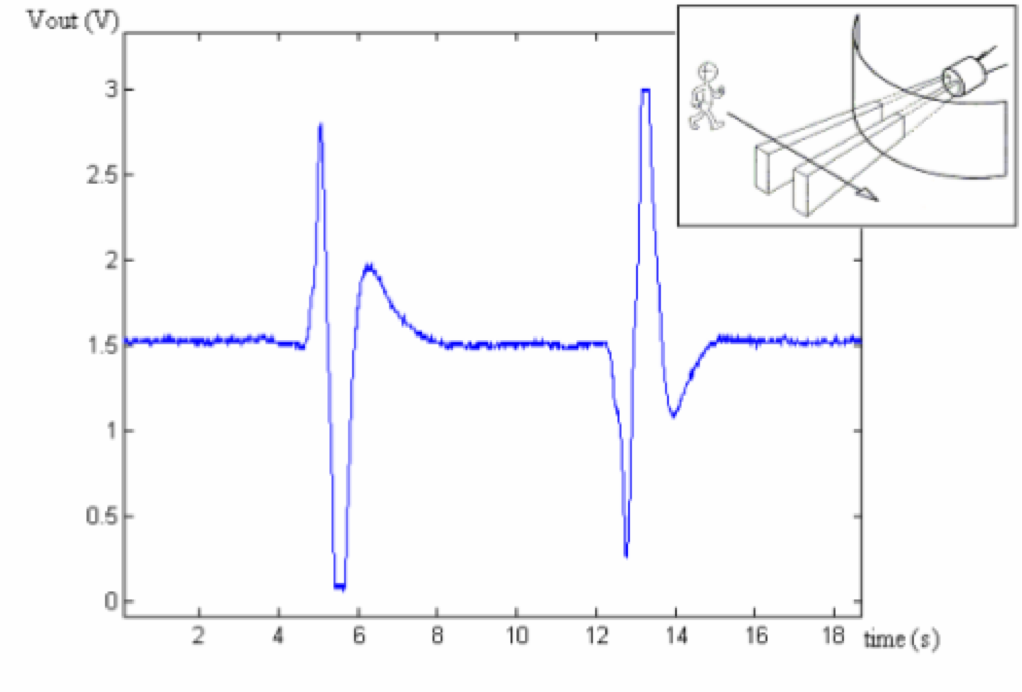

FIGURE 2.5: Output of a PIR sensor in the case of a person moving forward and backward. Image from [4].

A plain PIR sensor (with two components) can only split the detection area into two areas, each one associated to one pyroelectric component. This would allow to only detect bodies that move from one area to the other. In order to detect motions in a room, the field of view must be split into several areas, for this reason the lens is divided into multiple sections. Moreover, instead of using a conventional lens per section, Fresnel lenses are used to reduce the size of the plastic window, a typical lens is shown in figure 2.6. A Fresnel lens focuses the radiations on the sensors, just like a conventional lens, with the advantage of being much thinner. A schematic of a PIR detector together with a single Fresnel lens is provided in figure 2.7.

It is important to notice that the plastic window also acts as a filter by limiting the wavelengths to 8-14 micrometers, which is the range of infrared radiations emitted by humans beings.

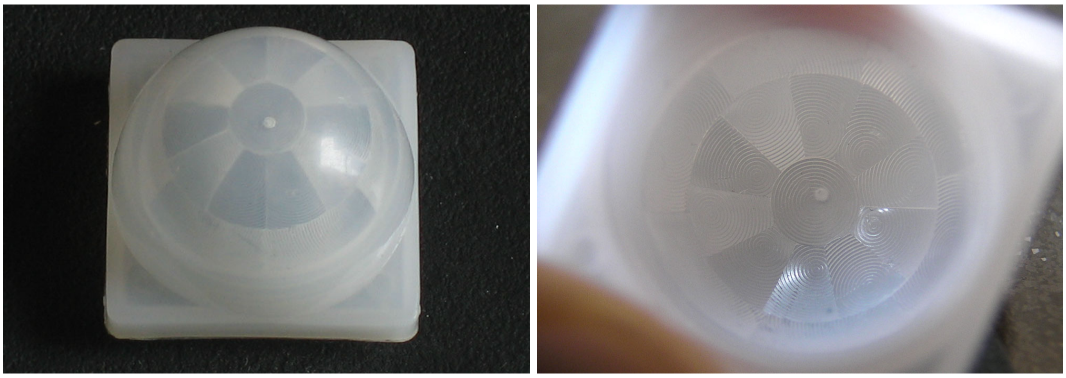

FIGURE 2.6: Multiple sections of the plastic window, each one is a Fresnel lens. Images from [5].

.

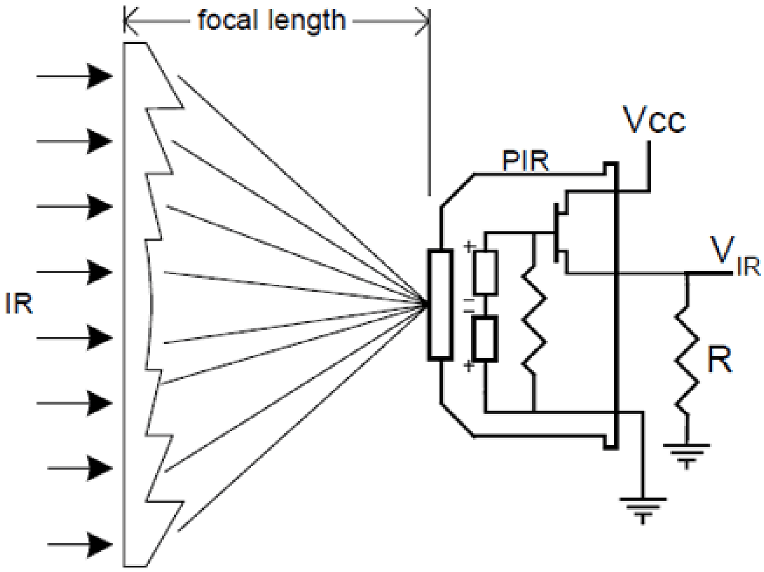

FIGURE 2.7: Schematic of a PIR sensor together with a single Fresnel lens. Image from [5].

2.3 Machine Learning

Machine Learning is a set of methods that can automatically detect patterns in data, and then use them to predict future data, or to make decisions under uncertainty [6]. Machine learning provides automated methods of data analysis which are needed in the new era of big data.

There are two main types of machine learning: supervised learning approach and unsupervised learning approach. The first one is also called predictive because its goal is to learn the relationship between inputs and outputs, given a set of labeled input-output pairs. This set is referred as training set. Each training input is a vector of numbers representing the features. The output or response variable can be categorical or real-valued. In the first case the problem is a classification problem, in the second case it is a regression problem.

The second type of machine learning is the unsupervised learning approach. Unlike the supervised learning approach, only the inputs are given and the goal is to detect patterns in the data. The main difference from the previous approach is that it is not possible to compare the predictions with the observed values.

2.4 Literature review

Human detection and motion tracking are very popular topics in fields like surveillance, industrial applications and smart environments. Typical tracking techniques use cameras and process big amounts of data to recognize people's position in a room. Even if these solutions are accurate, they are expensive under different points of view: hardware, software, and power consumption. For this reason and because of privacy, PIR-based solutions are preferred.

Some researchers have been working on detecting movement direction to keep track of the room occupancy using the digital output of PIR-based motion detectors [7]. In order to differentiate between entrances and exits, Wahl *et al* observed the time difference between output signals of multiple PIR sensors placed inside and outside the room. In a real case scenario, their solution detected all occupant movements to and from the office room with an error rate lower than 1%.

Other researchers have been working on the analog output of PIR-based motion detectors which carries more features. In article [4], Zappi *et al* used a pyroelectric infrared sensor based wireless network to distinguish the number of people walking in line, as well as side by side in a hallway. Their idea was to decrease the number of Fresnel lenses while increasing the number of PIR sensors. Metallic tape was used to cover the unwanted elements in the lens, the configuration is shown in figure 2.8.

The PIR array was made of three PIR sensors placed in row with different orientations (see figure 2.9).

A micro-controller was used for each sensor, providing connectivity to the network but also in charge of detecting the number of peaks present in the output.

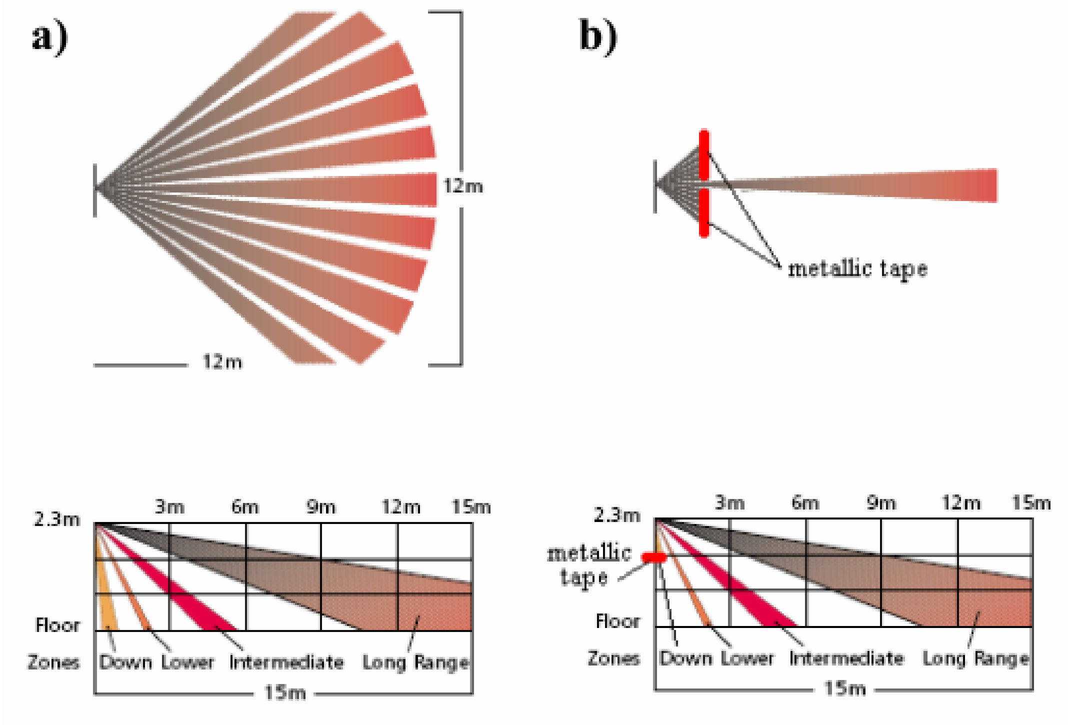

FIGURE 2.8: Customized Field of View a) original b) with metallic tape. The upper figures represent the top view while the lower ones the side view. Image from [4].

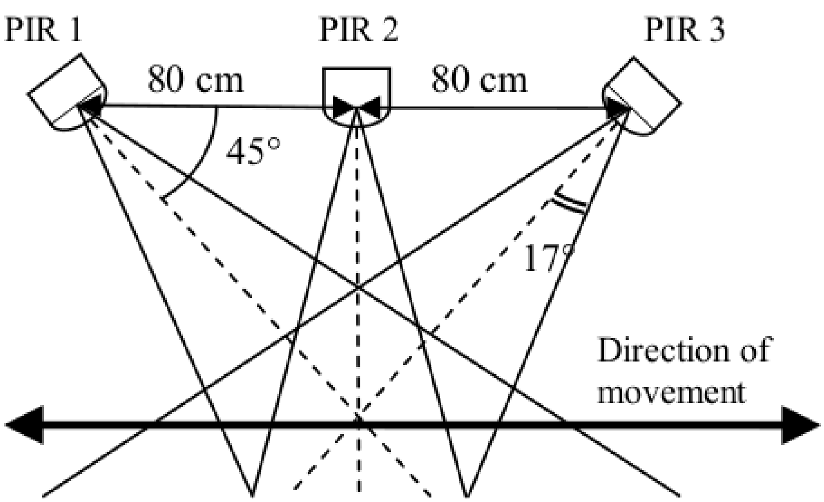

FIGURE 2.9: PIR array set-up. Image from [4].

With this configuration, Zappi *et al* achieved 100% correct detection of direction of movement and 89% correct detection of the number of people. In particular, 98.3% for people walking in line and 75% for people walking side by side.

Not only human movement detection was researched but also human identification, in article [8], Yun *et al* used machine learning algorithms to identify people passing in a hallway.

They developed a module with two pairs of PIR sensors perpendicularly placed and modified lenses. One module has been placed on the ceiling while the other two have been placed on the walls, facing each other (figure 2.10).

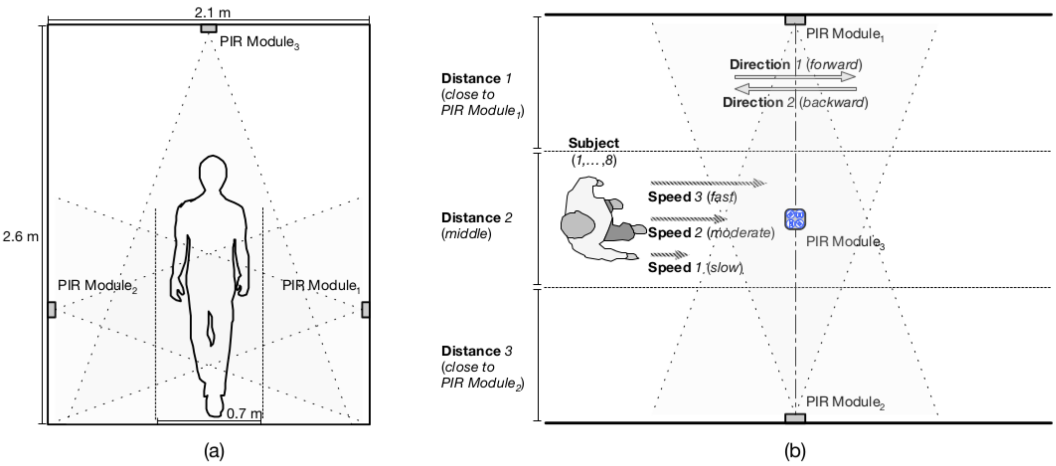

FIGURE 2.10: Experimental setup proposed in [8].

The data collected are related to eight subjects walking in different conditions: direction, distance from the sensors, and different speeds. Some samples are provided in figure 2.11.

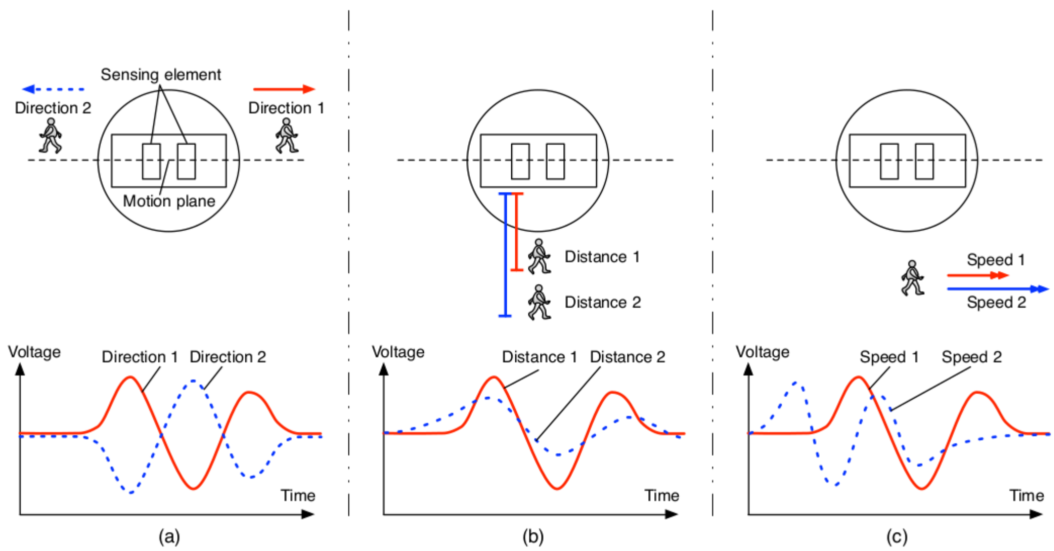

FIGURE 2.11: Output signal in the case of walking: a) in different direction, b) at different distances, c) at different speed levels. Image from [8].

Then classification was performed using amplitude, time to peaks, and event duration as features.

The results achieved by Yun *et al* show an accuracy better than 94% in identifying subjects and classifying the speed, distance and direction.

In article [9], instead, an application of a single PIR-based motion detector is discussed. The authors have designed a system to predict room occupancy by placing a micro-controller board, with a PIR sensor attached, in the middle of a conference room. As shown in figure 2.12, they found out that the statistical distribution of the sensor data can be described by a Dirac delta distribution centered at the maximum ADC output, and a Laplace distribution centered at the median value. The relationship between the Laplace spread parameter and the number of occupants was modeled by a linear model with Gaussian outputs.

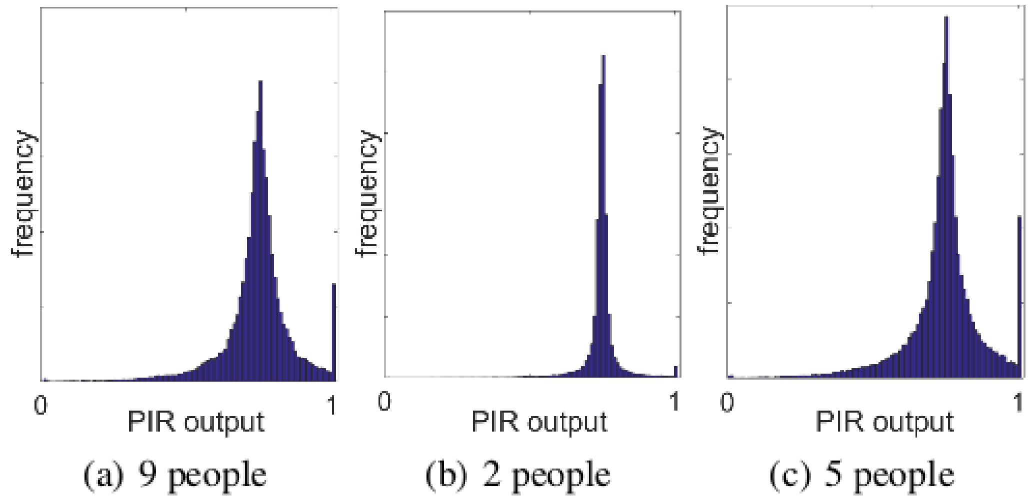

FIGURE 2.12: Histogram of raw data after 1 hour meeting. Image from [9].

For practical application the authors decided to split the one hour meeting in multiple time intervals and estimate the room occupancy for each one.

Since the spread parameter varied during the one hour meeting, the training data were clustered into groups by using the infinite hidden Markov model. The room occupancy estimation was then performed only on data belonging to clusters describing small movements.

The proposed system is able to estimate the number of people in a room ±1 with an accuracy of 80% and a minimum time window of 30 seconds.

After a careful literature review, only the last mentioned article regarding room occupancy evaluation through a single PIR-based motion detector was found. However, the system was designed for conference rooms, where the number of occupants does not vary for the whole duration of the meeting.

This thesis aims at evaluating the room occupancy in a dynamic context. Moreover, differently from other researches, the detector will not be placed to match the direction of motion, e.g., at middle height of a wall of a hallway. There are two reasons for this choice. First of all, the final device must be installable in every room and electric wires are not always available near the door. Second, it is possible that the detector would be obscured by a human body increasing the risk of miss-detection. Think about a person standing at the door while talking with someone inside the room.

Chapter 3

Methodology

3.1 Materials

The two PIR-based motion detectors used during this research were provided by the Legrand company.

The first motion detector was chosen because, according to the manufacturer, it should output two signals with a difference in phase, related to the direction of the moving object.

However, it did not show margins of improvement, as discussed in the following chapter, so another motion detector was tested.

In particular, a detector made of two PIR sensors having almost separated field of view was chosen. The advantage of this detector is that it could possibly split the room in two areas: one in proximity of the door and one made of the rest of the room except the door.

The detectors were connected to an analog-to-digital converter in order to transform the analog signal to a digital signal. The analog-to-digital converter was connected to an always powered on laptop. The laptop allowed to store, every certain amount of time, the digital signal in a file with csv (comma-separated values) format.

Through an application, it was possible to set the amount of time to record, which is proportional to the file size. Since some problems were encountered when reading the file with excel in case of a full day of recordings, the application was set to store and create a new file every 4 hours. A smaller time would lead to an unnecessary production of control data. Moreover, the sampling frequency was set to 50 Hz in order to respect the Nyquist Theorem. The resulting csv file is represented in table 3.1, where V_0 and V_1 represent the output of each sensor.

Date	Time	V0	V1
2019-03-13.08:00:02	0.00	1.016640	1.025030
2019-03-13.08:00:02	0.02	1.015995	1.025676
...	...	...	...

TABLE 3.1: Output file after digital conversion.

A web-cam was used to record the activity inside the room. Videos are the only way to assign a true label to the data corresponding to entrances, exits or other events.

Finally, Python was used to work on the csv files and apply machine learning methods. In particular, the following libraries were used:

- numpy;
- pandas;
- matplotlib;
- sklearn;
- tsfresh.

3.2 Placement

All the materials were installed on a moving cart in order to easily move the whole system. PIR-based motion detectors were placed at roof level not to be obscured by a person or an object. Moreover, the cart was parked at the corner of the room adjacent to the door and furthest from the hinge of the door, as shown in figure 3.1.

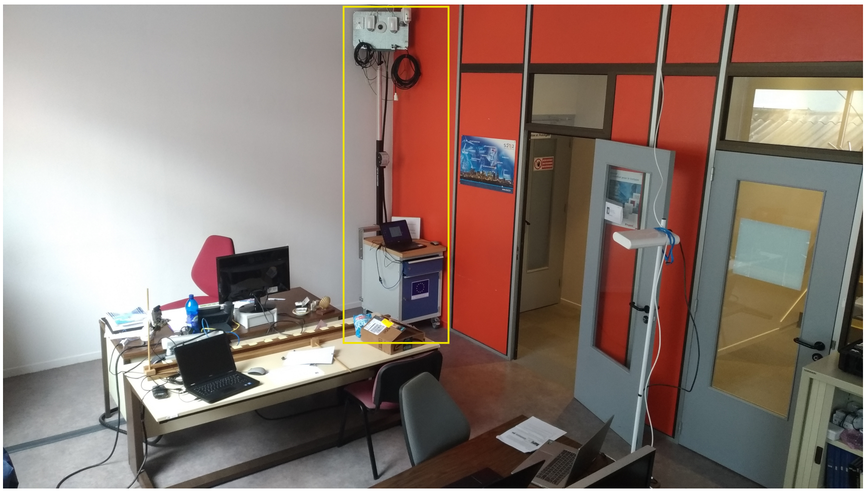

FIGURE 3.1: Placement of the cart.

This specific placement was chosen to ensure that the field of view of the detectors did not include the door. In fact, in case of opened door, the detector would be triggered by a person moving in the corridor. The decision related to the hinge of the door, instead, ensures that the radiations emitted by an entering or exiting human body reach the detectors without being blocked by the door itself.

3.3 Machine Learning System

As already said in the introductory chapter, the objective was to create an advanced PIR-based motion detector able to recognize room occupancy with a maximum delay of 5 seconds and with an accuracy higher than 90% when applied in offices.

In order to reach the objective through Machine Learning, it was important to choose the type of Machine Learning techniques.

Based on the amount of human supervision required during training, it is possible to have:

- supervised - all training data include the desired label;

- semi-supervised - partially labeled training data;
- unsupervised - unlabeled training data.

In the context of this thesis, supervised learning would mean to use only data acquired before final placement of the detector since later it would not be possible to label them. The data should be acquired for different seasons, people and rooms. In fact, only by training the machine with several scenarios, the advanced PIR-based motion detector could correctly predict room occupancy for a not previously seen scenario.

Due to the limited duration of the internship, supervised learning was not a possible solution. However, semi-supervised was a good solution since it is actually possible to label some data. In particular, it is possible to detect an entrance before which the room was empty (referred to as "first entrance") and, an exit after which the room was empty ("last exit"). The detection mechanism is explained in details in the following section.

The advantage of this solution with respect to supervised learning is that it allows to train the machine with data only related to the final placement, allowing to create a different machine for each motion detector. The disadvantage is that the training phase must be performed by the hardware mounted on the motion detector, instead of using external hardware and just replicate the machine inside the device. This means that a motion detector exploiting semi-supervised learning must have a higher computational power and larger memory size than one relying on supervised learning.

3.4 Automatic rule

As previously said, it is possible to detect a "first entrance" and a "last exit" by analyzing data a posteriori. The logic behind the algorithm is simple: if motion is detected after a certain period of inactivity then a "first entrance" occurred (figure 3.2). On the contrary, if after a detected motion there is inactivity for a certain amount of period then a "last exit" occurred (3.3).

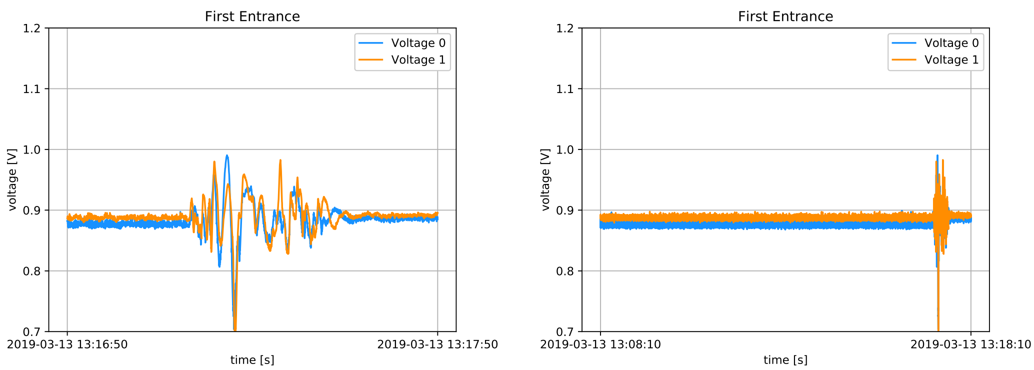

FIGURE 3.2: Automatic rule applied to a first entrance.

The automatic rule works well when the duration of the inactivity period is long. However, the longer it is the fewer the events detected. For example, by setting the time to one hour, a "last exit" followed by a "first entrance" half an hour later is not

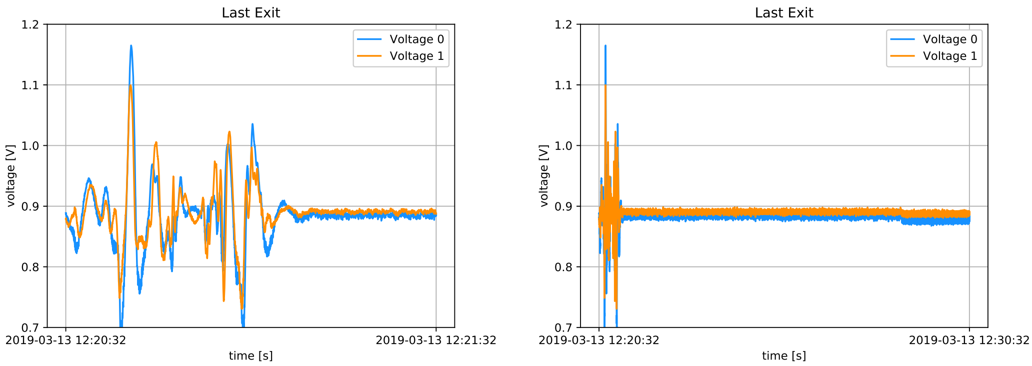

FIGURE 3.3: Automatic rule applied to a last exit.

detected. If it is too short then also a person that stands up after few seconds of being static leads to a "first entrance". And the contrary happens for a "last exit".

In a real application of the smart motion detector, the period would be set to long in order to make sure that there are no false events in the training set. Even if this means a longer learning time.

During the thesis, it was not possible to wait weeks in order to create the database so the time period was set to low and the video recordings were used to recognize the correct "first entrances" and "last exits".

In particular, the period of inactivity was set to ten minutes, where inactivity was associated to a RMS (Root Mean Square) value smaller than the one recorded during the night times one and half; and a maximum value smaller the the double of the one recorded during the night. Moreover, in the four seconds before the period of inactivity (in case of "last exit") or after (in case of "first entrance"), the RMS needed to be greater then ten times the one recorded during the night.

In the proposed solution, data were continuously monitored to detect the moment at which the previously mentioned conditions were met. After the detection, a sample of the signal made of ten seconds before the moment and five seconds after it was extracted for "first entrances", the vice versa was applied for "last exits". The time was set to respect the requirement of evaluating room occupancy with at most five seconds delay.

Chapter 4

Phase difference motion detector

4.1 Presence predicting logic

The original detector to be improved (figure 4.1) is characterized by a phase difference in the two voltages in output. This difference is related to the moving direction as shown in figure 4.2.

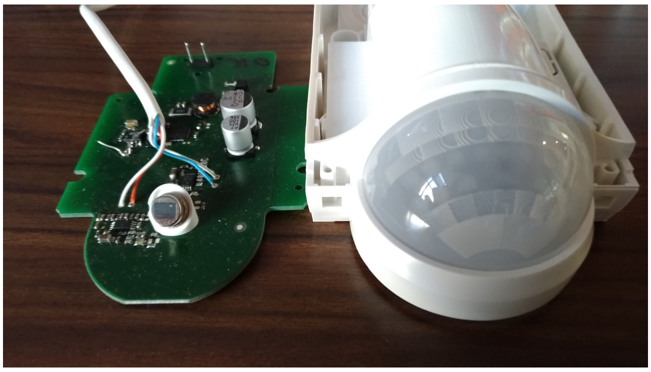

FIGURE 4.1: Phase difference motion detector.

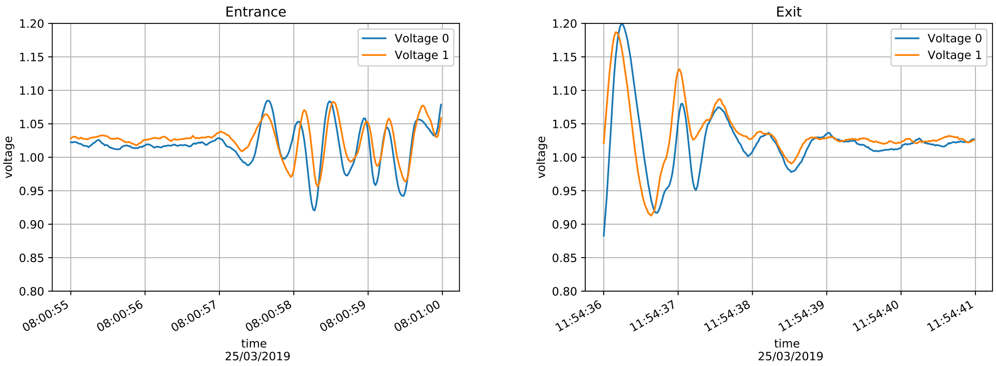

FIGURE 4.2: Entrance and Exit samples.

In figure 4.3 the block diagram of the proposed solution is shown. The motion detector was connected to the acquisition system already set up by the company. The file in output was used by the *Event Detector* in order to detect, by means of the automatic rule, "first entrances" or "last exits". In case the signal fulfilled the requirement of an event, some features were extracted creating a vector. The vectors were stored in a database and used to train a classifier. The classifier role was to detect if the signal coming from the motion detector corresponded to a first entrance or last exit. In order to be comparable to the features vectors, the data had to be extracted in a specific way and the same features had to be retrieved. The last block in the diagram is the *Occupancy evaluator* which is in charge of saving the state of the room i.e. empty or not-empty.

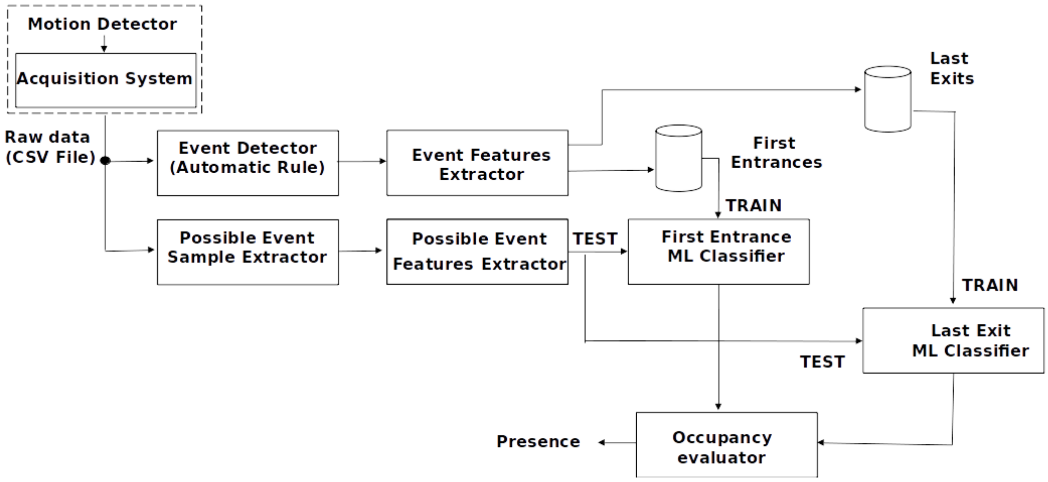

FIGURE 4.3: Block diagram of presence predicting logic

In order to decide which features to extract, the first step was to visualize some data.

By using the automatic rule mentioned in section 3.4, data related to last exits (160 samples), other exits (192 samples), and false exits (132 samples) were collected. Even if only "last exits" should be retrieved, also exits happening when someone else is in the room (other exits) and motions preceding a long period of inactivity (false exits) are detected by the rule. The reason behind starting the analysis with exits instead of entrances is that after the last exit, contrary to other exits, lights must be switched off, while every entrance leads to switching the lights on.

4.1.1 Data visualization

In figure 4.4, two samples from each class are shown.

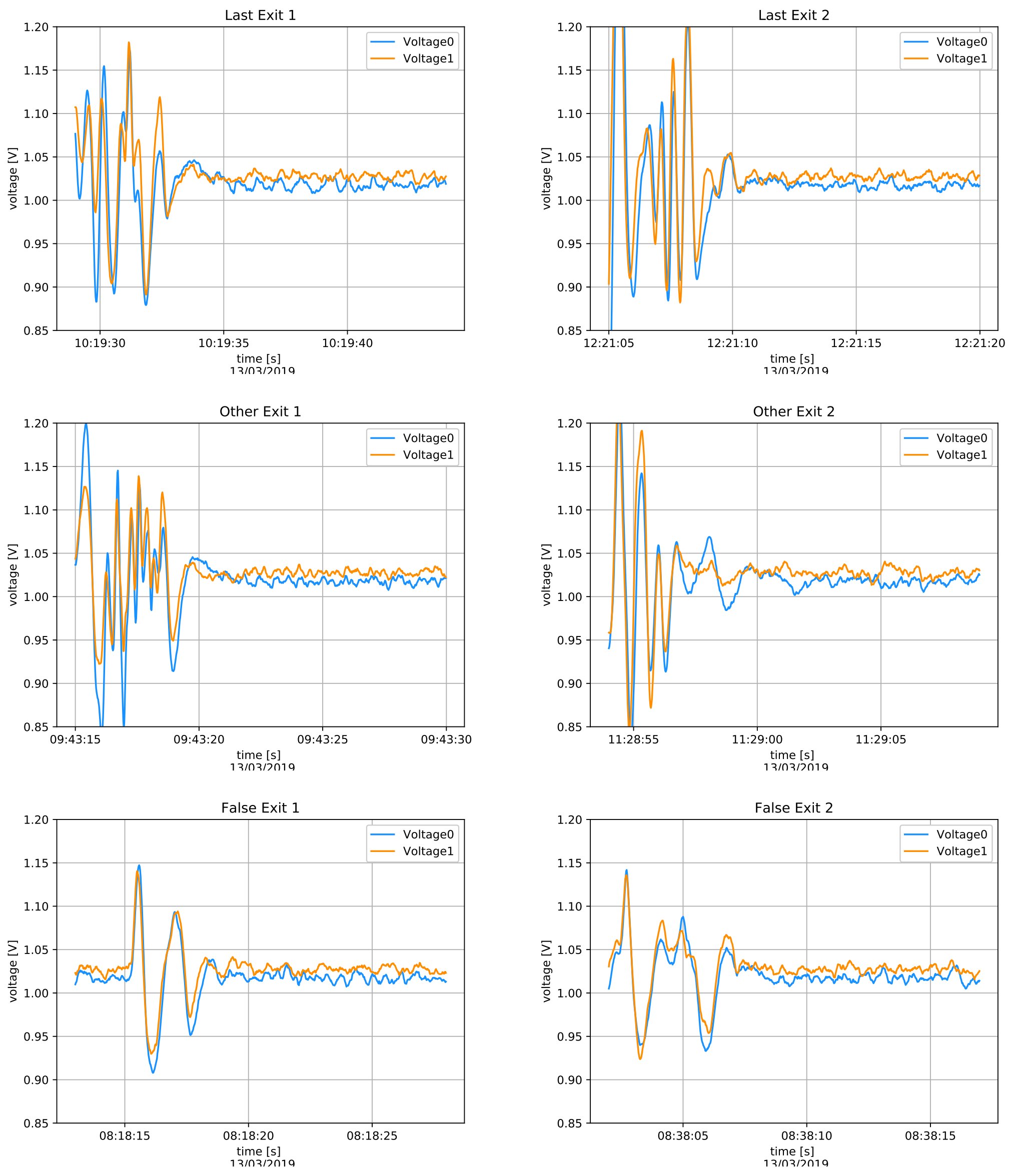

FIGURE 4.4: Last, other, and false exit samples.

In order to find some differences and possibly distinguish the last exits from the false and other exits, data were subtracted by their row-wise mean and analyzed in different domains (figure 4.5):

1. Time domain - the average between the values of voltage 0 belonging to the same class is taken at every time instance;
2. Fast Fourier Transform - the average between the magnitudes of the FFT belonging to the same class is taken at every frequency instance;
3. Spectrogram - the average between the spectrograms of the same class;

4. Energy - cumulative distribution function of the energy associated to each class;
5. Voltage - cumulative distribution function of the voltages 0 associated to each class;
6. Autocorrelation - the average between the normalized autocorrelations belonging to the same class is taken at every time instance;
7. Phase difference between V0 and V1 - cumulative distribution function of the amount of samples to be shifted in order to obtain the maximum cross-correlation between the two voltages.

By analyzing the plots one could see that false exits are distinguishable from other and last exits. Because the associated signal has lower energy in general, and lower magnitude components at frequencies grater than 0.5 Hz. On the contrary, other exits look very similar to last exits in every aspect. This is reasonable since the only difference is that there is a static person in the room, while for false exits, the difference is in the motion (e.g. sitting down).

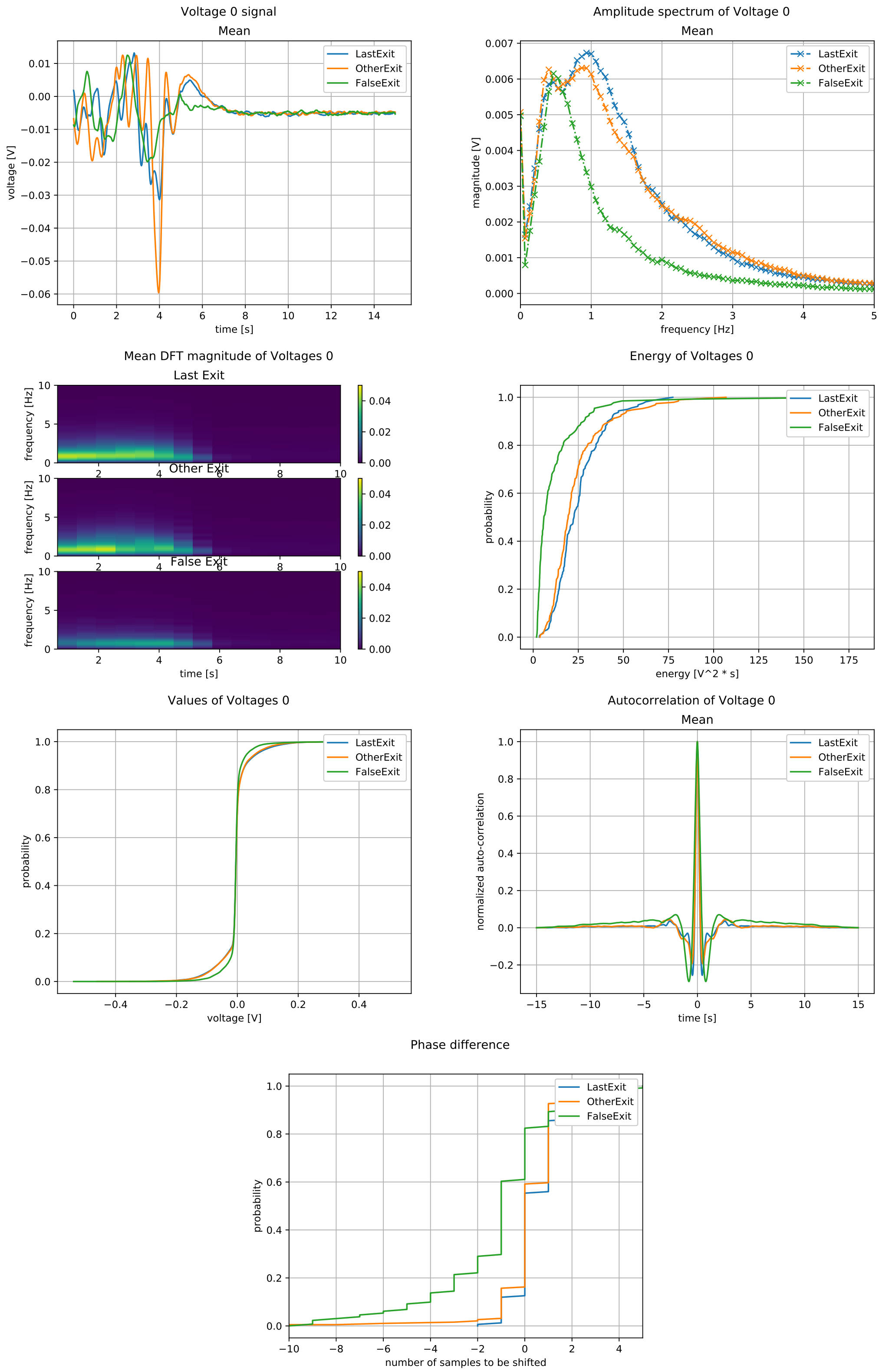

FIGURE 4.5: Data visualized in different domains

4.1.2 Feature extraction

In order to perform classification it was important to extract significant features from the raw data that best described each sample. Following the article "Machine Learning with Signal Processing Techniques" [10], the x and y value of the first N peaks of the time domain signal, the FFT, the power spectral density, and the autocorrelation, were used as features. N was set to four for the time signal and three for the other functions. In case of a smaller number of peaks detected in the signal, the x and y position of the last value of the sample were taken. An example is provided in figure 4.6.

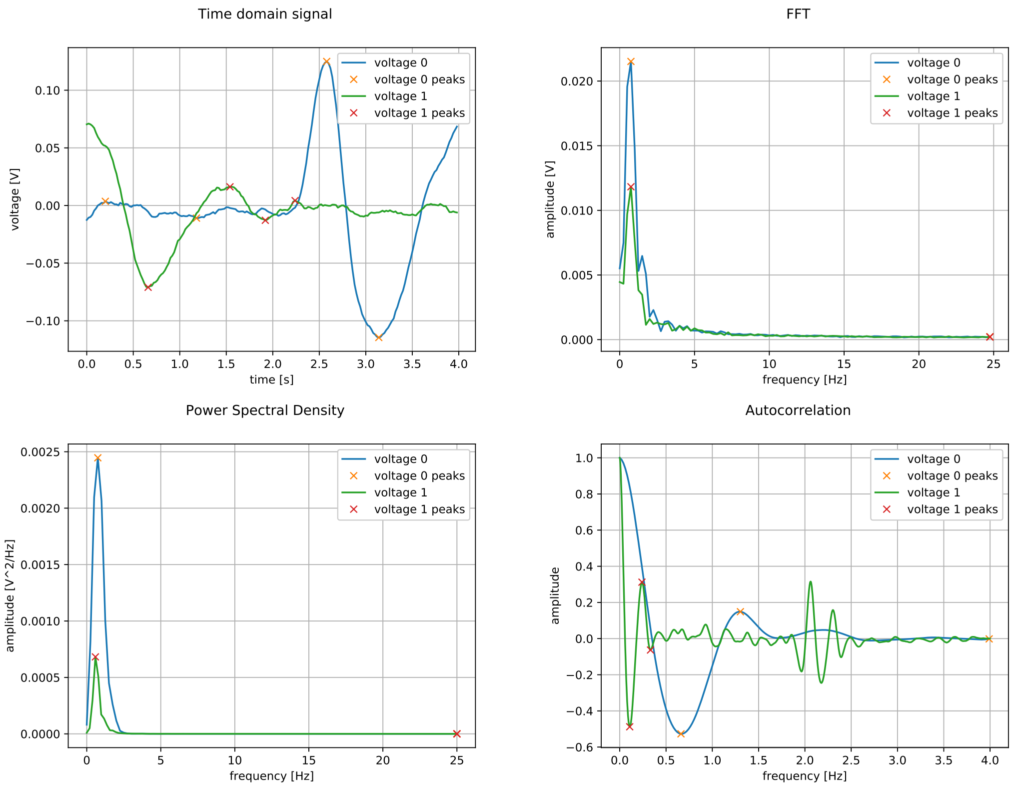

FIGURE 4.6: Feature extraction.

Moreover, also the phase difference between the two voltages and their respective energy were used as features, leading to the data structure represented in table 4.1.

Event	Time V0 x0	Time V0 y0	FFT V0 x0	FFT V0 y0	PSD V0 x0
LastExit	3.48	0.029	0.8	0.006	0.586
PSD V0 y0	Auto V0 x0	Auto V0 y0	Phase	Energy V0	...
0.001	0.58	-0.361	3	11.2793	...

TABLE 4.1: Row sample resulting from feature extraction

The total number of features was fifty-five as explained by the following calculation:

$$n_{features} = 1 \text{ phase} + 2 \text{ signals}* \\ [(3 \text{ peaks} * 4 \text{ characteristic} + 1 \text{ extra peak for time}) * 2 \text{ values} + 1 \text{ energy}] \\ = 55$$

4.1.3 Classification

First of all, data were split into training set (75% of the samples belonging to each class) and testing set (25%). Then, standardization was performed by subtracting the row-wise mean related to the training data and dividing by the standard deviation. At this point, classification was performed with a random forest classifier.

In section 3.3, it was written that the suitable machine learning system was the semi-supervised and it is true. A supervised classification was used in order to understand if the extracted features were significant or not. In fact, only by assuring that the last exits are distinguishable it is possible to perform semi-supervised classification.

In order to better explain random forest classifier is important to describe decision trees. Basically, decision trees recursively partition the input space and define a local model for each region. Each region corresponds to a leaf in the tree. This classifier is simple to interpret and can be visualized, the resulting tree is shown in figure 4.7.

During the split of the tree, the best feature and value for the feature are evaluated as follows [6]:

$$(j^*, t^*) = \arg \min_{j \in \{1,\dots,D\}} \min_{t \in T_j} cost(\{x_i, y_i : x_{ij} \leq t\}) + cost(\{x_i, y_i : x_{ij} > t\}) \tag{4.1}$$

$$\hat{\pi}_c = \frac{1}{|D|} \sum_{i \in D} \mathbb{I}(y_i = c) \tag{4.2}$$

$$cost(D) = \sum_{c=1}^{C} \hat{\pi}_c (1 - \hat{\pi}_c) \tag{4.3}$$

where D is the data in the leaf.

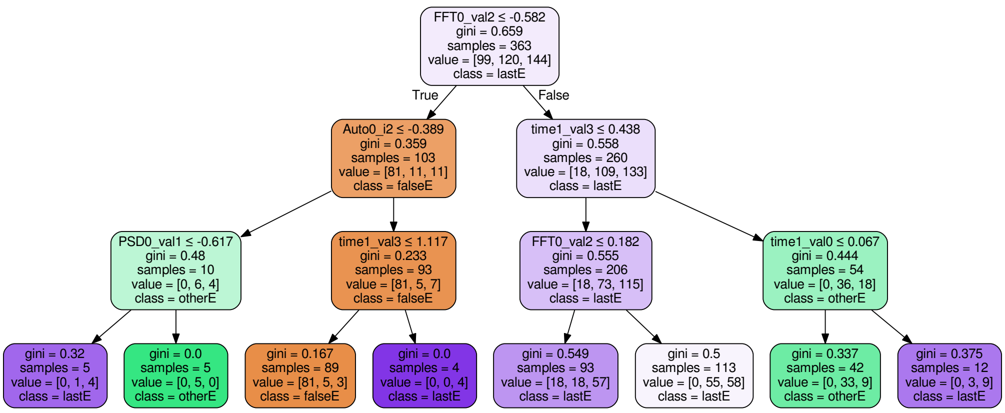

FIGURE 4.7: Resulting decision tree pruned at level 3.

However, decision trees are unstable: a small change to the input can have large effects on the structure of the tree. Random forest classifiers reduce this problem by training many trees on different subsets of the data, randomly chosen, and by averaging the result.

When applying a classifier is important to set its hyperparameters, parameters whose values are set before the learning process begins. In a random forest classifier the hyperparameter to be set is the number of decision trees to use. Usually, in order to set hyperparameters, the data set is split into training, validation and testing set. Then, the model is trained for different hyperparameters and the combination proving the best result for the validation set is considered. Finally, the test set is used to evaluate how well the model generalizes to new data.

Since the data set was made of only 484 samples, k-fold cross validation was used. This procedure consists in splitting a data set in k sets, training the model on all sets but one, testing the model on the excluded set and repeating the operations until each set is used as test set. In the following k was set to 5.

As it is possible to see from figure 4.8, the best average accuracy was 63%, provided by a number of trees of 39.

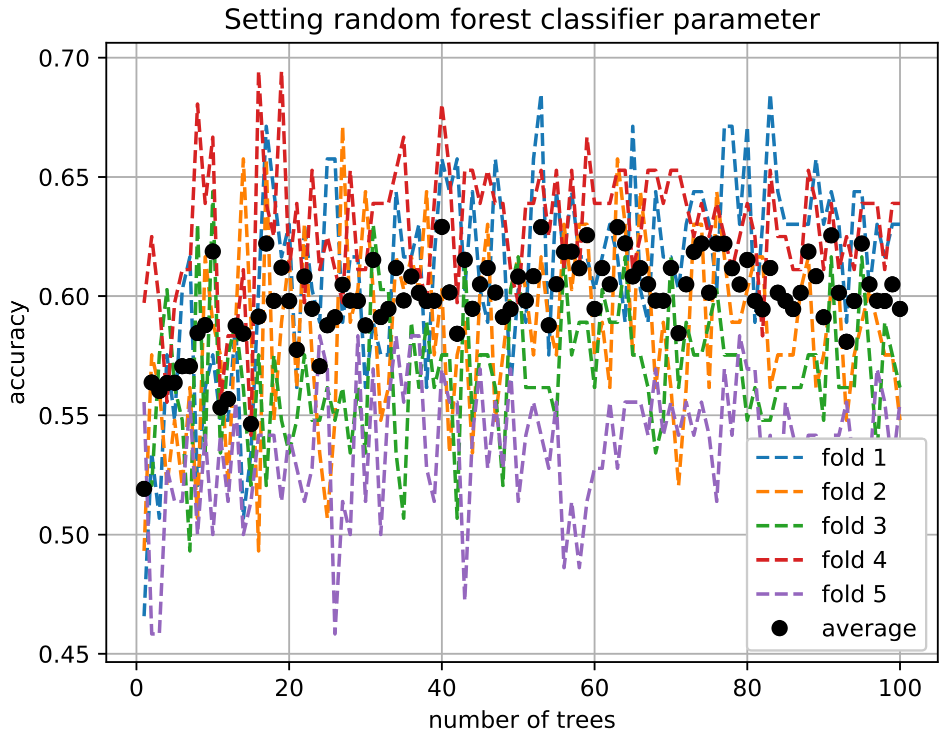

FIGURE 4.8

With 39 trees the classifier output the results shown in table 4.2 and, as already predicted by the previous analysis, it was possible to distinguish the false exits from the other and last exits but not between these two.

	pred: FalseExit	pred: OtherExit	pred: LastExit
true: FalseExit	30	0	3
true: OtherExit	6	20	14
true: LastExit	3	24	21

TABLE 4.2: Confusion matrix.

Table 4.3 provides a summary of classification performed and will be used during the rest of the thesis in order to keep track of the changes in the proposed solution.

	I
detector	phase difference
seconds per sample	15
predicting logic	presence
# features	55
classes	last, other, false exits
# train samples	120,144,99
# test samples	40,48,33
classification	RFC

TABLE 4.3: Summary of classification I.

In order to improve results, it was possible to change:

- classifier parameters;
- classifier;
- features;
- sample duration;
- predicting logic.

Since it seemed very difficult to differentiate other and last exits, the choice was to change the predicting logic; thus detecting not only the last exit but every exit (and entrance).

4.2 Counter predicting logic

In figure 4.9 is possible to see how the block digram changed using this logic. In the previous logic, the classifiers were used to detect first entrances and last exits in order to determine if the room was empty or not. With this new logic instead, the goal was to determine how many people are present in the room. The difference with respect to the previous logic is that the classifiers had to detect every entrance and exit.

While for the "presence predicting logic" it was possible to use the automatic rule in order to detect events which behaved like last exits, this time was not possible to easily retrieve events similar to exits. For this reason, some entrances and exits were extracted to see if it was possible to distinguish between them. In order to extract from the raw data some exits and entrances, the recorded video was watched and the time stamp related to each event was noted down. Later, four seconds of each event

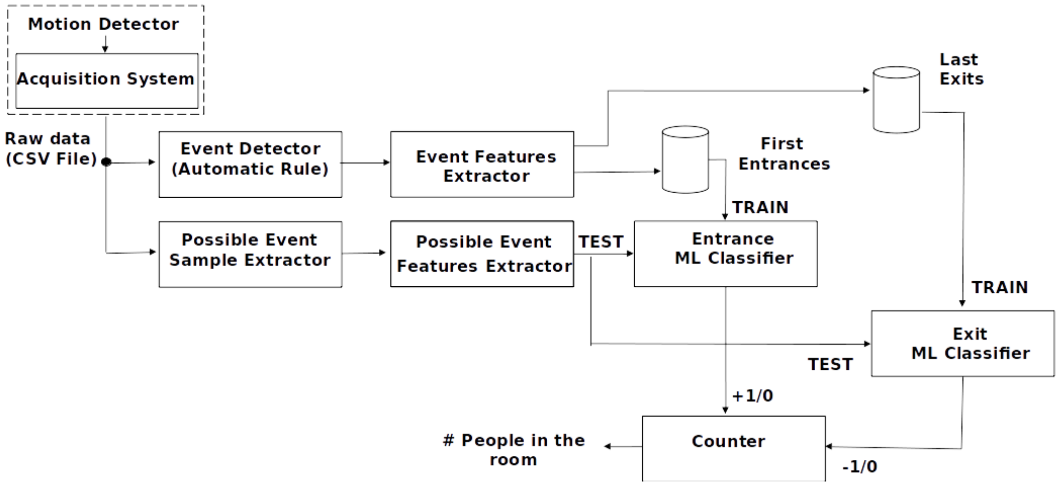

FIGURE 4.9: Block diagram of counter predicting logic.

were extracted (347 samples for both entrances and exits). The amount of seconds was chosen according to the average time for a person to cross the room.

In figure 4.10, two samples from each class are shown.

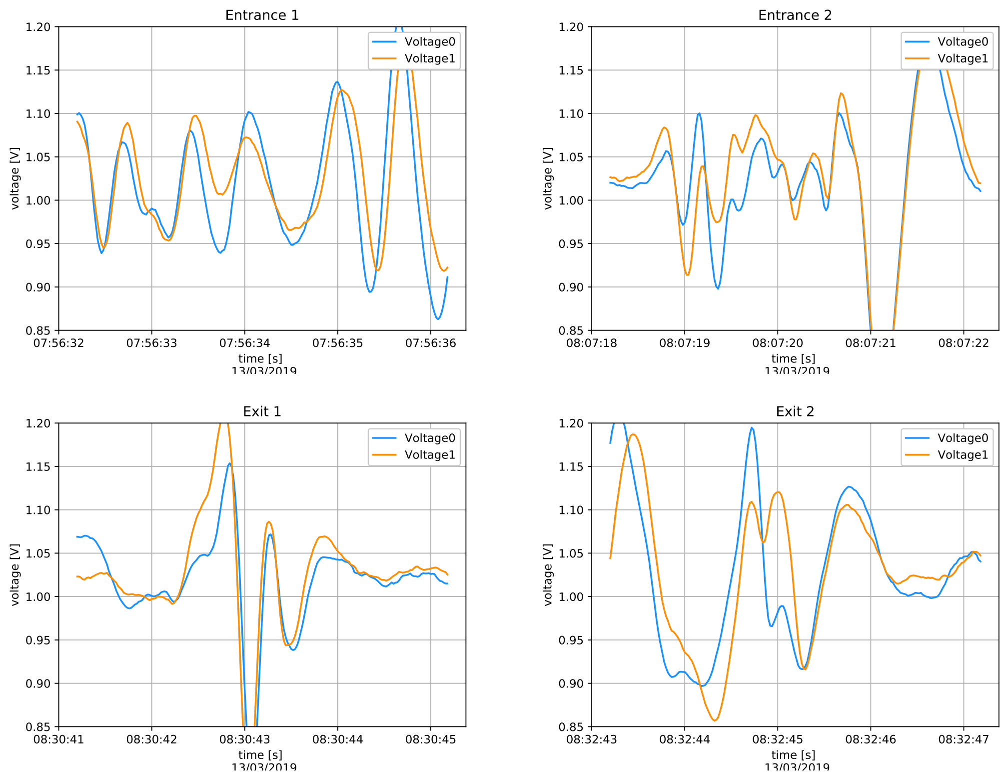

FIGURE 4.10: Entrance and exit samples.

As previously done for last, other, and false exits, also entrances and exits were analyzed in different domains and the resulting plots are shown in figure 4.11.

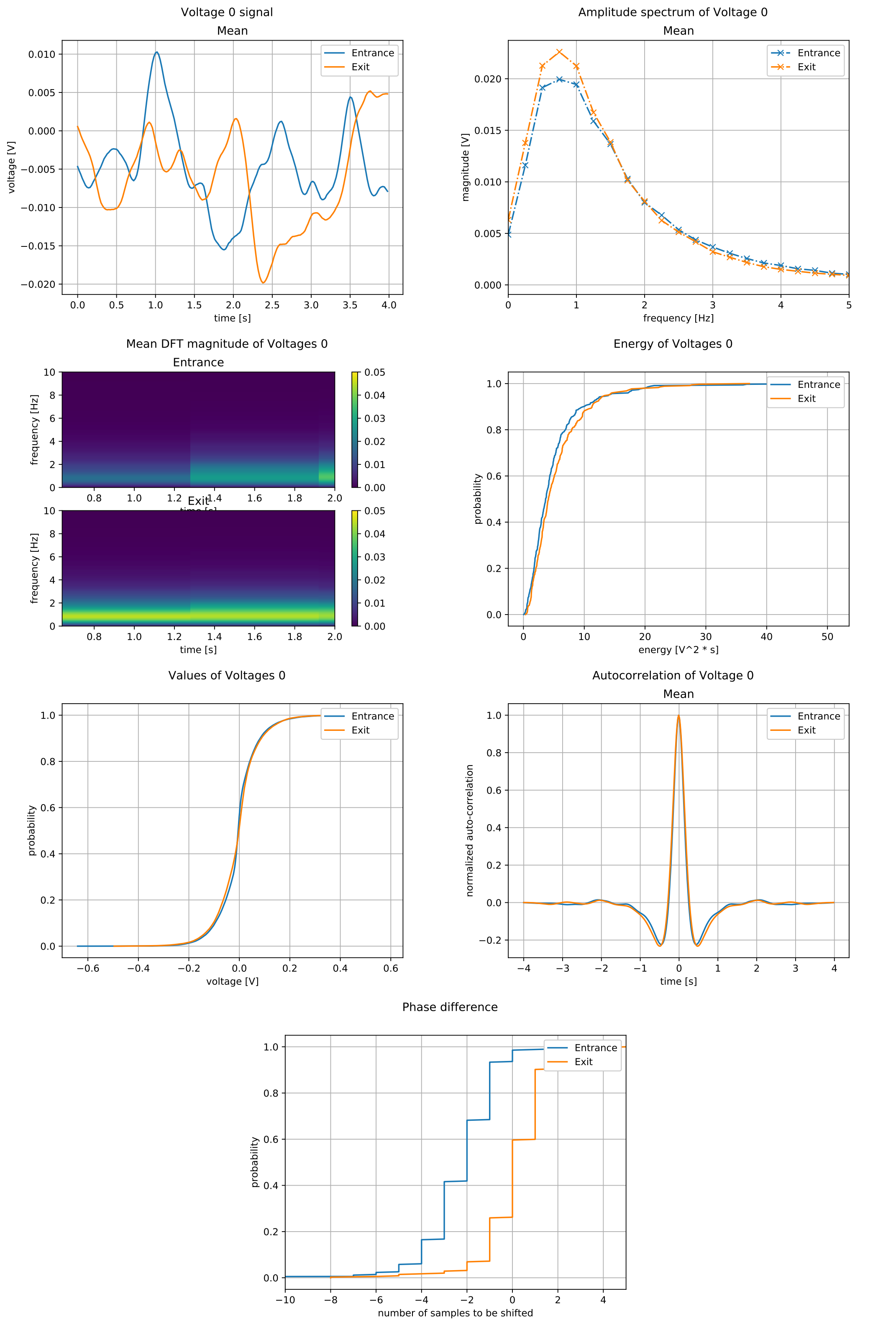

FIGURE 4.11: Entrances and exits visualized in different domains.

By analyzing the plots, it is possible to see a slight difference in the average frequency spectrum, with the exits having a bigger magnitude for frequencies around 1 Hz. Moreover, a different distribution of the phase difference between the two voltages is visible, as expected from the characteristics of this motion detector.

After the analysis, the features previously mentioned were extracted from the raw data and random forest classification was applied again. As done for the classification of false, last, and other exits, also this time the number of trees was set through a 5-fold cross validation.

By looking at figure 4.12 it is possible to see that the maximum accuracy was 86% given by 57 trees.

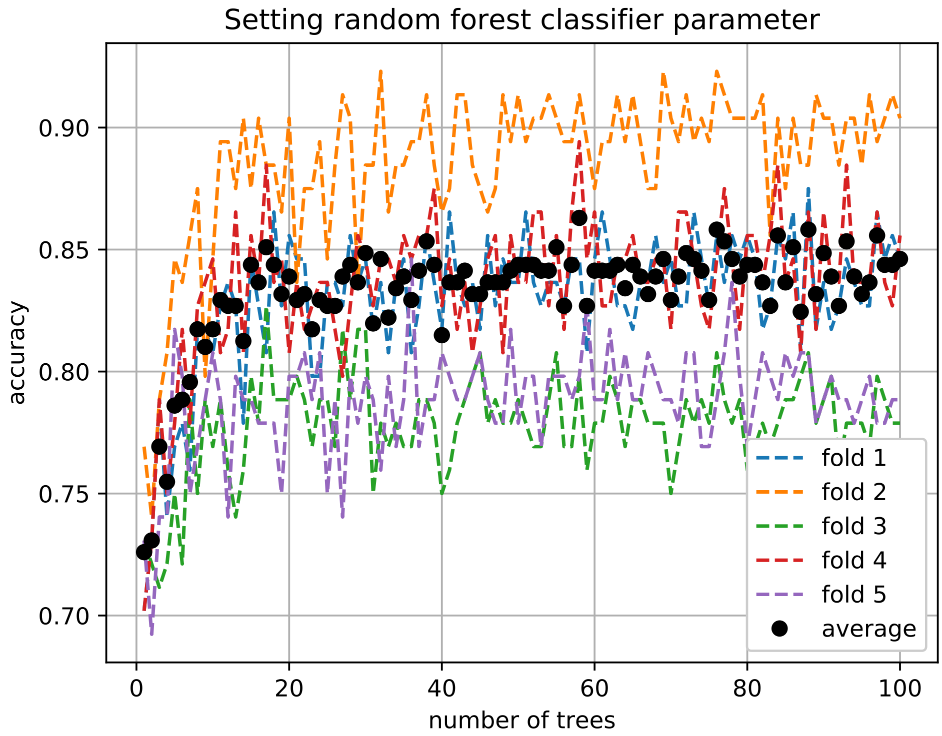

FIGURE 4.12

The confusion matrix obtained, after having set the hyperparameter to 57, is reported in table 4.4.

	pred: Entrance	pred: Exit
true: Entrance	74	13
true: Exit	17	70

TABLE 4.4: Confusion matrix.

Table 4.5 recalls the differences between this classification and the previous one.

With this results a semi-supervised classification is out of the way. The machine should learn only from entrances detected by the automatic rule and be able to detect every entrance without wrongly classify other events as entrances. It is enough to make one mistake that the counter would be wrong and the advanced motion detector could not even evaluate if the room is empty or not.

Further studies were conducted in order to better understand the source of missclassification. In particular, the number of people inside the room during the event

	I	II
detector	phase difference	phase difference
seconds per sample	15	4
predicting logic	presence	counter
# features	55	55
classes	last, other, false exits	entrances, exits
# train samples	120,144,99	260,260
# test samples	40,48,33	87,87
classification	RFC	RFC

TABLE 4.5: Summary of classification II.

was considered. From figure 4.13, it was possible to conclude that the motion detector mainly miss-classified entrances and exits that happened when someone else was in the room.

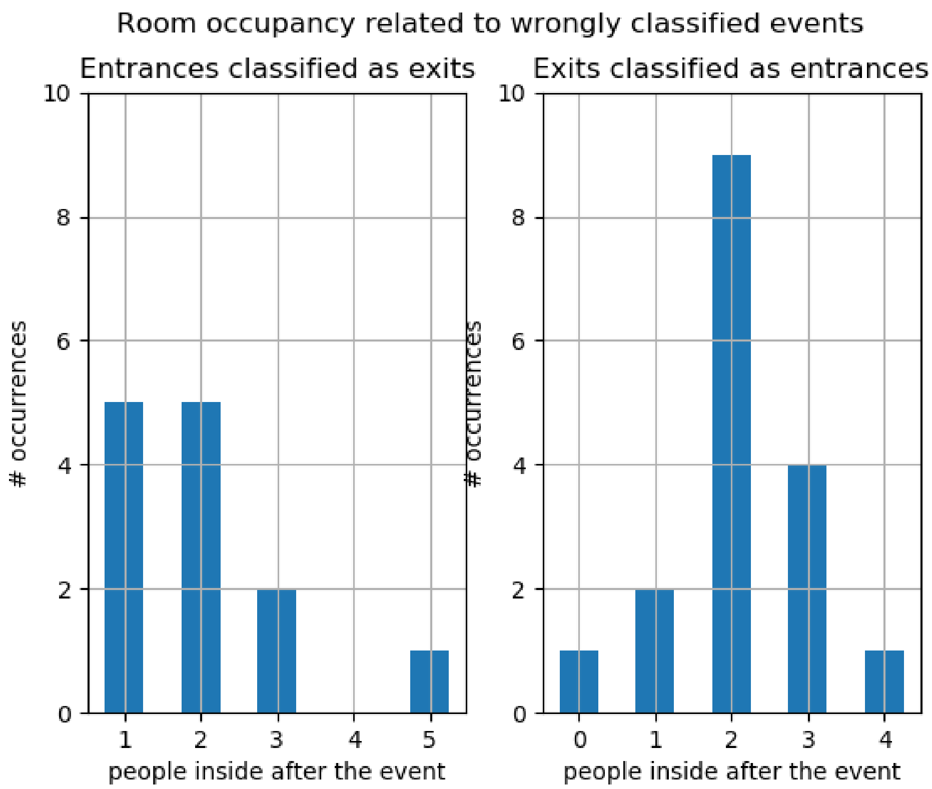

FIGURE 4.13: Entrances and exits visualized in different domains.

For this reason, a different motion detector was searched for, one that could possibly split the contribution of motions happening at the door, from those related to the rest of the room.

Chapter 5

Dual zone motion detector

5.1 Dual zone motion detector

The motion detector chosen to split the field of view in two regions is shown in figure 5.1. As it is possible to see, it is made of two passive infrared sensors placed at 90 degree angle, while in the previous motion detector they had the same orientation. Moreover, the plastic cover is different, being symmetrical for the two sensors.

FIGURE 5.1: Dual zone motion detector.

In figure 5.2, the output signal of the two motion detectors and related to a false entrance is shown. The difference is clear: when a motion happens after a period of inactivity, the voltage related to the room is the only one responding in the dual zone motion detector.

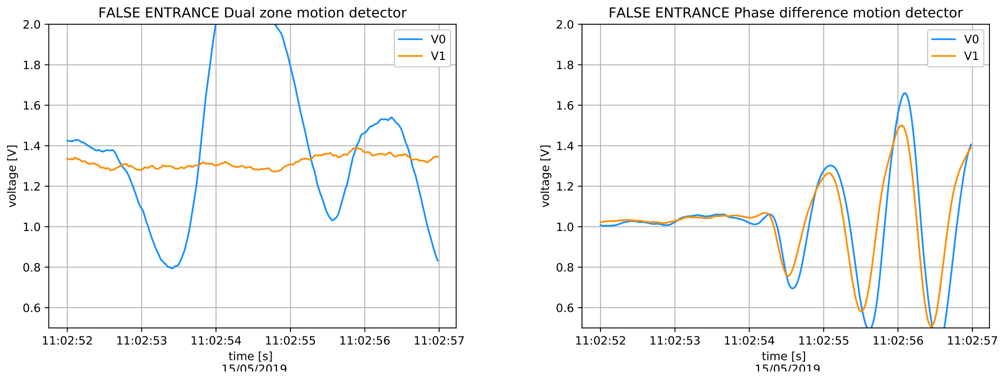

FIGURE 5.2: Signals of the two motion detectors related to a false entrance.

Some entrances and exits (36 samples each) were detected in the video recordings and their corresponding four seconds were extracted from the raw data. In figure 5.3, two samples from each class are shown.

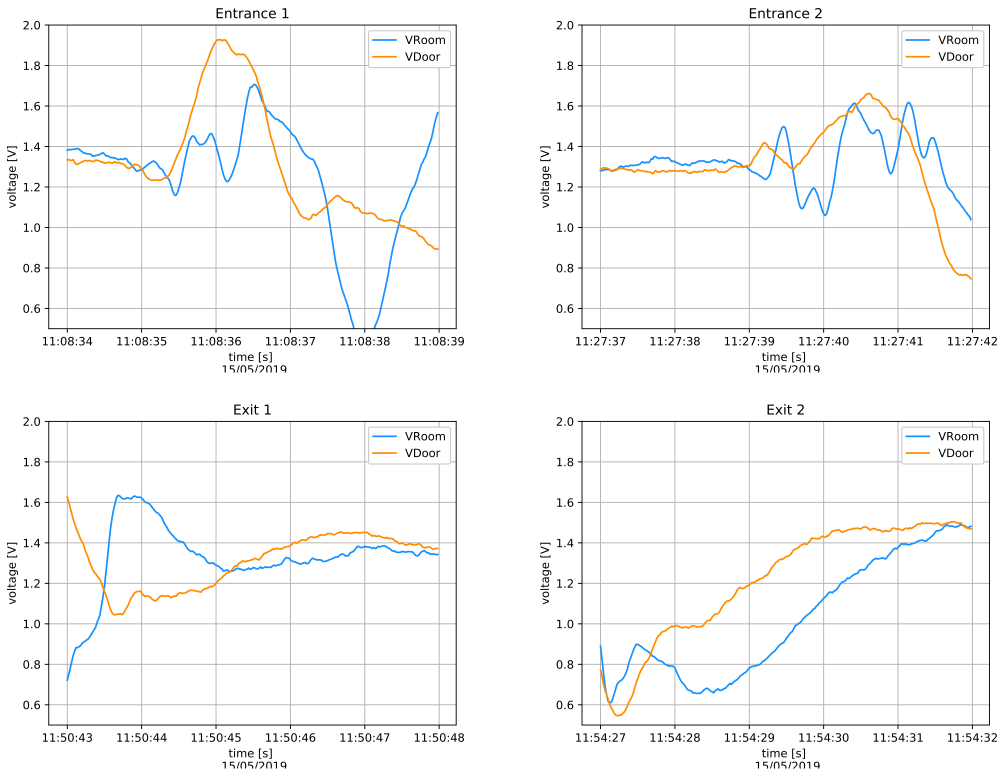

FIGURE 5.3: Entrance and exit samples.

Since no relevant characteristic was found in the time domain analysis, entrances and exits were analyzed in other domains and the resulting plots are shown in figure 5.4.

From the analysis it is clear that the features discussed in section 4.1.2 are not useful.

For this reason, different features were searched for.

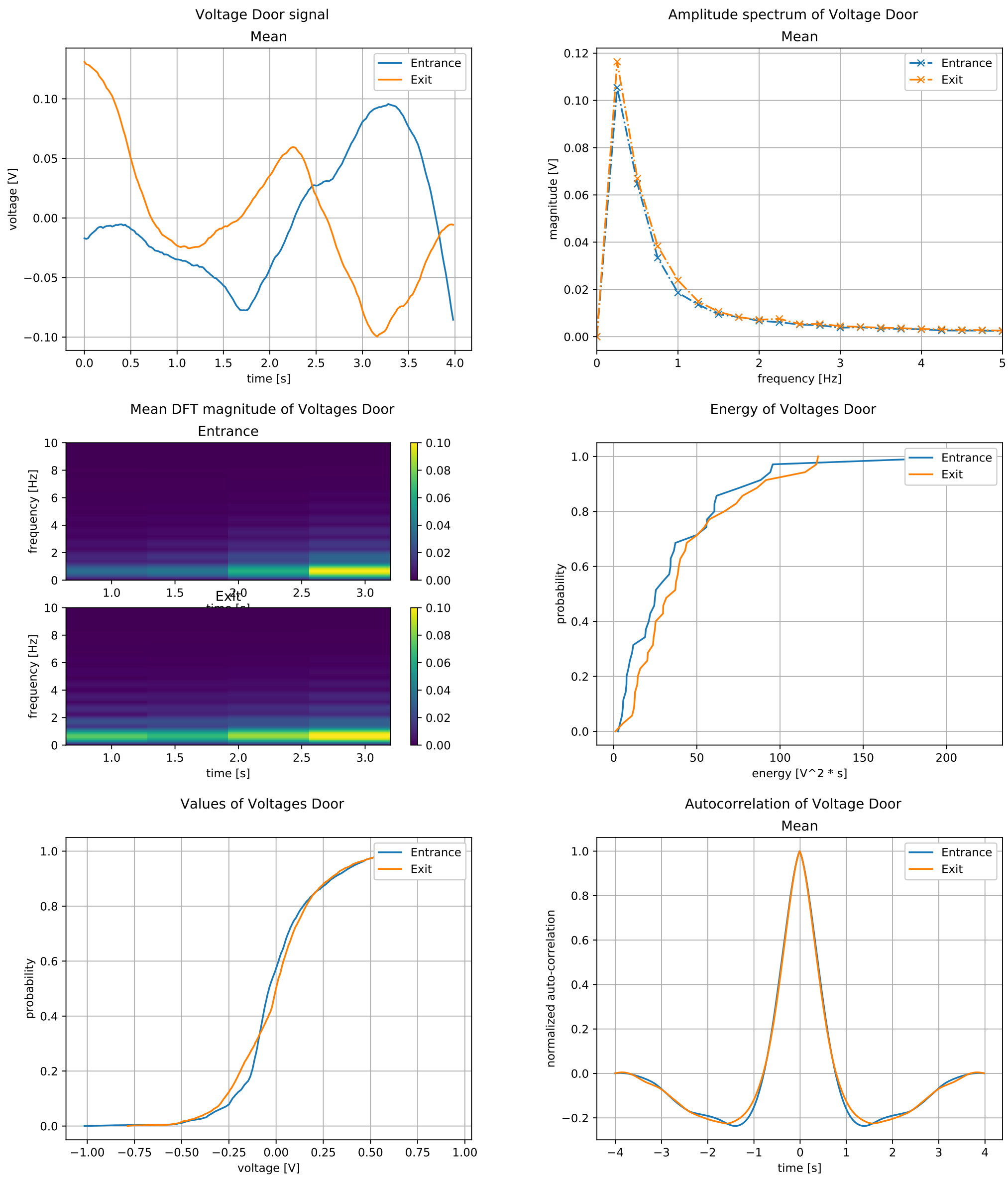

FIGURE 5.4: Entrances and exits visualized in different domains.

5.2 Time series features

After some researches, a Python package called *tsfresh* was found. This package automatically extracts a large number of features from a time series. The complete list of features together with their description is available on the web documentation [11]. Among the list, one can find simple features like the sum over the time series values, and more complex ones like the continuous wavelet transform for the Ricker wavelet.

In order to use the package, the raw data had to be formatted in a different way from the one provided by the data acquisition system (shown in table 3.1).

In particular, every event (made of 4 seconds related to entrances or exits) had to correspond to an id, since features are extracted individually for each one. The resulting data structure is provided in table 5.1.

id	time	V0	V1
0	0	1.016640	1.025030
0	1	1.015995	1.025676
...	...	...	...
1	0	1.024510	1.054130
...	...	...	...

TABLE 5.1: Input format required by tsfresh.

The *tsfresh* package also contains a function to find the relevant features among the extracted ones. This is done by means of the FRESH algorithm (FeatuRe Extraction based on Scalable Hypothesis tests [12]). The filtering process is represented in figure 5.5.

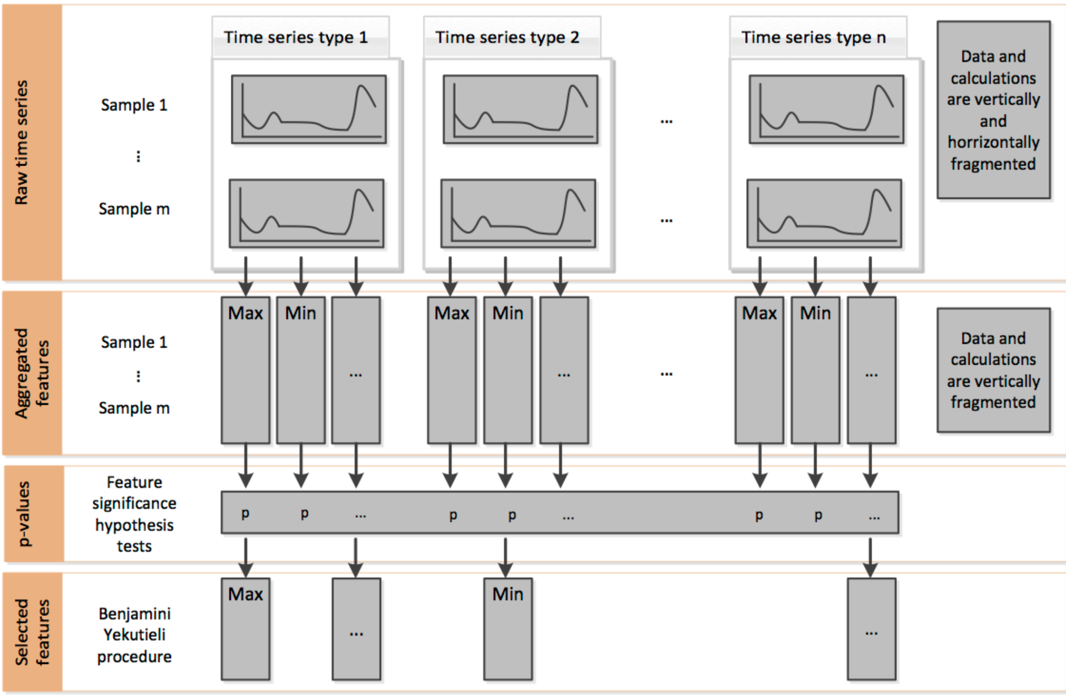

FIGURE 5.5: FRESH algorithm filtering process [11].

After features were extracted from the raw data, the Mann-Whitney U test [13] is used to assign a p-value to each feature. This value quantifies the significance of the feature for predicting the target and is defined as:

$$\text{p-value} = 2 * min\{P(X \leq x \mid H_0), P(X \geq x \mid H_0)\} \tag{5.1}$$

The null hypothesis H_0 is that the feature under test is not relevant and has no influence on the target. In general, it is rejected (resulting in the feature being relevant) when the p-value is smaller than 5%, thus when there is a smaller than 5% chance that the result is random.

In order to avoid that a small p-value happen by chance, and a feature is incorrectly considered relevant, the FRESH algorithm applies the Benjamini-Hochberg procedure [14].

This procedure decreases the false discovery rate:

$$FDR = \mathbb{E}\left[\frac{|\text{false rejections}|}{|\text{all rejections}|}\right] \tag{5.2}$$

5.3 Supervised classification

When applying *tsfresh* package to classification of entrances and exits, the extracted features are 1576 out of which only 57 are considered relevant, those that in figure 5.6 appear under the rejection line. It is interesting to compare the obtained feature significance with respect to the one evaluated by a random forest classifier. As shown in the figure, the two methods result in a different order of importance. In fact, at the right of the graph, features least relevant (according to the fresh algorithm) are present, but their normalized importance is not decreasing. However, both methods identify the same set of features as most significant, as shown in the left part of the figure.

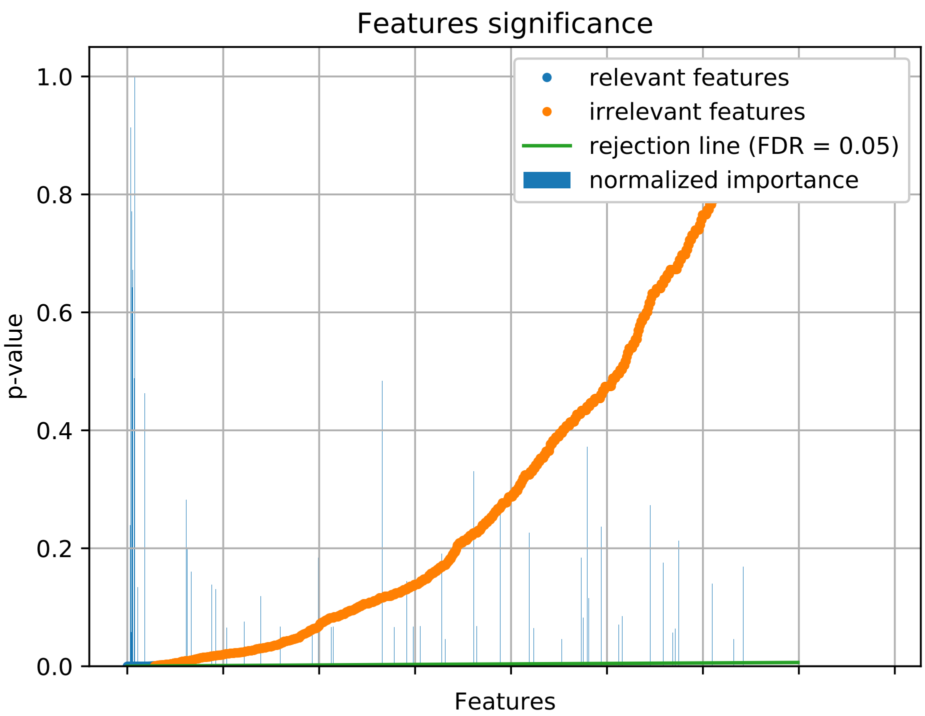

FIGURE 5.6: Features significance for classification task according to tsfresh method and random forest classifier.

Out of the 57 relevant features, according to the fresh algorithm, those considered not important by the random forest classifier were excluded, ending with only 14 features after the feature selection process.

These features were used to perform random forest classification and the results were optimal (table 5.2).

Table 5.3 recalls the differences between this classification and the previous ones.

	pred: Entrance	pred: Exit
true: Entrance	9	0
true: Exit	0	9

TABLE 5.2: Confusion matrix.

	I	II	III
detector	phase difference	phase difference	dual zone
seconds per sample	15	4	4
predicting logic	presence	counter	counter
# features	55	55	14
classes	last, other, false exits	entrances, exits	entrances, exits
# train samples	120,144,99	260,260	27,27
# test samples	40,48,33	87,87	9,9
classification	RFC	RFC	RFC

TABLE 5.3: Summary of classification III.

An optimal result was expected since features were selected using the same data set. A further test with a new data set was required but, since the company needed a result, it was time to pass to semi-supervised learning. In fact, until now, supervised learning was used since each data was labeled. However, once the advanced motion detector will be deployed, it will not be possible to label data except for the ones that comply with the automatic rule.

5.4 Semi-supervised classification

Thanks to the automatic rule (section 3.4), it was possible to extract samples of first entrances and last exits. Assuming that first entrances produce a similar couple of signals to the one of every entrance, then a machine trained with these samples should be able to detect every entrance. By making a similar assumption for last exits, another machine should detect every exit.

The two machines must be trained in a semi-supervised manner since samples for only one class are present (the class of the event under investigation). In fact, for the not-event class, it is not possible to choose all the samples excluded by the automatic rule because some entrances and exits could be part of them.

The choice of semi-supervised learning was suggested by a popular application of it: novelty detection, where the objective, is to decide whether, or not, a new observation belongs to the same distribution as training observations [15].

There are different machine learning techniques allowing to perform novelty detection, the one chosen to start with was the One-Class SVM.

Training samples were extracted from the raw data thanks to the automatic rule. In 20 analyzed days, 75 last exits were found and confirmed after checking the video recordings. At this point only exits were considered.

In order to test the performance, half an hour of data, during which many entrances and exits happened, was considered. In particular, one sample vector was created every second, with a duration of four seconds: two before and two after.

Moreover, the labels "Exit" and "Not-Exit" were assigned to each testing sample with the help of the recorded video. It was chosen to assign the label "Exit" at the sample centered at the corresponding second, not at the other samples containing it.

The results of classification with default parameters provided by *svm.OneClassSVM()* are shown in table 5.4.

	pred: Not Exit	pred: Exit
true: Not Exit	1718	65
true: Exit	10	5

TABLE 5.4: Confusion matrix.

In table 5.5 a summary of the differences between this classification and the previous ones is provided.

	I	II	III	IV
detector	phase difference	phase difference	dual zone	dual zone
seconds per sample	15	4	4	4
predicting logic	presence	counter	counter	counter
# features	55	55	14	14
classes	last, other, false exits	entrances, exits	entrances, exits	exit or not
# train samples	120,144,99	260,260	27,27	75
#test samples	40,48,33	87,87	9,9	half hour
classification	RFC	RFC	RFC	OneClassSVM

TABLE 5.5: Summary of classification IV.

By plotting the samples which were classified as exits (e.g. figure 5.7), it was found out that the voltage related to the door zone reacted also to some motions happening in the other zones of the office.

The cause of the problem was thought to be the reflection of radiations, coming from the room, inside the plastic cover of the motion detector. The solution, as shown in figure 6.1, was to apply a mask in order to block the radiations.

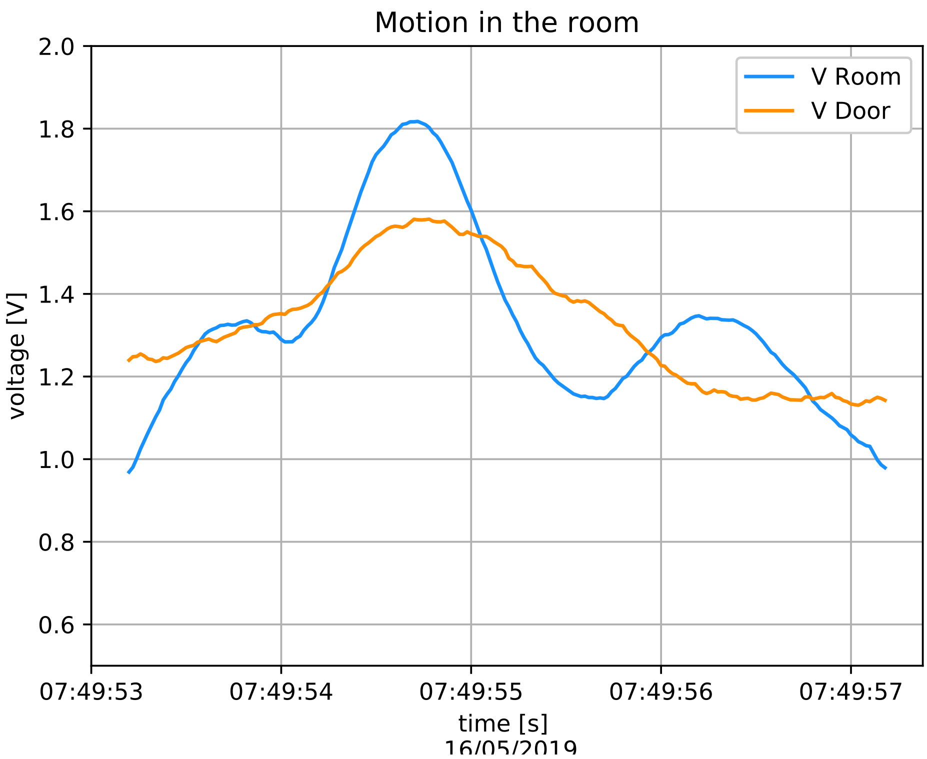

FIGURE 5.7: Voltage door reacting to motion inside the room.

Chapter 6

Dual zone with mask

FIGURE 6.1: Applied mask in order to better separate the two detection zones.

6.1 Samples extraction

Another problem found after plotting the miss-classified samples was that also the sample before or after the one corresponding to the exit, were sometimes classified as an exit.

For this reason, after the mask was applied, some entrances and exits were analyzed in order to study a better way of extracting samples.

In figure 5.4, "Step Entrance" and "Step Exit" represent a person entering or exiting the room with only one step and which stops after it. The difference between the two events is clearly visible. "Complete Entrance" and "Complete Exit" show that the voltage related to the door zone is quite similar during the two events, for this reason both voltages must be considered during the classification.

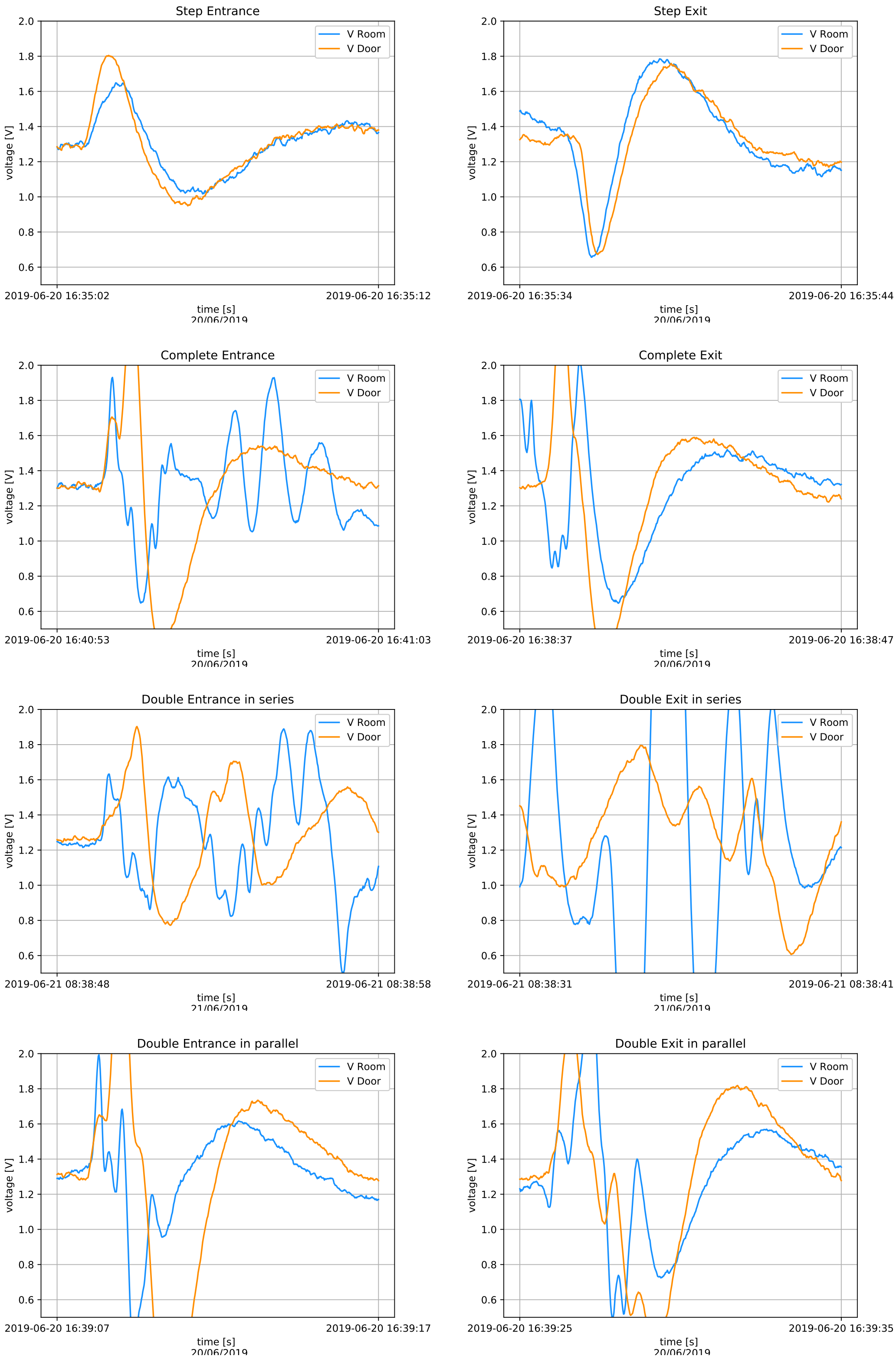

FIGURE 6.2: Output signals of dual zone motion detector with mask.

Regarding the portion of signals to extract, there is a similarity between "Step Entrance" and "Complete Entrance" in the raising part of the two voltages. It is more difficult to find a similarity between "Step Exit" and "Complete Exit", the idea was to consider the last portion of the signals.

Assume to train two separate machines, one for the entrances and one for the exits, with the aforementioned portion of signals as samples. The machine in charge of entrances would probably detect the first entrance in "Double Entrance in series" and in "Double Entrance in parallel", while the other one would probably detect the second exit out of the two happening during a double exit.

For this reason, it is possible to conclude that is impossible to detect every entrance and every exit, and thus, to have a perfect people counter, by training a machine with only "first entrances" and "last exits". However, it was interesting for the company and for research purposes to continue in the development of a smart motion detector.

By means of the automatic rule, some samples of entrances and exits were extracted from the raw data.

This time, however, in order to precisely extract the same portion of signals for each sample, the function *scipy.signal.find_peaks()* was used. Basically, once the automatic rule detects the point before or after which an event occurs, the segment between the preceding and following peak is extracted. Moreover, since the length of each sample is variable, only the last second is considered.

After having extracted the samples, their timestamps were used to check in the video recordings whether or not each sample was a real event. This operation was very useful also because it allowed to determine the number of people in the room after the event happened.

In table 6.1, the amount of training samples detected by the automatic rule, and filtered by video checking, is provided. The last column refers to the number of training samples that correspond to a "first entrance" or "last exit". These samples are the ones that should be detected by the automatic rule when the period of inactivity is set to be of the order of hours, as should be done in the deployed motion detector. In this phase it was set to minutes in order to detect as many "first entrances" and "last exits" as possible, with the disadvantage of detecting also false events or other entrances and exits (not first nor last).

Event	# Detected	# Real	# First/Last
Entrances	129	114	43
Exits	105	70	35

TABLE 6.1: Amount of training samples.

All the training samples (first entrances and last exits) are shown in figure 6.3.

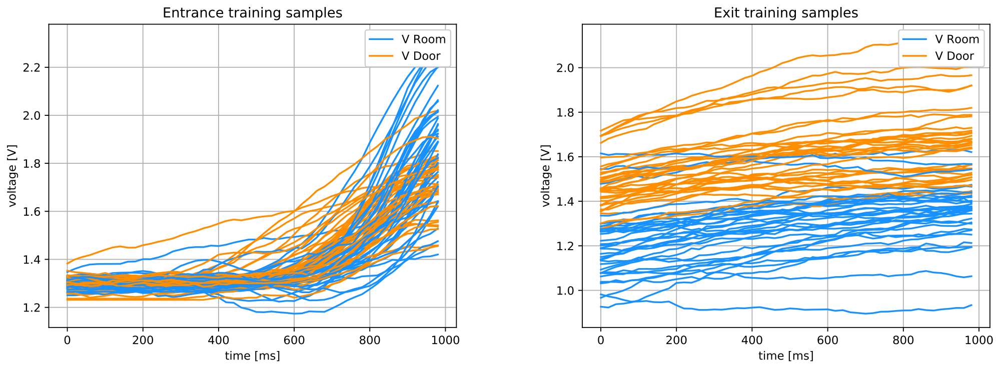

FIGURE 6.3: Training samples.

To better understand, in figure 6.4, an example of training sample per class is provided together with the signal from which it was extracted.

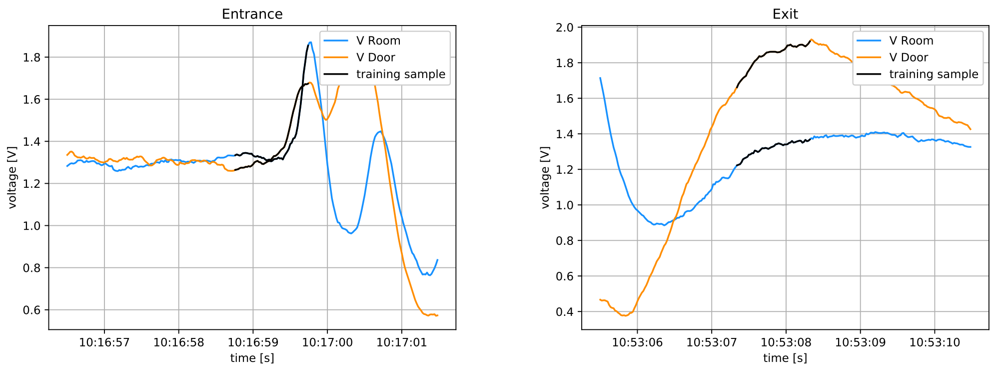

FIGURE 6.4: Examples of training samples before extraction.

The block diagram changed because of the different way to extract entrances and exits. The new version is shown in figure 6.5.

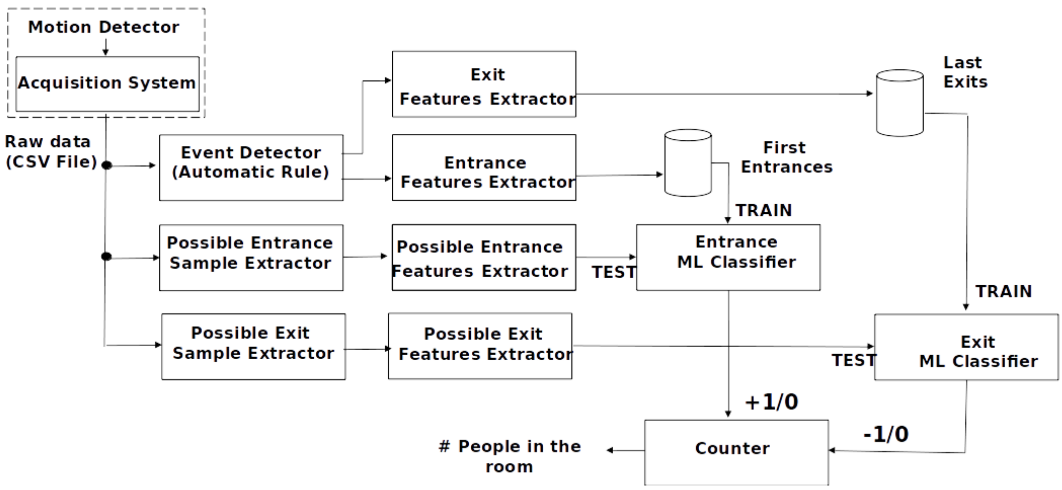

FIGURE 6.5: Block diagram of dual zone with mask.

6.2 Feature selection

As previously done in section 5.3, significant features were evaluated. Out of 942 features considered, 567 were relevant and only 11 were found to be important according to the Random Forest classifier. The features significance is represented in figure 6.6.

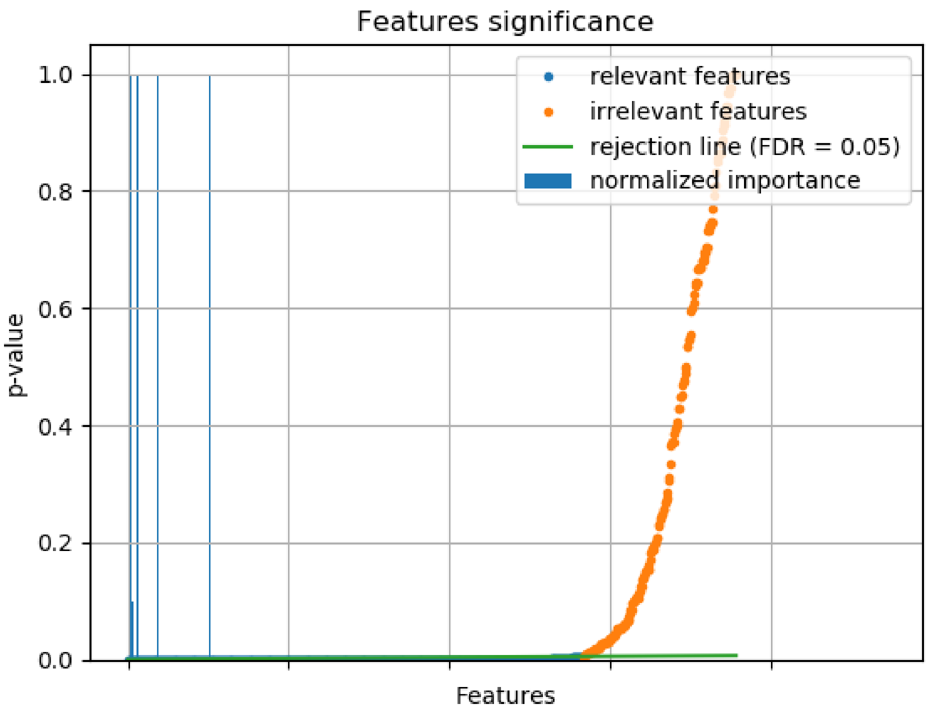

FIGURE 6.6: Feature significance for entrances and exits classification.

Then, semi-supervised classification was performed to classify other entrances and exits by learning only "first entrances" and "last exits". The results are provided in table 6.2.

	pred: Not Entrance	pred: Entrance
true: Not Entrance	0	0
true: Entrance	2	69

	pred: Not Exit	pred: Exit
true: Not Exit	0	0
true: Exit	2	33

TABLE 6.2: Confusion matrix.

As usual, the differences between this classification and the previous ones are provided in table 6.3.

	I	II	III	IV	V
detector	phase difference	phase difference	dual zone	dual zone	mask
seconds per sample	15	4	4	4	1
predicting logic	presence	counter	counter	counter	counter
# features	55	55	14	14	11
classes	last, other, false exits	entrances, exits	entrances, exits	exit or not	entrance or not, exit or not
# train samples	120,144,99	260,260	27,27	75	43,35
# test samples	40,48,33	87,87	9,9	half hour	71,35
classification	RFC	RFC	RFC	OneClassSVM	OneClassSVM

TABLE 6.3: Summary of classification V.

The results achieved prove that the selected features allow a machine trained only with "first entrances" to detect also other entrances and, a machine trained only with "last exits" to detect also other exits.

However, the samples used as test were extracted by means of the automatic rule, thus similar to the training samples. For this reason, a better test was needed.

6.3 Test set

In order to test the two machines, data related to an hour and half, during which many events happened, were selected. Then, test data were split into samples according to the peaks found by the function *scipy.signal.find_peaks()*.

The next step was to watch the video record and write down the time related to each entrance or exit. Thanks to these timestamps, it was possible to assign the correct label to each test sample. But, a difficulty was encountered: the web-cam used to record the office had a resolution in time only of seconds. Unfortunately, there was not a one-to-one correspondence with the test sample. Moreover, the portion of signal representing the exit usually happens few seconds after the real exit.

For these reasons, the samples were plotted and to each sample a label was assigned. In order to facilitate the process, each sample has been initially labeled as not entrance or not exit (one test set per machine). Then, the sample happening at the time given by the video event was plot together with some samples happening before and after it (figures 6.7 and 6.8). Finally, the sample with a signal shape corresponding to the event was selected and properly labeled.

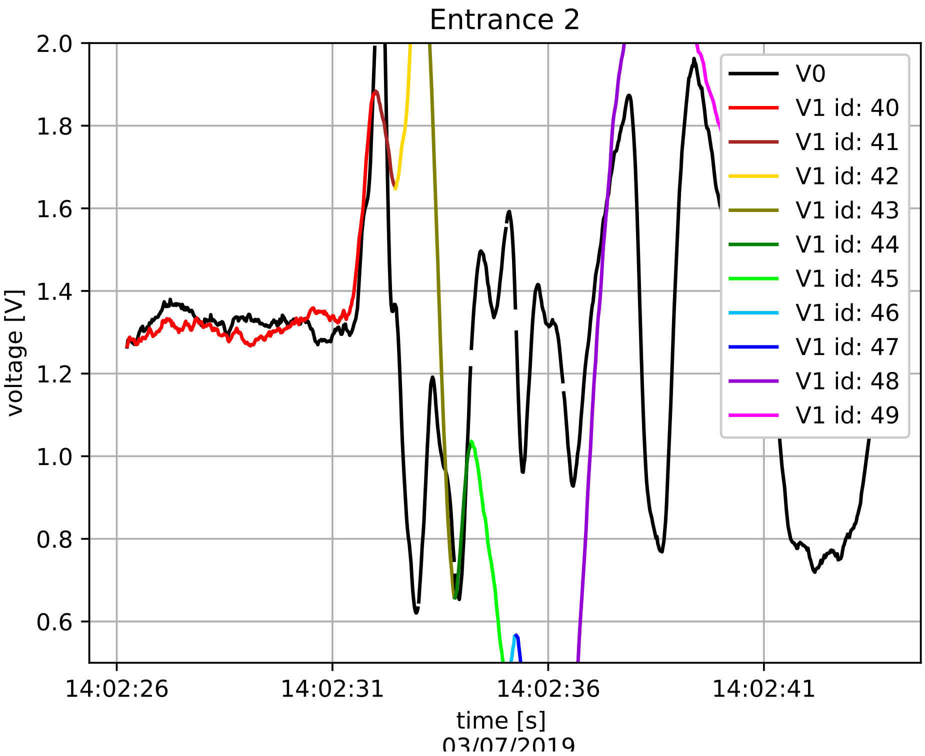

FIGURE 6.7: Example of label assignment for entrances.

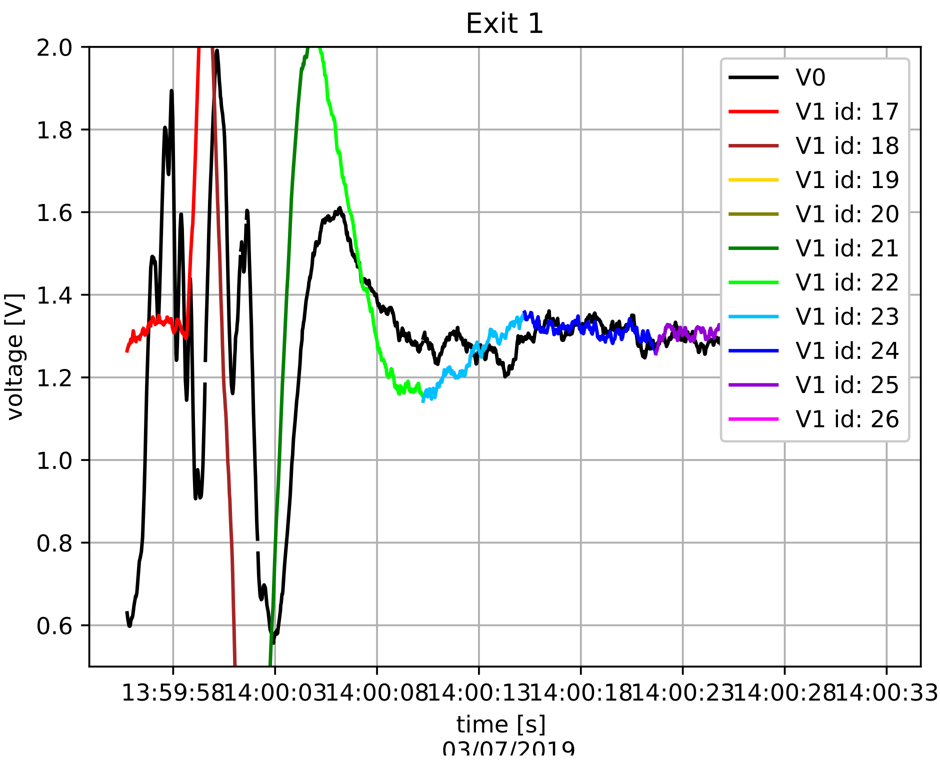

FIGURE 6.8: Example of label assignment for exits.

6.4 Classification

Semi-supervised classification was performed with two different machines. The first one, entitled to detect entrances, was trained with first entrances and tested on the test samples previously created. The possible labels for each sample were: entrance and not entrance. In a similar way, the second machine entitled to detect exits was trained with last exits and tested on samples with labels: exit and not exit.

The F_1 score was chosen as performance metric to extract a single value from the confusion matrix. It is evaluated in the following way:

$$F_1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{precision} = p(y = 1 | \hat{y} = 1)$$
$$\text{recall} = p(\hat{y} = 1 | y = 1)$$

The advantage of this performance metric is to not consider the true negatives. In fact, by considering them, a machine classifying every sample as "not event" would provide good results since the number of entrances and exits is way less than the number of not events

The results, provided in tables 6.4 and 6.5, show that the two machines have never classified a "not event" as an "event". However, more than half of the "events" were classified as "not events". In particular, the machine detecting entrances had a F_1 score of 44%, while the one detecting exits had 55%.

	pred: Not Entrance	pred: Entrance
true: Not Entrance	868	0
true: Entrance	5	2

TABLE 6.4: Confusion matrix.

	pred: Not Exit	pred: Exit
true: Not Exit	865	0
true: Exit	7	3

TABLE 6.5: Confusion matrix.

Table 6.6 recalls the differences between this classification and the previous ones.

	I	II	III	IV	V	VI
detector	phase difference	phase difference	dual zone	dual zone	mask	mask
seconds per sample	15	4	4	4	1	1
predicting logic	presence	counter	counter	counter	counter	counter
# features	55	55	14	14	11	11
classes	last, other, false exits	entrances, exits	entrances, exits	exit or not	entrance or not, exit or not	entrance or not, exit or not
# train samples	120,144,99	260,260	27,27	75	43,35	43,35
# test samples	40,48,33	87,87	9,9	half hour	71,35	one hour and half
classification	RFC	RFC	RFC	OneClassSVM	OneClassSVM	OneClassSVM

TABLE 6.6: Summary of classification VI.

The mis-classified samples were further analyzed by watching the video records and by checking the number of people inside the room at the moment during which the

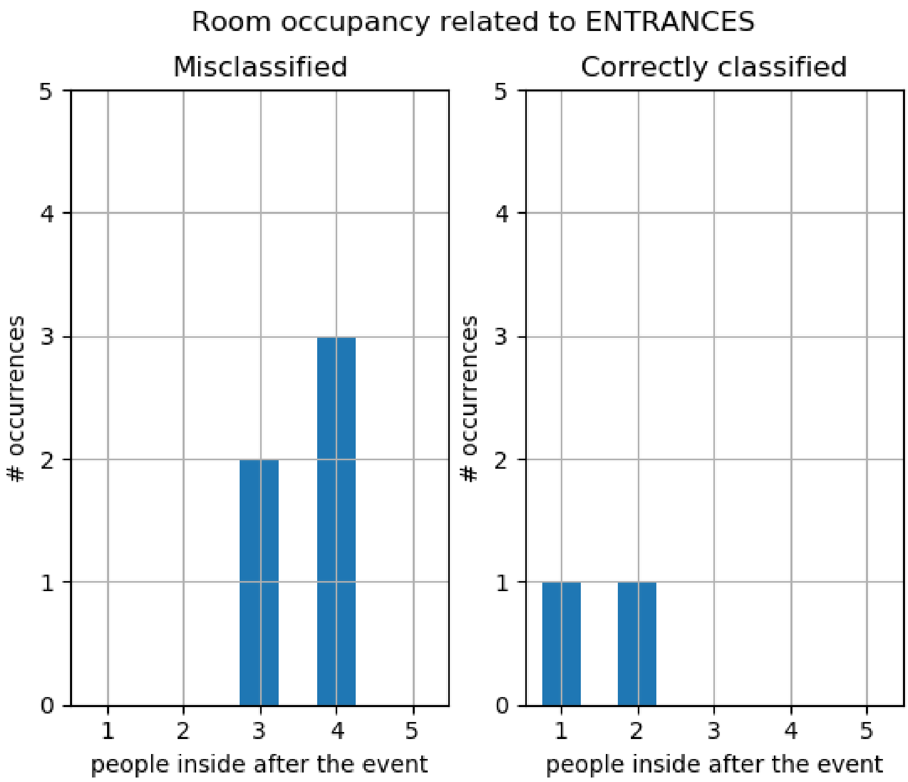

FIGURE 6.9: Room occupancy related to entrances.

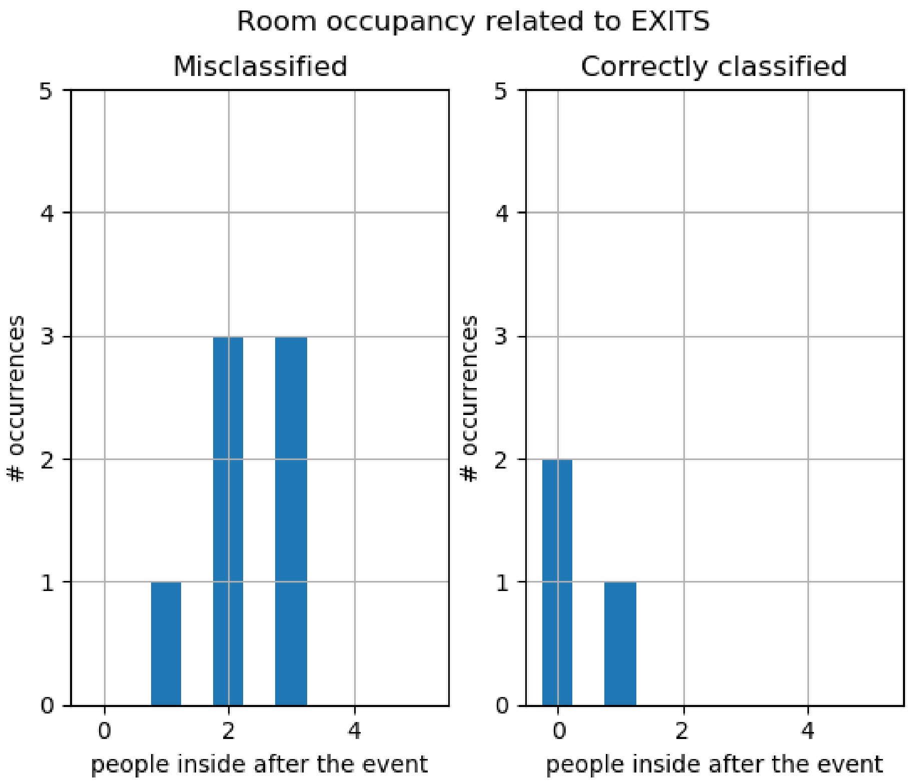

FIGURE 6.10: Room occupancy related to exits.

event happened. As shown in figures 6.9 and 6.10, the machines were able to correctly classify entrances and exits only when the room was empty or almost empty.

The fact that the machines have a poor F_1 score does not allow to evaluate room occupancy based on a counter because a single event not detected would lead to a wrong result. For example, the system could light on the office at the first entrance, not detect the second entrance and turn off the lights when the first person exits, leaving the second person in the dark.

Instead of using a counter, one could decide to detect only the first entrance and the last exit but, as shown in the previous figures, the machines also detect other entrances and exits.

Chapter 7

Conclusions and suggestion for further works

7.1 Conclusion

Ater the analysis of the phase difference PIR-based motion detector, it was concluded that it is not possible to distinguish between "last exits" and "other exits". For this reason, instead of detecting "first entrances" and "last exits" to evaluate if the office is empty or not, it was decided to detect every entrance and exit and build a counter of the number of people in the room.

Due to a misclassification between entrances and exits, the phase difference motion detector was replaced with a dual zone motion detector. Since the features extracted for the first detector were not useful, the *tsfresh* Python package was used. However, after feature selection, the semi-supervised classification did not provide good results.

At this point, it was thought that the reflection of radiations inside the plastic cover was interfering with the ability of the motion detector to see two different zones. For this reason, a mask was applied in order to block radiations. Moreover, due to a problem of double counting, a different method of sample extraction was searched.

After having analyzed specific motions, it was concluded that it is not possible to detect every entrance and exit by training a machine with only "first entrances" and another one with only "last exits". In particular, the F_1 scores obtained on a test set of an hour and half, during which many events happened, were 44% for entrances and 55% for exits.

In conclusion, the goal of evaluating room occupancy in almost real-time was not achieved due to the PIR technology and to the context of offices. This context was found hard to work with due to several reasons: people almost static for a long period, people standing by the door while talking with someone in the office, people entering and exiting just for the time of a hand-shake, and multiple people entering or exiting very close to each other.

The best results were obtained by focusing on a one second segment extracted from the raw data with specific rules.

The replicability of the results must be verified in different contexts (people, room, season), some parameters were set according to the analyzed context but they should not affect the replicability.

In conclusion, even if the goal was not achieved, the work done has been a success for the company. In fact, after the thesis, they learned the limit of PIR-based motion detectors. In particular, they learned that, exploiting classical machine learning, a motion detector can not become an event detector.

7.2 Possible improvements

Many different choices can be made with respect to the ones made during this thesis, probably the biggest change would be to apply deep learning. As already said, classical machine learning has been preferred due to the limited time available for this thesis. But, by training the two machines with "first entrances" and "last exits", it was not possible to detect double events (double entrance, double exits, and entrance followed by an exit or vice-versa). It does not seem possible with classical machine learning, thus, it would be interesting to see if double events are detectable by using deep learning.

Double events are not the only events not detected by the designed system. Also single events happening in a dynamic context are misclassified. In order to detect them, the suggestion is to analyze only the voltage related to the door in a first moment, and then use the voltage related to the office as a double check.

Bibliography

[1] Hans J. Keller. “30 Years of Passive Infrared Motion Detectors - a Technology Review”. In: (May 2000).

[2] Texas Instruments. “Advanced Motion Detector Using PIR Sensors Reference Design For False Trigger Avoidance”. In: (October 2017).

[3] J. Zhang et al. “Energy Savings for Occupancy-Based Control (OBC) of Variable-Air-Volume (VAV) Systems”. In: (January 2013).

[4] P. Zappi, E. Farella, and L. Benini. “Enhancing the spatial resolution of presence detection in a PIR based wireless surveillance network”. In: (September 2007).

[5] *How PIRs Work*. URL: https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/how-pirs-work.

[6] Kevin P. Murphy. *Machine Learning: a probabilistic perspective*. 2012.

[7] F. Wahl, M. Milenkovic, and O. Amft. “A green autonomous self-sustaining sensor node for counting people in office environments”. In: (May 2012).

[8] J. Yun and S. Lee. “Human Movement Detection and Identification Using Pyroelectric Infrared Sensors”. In: (May 2014).

[9] Y. P. Raykov et al. “Predicting room occupancy with a single passive infrared (PIR) sensor through behavior extraction”. In: (September 2016).

[10] *Machine Learning with Signal Processing Techniques*. URL: http://ataspinar.com/2018/04/04/machine-learning-with-signal-processing-techniques/.

[11] *Tsfresh python package documentation*. URL: https://tsfresh.readthedocs.io/en/latest/.

[12] M. Christ, A. Kempa-Liehr, and M. Feindt. “Distributed and parallel time series feature extraction for industrial big data applications”. In: (October 2016).

[13] *MannWhitney U test*. URL: https://en.wikipedia.org/wiki/Mann-Whitney_U_test.

[14] Yoav Benjamini and Daniel Yekutieli. “The control of the false discovery rate in multiple testing under dependency”. In: (2001).

[15] *Novelty and Outlier Detection*. URL: https://scikit-learn.org/stable/modules/outlier_detection.html.