UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE MÁSTER

Estimation of classroom occupancy using a multimodal sensory system

Máster Universitario en Investigación e Innovación en Tecnologías de la Información y las Comunicaciones (i2-TIC)

Autor: OMARI, Hind
Tutor: RODRÍGUEZ LUJÁN, Irene
Ponente: SERRANO JEREZ, Eduardo

Septiembre, 2017

ABSTRACT

The mammalian olfactory system had provided inspiration for a new class of electronic devices called electronic noses (e-nose) with applications in a wide variety of domains such as environmental monitoring, medical diagnosis, and industrial processes, among others. Electronic noses detect volatile chemical compounds, being more objective than human or canine experts and working continuously without exhaustion.

In this Master's Thesis, we will focus on the use a multimodal sensory network composed by an e-nose in order to detect presence and estimate the number of occupants in a classroom, which can be considered as an uncontrolled or semi-controlled environment. We have collected an extensive database from a multisensory network composed of 12 sensors.

To address the occupancy detection and occupancy estimation problems, we propose a model that combines a classification algorithm for occupancy detection followed by a regression algorithm for occupancy estimation. This model is applied over two types of datasets extracted from our e-nose records: the first type of data is formed by a set of statistical features summarizing the sensors' response behavior during a period of time, and the second type of data is defined by attributes modeling the rising and decaying portions of the sensors' resistance computed from the Exponential Moving Average of the signals.

On the one hand, the classification accuracy rates for the occupancy detection task vary from 93% to 100% using a Logistic Regression model. On the other hand, the best result for the occupancy estimation problem is obtained using a Random Forest algorithm that achieves a Mean Absolute Error of 5 people and a Mean Relative Error of 13%. The latter result corresponds to a dataset based on statistical variables, being the most relevant sensors the CO2 and wifi sensors, and the CO2, TGS 2600, and temperature sensors in the absence of the wifi sensor. The models trained with datasets formed by EMA features do not obtain competitive results as their error rates are very high in comparison with those achieved by the models

based on statistical variables.

In summary, this Masters Thesis presents promising results that demonstrate the ability of chemical sensors and wifi sensor to successfully address the presence detection and occupancy estimation problems. The main novelty of this work compared to other studies in the literature relies on the use of Metal Oxide (MOX) sensors in the sensoring network as well as the recording of data during several months.

ACKNOWLEDGMENT

First of all I would like to thank my parents for their endless love, support and help throughout my life. Thanks a lot for everything you have done.

Thanks to my brother and my sisters for all their help during all my life especially writing this thesis, thank you for correcting my English errors and vocabulary.

I would like to sincerely thank my tutor Dr. Irene Rodríguez Luján for giving me the chance to work in this project, and for her guidance, support and encouragement throughout this year. I would also like to thank Dr. José R. Dorronsoro Ibero for giving me the opportunity to join the Machine Learning group. Also, I would like to thank Dr. Francisco De Borja Rodríguez Ortiz, Dr. Eduardo Serrano Jerez, and Dr. Pablo Varona Martínez for their helpful comments and advice throughout this Master's Thesis.

I also thank my labmates in the Machine Learning group (GAA), for all the times that we spent together during this last year, thanks for you help and encouragement.

Last but not least, I would like to thank ERASMUS MUNDUS Battuta program for giving me this amazing chance to study this master, that was such an amazing experience.

Thank you God.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The sense of smell is a chemical sense that detects and analyses volatile chemical substances (odors) present in the air. Smell is a very powerful sense due to its ability to change our heart rate, attract us to a mate or stir our memories to different times in our lives, and it can also alert us from danger such as gas leak, fire or rotten food.

The sense of smell plays an important role in a wide variety of activities including daily hygiene, industry, and medical diagnostics.

Given the importance of odors and the sense of smell, it has been thought to develop an electronic alternative that acts the same way as the mammalian smell system occupying the positions of humans and dogs by detecting different odors. The most important advantage of this alternative is that it works continuously without tiredness and with objectivity. In addition, all the data captured will be automatically stored in a database.

1.1 Artificial noses

Perhaps the earliest attempt to develop a machine capable of detecting fragrances was in the year 1920 when Hogewind, F and H. Zwaardemaker [Hogewind and Zwaardemaker, 1920], suggested that odors can be detected by mea-

suring the electric charge in a fine stream of water containing an aqueous solution.

The "electronic nose" concept was firstly discussed by Wilkens and Hatman in 1964 [Wilkens and Hartman, 1964], but the term "electronic nose" was not mentioned until later during a conference in 1987 [Gardner et al., 1988]. Ten, the artificial olfactory system design was established by Gardner et al. [Gardner et al., 1990], and the first conference dedicated to e-nose was celebrated in 1990 [Gardner, 1991].

From a technical point of view, artificial noses are electronic systems with analytical capacities whose purpose is to detect volatile organic compounds (VOCs) that are part of an odorous sample and can thus recognize or discriminate them within a set of odorous substances. Electronic noses are basically composed of a multisensor array, which in turn consists of different sensors that respond to a wide range of chemical gases. Figure 1.1 and Table 1.1 show the similarities between an electronic nose and a biological nose. The main objective of an e-noses is to be able to identify and/or quantify some type of aroma or gaseous sample. An e-nose typically consists of a sensing or detection system responsible for sensing odour and collecting data, and a computing system aimed at preprocessing and analyzing the signal by means of pattern recognition algorithms focused on discriminating among odors and/or quantifying the amount of different substances. In laboratory experiments, an electronic nose usually includes a sample delivery system that generates samples to be analyzed. Therefore, the e-nose workflow is formed by the following components:

- **The sample delivery system:** The sample delivery system enables the generation of volatile compounds. These volatile compounds are sent to the sensing system of the e-nose.

- **The sensing system :** A system consisting of a multisensor array or a group of sensors which is able to generate electrical signals in response to either simple or complex volatiles compounds present in the gaseous sample, and then transforms these signals into digital values.

- **The computing system :** A mechanism for pattern recognition that mimics the human brain: it combines the responses of all sensors to produce results that can be analyzed to identify/quantify the chemical volatiles exposed to the e-nose.

Table 1.1: Comparison between electronic and biological nose.

Biological nose	**E-nose System**	**Functions**
Nostril	Sampler	Serves as gas detection chamber
Olfactory receptor	Sensory array	Sense odor and collect data
Olfactory bulb	Signal conditioning & data preprocessing	Analyze and process data
Brain or Olfactory cortex	Pattern Recognition	Classify the smell and/or quantify its chemical compounds

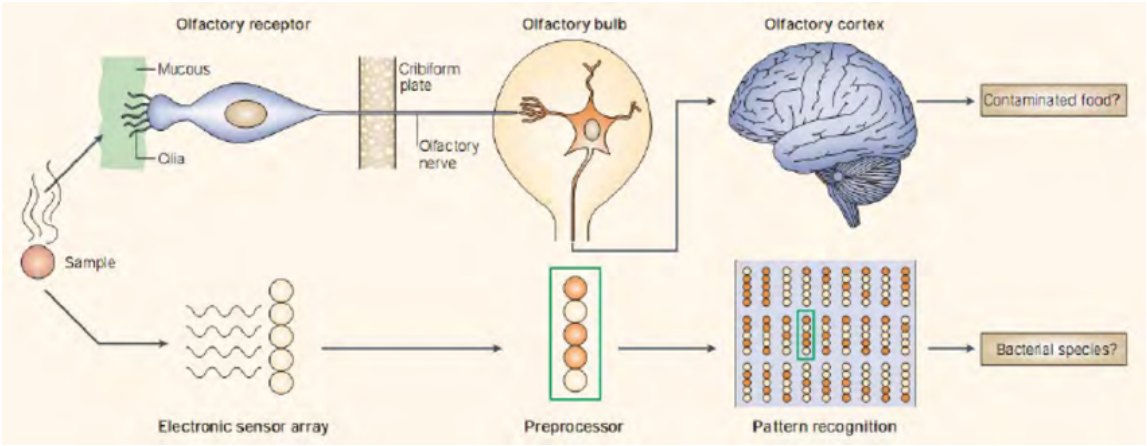

Figure 1.1: Electronic nose devices mimic the human olfactory system. Source: [Zhao and Yongxin, 2012].

Among the different modules involved in an e-nose system, this Master's thesis is focused in the computing system responsible for processing the data coming from the

sensors in order to detect and quantify the gas exposed to the e-nose. In particular, the objective of this work is to analyze the data coming from an e-nose located in a classroom by means of machine learning algorithms in order to detect and estimate the classroom occupancy. The machine learning techniques used in this work will be explained in Chapter 2, and the obtained results will be presented in Chapter 4.2.

Electronic noses were originally used by the food, beverage and cosmetic industries in order to control the quality of their products as e-noses can detect hazardous or poisonous gases [Chilo et al., 2016, Pathange et al., 2006]. Current applications include detection of odors specific to diseases for medical diagnosis [D'Amico et al., 2010], and detection of pollutants and gas leaks for environmental protection [Baby et al., 2000].

1.2 Goals and Outline

The goal of this work is to apply machine learning techniques to data from an uncontrolled or semi-controlled environment coming from a multimodal sensor system installed in a classroom in the Autonomous University of Madrid to detect human presence in the classroom (occupancy detection) and to estimate the number of occupants (occupancy estimation). As far as we know, some of the sensors used in this work have never been used to solve these type of problems.

In order to achieve this objective, we will proceed as follows: first, we will study the state of the art to determine the pattern recognition techniques used for the analysis of data from artificial noses in both controlled and uncontrolled environments, and their particular application in the detection and estimation of occupation in different environments such as offices and classrooms. Second, taking into account the results from previous works in the literature, we will prepare the data collected from our electronic noses records to build an input dataset to be analyzed by the machine learning techniques. Next, we will apply a series of data analysis and machine learning techniques over our datasets to solve the occupancy detection and the

occupancy estimation problems.

The organization of this Master's thesis is as follows: in Chapter 2, we present an overview of some previous works that have used machine learning techniques with data coming from e-nose and introduce the pattern recognition techniques used in this work including feature selection algorithms as well as classification and regression models. Different metrics used to evaluate the performance of the machine learning models are also presented. Chapter 3 explains how data were collected using the e-nose installed in a classroom in the Autonomous University of Madrid. This section also describes the four datasets that were generated using the information captured by the sensors. The experimental setup and the empirical results are presented in Chapter 4. Finally, the conclusions of our investigation and some suggestions of further research lines are presented in Chapter 5.

Chapter 2

Related work and pattern recognition techniques

This chapter present an overview of some previous works that have used e-nose to detect or to estimate the occupancy of a room, as will as, a resume of the machine learning techniques used in our work.

2.1 Related work

There are several studies that have successfully applied machine learning techniques in controlled environments [Fonollosa et al., 2014, Muezzinoglu et al., 2009], but their application in uncontrolled environments is more limited due to the greater difficulty of the problem and the lack of data [Monroy et al., 2016].

In [Candanedo and Feldheim, 2016], the authors study the problem of detecting whether an office is occupied or not using data from light, temperature, humidity and CO2 sensors. They achieve a classification accuracy rate of 97% using a Linear Discriminant Analysis (LDA) model based on the information coming from a combination of two sensor namely: temperature and light, light and CO2 and light and humidity, light and humidity ratio. They also show that regardless of the classifier used, the high classification accuracy is always obtained when using the light sensor which means that the light sensor is the most relevant for the classification problem.

To solve the same problem, [Hailemariam et al., 2011] proposed to use information captured by CO2, light, motion, sound and electrical current (Power consumption of two computers) sensors. They achieve a classification accuracy rate that ranges from 81.019% using only the light sensor, to 98.44% using only the motion sensor. In both cases, the classification is performed by a decision tree model.

A more complicated problem is to estimate the number of occupants in a room. To deal with this problem, Yang et al. [Yang et al., 2012] propose to use a Radial Basis Function (RBF) Neural Network model with a combination of sensors that detect indoor temperature, humidity, CO2, light, sound and motion to estimate the number of occupants in two laboratories with a maximum number of occupants of 5 and 8, respectively. They achieve a classification accuracy rates of 88.74% and 86.50%, respectively. To get a classification accuracy rate from the regression problem, a prediction is considered as correct if the difference between the estimated and real value is less or equal to 1. A more general approach is presented in [Rodrigues et al., 2017], where they propose to use three types of environmental variables (relative humidity, air temperature and CO2 concentration) to estimate the number of occupants in a classroom. They propose to use a MultiLayer Perceptron (MLP) model trained with information coming only from two of the three environmental variables.

More precisely, the features used as input for the MLP model were defined as the average of the last 5 samples of each sensor's response [Rodrigues et al., 2017]. Their results show that the models in which the CO2 sensor was included were the ones with the lowest Mean Absolute Error (MAE), reporting a MAE of 1 occupant. However, it should be noted that the performances obtained in this work were calculated considering also data corresponding to periods of time when the classroom is empty, such as weekends or nights, so their error estimates may be positive biased. In [Ekwevugbe et al., 2013], a back propagation Artificial Neural Network (ANN) algorithm is applied over data coming from the following sensors: temperature, light, humidity, CO2, sound and PIR (motion). Data are captured for a period of 8 days to detect occupancy in a room. They achieve a classification accuracy rate of 84.59%

and the most effective sensors were the sound and the motion ones.

As far as we know, the problems of occupancy detection and occupancy estimation have not been addressed using a network of multimodal sensors like the one employed in this work and capturing data over such a prolonged period of time.

The problem of occupancy detection can be also addressed from other points of view without using chemical sensors. For example, in [Kleiminger et al., 2013], the electricity consumption in 5 households was used to detect their occupancy. The classification accuracy was on average above 80%.

Table **??** summarizes the related works by showing the used sensors and algorithms as well as the classification accuracies or the accuracies of number of occupants

Regardless of the type of sensors to be used to detect/estimate occupancy, different strategies can be applied to extract features from the sensors' response signal that will serve as the input for the machine learning algorithms. On the one hand, in [Monroy et al., 2016], the authors show that features generated form a moving average and a moving variance techniques with sliding windows allow improving the performance of a classifier trained over sensor's response raw data up to 6% as past information is taken into account in the features. On the other hand, in [Muezzinoglu et al., 2009] and [Vergara et al., 2012], features are created using an Exponential Moving Average (EMA) approach to solve the problem of gas classification . The Exponential Moving Average (EMA) is a type of infinite impulse response filter that applies weighting factors that decrease exponentially using the following equation:

$$y[t] = (1 - \alpha)y[t - 1] + \alpha(r[t] - r[t - 1]) \tag{2.1}$$

where t=1,2,...T, being T duration of the experiment, r[t] is the raw sensor's response at time t, and y[t] is the EMA's filter for r at time t with initial condition y[0]=0. Finally, the parameter α is a smoothing parameter that takes values between 0 and 1. Muezzinoglu et al. conclude that the transient features from EMA provide a classification accuracy of 92.9% when used together with SVM classifier, while the

steady state features –features that describe the dynamic process of the whole signal– only achieve a classification accuracy of 63.3%.

However, in the scope of our work, EMA's features are not helpful to either have a good classification accuracy in the occupancy detection task or to obtain a good regression model for occupancy estimate problems. This point will be discussed in detail in Chapter 4.2.

2.2 pattern recognition techniques

Machine Learning is the sub-field of computer science and a branch of artificial intelligence whose goal is to develop theoretical foundations, models, and procedures that allow computers to learn from data. More concretely, machine learning algorithms are capable of generalizing behaviors from provided data. Machine learning algorithms can be divided into supervised and unsupervised models as a function of the learning paradigm they address.

Supervised learning is a family of techniques for deriving a function from training data. The training data consist of object pairs, the input data of the algorithm (usually vectors), and the desired outputs. These algorithms are typically used to solve classification and estimation or regression problems. A simple example would be to detect if there is someone in a room or not, for learning this we should train our machine with different tagged data. On the other hand the unsupervised learning is a family of techniques where the model is adjusted to the input data, These algorithms are typically used in clustering and representation learning.

In this section we will introduce the machine learning algorithms used in this work which include feature selection, classification, and regression models. In all cases, we will assume a labeled dataset $(\mathbf{x}^{(n)}, y^{(n)})$ for n = 1,2,...,N, where $\mathbf{x}^{(n)} \in \mathbb{R}^D$ represents each data sample formed by D features, and $y^{(n)} \in \{0,1\}$ is the label associated to the sample. In the case of binary classification problems, , while for regression problems $y^{(n)} \in \mathbb{R}$.

2.2.1 Feature Selection

Feature selection in machine learning is a process of selecting a subset of features or variables to reduce the dimensionality of the patterns to be used in the classification or the regression model. Feature selection helps to simplify the model, accelerate the training and/or testing processes, and reduce over-fitting.

The main premise when using a feature selection algorithm is that the data contain many features that are irrelevant or redundant, and can thus be removed without incurring much loss of information.

2.2.1.1 Quadratic Programming Feature Selection

Quadratic Programming Feature Selection (QPFS) is a feature selection algorithm that formulates feature selection as a quadratic programming problem. It has been shown to be competitive with state-of-the-art feature selection methods in terms of classification accuracy, while it reduces the training times of several multivariate filter-type feature selection methods thanks to the use of the Nyström approximation [Rodriguez-Lujan et al., 2010].

Given a dataset with D features, the QPFS formulation is

$$\begin{aligned} &\min_{\mathbf{w}} \tfrac{1}{2}(1-\alpha)\mathbf{w}^T Q\mathbf{w} - \alpha F^T\mathbf{w}, \\ &\text{subject to} \quad w_i \geq 0 \forall \quad i = 1...D \quad ||w_1|| = 1 \end{aligned} \tag{2.2}$$

where $\mathbf{w}$ is a D dimensional vector that represents the weights gived to the features, Q is a $D \times D$ symmetric positive semi-definite matrix that represents the redundancy among the features, and F is a D dimensional vector of non-negative values that measures the correlation between each feature and the target class.

Considering the components of $\mathbf{w}$ as the weight or importance of each feature, its optimal value represent the feature ranking, thereby the features with the higher weight are the most relevant ones.

2.2.2 Classification

Among supervised machine learning, techniques, classification algorithms try to identify which categories or classes a new observation belongs to, based on a training set of data containing observations (samples or patterns) whose category is already known. In this work, we use a linear classifier, the Logistic Regression model, due to its simplicity and good results in our datasets.

2.2.2.1 Logistic Regression

The logistic regression is possibly the **well-known** statistical model of binary classification for its simplicity and good performance in simple problems.

The Logistic Regression model estimates the probability $P(Y = 1|X = \mathbf{x}^{(n)})$ as a function of $\mathbf{x}^{(n)}$ as follows:

$$P(Y = 1|X = \mathbf{x}^{(n)}) = \frac{1}{1 + e^{-f(\mathbf{x}^{(\mathbf{n})})}} \tag{2.3}$$

where $f(\mathbf{x})$ is a linear function on the training samples that can be expressed as follows

$$f(\mathbf{x}^{(n)}) = w_0 + \mathbf{w}^T\mathbf{x}^{(n)} \tag{2.4}$$

This model predict y=1 if $P(Y = 1|X = \mathbf{x}^{(n)}) \geq 0.5$ and y=0 otherwise. Thereby, logistic regression defines a linear classification model where the decision boundary is defined by the solution of $f(\mathbf{x}) = 0$, which is equivalent to $P(Y = 1|X = \mathbf{x}^{(n)}) = 0.5$. Figure 2.1 shows $P(Y = 1|X = \mathbf{x}^{(n)})$ as a function of $f(\mathbf{x})$.

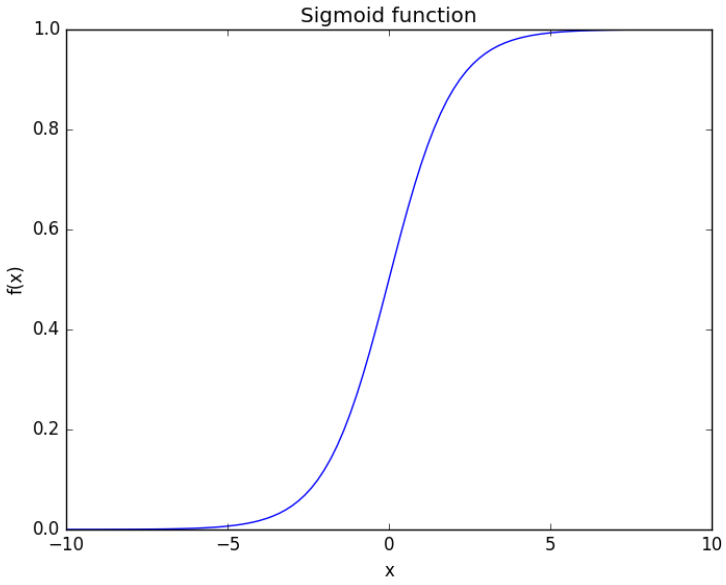

Figure 2.1: The sigmoid output in function of $f(\mathbf{x})$

To train logistic regression model and estimate the parameters $\mathbf{w}$, well-known methods such as maximum likelihood or ordinary least squares [Bishop, 2006, Hastie et al., 2009] are commonly used.

2.2.3 Regression

Regression models are supervised machine learning techniques that estimate a function $f(x)$ that maps from an input data-point to a real number based on training data. In general, a regression model can be written as follows

$$y^{(n)} = f(\mathbf{x}^{(n)}, \mathbf{w}) + \nu \tag{2.5}$$

where $f(\mathbf{x}, \mathbf{w})$ is the function that represent the regression model and ν is the noise term.

In this work, we use four different linear regression models (Linear Regression, Ridge Regression, Lasso Regression and Linear Support Vector Regression Machine),

and one non-linear regression model (Random Forest). These models will be briefly described in the following sections.

2.2.3.1 Linear Regression

Linear regression is one of the most popular regression models because of its simplicity and easy interpretability. This model assumes that the variable $y^{(n)}$ can be represented as a a linear combination of the input variables. Each data sample is represented as $\mathbf{x}^{(n)} = [x_0^{(n)}, x_1^{(n)}, ..., x_D^{(n)}]^T$ with $(x_0^{(n)} = 1)$ is used for modeling the bias. Therefore, the goal of a linear regression model is to find a vector of coefficients $w = [w_0, w_1, ..., w_D]^T$.

linear combination of the variables $\mathbf{x}^{(n)} = [x_0^{(n)}, x_1^{(n)}, ..., x_D^{(n)}]^T (x_0^{(n)} = 1)$ that represents the data samples formed by D features, with coefficients $\mathbf{w} = [w_0, w_1, ..., w_D]^T$. Then, the decision function of a linear regression model can be expressed as follows

$$\widehat{y}^{(n)} = f(\mathbf{x}^{(n)}, \mathbf{w}) = \sum_{i=0}^{D} w_i x_i^{(n)}, \tag{2.6}$$

where $\widehat{y}^{(n)}$ is the estimated value of $y^{(n)}$. Equation (2.6) can be rewritten in a matrix form by setting $Y = [y^{(1)}, y^{(2)}, ..., y^{(N)}]^T$ and X being a $N \times (D+1)$ matrix formed by all the patterns $x^{(n)}$, where each row represent a single pattern with an extra column with all entries equal to 1 to represent $x^{(0)}$. Then, the decision function of a linear regression model can be expressed as follows

$$\widehat{\mathbf{y}} = \mathbf{w}^T X. \tag{2.7}$$

To train this simple model and find the regression coefficients $\mathbf{w}$, we can use the *normal equations* for the least squares problem (See section 3.1.1 of [Bishop, 2006]):

$$\mathbf{w} = (X^T X)^{-1} X^T Y. \tag{2.8}$$

This equation gives us the vector of coefficients that minimizes the mean squared

error (MSE) between $\widehat{\mathbf{y}}$ and $\mathbf{y}$

$$MSE = \frac{1}{N}\sum_{n=1}^{N}(\widehat{y}^{(n)} - y^{(n)})^2 \tag{2.9}$$

2.2.3.2 Regularized Linear Regression

Regularization is a process that reduces over-fitting and improves the generalization capabilities of the machine learning model by adding a complexity penalty to the cost function [Hastie et al., 2009].

The least square regression method described in the previous section minimizes the sum of residual squares; however, it may be unstable and may produce over-fitting [Hastie et al., 2009, Mustafa et al., 2014]. Therefore, the inclusion of a regularization term in this model is highly advisable. One of the simplest form of regularization is to constrain the the magnitude of each weight w_j, thus favoring small values for w_j. The regularized cost function can be written as follows:

$$\min_{\mathbf{w}} \frac{1}{2}\sum_{i=1}^{N}(\widehat{y}^{(n)} - y^{(n)})^2 \qquad \text{subject to} \qquad |\mathbf{w}|_q < t \tag{2.10}$$

or in the Lagrangian form:

$$\min_{\mathbf{w}} \frac{1}{2}(\sum_{i=1}^{N}(\widehat{y}^{(n)} - y^{(n)})^2 + \lambda||\mathbf{w}||_q) \tag{2.11}$$

where $\widehat{y}^{(n)}$ is the predicted value for the i-th pattern, and $y^{(n)}$ is the real value, $\mathbf{w}$ is the vector of the regression coefficients, and $\lambda > 0$ is the regularization parameter. The larger the λ parameter, the more regularization is imposed in the model. The most used values for q are 1 and 2 corresponding to Lasso Regression and Ridge Regression, respectively [Hastie et al., 2009].

Ridge Regression

As mentioned above, the Ridge Regression model corresponds to $q = 2$ in the equation (2.10), also known as L2 regularization or quadratic regularization. Ridge

Regression was firstly introduced in statistics in 1970 [Hoerl and Kennard, 1970] to provide a solution in the case where $X^T X$ is not of full rank and, thus, it is not invertible and the solution to the least squares linear regression (equation 2.8) cannot be computed.

We can write Equation (2.11) in a matrix form as follows

$$\mathbf{w}_{ridge} = \min_{\mathbf{w}} \frac{1}{2}((\widehat{Y} - Y)^T(\widehat{Y} - Y) + \lambda||\mathbf{w}||_2^2) \tag{2.12}$$

or

$$\mathbf{w}_{ridge} = \min_{\mathbf{w}}(X \cdot \mathbf{w} - Y)^T(X \cdot \mathbf{w} - Y) + \lambda \mathbf{w}^T\mathbf{w} \tag{2.13}$$

Where X is the input matrix whose lines are the patterns, therefore X is s N $\times$ D dimensional matrix. Y is the vector of the N real values, $\widehat{Y}$ the vector of the N predicted values, and $\mathbf{w}$ the coefficient vector.

To train this model and find the regression coefficients $\mathbf{w}$, we can proceed in a similar way as in linear regression (See section 3.1.1 of [Bishop, 2006]) and obtain a closed form solution:

$$\mathbf{w}_{ridge} = (X^T X + \lambda I)^{-1} X^T Y \tag{2.14}$$

where I is the $N \times N$ identity matrix. Since the Ridge Regression model has a quadratic penalty $\mathbf{w}^T\mathbf{w}$, its solution is also a linear function of $\mathbf{x}$. Ridge Regression will not yield sparse models as all coefficients are shrunk by the same factor (none are eliminated) [Hastie et al., 2009].

Lasso Regression

The Lasso regression model uses the L1 regularization method that corresponds to $q = 1$ in Equation (2.10) . It can be shown that if λ is sufficiently large, some of the coefficients w_j are driven to zero, which makes Lasso to produce sparse models. The Lasso problem can be written in the equivalent *Lagrangian form* as follows

$$\mathbf{w}_{lasso} = \arg\min_{\mathbf{w}} \left(\frac{1}{2} \sum_{i=1}^{N} (y^{(n)} - \widehat{y}^{(n)})^2 + \lambda ||\mathbf{w}||_1 \right). \tag{2.15}$$

This penalty term makes the solutions nonlinear in $y^{(n)}$ so there is no closed form solution as in linear regression and ridge regression models. To compute the Lasso solution, we need to solve a quadratic programming problem, which have a high computational cost, although, there are efficient algorithms with the same computational cost of ridge regression, see Section 3.4.4 of [Hastie et al., 2009] for more details.

The most attractive characteristic of Lasso regression is the sparsity of its solutions, which not only reduces the number of operations needed to calculate the Lasso estimation, but also makes it possible to use Lasso as a feature selection algorithm [Hastie et al., 2009].

2.2.3.3 Support Vector Machines

Support Vector Machines (SVM) are considered as one of the best "off-the-shelf" supervised machine learning algorithms. Initially developed for binary classification problems [Burges, 1998], SVMs have been extensively researched by the machine learning community for the last decade, giving as a result SVMs' extensions to other type of problems such as Support Vector Regression (SVR) [Smola and Schölkopf, 2004] and Ranking SVM (or RankSVM) [Herbrich et al., 2000, Yu, 2005]. The main advantages of the SVM formulation are the generalization capability of the model guaranteed by the margin maximization, and the easy extension of SVMs to nonlinear functions by means of the kernel trick.

In this work, Support Vector Regression (SVR) models are used to estimate the number of people in a classroom. However, before explaining the fundamentals behind SVR, we need to introduce the binary SVM and the concept of margin.

SVM Classification

In a linearly separable problem, a binary linear classifier separates the input data in two classes or categories with an hyper-plane that may not be unique. For example, in Figure 2.2, it can be seen how there exist several hyper-planes capable of discriminating the two classes with a classification error equals to zero. Among these hyper-planes, we should try to find the one with the best generalization capability, that is, the hyper-plane that will correctly classify "unseen" or testing data with the highest probability. To do so, SVMs try to find the hyper-plane with the maximum separation between the two classes, or, in other words, the hyper-plane that has the largest margin. The margin is defined as the smallest distance between the decision boundary (hyper-plane) and any of the training samples, as illustrated in Figure 2.3.

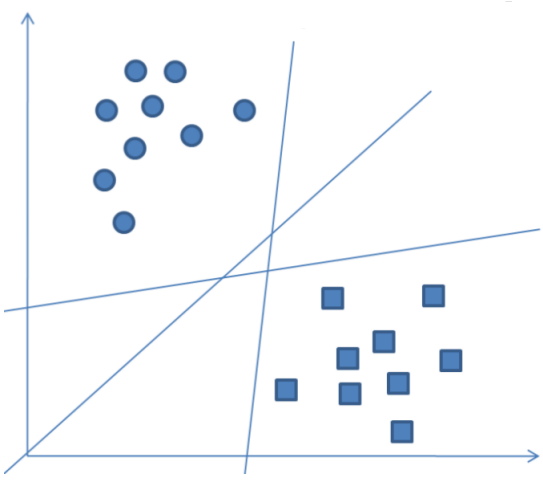

Figure 2.2: Linear classifiers (hyper-plane) in two-dimensional spaces

In the binary SVM formulation, the target variable is assumed to take the value -1 or 1; that is, $y^{(n)} \in \{-1, 1\}$. The SVM decision function $F(\mathbf{x})$ takes the form

$$F(\mathbf{x}) = \mathbf{w}\mathbf{x} - b \tag{2.16}$$

where $\mathbf{w}$ is the weights or coefficients vector and b is the bias, both will be determined during the SVM training phase. To correctly classify a pattern $\mathbf{x}^{(n)}$, $F(\mathbf{x}^{(n)})$ must

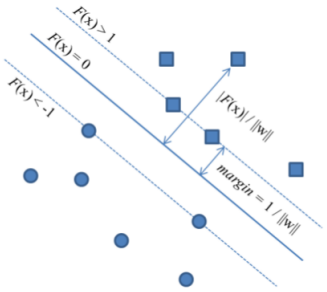

Figure 2.3: SVM classification function: the hyper-plane maximizing the margin in a two-dimensional space. Source: [22].

return a positive value if $y^{(n)} = 1$, and a negative value when $y^{(n)} = -1$. Therefore, the following inequality must hold:

$$y^{(n)}(\mathbf{w} \cdot \mathbf{x}^{(n)} - b) > 0, \qquad \forall(\mathbf{x}^{(n)}, y^{(n)}). \tag{2.17}$$

If the dataset is linearly separable, we can rewrite the condition in Equation 2.17 by adding the restriction $\mid F(\mathbf{x}) \mid \geq 1$ as

$$y^{(n)}(\mathbf{w} \cdot \mathbf{x}^{(n)} - b) \geq 1, \qquad \forall(\mathbf{x}^{(n)}, y^{(n)}). \tag{2.18}$$

The distance from the hyper-plane F to a vector $\mathbf{x}^{(n)}$ is given by $\frac{|F(\mathbf{x})|}{\|\mathbf{w}\|}$ and the margin becomes

$$margin = \frac{1}{\|\mathbf{w}\|}. \tag{2.19}$$

Therefore, in order to maximize the margin, we can minimize $||\mathbf{w}||$ under the restriction given in Equation (2.18). Thus, the training problem in SVM becomes a constrained optimization problem as follows,

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad & y^{(n)}(\mathbf{w} \cdot \mathbf{x}^{(n)} - b) \geq 1 \qquad n = 1, ..., N. \end{aligned} \tag{2.20}$$

This optimization problem, known as hard-margin SVM, does not have a solution if the dataset is not linearly separable, which is the common case in practice. To deal with such cases, we need to introduce some modifications, in this new formulation, known as soft-margin SVM, we introduce new variables $\xi^{(n)}$, called slack variables, which measure the degree of misclassification of the sample $\mathbf{x}^{(n)}$ in terms of how far it is x(n) from the correct side of its corresponding margin [Yu and Kim, 2012, Rao, 2013].

The soft-margin SVM formulation can be writing as :

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi^{(n)} \\ \text{s.t.} \quad & y^{(n)}(\mathbf{w}\cdot\mathbf{x}^{(n)} - b) \geq 1 - \xi^{(n)} \\ & \xi^{(n)} \geq 0 \qquad n = 1, ..., N. \end{aligned} \tag{2.21}$$

Slack variables $\xi^{(n)}$ in Equation (2.21), allow data samples to be misclassified to certain degree, and the amount of misclassification quantified by the slack variables will be minimized while the margin is maximized. The new hyperparameter $C \geq 0$ determines the tradeoff between the margin size and the amount of misclassification in training.

Support Vector Regression

SVM Regression is a modification of the soft-margin SVM for regression problems in which $y^{(n)} \in \mathbb{R}$. The key idea in SVR is the introduction of a ϵ-insensitive error function that is equals to zero if the absolute difference between the target $y^{(n)}$ and the predicted value $\hat{y}^{(n)}$ is less than ϵ, where $\epsilon > 0$. More precisely, the ϵ-insensitive error function is given by:

$$E(y^{(n)}, \hat{y}^{(n)}) = \begin{cases} 0 & \text{if } \left|y^{(n)} - \hat{y}^{(n)}\right| \\ \left|y^{(n)} - \hat{y}^{(n)}\right| & \text{otherwise} \end{cases} \tag{2.22}$$

The SVR formulation is given by:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}(\xi^{(n)} + \xi_*^{(n)}) \\ \text{s.t.} \quad & y^{(n)} - \mathbf{w}\cdot\mathbf{x}^{(n)} - b \leq \epsilon + \xi^{(n)} \\ & \mathbf{w}\cdot\mathbf{x}^{(n)} + b - y^{(n)} \leq \epsilon + \xi_*^{(n)} \\ & \xi^{(n)}, \xi_*^{(n)} \geq 0 \qquad n = 1, ..., N. \end{aligned} \tag{2.23}$$

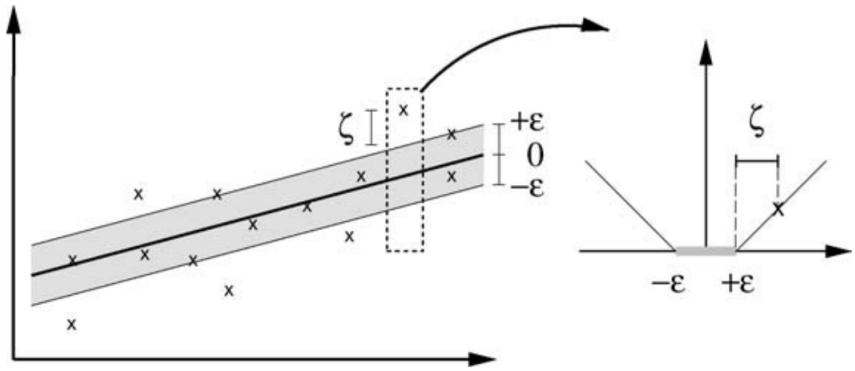

Figure 2.4: The soft margin loss setting for a linear SVM. Source [Smola and Schölkopf, 2004].

Figure 2.4 illustrates the idea behind the optimization problem in Equation 2.1 only the points outside the dark region contribute to the SVR cost function, outside this area the penalty increase in a linear fashion with the sample prediction error. This optimization problem can be solved more easily in its dual formulation [Yu and Kim, 2012, Smola and Schölkopf, 2004].

2.2.3.4 Random Forest

Regression Random Forest (RF) is an ensemble method, which means that it is a fusion of multiple weak learners [Breiman, 2001]. Random forest is built by training a large number of regression trees, and the RF output is computed as the average of individual trees output. To exploit the maximum from this fusion, we need to

guaranteed a diversity between individual trees. To do that RF uses two diversity techniques, Bootstrap Sampling and randomized feature selection. To explain how Random Forest algorithm works, we will first introduce its three principal elements : regression trees, bootstrap sampling or Bootstrapping, and Random Subspace method.

Regression trees

A Regression tree is a recursive partitioning regression model that consists on partitioning the space into smaller regions until we finally get pieces of space where the problem is sufficiently simple that we can fit simple models for them [Breiman et al., 1984, Hastie et al., 2009]. A regression tree is a binary tree, it starts by the root node from which outcome two branches, every branch ends in a new node from which outcome two new branches. Every inner node represent a condition over one problem variable, and the branches between represent the answers to this condition. this sequence of condition divide the input space in set of sub-spaces.

Figure 2.5 shows a regression tree for a problem with three input features and one output $\widehat{y}$ that is a continuous variable. We first start by dividing the input space into two subspaces according to the values of 'feature 1'; then, we divide again these two resulting spaces into two subspaces each one, one according to the values of 'feature 1' and the other one according to the values of 'feature 2' and so on. The predicted value $\widehat{y}$ in a leaf-node is the mean of the samples' targets belonging to that leaf-node.

Bootstrapping

Proposed by Efron in 1979 [Efron, 1979], Bootstrapping is a technique for reducing the variance of an estimated prediction function and improving the stability and accuracy of machine learning algorithms. It is used in both classification and regression problems.

The basic idea of Bootstrapping is to produce several data subsets from the original training set of the same size using random sampling with replacement. It

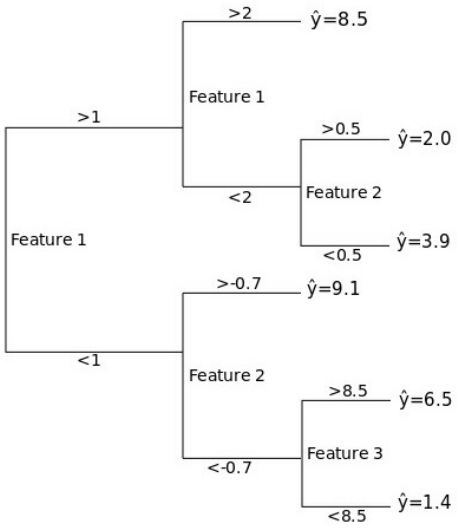

Figure 2.5: Regression tree for predicting a continuous output $\widehat{y}$ as a function of three input features.

means that some original training samples appear more than once, while some other are not present. A machine learning algorithm is trained over every generated subset of data, and the predictions of each model are combined by some voting scheme in classification problems, or by averaging the predictions in regression problems.

Random subspace method

Also know as Random subspace method [Bryll et al., 2003, Ho, 1998], the random subspace method is very similar to bagging except that it randomly samples features, not samples, for each learner. In this way, individual learners do not over-focus on features that seem to be highly predictive in the training set, but fail to be as predictive for points outside that set. This strategy prevents from over-fitting.

Random Forest

After introducing the different element of Regression Random Forest, we will

now describe how it works. Given a dataset of N samples and D features, a Random Forest Algorithm composed of B trees can be summarized as Algorithm 1:

for *b=1 to B:* **do**

1 - Draw a bootstrap sample of size N from the training data.

2 - Select randomly d variables form the D variables.

3 - Train a new tree with the bootstraped samples only with the d selected variables .

end

Averaging the individual tree outputs.

Algorithm 1: Random Forest algorithm for regression problem

We start by generating a new training dataset using a Bootstrapping technique [Hastie et al., 2009]. After that we randomly select d features from the original D-dimensional space, and we finally train a regression tree with the generated dataset. This procedure is repeated B times. After training the B trees, we aggregate them by averaging the individual tree output.

2.2.4 Evaluation Procedures

This section presents the evaluation procedures used in this work such as strategies for splitting our dataset in training/validation/test partitions to properly measure the generalization capability of the machine learning models, the evaluation metrics used to quantify the performance of these methods, and the hyperparameter search strategy followed to obtain the best configuration for each pattern recognition algorithm.

2.2.4.1 Hold-out

The hold-out method randomly splits the data into two subsplits, the first one is used for training the machine learning model, while the second partition is used as test set to measure the performance of the algorithm in "unseen data, and thus, estimate its generalization capability.

2.2.4.2 Cross Validation

Cross-Validation (CV) is a method used to estimate the reliability of a model. CV consist in splitting the training data in k splits, named folds, train the model with k-1 folds and test it with the remaining fold. This procedure is repeated until each fold has been used once as test set. The cross validation error is obtained as the average over the errors obtained in the test partition in each of the k iterations of the algorithm. Figure 2.6 shows an example of a 10-fold Cross Validation.

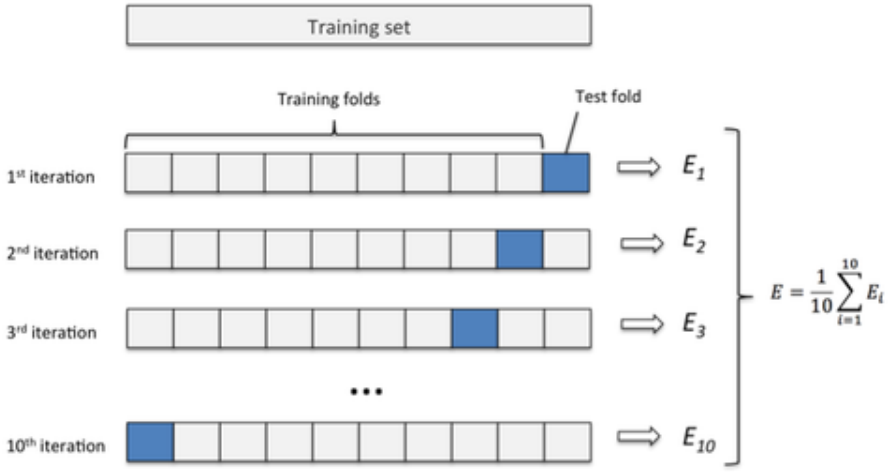

Figure 2.6: Example of a 10/-fold Cross Validation [Raschka, 2017]

2.2.4.3 Grid Search

Many machine learning algorithms have some hyperparameters or no-trainable parameters which have to be specified by the user. To select no-trainable parameters, we have to use an hyperparameter optimization method.

One of the most simple methods to perform hyperparameter optimization is the Grid Search algorithm that consists in an exhaustive searching through a subset of the hyperparameters of the learning algorithm.

To perform a grid search, we have first to define a range of values for each no-

trainable parameters and create all the possible combinations of no-trainable parameters. Then, we calculate the cross validation error for every combination, and, finally, we select those parameters that provide the best cross validation error.

2.2.5 Evaluation Metrics

This section contains a brief definition of the evaluation metrics that have been used to measure the performance of the classification and regression models such as classification accuracy, mean absolute error and mean relative error.

2.2.5.1 Classification accuracy

The classification accuracy $C_{accuracy}$ of an classification algorithm is defined as the proportion of samples correctly classified and it is evaluated by the following formula:

$$C_{accuracy} = \frac{m}{N} \times 100, \tag{2.24}$$

where m is the number of samples correctly classified and N is the total number of samples.

2.2.5.2 Mean Absolute Error

The Mean Absolute Error (MAE) is a measure of difference between two continuous variables. In our case, these variables are the target variable y that represent the number of people in the classroom and its predicted value $\widehat{y}$. Formally, MAE can be defined as :

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |\widehat{y}^{(n)} - y^{(n)}|, \tag{2.25}$$

where N is total number of patterns.

2.2.5.3 Mean Relative Error

Mean Relative Error (MRE) is computed to evaluate the goodness of a model, and it is defined as the mean of the relative errors (RE) of each sample.

$$MRE = \frac{1}{N}\sum_{i=1}^{N}\frac{|\widehat{y}^{(n)} - y^{(n)}|}{y}, \tag{2.26}$$

Where N is the total number of samples.

Chapter 3

Data collection and datasets description

This chapter describes the data collection and feature extraction processes carried out. As a result, four different datasets were produced, which differ in the recording periods, the input sensors, and the generated features. These datasets will serve as input for pattern recognition techniques to be used to solve the occupancy detection and occupancy estimations problems.

3.1 Data collection

In classroom number 5 of the Polytechnical school of the Autonomous University of Madrid (UAM), an e-nose device (see Figure 3.1) composed of several sensors that monitorize different chemical substances in the air (Analog CO2, Digital CO2, air quality, and the records of the TGS 2620, TGS 2600, TGS 2611, TGS 2603 and TGS 2602), humidity, luminosity and the temperature sensors was installed in April 2016. In addition to a wifi sensor that allows us to estimate the number of electronic devices connected to the wifi network in the proximity of the classroom. Finally, other sensors that detect when the door of the classroom and the door of the computer case get opened or closed were also installed in the same classroom. In what follows, we will call e-nose to this multisensory sensing network. The information about all

the sensors used can be found in Table 3.1.

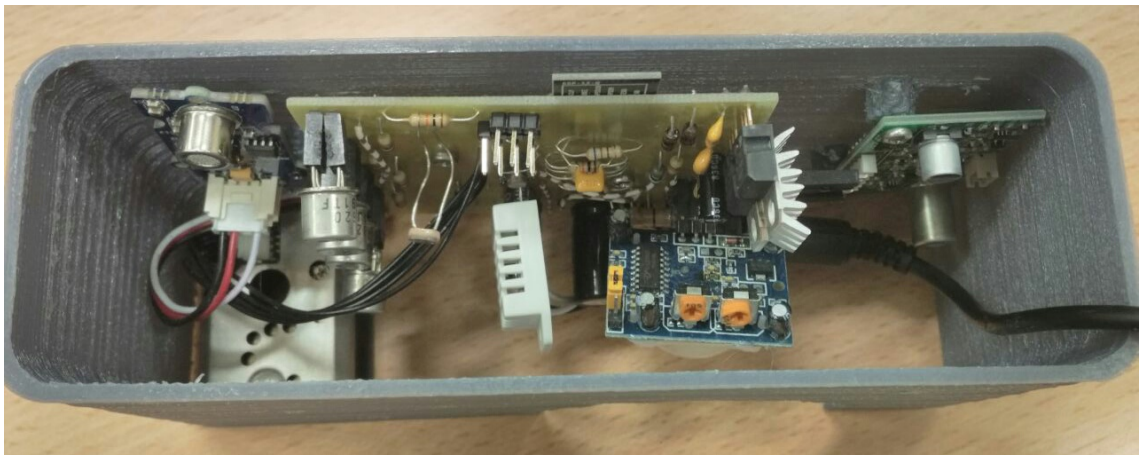

Figure 3.1: The e-nose installed in the classroom number 5.

Table 3.1: Sensors included in the main device installed in classroom number 5 (EPS-UAM)

Sensor	Description
Figaro CDM4161A	Analog CO2 sensor
DHT22	Humidity sensor
LDR	Luminosity sensor
TGS 2600	Detection of pollutants in the air (eg, hydrogen and carbon monoxide).
TGS 2620	Detection of vapors from organic solvents and other volatile vapors.
TGS 2011	Methane sensor
TGS 2602	Detection of Air Pollution (Air Purifiers - Ventilation Control)
TGS 2603	Detection of odors and air pollution (Air cleaners - Ventilation control)
Winsen MP503	Air-Quality. Detection Gas: Alcohol, Smoke, VOC ects air quality elements
Figaro CDM4160	Digital CO2
WiFi Module - ESP8266	low-cost Wi-Fi chip with full TCP/IP stack and MCU (microcontroller unit) capability

The e-nose sends the data recorded by the different sensors to a remote database every 10 seconds. To access the registered data, we dispose of a web tool that allows us to visualize and download the data (See Figure 3.2).

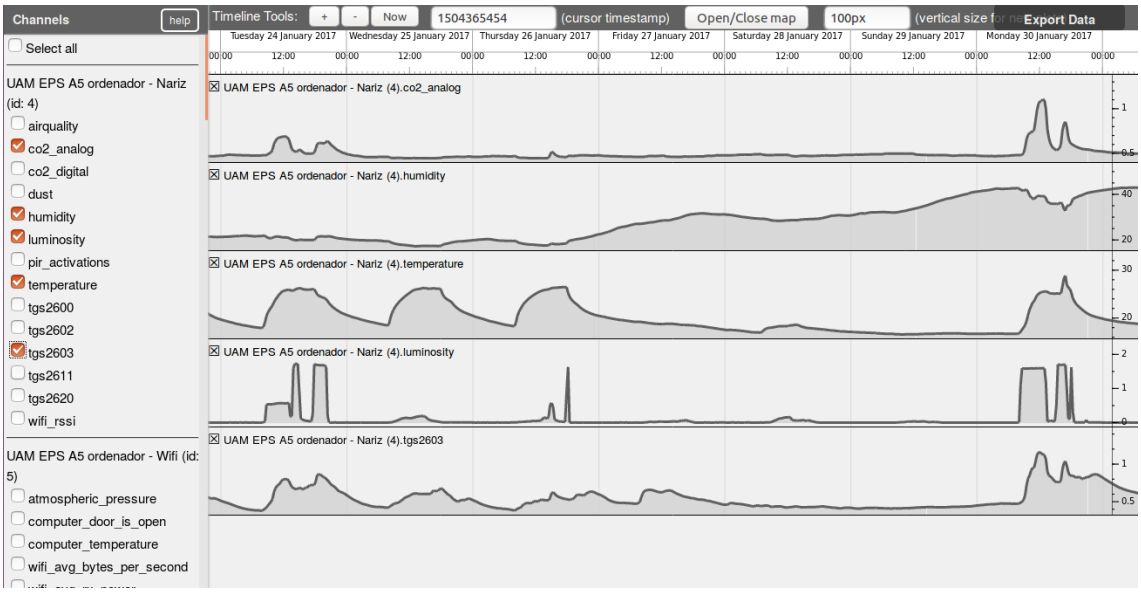

Figure 3.2: The web tool that allow us to visualize and download the e-nose records.

In addition to the sensors constituting the multisensory network, some information about about the activities taking place in the classroom was captured with the collaboration collaboration of some professors that provided information about the number of people in the classroom, the type of activity carried out such as keynote lecture or exam, or some modifications in the class schedule. These variables were introduced manually into the database and some of them (classroom attendance) will be used as target variable in the machine learning models.

3.2 Datasets and Feature extraction

This section describes the features extracted from sensors' responses in order to generate the datasets that will serve as input to the machine learning models. More precisely, in this work we have constructed two types of datasets as a function of the nature of their attributes: statistical variables summarizing sensors' response for a period of time and variables based on the Exponential Moving Average of the signals that try to reflect the sensor dynamics of the increasing/decaying transient portion

of the sensor response for a period of time.

3.2.1 Datasets with statistical variables

To construct the datasets using the statistical variables, we considered the following features: the median of the signal from which we subtract the its minimum value denoted median-min to subtract the baseline, the mean of the signal from which we subtract its minimum value to remove the baseline denoted mean-min. Removing the baseline signal at the beginning of the activity is especially crucial when we have two classes in a row, because when having two classes in a row, the initial condition of the sensor for the second class will not be the same as in the first one, so it is expected the values of the mean and the mean to be very different even in those cases in which the number of occupants is similar. The standard deviation (Std), the maximum (Max), the minimum (Min), and the difference between the maximum and the minimum value of the signal (Max-Min) are variables also included in the dataset. All these statistical features are computed during the period of time in which the class/activity takes place or during predefined periods of time in which the classroom is empty.

Table 3.2: Main characteristics of the datasets used in this work

Name	Start date	End date	Dimension	Patterns	Sensors
DS1	20/04/2016	30/06/2016	54	56	Temperature, Analog CO2, Humidity, Luminosity, TGS 2620, TGS 2600, TGS 2611, TGS 2602, Air quality
DS2	20/09/2016	31/01/2017	66	78	Temperature, Analog CO2, Humidity, Luminosity, TGS 2620, TGS 2611, Air quality, TGS 2602, Digital CO2, TGS 2603 and ESP8266(wifi)
DS3	20/04/2016	31/01/2017	48	135	Temperature, Analog CO2, Humidity, Luminosity, TGS 2620, TGS 2611, Air quality, TGS 2602

To obtain the corresponding patterns at times when the classroom is empty, we computed the same features for two periods of one hour during the weekends. We

chose the period of time between 13pm and 14pm on Saturdays and Sundays in order to ensure that the classroom is, in general, empty and to avoid any kind of bias regarding the luminosity. Under this experimental setup, we considered three different databases that differ in both the period of tie in which data were captured, and the sensors installed in the classroom during this period of time as some sensors changed while this work was under development. Table 3.2 shows the characteristics of the three datasets used in this work.

3.2.2 Datasets with Exponential Moving Average variables

An exponential moving average (EMA), also known as the exponentially weighted moving average is similar to a simple moving average, except that more weight is given to the most recent data points.

The Exponential Moving Average is computed using the equation (2.1). The EMA transformation generates a time serie with a single peak that corresponds to the increasing/decreasing transient portion of original signal, as can be seen in Figure 3.3, the exact location of the peak or the maximum value depends on the value of the smoothing parameter α.

Muezzinoglu et al. [Muezzinoglu et al., 2009] proposed the use of the EMA over chemical sensors' signals because their responses are slow when exposed to a constant concentration of a stimulus, and the same happens when the stimulus is removed. Given that EMA's features lead to successful results in other electronic nose data [Muezzinoglu et al., 2009, Vergara et al., 2012], we have applied the features proposed by Muezzinoglu et al. over our original signals to study whether EMA's variables provide additional information to the statistical features described in the previous section. EMAs variables were only computed over the chemical sensors (Analog CO2, Digital CO2, TGS 2620, TGS 2611, TGS 2602, TGS 2603 and Air quality) because the rest of the sensors like the wifi sensor are not expected to have this behaviour.

The features considered in this dataset are the maximum value of the signal

corresponding to a class period minus its minimum value denoted "Max-Min" and the maximum value minus the minimum value divided by the minimum value of the signal corresponding to a class period "$\frac{\text{Max-Min}}{\text{Min}}$" as steady-state features in addition to the EMA's features that are the maximum value of the increasing portion and the minimum value of the decaying portion corresponding to the three values of α ($\alpha = \{0.1, 0.01, 0.001\}$) as transient features (See Table 3.3).

For pattern corresponding to times when the classroom is empty, we computed the same features for two periods of one hour during the weekends as we did in the previous section (Section 3.2.1 for the three datasets with statistical variables).

Table 3.3: Feature extracted for the EMA's datasets.

Statistical features	EMA's features	
	Rising portion	Decaying portion
Max-Min	$\max_{\mathbf{k}} ema_{\alpha=0.001}(r[k])$	$\min_{\mathbf{k}} ema_{\alpha=0.001}(r[k])$
Max-Min/Min	$\max_{\mathbf{k}} ema_{\alpha=0.01}(r[k])$	$\min_{\mathbf{k}} ema_{\alpha=0.01}(r[k])$
	$\max_{\mathbf{k}} ema_{\alpha=0.1}(r[k])$	$\min_{\mathbf{k}} ema_{\alpha=0.1}(r[k])$

Figure 3.3 shows the EMA's signals of the Analog CO2 corresponding to an isolated class, where we can clearly see the values of the EMA's features -the maximum of the rising portion and the minimum of the decaying portion-.

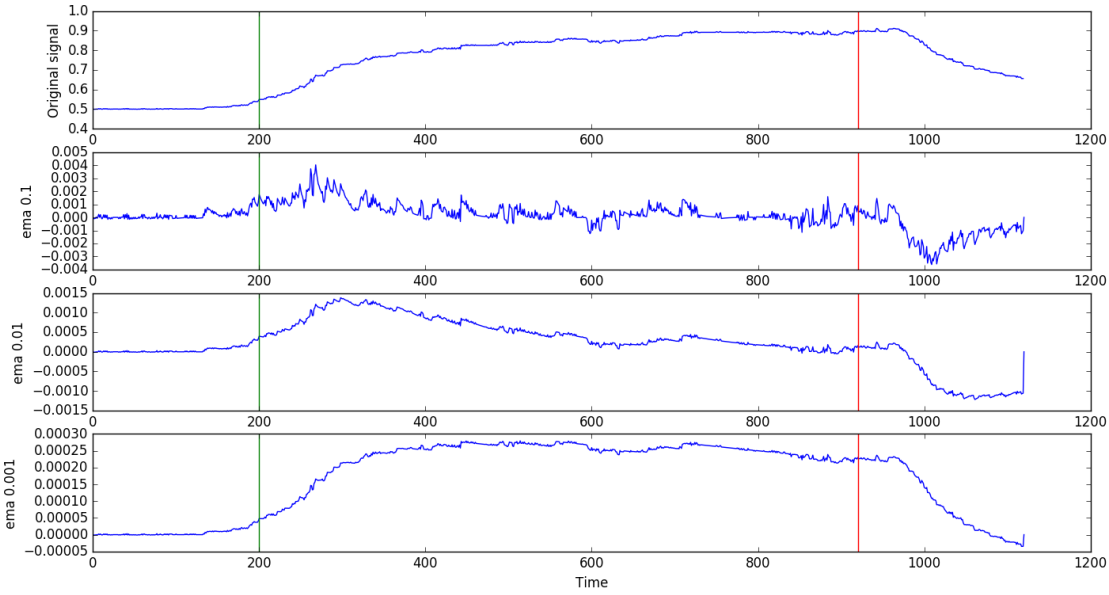

Figure 3.3: Real signal and its ema_α transform for $\alpha = 0.1, 0.01, 0.001$ corresponding to the Analog CO2 of an isolated class where there is no other class directly before or after. The vertical lines indicate the beginning (green) and the end (red) of the class period.

Chapter 4

Experiments and results

In this chapter we present the proposed architecture to solve the occupancy estimation problem. The proposed architecture consists in an hierarchical model, that firstly detect if the classroom is empty or not (occupancy detection problem) and finally estimate the number of occupant if the classroom is occupied (occupancy estimation problem). We also present and analyze the results of applying this model to the datasets presented in 3.2.

4.1 Experimental setup

To be able to evaluate our models, we start first by splitting the data into a training set and a test set, where the training set represents the 90% of our dataset, and the remaining 10% corresponds to the test set. Data are normalized to have zero mean and unit variance. As we want to predict the number of people in the classroom during a class and given that the classroom was empty the majority of the time, we decided to design a hierarchical model that in the first stage attempts to separate the samples corresponding to the empty classroom from the other scenarios by means of a classification algorithm, and in the second stage provides an estimation of the classroom occupancy using a regression model. Figure 4.1 shows the steps of the

experimental setup that we will explain in more detail in the following sections.

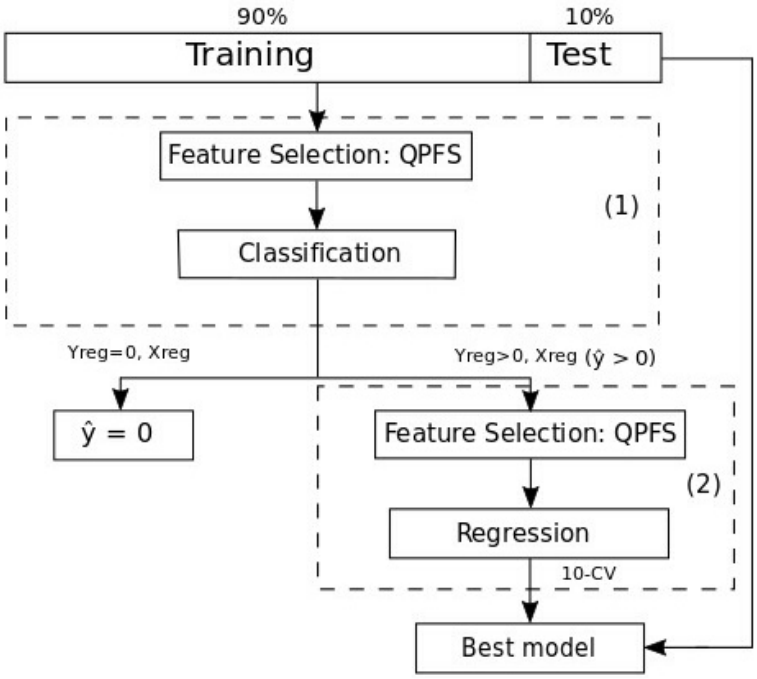

Figure 4.1: Proposed hierarchical model with the occupancy detection stage to detect if the classroom is empty or not followed by the occupancy estimation stage to estimate the number of occupant.

First stage: Occupancy detection

First of all, we perform a feature selection step using the QPFS algorithm [Rodriguez-Lujan et al., 2010] to identify the most relevant features (sensors) for the classification process, then we use the selected features to train the Logistic Regression algorithm (block 1 in Figure 4.1). In this stage we do not need to use any hyperparameter optimization method because the logistic regression model does not require to adjust any hyperparameter. Our model predict "0 occupants" for patterns that were classified as "empty classroom", and the remaining patterns will pass to the second stage.

Second stage: occupancy estimation

Once the classification is done, we consider a subset of the initial data made up of patterns in which the classifier predicts the presence of people in the classroom. Then, a feature selection algorithm over all the features and a regression model over this data subset is applied in order to estimate the number of people in the classroom (block 2 in Figure 4.1).

Considering the available features, we apply again the QPFS algorithm [Rodriguez-Lujan et al., 2010] for feature selection but this time the target variable is the number of people in the classroom. Once the relevant features for the regression problem are detected, the performance of five regression models to estimate the number of occupants in the classroom, namely: Linear Regression, Linear Support Vector Regression Machine, Random Forest, Ridge Regression and Lasso Regression is evaluated.

Table 4.1: Values of the Grid search for the regression models used in this work where λ is the penalty parameter for Lasso Regression and Ridge Regression (Section 2.2.3.2), number of trees refers to the number of decision trees in the Random forest (Section 2.2.3.4), ϵ is the margin tolerance, and C is the regularization parameter that establishes a trade-off between the margin size and the minimization of the loss function for the SVM (Section 2.2.3.3).

Regression model	Grid of hyperparameters
Linear Regression	-
Linear SVM	C = 2^a where a varies from -12 to 12 with a step of size 1. $\epsilon = 2^b$ where b varies from -8 to 0 with a step of size 1.
Random Forest	number of trees varies from 10 to 100 with a step of size 1.
Ridge Regression (Section 2.2.3.2)	$\lambda = 2^a$ where a varies from -12 to 12 with a step of size 1.
Lasso Regression	$\lambda = 2^a$ where a varies from -12 to 12 with a step of size 1.

An important task before training the above mentioned regression models except

for linear regression is to choose the best value of the hyperparameters that set up the model; this is done by performing a search over a grid that represents some discrete values for the parameter space using logarithmic scale. Table 4.1 shows the hyperparameter grid considered for each regression model.

The search over the grid is guided by a 10-fold cross validation procedure over the training set. That is, 10-CV is performed for every combination of the grid parameters and the error associated with such combination is obtained as the mean error in the ten folds. The combination of hyperparameters with the lowest Mean Average Error is used to train the final regressor over the whole training set. This procedure is repeated ten times over 10 different training/test partitions of the initial dataset, and the reported errors are obtained by averaging the errors obtained in the test sets of the ten permutations. Section 4.2 will show the results obtained after applying the above mentioned algorithms over our datasets.

4.2 Results

This section presents a description of the results obtained after applying the experiments described in Section 4.1 over our datasets both with statistical variables (Section 3.2.1) and EMA's variables (Section 3.2.2).

As already described in Section 4.1, before estimating the number of occupants in the classroom, we start by classifying our samples into two classes using the Logistic Regression model (Section 2.2.2.1), and then, estimate the number of occupants for those patterns in which the classifier predicts the presence of people in the classroom using five regression models: Linear Regression, Support Vector Regression Machine, Random Forest, Ridge Regression and Lasso Regression. The results obtained by these models are compared in order to determine the model with the best performance"

In order to estimate the number of people in the classroom, we apply five regression models over our complete datasets considering all the features and then removing

some features corresponding to some specific sensors: CO2, luminosity and wifi. The reasons why we decided to analyze the performance of the models without taking into account these sensors are:

- Variables related to CO2 and Wifi are the most relevant for the regression problem when all the available features were initially considered. By removing these features, we wanted to see to what extent the MOX sensors are capable of performing this task since, as far as we know, the is not any work in the literature focused on occupancy detection by means of MOX sensors.

- Luminosity data were used in other work to detect the presence of people in an office [Candanedo and Feldheim, 2016]. In this work, luminosity is the most relevant feature to determine whether there are people inside the room. However, it should be mentioned that almost all the patterns associated to an empty room in this work corresponds to hours of the day in which there is not natural light outside. Although for the occupancy detection problem we do not expect luminosity to be a particularly relevant variable, we decided to analyze the performance of the whole system (classification and regression steps) when luminosity information is not taken into account in order to dismiss a possible source of bias.

4.2.1 DS1 dataset

The DS1 dataset is made of data coming from the following sensors: Temperature, Analog CO2, Humidity, Luminosity, TGS 2620, TGS 2600, TGS 2011, TGS 2602 and Air quality(Winsen MP503) captured from 20/04/2016 to 30/06/2016 (See Table 3.2).

As shown in Figure 4.1, feature selection is carried out before both occupancy detection and occupancy estimation models. The motivation behind the application of a feature selection algorithm is to reduce the risk of over-fitting, but mainly, to improve the interpretability of the obtained models by determining the most relevant

sensors for each problem. In this regard, Table 4.2 shows the features selected by QPFS for the classification problem when all the features are initially considered. According to these results, analog CO2 is one of the most relevant features since it is selected 60% of the time. None of the features that comes from the luminosity sensor are relevant for the classification problem, so there is not a luminosity bias between empty and non-empty scenarios. Finally, it is remarkable that we obtain a 100% classification accuracy in all cases as shown in Table 4.3.

Table 4.2: Percentage of times that a feature is selected by QPFS for the classification problem over the 10 iterations of experimental procedure presented in Figure 4.1 when the DS1 dataset is used. Different subsets of initial features are considered as input for the feature selection algorithm: All: DS1 dataset with all the features; All-CO2: DS1 dataset without the CO2 data, All-Luminosity: DS1 dataset without the luminosity data and All-{CO2, Luminosity}: DS1 dataset without both the CO2 and the luminosity data.

Features selected for classification	Initial set of features			
	All	All-CO2	All-Luminosity	All-{CO2, Luminosity}
Max Analog CO2	60%	-	50%	-
Max TGS 2600	50%	30%	30%	50%
Max TGS 2620	100%	100%	100%	100%
Max Air quality	100%	100%	100%	100%
Max TGS 2602	0%	0%	0%	10%

Regarding the occupancy estimation problem, we can clearly see in Table 4.4 that the Analog CO2 sensor is the most relevant one as it is selected by the QPFS in all the permutations. It means that CO2 information is essential to get a good regression model. This fact is also clear when looking at the Mean Absolute Errors and the Mean Relative Errors of the regression models in Table 4.3 when CO2's features are not considered by the regression algorithms, their performances get worse.

Table 4.3: Results after applying the classification and the regression models to the DS1 dataset using different evaluation metrics. MAE test: Mean Absolute Error over all the test set, MAE test >0: Mean Absolute Error over samples predicted as positive by the classification model, MRE test >0: Mean Relative Error over samples predicted as positive by the classification model. The column Input features indicates the subset of variables considered as the input of the feature selection algorithm. The best result in terms of the lowest MAE is shown in bold.

Input features	Model	MAE test >0	MAE test	MRE test>0	classification accuracy
ALL	Linear regression	9.38±1.27	9.38±1.27	0.29±0.06	100%
	Linear SVM	10.34±1.04	10.34±1.04	0.29±0.05	
	Random forest	9.61±0.80	9.61±0.80	0.29±0.04	
	Lasso	**8.34±1.07**	**8.34±1.07**	**0.23±0.04**	
	Ridge regression	9.37±1.24	9.37±1.14	0.28±0.06	
ALL-CO2	Linear regression	10.00±1.44	10.00±1.44	0.30±0.07	100%
	Linear SVM	10.01±1.05	10.01±1.05	0.3±0.05	
	Random forest	11.54±1.18	11.54±1.18	0.32±0.04	
	Lasso	10.80±1.29	10.80±1.29	0.31±0.05	
	Ridge regression	9.93±1.43	9.93±1.43	0.30±0.01	
ALL-Luminosity	Linear regression	8.98±1.14	8.98±1.14	0.27±0.05	100%
	Linear SVM	10.38±1.02	10.38±1.02	0.28±0.05	
	Random forest	10.3±0.82	10.30±0.82	0.3±0.03	
	Lasso	8.40±1.37	8.40±1.37	0.23±0.04	
	Ridge regression	9.00±1.13	9.00±1.13	0.27±0.05	
ALL-{CO2 , Luminosity}	Linear regression	9.64±1.39	9.64±1.39	0.29±0.01	100%
	Linear SVM	10.38±1.20	10.38±1.20	0.31±0.06	
	Random forest	11.68±1.15	11.68±1.15	0.33±0.05	
	Lasso	10.95±0.92	10.95±0.92	0.29±0.03	
	Ridge regression	9.64±1.39	9.64±1.39	0.29±0.01	

Results in Table 4.3 and Figure 4.2 show that the lowest MAE is obtained by Lasso considering all the features. This model gets a MAE of 8.34 ± 1.07 and a MRE

of 0.23 ± 0.04. In addition, the models without the Analog CO2 are clearly the worst, which confirms that CO2 features are the most relevant ones when predicting the number of people in the classroom. It makes sense as the level of CO2 rises as a function of the number of people [Jiang et al., 2016].

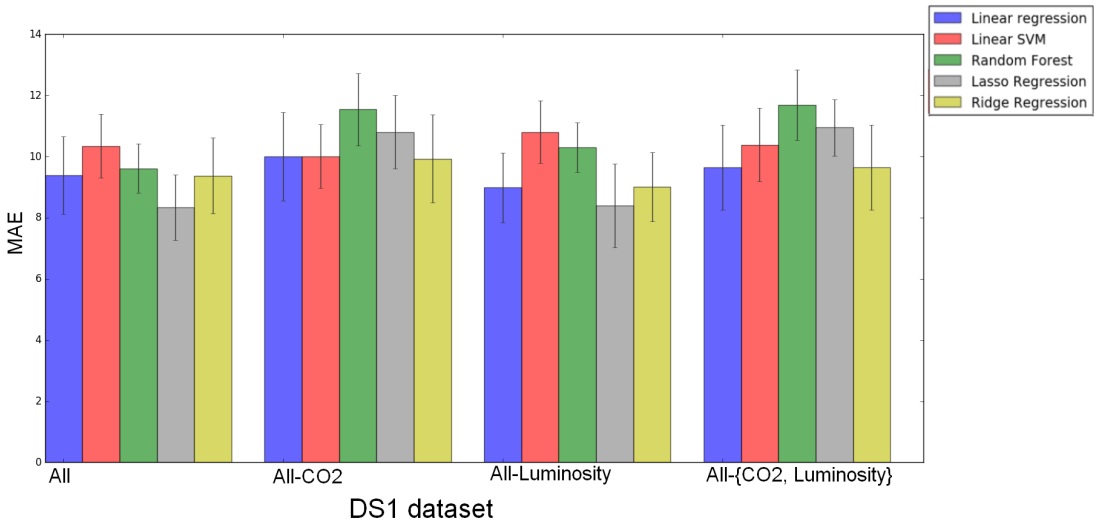

Figure 4.2: Bar plot comparing the Mean Absolute Error of the five regression models used in this work for the DS1 dataset when considering different subsets of sensors. The x-axis represents the input features, and the y-axis represents the Mean Absolute Error obtained by each of the models for each feature subset in the occupancy detection problem.

As Lasso is the model with the best performance, and it has its own feature selection process, Figure 4.3 shows the importance of each sensor according to the Lasso's coefficients. This importance is calculated by summing the absolute values of the Lasso's coefficients over the 10 permutations of the experimental setup (Figure 4.1). This sum is divided by 10 to normalize. After that we sum the coefficients of the 6 variables that correspond to the same sensor in a single variable and we divide it by 6 to normalize it again.

Figure 4.3 reveals that the analog CO2 sensor is the most important one for the

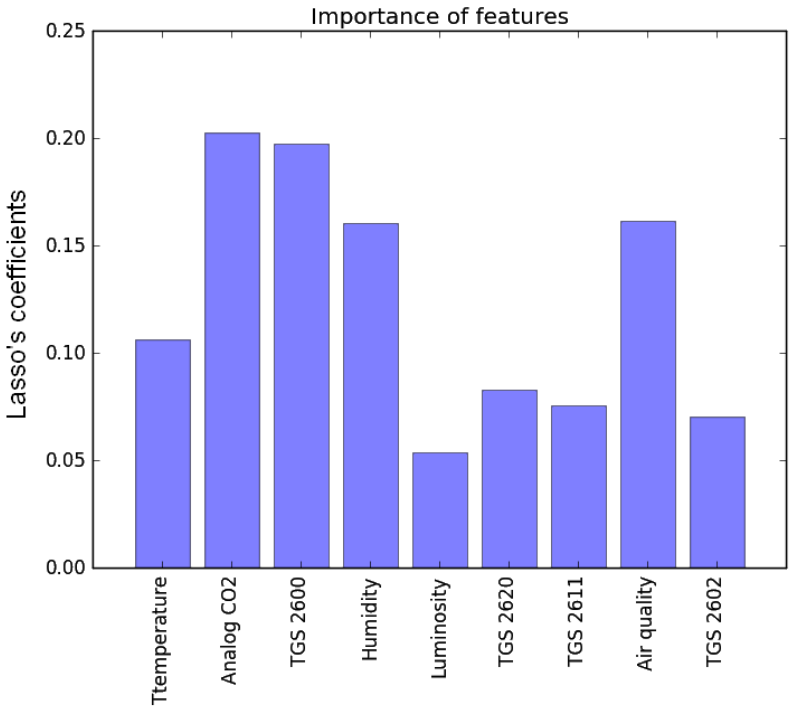

Figure 4.3: Importance of the features according to the according to Lasso's coefficients for the 10 permutations of the experimental setup (Figure 4.1).

regression task together with the TGS 2600 sensor. These results are consistent with those obtained when using a QPFS for the regression problem (Table 4.4). However, there are some differences between both approaches: while air quality features are very important for Lasso, they are not so relevant for QPFS. This may be due to the differences on how both algorithms deal with features' collinearity.

Finally, Figure 4.4 is a comparison between the actual number of occupants in the classroom and the prediction of the Lasso regression model with all features as it is the model with the lowest Mean Absolute Error. In Figure 4.4 we can see that training and test points are close to the bisector, so the model provides a good estimation of the number of people in the classroom.

Table 4.4: Percentage of times that a feature is selected by QPFS for the regression problem over the 10 iterations of experimental procedure presented in Figure 4.1 when the DS1 dataset is used. Different subsets of initial features are considered as input for the feature selection algorithm: *All*: DS1 dataset with all the features; *All-CO2*: DS1 dataset without the CO2 data, *All-Luminosity*: DS1 dataset without the luminosity data and *All-{CO2, Luminosity}*: DS1 dataset without both the CO2 and the luminosity data.

Features selected for regression	Initial set of features			
	All	All-CO2	All-Luminosity	All-{CO2, Luminosity}
Min Temperature	90%	90%	90%	90%
Median-Min Analog CO2	30%	-	30%	-
Max Analog CO2	100%	-	100%	-
Std TGS 2600	60%	70%	80%	100%
Max TGS 2600	100%	100%	100%	100%
Std Luminosity	20%	70%	-	-
Max-Min Luminosity	10%	0%	-	-
Max-Min TGS 2600	0%	0%	10%	0%
Median-Min TGS 2620	10%	0%	10%	0%
Mean-Min TGS 2620	0%	10%	0%	10%
Std TGS 2611	0%	20%	0%	30%
Max Air quality	0%	10%	0%	20%
Max TGS 2602	10%	70%	10%	70%

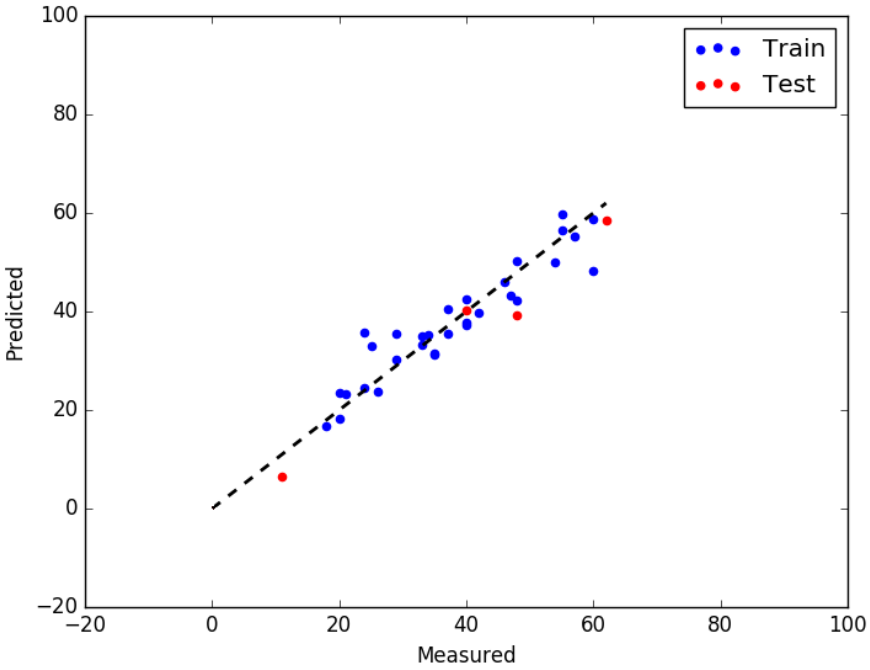

Figure 4.4: Estimation of the number of people in the classroom corresponding to the DS1 dataset without the luminosity's data using Lasso Regression. The x-axis represents the actual number of people in the classroom, the y-axis represents the predicted values, blue points represent training patterns, and red points are associated with est patters. The bisector represents the perfect prediction.

4.2.2 DS2 dataset

In this dataset, we include data coming from three new sensors. The recording period (from 9 September 2016 to 31 January 2017) is different comparing to DS1 dataset while the used sensors are: Temperature, Analog CO2, Humidity, Luminosity, TGS 2620, TGS 2611, Air quality, TGS 2602, Digital CO2, TGS 2603 and wifi sensor (See Table 3.2). Since one of these sensors allows us to estimate the number of devices connected to the wifi network of the classroom, we expect improvements in the performances of the regression models.

Table 4.5: Percentage of times that a feature is selected by QPFS for the classification problem over the 10 iterations of experimental procedure presented in Figure 4.1 when the DS2 dataset is used. Different subsets of initial features are considered as input for the feature selection algorithm: *All*: DS2 dataset with all the features; *All-CO2*: DS2 dataset without the CO2 data; *All-Luminosity*: DS2 dataset without the luminosity data; *All-Wifi* : DS2 dataset without the wifi data; *All-{CO2, Luminosity}*: DS2 dataset without both the CO2 and the luminosity data, *All-{Wifi, Luminosity}*: DS2 dataset without both the wifi and the luminosity data and *All-{CO2, Wifi}*: DS2 dataset without both the CO2 and the wifi data.

Features selected for classification	Initial set of features						
	All	All-CO2	All-Luminosity	All-Wifi	All-{CO2, Luminosity}	All-{Wifi, Luminosity}	All-{Wifi, CO2}
Max Temperature	0%	0%	0%	0%	0%	10%	0%
Max Analog CO2	0%	-	0%	20%	-	30%	-
Max Luminosity	100%	100%	-	100%	-	-	100%
Max TGS 2620	0%	0%	0%	30%	0%	80%	20%
Max Air quality	100%	100%	100%	100%	100%	100%	100%
Min Air quality	0%	0%	0%	0%	0%	10%	0%
Max TGS 2602	10%	10%	90%	10%	90%	100%	10%
Median-Min TGS 2603	0%	0%	0%	0%	0%	30%	0%
Std TGS 2603	0%	0%	0%	0%	0%	10%	0%
Max TGS 2603	0%	0%	0%	0%	0%	20%	0%
Max-Min TGS 2603	0%	0%	0%	0%	0%	30%	10%
Max Wifi	100%	100%	100%	-	100%	-	-

In terms of the occupancy detection task, Table 4.5 shows the percentage of times that each feature is selected by QPFS algorithm. According to the results presented in this table, wifi and luminosity are always chosen by the feature selection algorithm when available, which reveal their importance and matches with our initial expectation with regards to the wifi.

Table 4.6 is similar to Table 4.3 and it shows the classification accuracy of the Logistic Regression model for the occupancy detection problem along with the MAE

and MRE of the five regression models considered in this work. The classification results in Table 4.6 are slightly worse than those presented in Table 4.3 with an accuracy of 97.5% since the DS1 and DS2 do not actually have the same dimension nor the same number of patterns (see Table 3.2) as they do not have the same sensors and the data were not recorded in the same period of time, in addition to the presence of the wifi chip that estimates the number of devices connected to the wifi network of the classroom, this chipset can also detect the devices connected to the wifi network in the corridor near the classroom which means that we can have a positive values although the classroom is empty. The wifi's features are really relevant being that the Max wifi is selected by the QPFS for the classification problem in the 10 permutation as shown in Table 4.5.

Considering the complete dataset with all the features, the results improve considerably when predicting number of people: the Mean Absolute Error and the Mean Relative Error significantly drop as it can be seen in 4.6. For example, the MAE of the test set of the Random Forest model decreases from 9.61±2.53 to 5.39±1.97 and the MRE drops from 0.29±0.04 to 0.13±0.03. As for the most relevant sensors, we find that analog CO2 and wifi's features are selected in all the permutations, and TGS 2603 sensor is selected in 80% of times as shown in Table 4.7.

Although these results are very promising, we repeat the same procedure as for the DS1 dataset in which we remove features from the most relevant sensors in order to check the usefulness of the chemical sensors not used in previous works. We start by removing the data coming from both analog and digital CO2 sensors, then we remove the data coming from the luminosity sensor, and later we do not take into account the wifi sensor's data. Once the analysis of the relevance of the individual sensors are completed, we remove the already cited sensors in pairs: CO2 and luminosity, luminosity and wifi, and finally CO2 and wifi. Table 4.6 shows the results obtained on these experiments. On the one hand, the model with the lowest MAE (5.09±0.69) and MRE (0.15±0.1) is the Random Forest without the luminosity

Table 4.6: Results after applying the classification and the regression models to the DS2 dataset using different evaluation metrics. MAE test: Mean Absolute Error over all the test set, MAE test >0: Mean Absolute Error over samples predicted as positive by the classification model, MRE test >0: Mean Relative Error over samples predicted as positive by the classification model. The column Input features indicates the subset of variables considered as the input of the feature selection algorithm. The best result in terms of the lowest MAE is shown in bold.

Input features	Model	MAE test >0	MAE test	MRE test>0	classification accuracy
ALL	Linear regression	7.04±0.95	6.74±0.96	0.15±0.02	97.5%
	Linear SVM	8.07±0.88	7.90±0.84	0.19±0.04	
	Random forest	5.39±0.62	5.14±0.38	0.13±0.03	
	Lasso	9.67±1.28	9.51±1.28	0.24±0.05	
	Ridge regression	7.05±0.95	6.75±0.95	0.15±0.02	
ALL-CO2	Linear regression	8.96±0.61	8.52±0.62	0.20±0.03	97.5%
	Linear SVM	9.57±0.60	9.46±0.60	0.21±0.03	
	Random forest	8.41±0.85	8.28±0.86	0.20±0.03	
	Lasso	11.08±0.72	11.03±0.73	0.26±0.04	
	Ridge regression	8.69±0.61	8.53±0.62	0.20±0.03	
ALL-Luminosity	Linear regression	6.83±0.95	6.55±0.96	0.15±0.082	97.5%
	Linear SVM	8.23±0.92	8.05±0.88	0.19±0.04	
	Random forest	**5.09±0.69**	**5.29±0.70**	**0.15±0.03**	
	Lasso	8.25±0.94	7.93±0.93	0.19±0.03	
	Ridge regression	6.84±0.95	6.55±0.95	0.15±0.03	
ALL-Wifi	Linear regression	9.78±0.73	10.02±0.75	0.23±0.05	97.5%
	Linear SVM	11.19±0.84	11.37±0.90	0.27±0.05	
	Random forest	7.06±0.73	7.14±0.70	0.18±0.04	
	Lasso	9.90±1.05	9.57±1.06	0.26±0.02	
	Ridge regression	9.74±0.73	9.98±0.76	0.23±0.05	
ALL-{CO2, Luminosity}	Linear regression	8.67±0.63	8.52±0.64	0.19±0.03	97.5%
	Linear SVM	9.67±0.57	9.48±0.56	0.22±0.03	
	Random forest	8.89±1.05	9.10±1.04	0.20±0.04	
	Lasso	10.02±0.58	10.10±0.86	0.22±0.04	
	Ridge regression	8.69±0.63	8.53±0.63	0.20±0.03	
ALL-{Wifi, Light}	Linear regression	9.90±0.62	10.07±0.59	0.23±0.04	97.5%
	Linear SVM	10.36±0.97	10.46±0.98	0.26±0.05	
	Random forest	6.40±0.64	6.50±0.61	0.16±0.04	
	Lasso	9.51±0.80	9.17±0.88	0.21±0.02	
	Ridge regression	9.87±0.62	10.04±0.59	0.23±0.04	
ALL-{Wifi, CO2	Linear regression	11.53±1.27	11.93±1.27	0.28±0.06	97.5%
	Linear SVM	11.64±1.38	11.97±1.46	0.29±0.07	
	Random forest	11.07±1.22	11.44±1.27	0.28±0.05	
	Lasso	13.45±2.43	13.7±2.38	0.33±0.65	
	Ridge regression	11.56±1.25	11.98±1.26	0.29±0.06	

data. The comparison between the actual number of people in the classroom and the number of occupants predicted by this model is depicted in Figure 4.5.The most relevant sensors for this model are Analog CO2 and wifi that are selected in 100% of times and TGS2603 (amines) that is selected in 9 out of 10 permutations (see Table 4.7). On the other hand, the model with the lowest MRE (0.13±0.09) is the Random Forest trained with all tall the available features. The MAE of this model (5.39±1.93) is very close to the MAE obtained by the best model in terms of the MAE. We can also see in Table 4.3 that when we remove CO2 or wifi's features, the performance of the models drops significantly. This worsening is even more pronounced when both features are eliminated at the same time. It is also remarkable that the feature representing the variability of the amines sensor (TGS2603) is highly important in all cases. As far as we know, this type of sensor has been never applied to the occupancy detection and estimation problems.

As a graphical summary of Table 4.6, Figure 4.6 shows bar plots comparing the Mean Absolute Errors of the five regression models used for the DS2 Dataset when considering different subsets of features. Figure 4.6 makes clear that the worst model is the one without both CO2 and wifi's data and the best model is the one that does not take into account luminosity data. These results are in line with those obtained in Section 5.2.1

From Table 4.6 and Figure 4.6 we can see that Random Forest models always give the lowest error MRE regardless off the input set of features. It is because Random Forests can handle problems with a large number of variables with a relatively small number of observations (see Table 3.2) [Fernández-Delgado et al., 2014]. We can also deduce from Figure 4.6 that wifi's features are very useful to estimate the number of people in the classroom as the wifi sensor allows us to estimate the number of devices connected to the wifi network of the classroom. The importance of CO2 and Wifi data is also evident from Table 4.7, which shows that the CO2 and the wifi are always selected by the QPFS for the regression problems.

Table 4.7: Percentage of times that a feature is selected by QPFS for the regression problem over the 10 iterations of experimental procedure presented in Figure 4.1 when the DS2 Dataset is used. Different subsets of initial features are considered as input for the feature selection algorithm: All: DS2 dataset with all the features; All-CO2: DS2 dataset without the CO2 data; All-Luminosity: DS2 dataset without the luminosity data; All-Wifi : DS2 dataset without the wifi data; All-{CO2, Luminosity}: DS2 dataset without both the CO2 and the luminosity data, All-{Wifi, Luminosity}: DS2 dataset without both the wifi and the luminosity data, and All-{CO2, Wifi}: DS2 dataset without both the CO2 and the wifi data.

Features selected for regression	Initial set of features						
	All	All-CO2	All-Luminosity	All-Wifi	All-{CO2, Luminosity}	All-{Wifi, Luminosity}	All-{Wifi, CO2}
Median-Min Temperature	0%	0%	0%	0%	0%	10%	0%
Max Temperature	0%	0%	0%	0%	0%	0%	20%
Median-Min Analog CO2	0%	-	10%	90%	-	90%	-
Mean-Min Analog CO2	40%	-	20%	0%	-	0%	-
Max Analog CO2	100%	-	100%	100%	-	100%	-
Std TGS 2620	0%	0%	0%		70%	0%	50%
Max TGS 2620	0%	60%	0%	0%	0%	0%	80%
Std TGS 2611	0%	0%	0%	0%	0%	0%	20%
Max TGS 2611	0%	30%	0%	0%	0%	0%	0%
Std TGS 2602	0%	0%	0%	0%	10%	0%	0%
Max TGS 2602	0%	0%	0%	10%	0%	0%	10%
Median-Min TGS 2603	0%	0%	0%	20%	0%	20%	0%
Std TGS 2603	90%	100%	90%	100%	100%	100%	100%
Max TGS 2603	0%	0%	0%	80%	0%	30%	100%
Max-Min TGS 2603	0%	0%	0%	0%	30%	0%	10%
Max Wifi	100%	100%	100%	-	100%	-	-

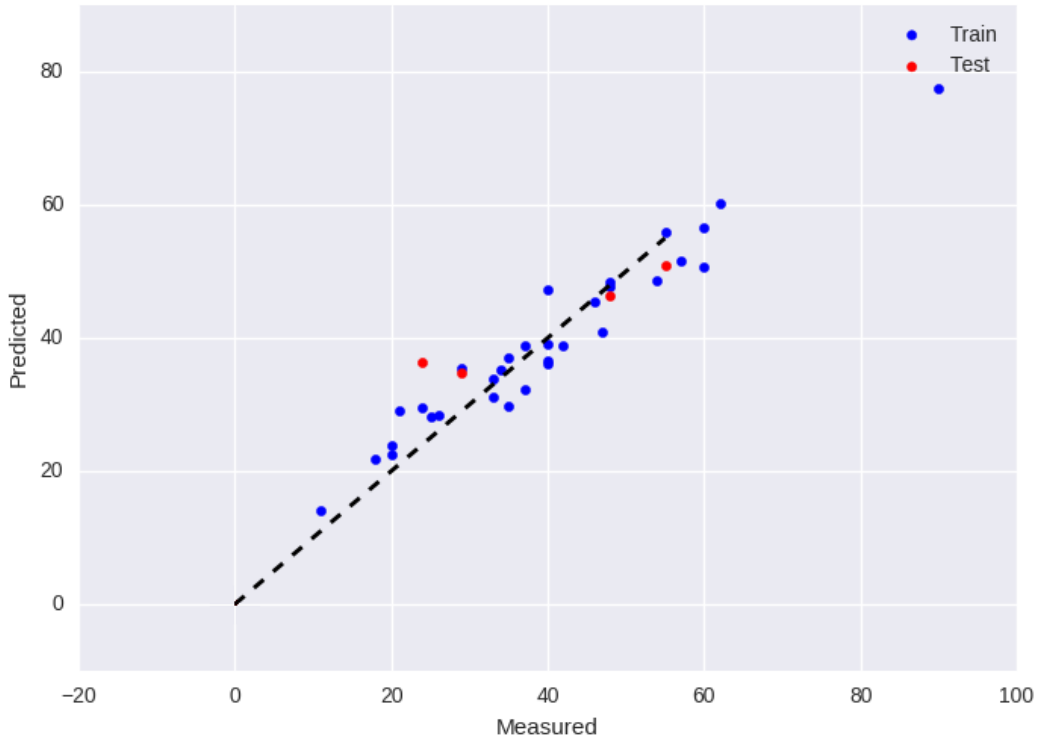

Figure 4.5: Estimation of the number of people in the classroom corresponding to the DS2 Dataset without the luminosity data using Random Forest. The x-axis represents the actual value of occupants, the y-axis represents the predicted values, blue points represent training patterns, and red points are associated with est patters. The bisector represents the perfect prediction.

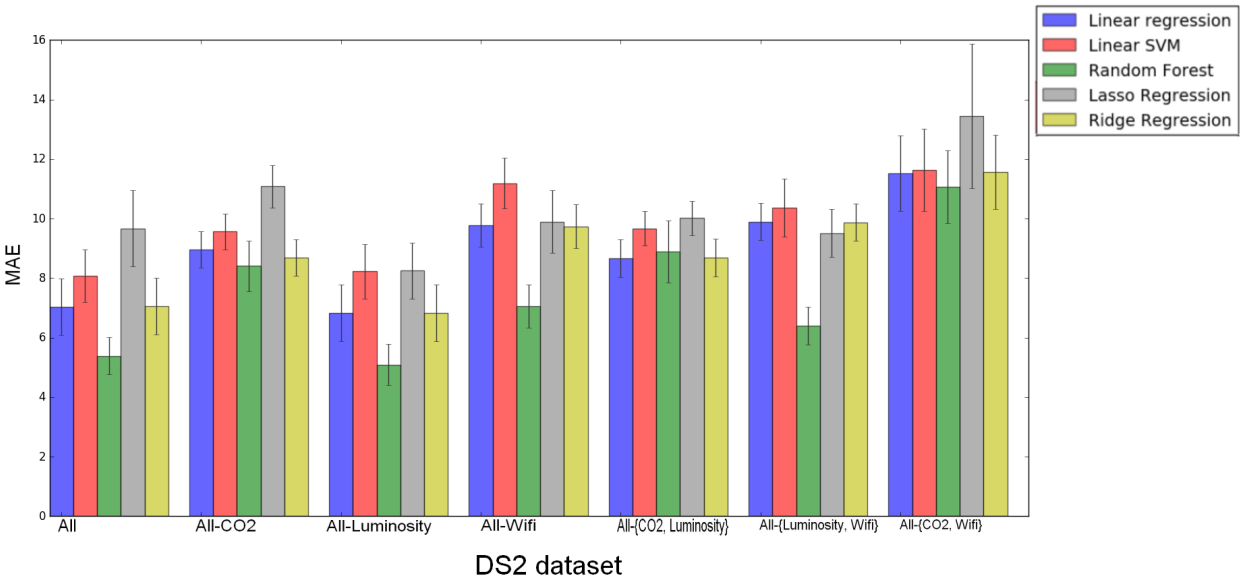

Figure 4.6: Bar plot comparing the MAE of the five regression models used in this work for the DS2 Dataset when considering different subsets of sensors.The x-axis represents the input features, and the y-axis represents the MAE obtained by each of the models for each feature set in the occupancy detection problem.

4.2.3 DS3 dataset

The recording period of the DS3 dataset is the combination of the recording periods of DS1 and DS2 dataset, and the sensors used are those available in both datasets, namely: Temperature, Analog CO2, Humidity, Luminosity, TGS 2620, TGS 2611, Air quality, TGS 2602 and TGS 2603).

As shown in Table 4.8, the most relevant sensors according to QPFS for the classification problem are TGS 2620 (Alcohol, Solvents Vapors) and Air quality - their variables are selected in all the permutations - followed by the Analog CO2 sensor that is selected 50% of the times. As shown in Table 4.9, the Logistic Regression model achieves a classification accuracy of 99.29% when considering all the available features.

Table 4.8: Percentage of times that a feature is selected by QPFS for the classification problem over the 10 iterations of experimental procedure presented in Figure 4.1 when the DS3 Dataset is used. Different subsets of initial features are considered as input for the feature selection algorithm: All: DS3 Dataset with all the features; All-CO2: DS3 Dataset without the CO2 data, All-Luminosity: DS3 Dataset without the luminosity data and All-{CO2, Luminosity}: DS3 Dataset without both the CO2 and the luminosity data.

Features selected for classification	Initial set of features			
	All	All-CO2	All-Luminosity	All-{CO2, Luminosity}
Max Analog CO2	50%	-	50%	-
Max TGS 2620	100%	100%	100%	100%
Max Air quality	100%	100%	100%	100%
Max TGS 2602	0%	30%	20%	10%

In terms of prediction the number of occupants, taking into account all the available features, Random Forest is the model with the lowest Mean Absolute Error (9.49±1.22), and a Mean Relative Error of 21%, while Lasso is the model with the

lowest MRE (20%) and a MAE of 9.57±0.58 as shown in Table 4.9. As shown in Table 4.10, the most relevant sensor for these models is the CO2 sensor as its features are selected in all the 10 permutations. The temperature sensor is also informative as its features are chosen in 80% of the times as shown in Table 4.10.

Table 4.9: Results after applying the classification and the regression models to the DS3 dataset using different evaluation metrics. MAE test: MAE over all the test set, MAE test >0: MAE over samples predicted as positive by the classification model, MRE test >0: MRE over samples predicted as positive by the classification model. The column Input features indicates the subset of variables considered as the input of the feature selection algorithm. The best result in terms of the lowest MAE is shown in bold.

Input features	Model	MAE test >0	MAE test	MRE test>0	classification accuracy
ALL	Linear regression	10.99±0.94	11.10±0.94	0.26±0.02	99.29%
	Linear SVM	11.88±0.82	12.06±0.83	0.28±0.02	
	Random forest	**9.49±1.22**	**9.53±1.22**	**0.21±0.02**	
	Lasso	9.57±0.58	9.59±0.58	0.20±0.01	
	Ridge regression	11.00±0.94	11.11±0.94	0.26±0.02	
ALL-CO2	Linear regression	13.19±0.99	13.34±1.02	0.32±0.03	99.29%
	Linear SVM	13.3±0.86	13.54±0.92	0.33±0.04	
	Random forest	12.81±1.12	13.03±1.22	0.32±0.04	
	Lasso	13.32±0.73	13.43±0.81	0.33±0.03	
	Ridge regression	13.18±0.99	13.33±1.02	0.32±0.03	
ALL-Luminosity	Linear regression	10.63±0.94	10.74±0.73	0.25±0.02	99.29%
	Linear SVM	11.58±0.75	11.72±0.75	0.28±0.02	
	Random forest	9.82±1.06	9.87±1.07	0.22±0.02	
	Lasso	13.67±0.78	13.79±0.82	0.33±0.03	
	Ridge regression	10.63±0.94	10.74±0.94	0.25±0.02	
ALL-{CO2, Luminosity}	Linear regression	13.56±1.01	13.77±1.07	0.32±0.03	99.29%
	Linear SVM	13.79±0.93	14.05±0.99	0.34±0.03	
	Random forest	12.52±1.01	12.75±1.12	0.32±0.09	
	Lasso	13.52±0.80	13.64±0.84	0.32±0.03	
	Ridge regression	13.65±0.98	13.86±1.04	0.33±0.03	

As done for the DS1 and DS2 datasets, we repeated the same experiment after removing CO2 and luminosity sensors to check the usefulness of the chemical sensors. We started by removing data coming from CO2 sensor, then we eliminated the luminosity data, and finally, we discarded both luminosity and CO2 data.

Table 4.10: Percentage of times that a feature is selected by QPFS for the regression problem over the 10 iterations of experimental procedure presented in Figure 4.1 when the DS3 Dataset is used. Different subsets of initial features are considered as input for the feature selection algorithm: All: DS3 Dataset with all the features; All-CO2: DS3 Dataset without the CO2 data, All-Luminosity: DS3 Dataset without the luminosity data and All-{CO2, Luminosity}: DS3 Dataset without both the CO2 and the luminosity data.

Features selected	Initial set of features			
for regression	All	All-CO2	All-Luminosity	All-{CO2, Luminosity}
Max Temperature	80%	100%	90%	100%
Min Temperature	0%	10%	0%	20%
Median-Min Analog CO2	100%	-	100%	-
Max Analog CO2	100%	-	100%	-
Median-Min Luminosity	0%	20%	-	-
Std Luminosity	40%	60%	-	-
Max Luminosity	10%	100%	-	-
Median-Min TGS 2611	20%	50%	20%	60%
Mean-Min TGS 2611	0%	10%	0%	10%
Std TGS 2611	0%	100%	10%	100%
Max TGS 2611	30%	40%	40%	100%
Max Air quality	0%	0%	0%	60%
Max TGS 2602	10%	20%	10%	20%

The most relevant sensors for the classification problem did not change since

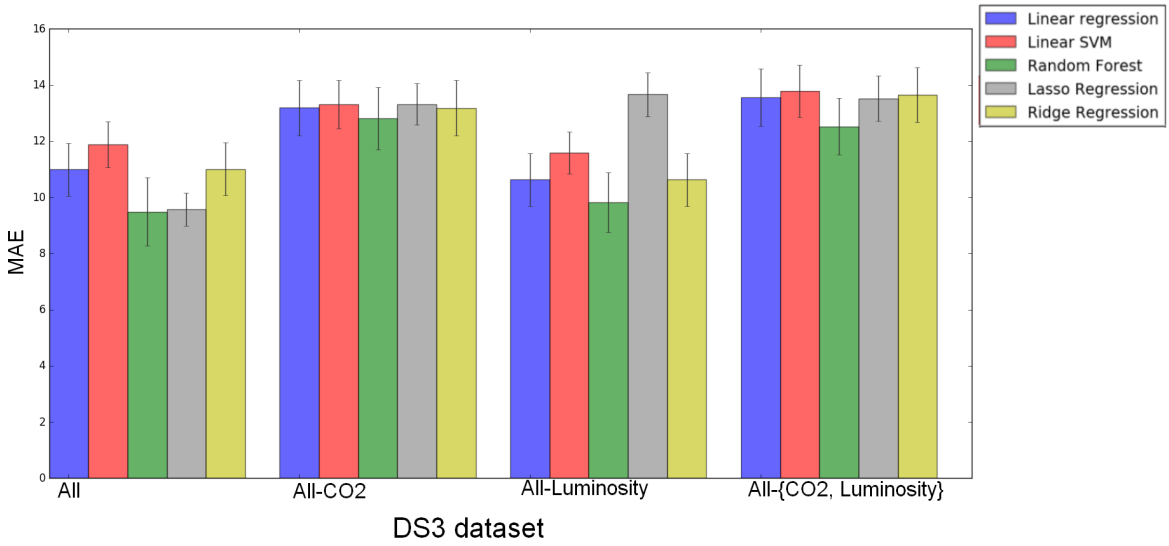

Figure 4.7: Bar plot comparing the Mean Absolute Errors of the five regression models used in this work for the DS3 Dataset when considering different subsets of sensors. The x-axis represents the input features, and the y-axis represents the Mean Absolute Error obtained by each of the models for each feature subset in the occupancy detection problem.

we did not remove any of them (see Table 4.8). Regarding the regression problem, Table 4.10 shows that the most relevant sensors when removing the CO2 data are temperature, luminosity and TGS 2611 (methane) that are selected all the times. Again, these results reveal the usefulness of MOX sensors for the people estimation problem. Table 4.9 and Figure 4.7 show that the models with the lowest MAE and MRE are still those considering all the available features.

Figure 4.8 shows a comparison between the actual number of occupants and the prediction of the Random Forest model trained with all the available features as it is the model with the lowest MAE and MRE. Red points represent the test patterns and they are close to the bisector, which indicates accurate predictions.

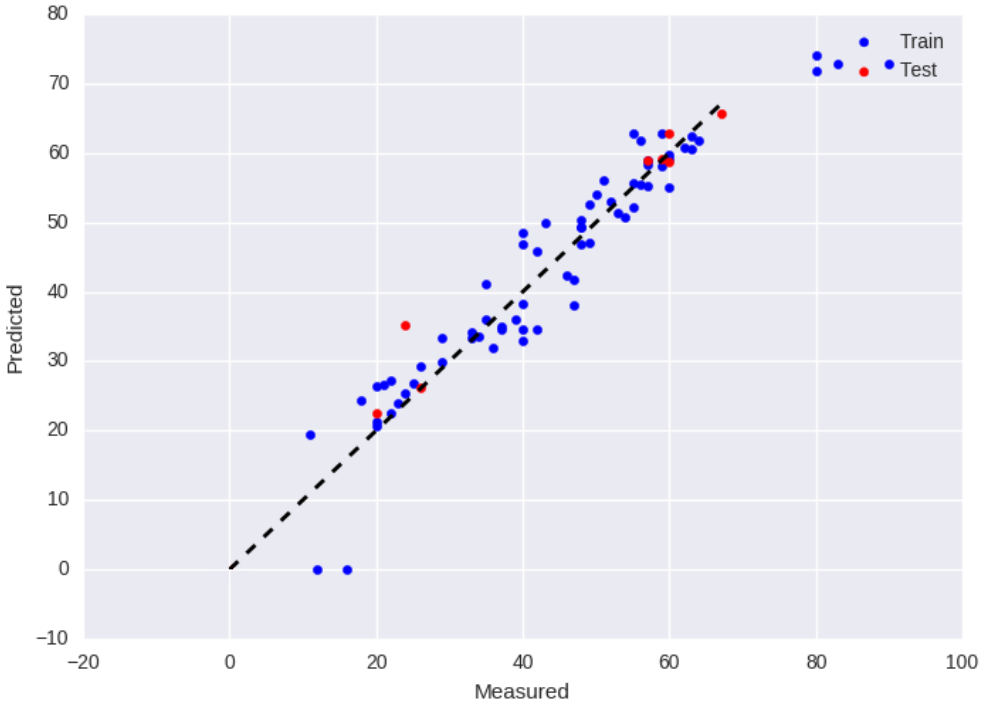

Figure 4.8: Estimation of the number of people in the classroom corresponding to the DS3 dataset using Random Forest. The x-axis represents the actual value of occupants, the y-axis represents the predicted values, blue points represent training patterns, and red points are associated with est patters. The bisector represents the perfect prediction.

4.2.4 EMA's Dataset

Looking upon the EMA's dataset, the recording period is the same as DS2 dataset (from 9 September 2016 to 31 January 2017) but, we only have data coming from the following chemical sensors: analog CO2, Digital CO2, TGS 2620, TGS 2611, TGS 2602, TGS 2603 and Air quality (see Section 3.2).

Table 4.11: Results after applying the classification and the regression models to the EMA'S dataset using different evaluation metrics. MAE test: Mean Absolute Error over all the test set, MAE test >0: Mean Absolute Error over samples predicted as positive by the classification model, MRE test >0: Mean Relative Error over samples predicted as positive by the classification model.

Input features	Model	MAE test >0	MAE test	MRE test>0	classification accuracy
ALL	Linear regression	17.15±1.65	16.22±1.65	0.49±0.05	93.75%
	Linear SVM	16.77±1.79	15.52±1.89	0.54±0.07	
	Random forest	14.80±1.64	12.55±1.80	0.43±0.05	
	Lasso	16.82±2.28	15.84±2.94	0.45±0.05	
	Ridge regression	17.09±1.64	16.14±1.64	0.49±0.05	

Regarding this dataset, we obtain a low classification accuracy of 93.75% and high Mean Absolute Errors and Mean Relative Errors when predicting the number of people in the classroom compared to the results obtained for the three datasets (DS1, DS2, and DS3) with statistical variables. More precisely, the Mean Absolute Error of the regression models varies from 14.80 to 17.50 as shown in Table 4.11. The poor performance of the models when using EMA's features may be due to sensors do not return to their baseline signal when there are two or more consecutive classes/activities, which has an effect on the rising and decaying portions of the EMA's signal, and it produces meaningless variables. To illustrate this scenario, Figure 4.9 shows the original and EMA's signals associated with the Analog CO2

sensor response in a class that takes place before and after other classes. It can be seen that the beginning and the end of the class are not properly identified by the EMA's signals. In short, EMA's features are appropriated to model isolated events (see Figure 3.3 in Chapter 3) in which the rising and decaying portions can be easily identified, but they are not suitable for semi-controlled or uncontrolled environments in which a series of events may occur without pause.

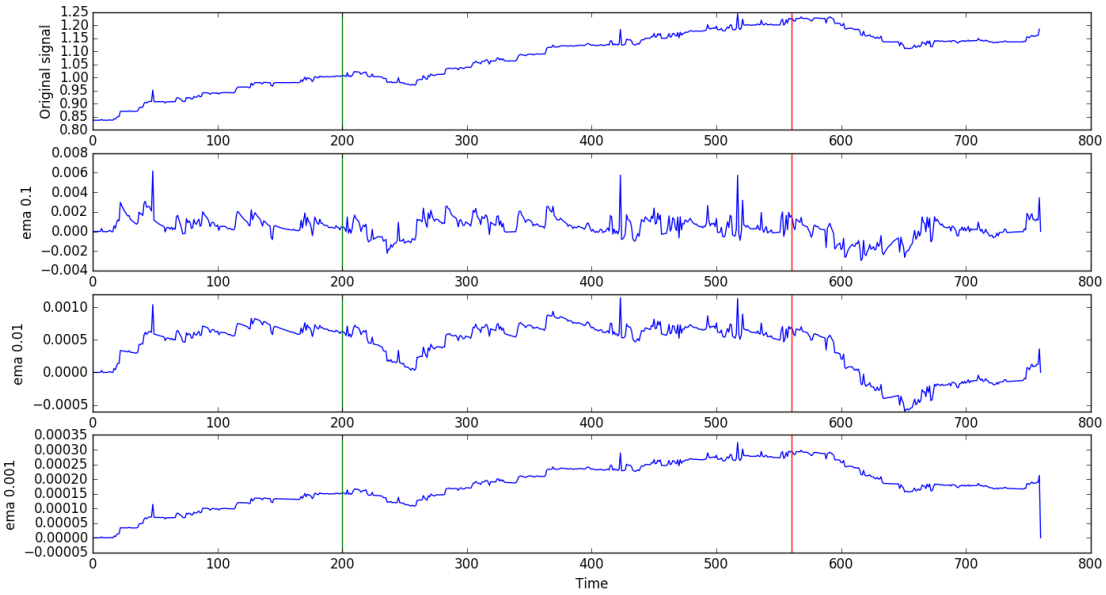

Figure 4.9: Original signal and its ema_α transformation for $\alpha = 0.1, 0.01, 0.001$ corresponding to the Analog CO2 sensor's response during a class that takes place before and after other classes. The vertical lines indicate the beginning (green)and the end (red) of the class period.

Chapter 5

Discussion and Further Work

The main objective of this Master's Thesis was to study the use of machine learning techniques along with electrical noses (e-noses) that mimic the human olfaction. In this work, we wanted to estimate the number of people in a classroom in the Polytechnical school of the Autonomous University of Madrid (UAM) using an e-nose. Our e-nose device was composed by several sensors that monitorize different chemical substances in the air, luminosity and temperature, in addition to a device that measures the number of electronic devices connected to the wifi network in the proximity of the classroom.

In order to address the occupancy estimation problem, firstly, we started by collecting information from the e-nose during a large period of time to obtain enough data to be able to train a machine learning model. The recording periods were April 2016 to June 2016, and from September 2016 to January 2017. Secondly, we constructed two different datasets based on different feature extraction methods. More precisely, we built three datasets formed by statistical variables obtained from sensors' responses, and as a proof of concept, we made up a dataset whose attributes were based on the Exponential Moving Average of chemical sensors' response. The datasets with statistical variables differentiate between themselves in the sensors used and in the recording period. All the datasets include patterns that correspond to

times when the classroom was empty and patterns that correspond to times when there were people in the classroom.

We have proposed a two step procedure to estimate the occupancy of the classroom. The first step is formulated as a supervised classification problem to detect the presence of people in the classroom. In this stage, we applied a feature selection algorithm (QPFS) followed by a Logistic Regression model. The second step aims at estimating the number of people in the classroom, and it is defined as a feature selection algorithm (QPFS) followed by a regression model. In this step, we tried five different regression models: Linear Regression, Support Vector Regression, Random Forest, Lasso and Ridge Regression.

Both the classification and the regression results differ from a dataset to another. In terms of the occupancy detection problem (classification), the best result corresponds to a classification accuracy of 100% using information coming from different sensors.Among all the information used by the classification model, the most relevant sensors are the Air quality and TGS2620 sensors, which are selected in all the permutations. The incorporation of this kind of sensors is one of the main contributions of the data used in this Mater's Thesis. On the other hand, the worst classification result was obtained when we included data coming from a wifi sensor that allows us to estimate the number of devices connected to the wifi network in the classroom. A possible explanation to the poor performance of the model when including this information is that the wifi sensor may be also detecting devices connected to the wifi network in the corridor near to the classroom.

Regarding the occupancy estimation problem (regression), the best result was obtained when using a Random Forest algorithm with the DS2 dataset (September 2016 - January 2017), which includes the wifi sensor information. In this case, the Mean Absolute Error of the model was 5.09±0.69 and the Mean Relative Error was 0.12±0.03. Regarding the most relevant sensors, Analog CO2, wifi, and TGS2600 sensors are always selected when present, and the TGS 2603 sensor (amines) is selected 90% or 100% of times depending on the input features. The TGS 2611 sensor

(methane) is also considered as relevant as it is selected between 30% and 100% of times depending on the input features. Its importance increases when other relevant sensors are removed. The usefulness of the CO2 sensor for estimating occupancy was already known in the literature [Rodrigues et al., 2017, Jiang et al., 2016] , while the importance of the wifi sensor was expected as it allows us to estimate the number of devices connected to the wifi network in the classroom.

With regard to the EMA's dataset, in which we only considered gas sensors, we got significantly worse results in both the classification and regression problems than those obtained by the datasets with statistical variables. This is due to to insufficient isolated classes, so our sensors are not able to recover their baseline status, and the rising and decaying portions of the signal are not easily identifiable. Overall, the classification accuracy of the Logistic Regression model for all the datasets was very high, being all the classification accuracies rates above 93.75%

In conclusion, in this Master's thesis we have shown that the application of machine learning techniques to data coming from multimodal sensoring data allows solving the occupancy detection and occupancy estimation problems achieving competitive performances in comparison with other works in the literature. These problems are especially interesting in industrial applications such as human activity monitoring or the management of energy-efficient buildings, among others [Verriele, 2016, Pan and Yang, 2009]. The principal novelty of this work relies on the use of data coming from a multimodal sensory network that includes some metal oxide (MOX) sensors and a wifi sensor that had never been used before to solve these type of problems. In fact, our experimental results show that these new sensors provide relevant information for the occupancy detection and occupancy estimation problems.

5.1 Further work

One of the problems that researchers find when working on multisensory networks is the lack of comprehensive, labeled, and reliable data sources. Our work represents a significant advance in this direction as we have built a dataset that includes several months of multimodal sensors' data. Nevertheless, we consider that the availability of more data will help to improve the machine learning models, so the first step of our further work is to collect more data. Another advantage of increasing the amount of data is the possibility of training more powerful state-of-the-art models such as Recurrent Neural Networks (RNN), which are specially designed for working with time series [Walid and Alamsyah, 2017]. Other line of future work is based on the study of other feature extraction methods for e-nose data such as the sliding window methodology proposed by Monroy et al. [Monroy et al., 2016]. We believe that finding the appropriate representation of the data will significantly improve the performance of the regression models.

Another interesting line of research is to extend our work to other problems like the identification of the type of activity that takes place in the room - for example, master class versus exams) - , and real-time occupancy estimation.

Bibliography

[Baby et al., 2000] Baby, R., Cabezas, M., and de Reca, E. W. (2000). Electronic nose: a useful tool for monitoring environmental contamination. *Sensors and Actuators B: Chemical*, 69(3):214 – 218. Proceedings of the International Symposium on Electronic Noses.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[Breiman et al., 1984] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

[Bryll et al., 2003] Bryll, R., Gutierrez-Osuna, R., and Quek, F. (2003). Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition*, 36(6):1291 – 1302.

[Burges, 1998] Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

[Candanedo and Feldheim, 2016] Candanedo, L. M. and Feldheim, V. (2016). Accurate occupancy detection of an office room from light, temperature, humidity and {CO2} measurements using statistical learning models. *Energy and Buildings*, 112:28 – 39.

[Chilo et al., 2016] Chilo, J., Pelegri-Sebastia, J., Cupane, M., and Sogorb, T. (2016). E-nose application to food industry production. *IEEE Instrumentation Measurement Magazine*, 19(1):27–33.

[D'Amico et al., 2010] D'Amico, A., Pennazza, G., Santonico, M., Martinelli, E., Roscioni, C., Galluccio, G., Paolesse, R., and Natale, C. D. (2010). An investigation on electronic nose diagnosis of lung cancer. *Lung Cancer*, 68(2):170 – 176.

[Efron, 1979] Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *Ann. Statist.*, 7(1):1–26.

[Ekwevugbe et al., 2013] Ekwevugbe, T., Brown, N., Pakka, V., and Fan, D. (2013). Real-time building occupancy sensing using neural-network based sensor network. In *2013 7th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, pages 114–119.

[Fernández-Delgado et al., 2014] Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15:3133–3181.

[Fonollosa et al., 2014] Fonollosa, J., Rodríguez-Luján, I., Trincavelli, M., Vergara, A., and Huerta, R. (2014). Chemical discrimination in turbulent gas mixtures with mox sensors validated by gas chromatography-mass spectrometry. *Sensors*, 14(10):19336–19353.

[Gardner et al., 1988] Gardner, J., Bartlett, P., Dodd, G., and Shurmer, H. (1988). Pattern recognition in the warwick electronic nose. In *Proc. 8th Intl Congress of European Chemoreception Research Organisation*, University of Warwick, UK.

[Gardner et al., 1990] Gardner, J. W., Bartlett, P. N., Dodd, G. H., and Shurmer, H. V. (1990). *The Design of an Artificial Olfactory System*, pages 131–173. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Gardner, 1991] Gardner, J. W.; Bartlett, P. N. Sensors and sensory systems for an electronic nose. In *Proceedings of the NATO Advanced Research Workshop*, pages 5–8.

[Hailemariam et al., 2011] Hailemariam, E., Goldstein, R., Attar, R., and Khan, A. (2011). Real-time occupancy detection using decision trees with multiple sensor types. In *Proceedings of the 2011 Symposium on Simulation for Architecture and Urban Design*, SimAUD '11, pages 141–148, San Diego, CA, USA. Society for Computer Simulation International.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition.

[Herbrich et al., 2000] Herbrich, R., Graepel, T., and Obermayer, K. (2000). Large margin rank boundaries for ordinal regression. In Bartlett, P. J., Schölkopf, B., Schuurmans, D., and Smola, A. J., editors, *Advances in Large Margin Classifiers*, pages 115–132. MIT Press.

[Ho, 1998] Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844.

[Hoerl and Kennard, 1970] Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 42(1):80–86.

[Hogewind and Zwaardemaker, 1920] Hogewind, F. and Zwaardemaker, H. (1920). On spray-electricity and waterfall-electricity. In *Proc. 22th KNAW*, pages 429–437, Amsterdam.

[Jiang et al., 2016] Jiang, C., Masood, M. K., Soh, Y. C., and Li, H. (2016). Indoor occupancy estimation from carbon dioxide concentration. *CoRR*, abs/1607.05962.

[Kleiminger et al., 2013] Kleiminger, W., Beckel, C., Staake, T., and Santini, S. (2013). Occupancy detection from electricity consumption data. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, BuildSys'13, pages 10:1–10:8, New York, NY, USA. ACM.

[Monroy et al., 2016] Monroy, J. G., Palomo, E. J., Lpez-Rubio, E., and Gonzalez-Jimenez, J. (2016). Continuous chemical classification in uncontrolled environments with sliding windows. *Chemometrics and Intelligent Laboratory Systems*, 158:117 – 129.

[Muezzinoglu et al., 2009] Muezzinoglu, M. K., Vergara, A., Huerta, R., Rulkov, N., Rabinovich, M. I., Selverston, A., and Abarbanel, H. D. (2009). Acceleration of chemo-sensory information processing using transient features. *Sensors and Actuators B: Chemical*, 137(2):507 – 512.

[Mustafa et al., 2014] Mustafa, S., Riaz, K., and Perveen, Q. (2014). Stability of linear regression models. *Science International (Lahore)*, 313:73–76.

[Pan and Yang, 2009] Pan, L. and Yang, S. X. (2009). An electronic nose network system for online monitoring of livestock farm odors. *IEEE/ASME Transactions on Mechatronics*, 14(3):371–376.

[Pathange et al., 2006] Pathange, L. P., Mallikarjunan, P., Marini, R. P., OKeefe, S., and Vaughan, D. (2006). Non-destructive evaluation of apple maturity using an electronic nose system. *Journal of Food Engineering*, 77(4):1018 – 1023.

[Rao, 2013] Rao, R. P. N. (2013). *Brain-Computer Interfacing: An Introduction*. Cambridge University Press, New York, NY, USA.

[Raschka, 2017] Raschka, S. ((accessed June 3, 2017)). Machine Learning FAQ. `https://sebastianraschka.com/faq/docs/evaluate-a-model.html`.

[Rodrigues et al., 2017] Rodrigues, E., Pereira, L. D., Gaspar, A. R., Gomes, Á., and da Silva, M. C. G. (2017). Estimation of classrooms occupancy using a multi-layer perceptron. *CoRR*, abs/1702.02125.

[Rodriguez-Lujan et al., 2010] Rodriguez-Lujan, I., Huerta, R., Elkan, C., and Cruz, C. S. (2010). Quadratic programming feature selection. *J. Mach. Learn. Res.*, pages 1491–1516.

[Smola and Schölkopf, 2004] Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, 14(3).

[Vergara et al., 2012] Vergara, A., Vembu, S., Ayhan, T., Ryan, M. A., Homer, M. L., and Huerta, R. (2012). Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 166:320 – 329.

[Verriele, 2016] Verriele, M., S. C. H. B. L. N. G. S. G. V. L. N. (2016). The mermaid study: indoor and outdoor average pollutant concentrations in 10 low-energy school buildings in france.

[Walid and Alamsyah, 2017] Walid and Alamsyah (2017). Recurrent neural network for forecasting time series with long memory pattern. *Journal of Physics: Conference Series*, 824(1):012038.

[Wilkens and Hartman, 1964] Wilkens, W. F. and Hartman, J. D. (1964). An electronic analog for the olfactory processesa. *Journal of Food Science*, 29(3):372–378.

[Yang et al., 2012] Yang, Z., Li, N., Becerik-Gerber, B., and Orosz, M. (2012). A multi-sensor based occupancy estimation model for supporting demand driven hvac operations. In *Proceedings of the 2012 Symposium on Simulation for Architecture and Urban Design*, pages 2:1–2:8, San Diego, CA, USA. Society for Computer Simulation International.

[Yu, 2005] Yu, H. (2005). Svm selective sampling for ranking with application to data retrieval. In *Proceedings of the Eleventh ACM SIGKDD International Conference*

on Knowledge Discovery in Data Mining, KDD '05, pages 354–363, New York, NY, USA. ACM.

[Yu and Kim, 2012] Yu, H. and Kim, S. (2012). SVM tutorial - classification, regression and ranking. In *Handbook of Natural Computing*, pages 479–506. Springer.

[Zhao and Yongxin, 2012] Zhao, Y. and Yongxin, Y. (2012). Electronic nose integrated with chemometrics for rapid identification of foodborne pathogen. In Varmuza, K., editor, *Chemometrics in Practical Applications*, chapter 9. intech. `https://www.intechopen.com/books/howtoreference/chemometrics-in-practical-applications`.