Occupancy forecasting in HVAC systems

Written by,

Clément CRUVEILLER
Anas DEHMANI
Vasish ARVIND
Ozgur POLAT

Project supervisor : Manar AMAYRI

Abstract:

Electricity plays an important role in countries' residential and non-residential building.Using various forms of measures such as CO2 concentration or power usage, a general method is proposed to estimate the number of occupants in a particular location. The recommended technique is inspired by machine learning. It begins by deciding those that are the most useful among various metrics by measuring the gains in information. Then, there is a suggested estimation algorithm.Here,different algorithms have been implemented to forecast the future occupancy; among the common ones are Deep Neural Networks, CNN, LSTM, Regression-based approaches, and ARIMA. In this article, various algorithms are explored so that a thorough analysis can be carried out between them.A collection of real-world data, namely hourly load consumption as well as the daily energy usage of an office setting in $Ense^3$ building, which is used to test and compare the different algorithm models presented.

Introduction:

Occupancy behaviour is a significant activity that influences the use of building energy as occupants constantly produce heat because of their metabolism and multiple activities. Almost 40 percent of global energy and 20 percent of global greenhouse gas emissions are accounted for by both residential and non residential buildings. In order to reduce the energy consumption, using artificial intelligence and smart equipment, it is possible to control the heating system remotely. However on a daily basis it is required to tune the HVAC system manually. Recent research shows that based on determining the number of occupants, it is possible to save HVAC energy in the building. There are some research works which focus mainly on different behavior patterns through sensor measurements, which can be used to build models for occupants.There are some privacy issues involved in this technique and it is expensive. And it is not easy to determine the number of occupants using these features. So in this article,different approaches to developing a model for occupant comfort were based on temperature, power consumption,motion detection, humidity and CO2 data. Using different measurements and combining them makes the estimator more reliable because focusing on one measurement is not reliable enough to predict the occupants.The key aim of this project is to explore the issue of energy reduction using an HVAC system-based occupant detection system.

Background study:

There is a vast and increasing body of work on occupancy estimation schedules of various rooms inside a building.In previous research works, several sensor measurements have been used. However, for multiple uses, such as behaviour pattern or context analysis,and knowledge about the presence or absence of persons is not adequate in an office room, and an assessment of the number of individuals occupying the space is necessary.This issue is investigated by (Lam et al., 2009) in open offices, predicting Occupancy and human behaviors, using a range of data, and performance comparison where made between Artificial neural networks and SVMs. None of these approaches, though, produce human-understandable rules that can be useful to building administrators.An interesting method attempts to consider the connections between the accumulation of carbon dioxide, air quality in the room and the number of inhabitants.The physical CO2 paradigm based on

sensor networks has been commonly used to maximize occupant satisfaction and minimize the use of building energy. The model has been analyzed in this project to figure out the importance of using it in occupancy estimation. Most of the work in relation to it relies on the use of techniques for machine learning and time series analysis to identify human activity and estimate the number of inhabitants in a multi-story building. The methodology used here is similar to data driven modeling technique where the accuracy of different algorithm methods using the given data sets with various sensor data.

Testbed setup:

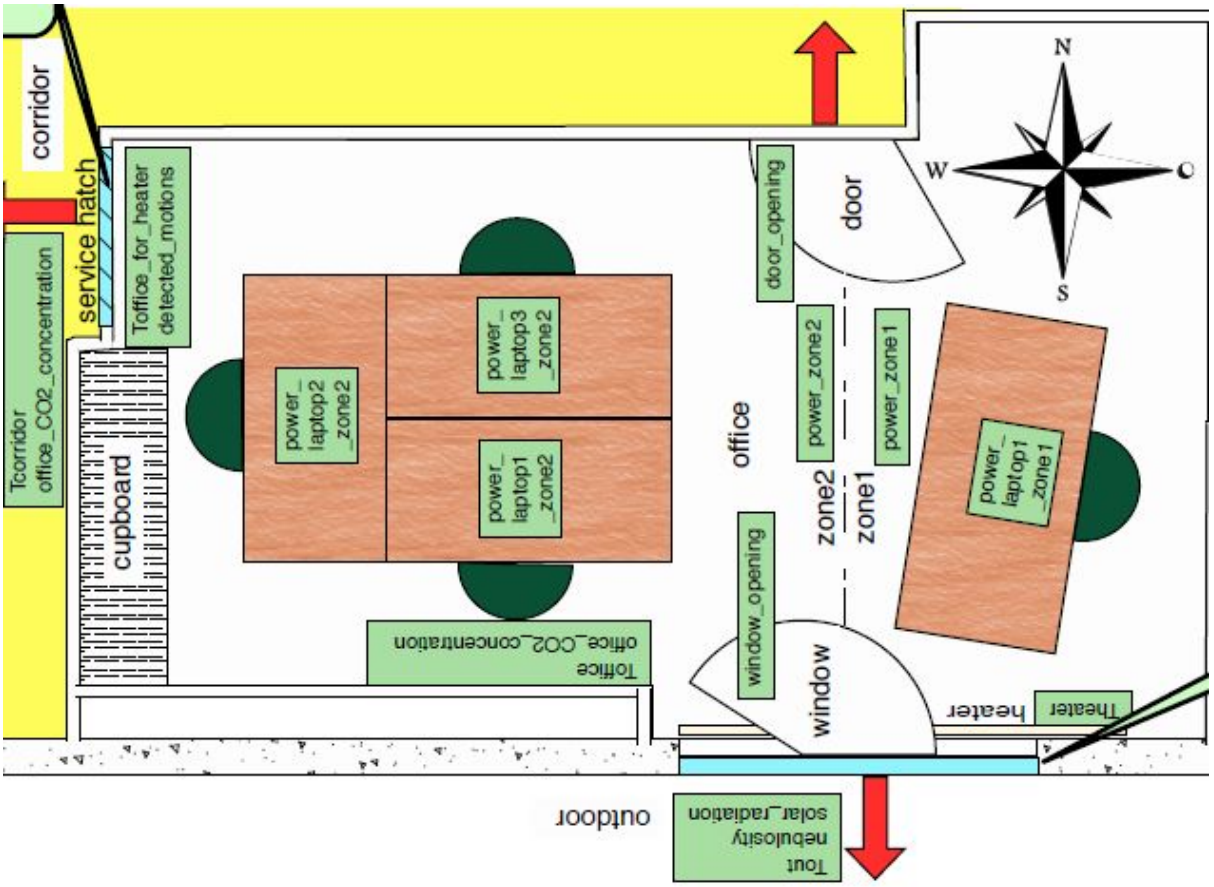


Figure 1: Smart office in Ense3 building

The test bed setup is at the Grenoble Institute of Technology.It is an office that accommodates a professor and three PhD students. The office has regular visits during the week, with a lot of meetings and presentations. The setup includes different sensing networks that monitor indoor and outdoor temperature, humidity, motion detection,CO2 concentration and acoustic pressure.It also contains a web application with a consolidated database for continuinal data retrieval from various sources.In machine learning,all the retrieved information that can be used to predict occupancy is referred to as features.

Occupancy Estimation: Different Methods Comparison

The office data:

The use of machine learning methods needs a lot of datas to analyze in order to optimize and enhance the results. Thus, we work on the office datas which contains all the records from the sensors for the year 2015.
However, for our study, we only have the label which stands for the occupancy record on the office, the Indoor Temperature room, the humidity, the motion detection, the power, the CO2 concentration and the door opening ratio for 2 weeks in May 2015 to fit our model.
Thus,we estimated the occupancy profile using datas for this period.

The first step is linked to the understanding of sensor datas correlation and the influence of each feature for the building of different models.

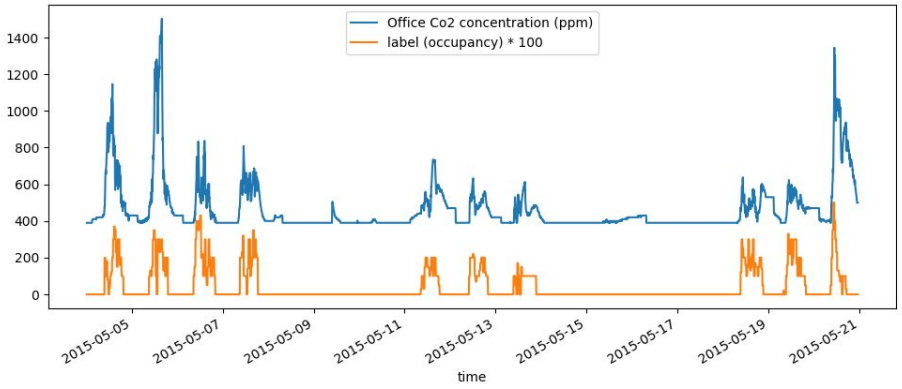


Figure 2: Plot of the office concentration (ppm) and occupancy (label)

Here for example, we plot the office CO2 concentration (ppm) and the label (occupancy) with a factor of 100 to adapt the view of both features. We can see that when the CO2 concentration varies, the label is evolving too. Thus, it helps to understand that this feature might be useful in our models. To get a first look at which features are important for our

estimation model, we can simply check the data set matrix correlation:

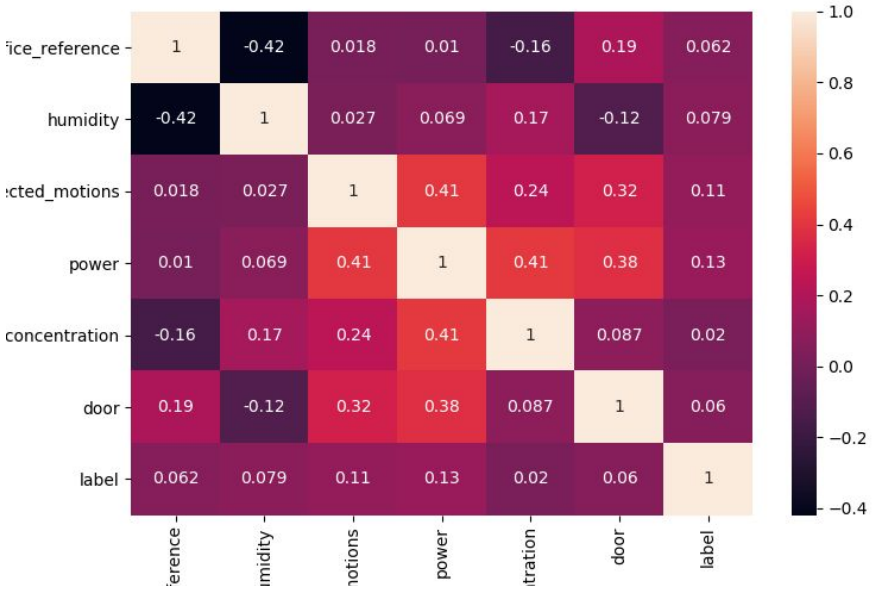


Figure 3: Correlation matrix of the dataset

As we can see on the last row “label” that interests us, power consumption and detected motions are features that matters the most and affects our estimation of occupancy, more detailed methods such as Univariate Selection technique and feature importance can be used to determine which feature is more important based on what estimation model we use, but overall, the matrix correlation gives a good and accurate first look.

Model evaluation:

Before starting to study the different models in order to estimate the occupancy profile, we have to choose model evaluations. Thus, there are two ways to manage to do that:

- Graphically: the aim is to plot the estimated occupancy profile and the real one in order to observe the linearity between them.

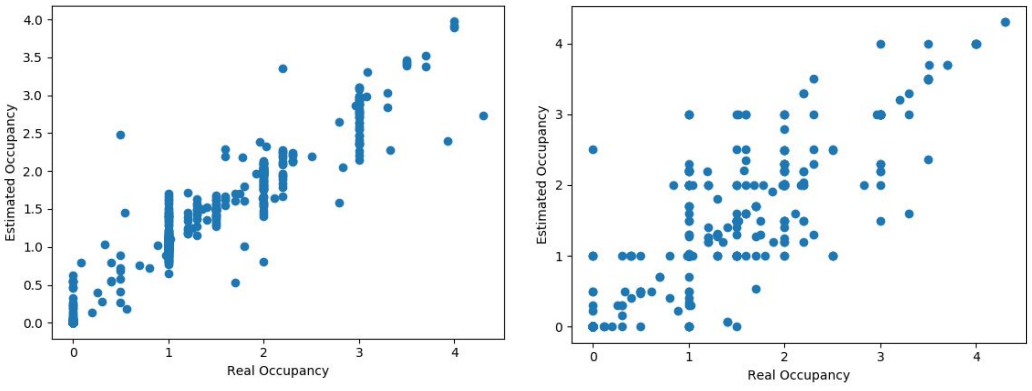


Figure 4: Plot of the estimation occupancy and real occupancy

Here, for example, we can observe that the model performance of the left one is better than the right one because the trend curve is more linear.

- Mathematically: the aim is to use different parameters to calculate the performance of the model.
  Thus, we decide to work with 2 parameters:
  - Mean Squared Error (MSE), a risk metric corresponding to the expected value of the squared (quadratic) error or loss: the best value is 0.0
  - R2 score (R2S), which is the coefficient of determination (R^2) : the best value is 1.0

Therefore, the best option to study a model's performance is to combine the two previous methods. The first one is to have a quick idea of the performance and the second one enables us to better compare the different models and to fit them to have a good model for the estimation occupancy profile.

Model comparison:

When the model evaluation is built, we can manage to compare the different models and determine which one is the best to estimate the occupancy profile.
To start, we will present each model and analyze graphically the performance of the model. Then, we will compare all the methods at then end using the error evaluation parameters.

Linear Regression:

The first model is Linear Regression which fits a linear model with some coefficients in order to minimize the residual sum of squares (Least Squared Error) between the measured features in the dataset, and the targets predicted (occupancy) by a linear approximation.

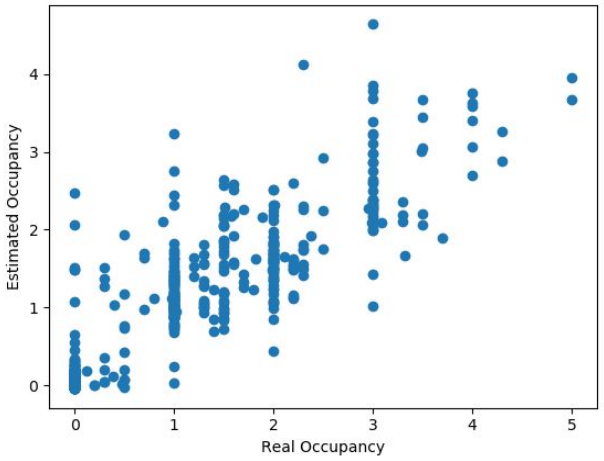


Figure 5: Plot of the estimation occupancy and real occupancy (Linear Regression)

Here, we can observe that the method seems to be relevant even though we have dispersions.

Gradient Boosting Regressor:

It builds an additive model in a forward stage-wise fashion. It allows for the optimization of arbitrary differentiable loss functions. In each stage, a regression tree is fit on the negative gradient of the given loss function.

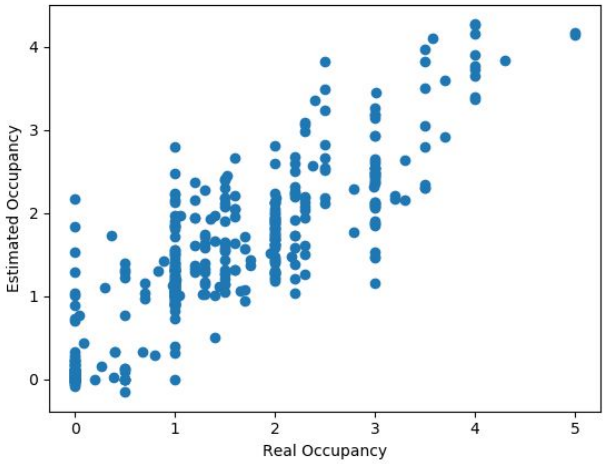


Figure 6: Plot of the estimation occupancy and real occupancy (Gradient Boosting Regressor)

With this model, we can see that we curb a little the dispersion that we had with the previous model. However, there are still some dispersion points.

Random Forest Regressor:

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

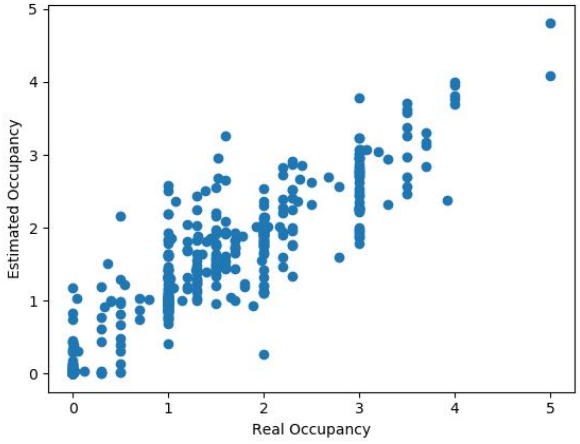


Figure 7: Plot of the estimation occupancy and real occupancy (Random Forest Regressor)

Finally, with this method, it seems that we have a good estimation using the graphic method. The dispersion is really low compared to the other ones.

Now, we will try to go further in our analysis using the error evaluation parameters.

	MSE	R2S
Linear Regression	0.1361	0.8128
Gradient Boosting Regressor	0.0841	0.8970
Random Forest Regressor	0.0490	0.9352

Figure 8: Table of mean squared error (MSE) and R2 score (accuracy) for different models

As we can observe, the Random Forest Regressor is the best method in order to have the minimum error to estimate the occupancy profile with a Mean Squared Error (MSE) close to 5% and a R2 score (R2S) rising 93.52%.
Finally, combining both methods, Random Forest Regressor stands for a nice method to achieve our objective.

Model Optimization:

Finding the method, we can work on the optimization in order to reduce the errors and so to have a better estimation.

Model parameters:
Each method has different parameters as inputs which it enables to better fit the model. For the Random Forest Regressor, we can change for example the number of trees in the forest represented by the parameter n_estimators. Previously, this parameter was set by default to 100. Thus, we can study the impact of changing this parameter on our models:

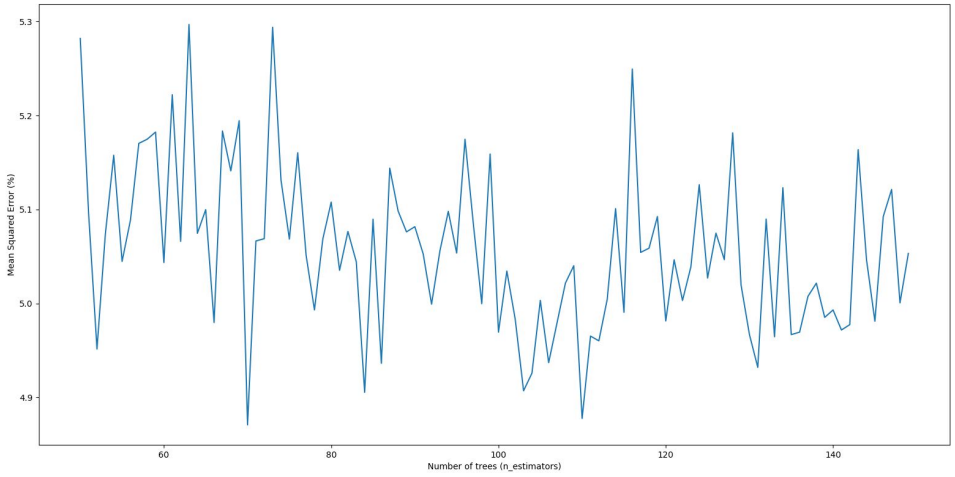


Figure 9: Plot of Mean squared error in function of the number of trees

As we can see, we are able to fit our model better by changing the number of trees even though we reduce just a little the error.

Adding sensors:

As we said earlier, in order to estimate the occupancy profile for the whole year, we only had some features to work on building the model. However, if we focus on the 2 weeks of May, we have more features measured that we can use to enhance our model. Besides, we add the outdoor Temperature, the CO2 corridor concentration and the acoustic pressure of the room.

	MSE	R2S
Random Forest Regressor	0.0239	0.9617

Figure 10: Table of mean squared error (MSE) and R2 score (accuracy) with added sensors

As we can see, we divided the MSE value by two that shows how relevant it is to add features to enhance our model. However, we didn't have those features measured for the year. Thus, complexity and the cost of having some features have to be studied even though it really increases the model performances.

Application of Occupancy estimation:

To have an idea of why the occupancy profile estimation is useful, we imagine an easy application:

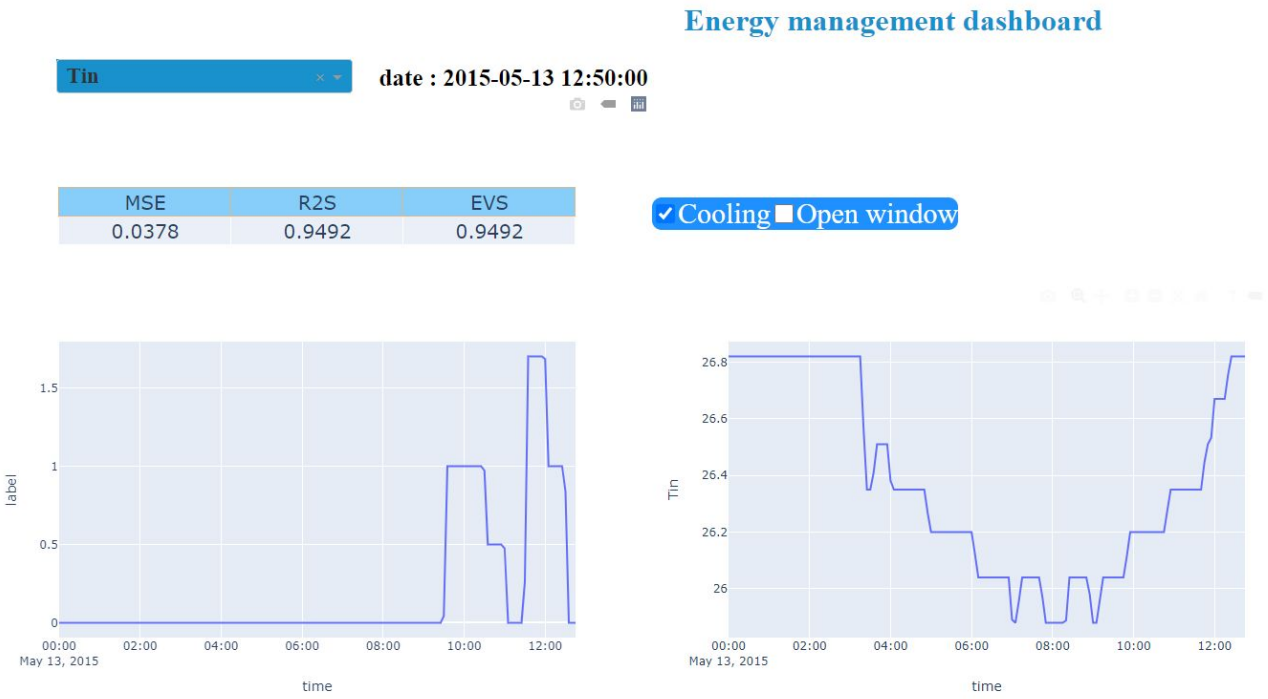


Figure 11: User interface example for an energy management application using the estimation model

The energy manager can control the occupancy and the CO2 concentration level. Here for example, the occupancy at 12.50 falls down to zero, so he can stop the cooling system and open the window to regulate the CO2 concentration level to save energy.

However, the comfort of the worker might be impacted because the temperature can vary a lot. Thus, forecast occupancy seems to be imperative in order to combine energy saving and ensure comfort of workers.

Occupancy Forecasting: Different Methods Comparison

Now that we have estimated the data for the rest of the year the idea is to try to build a model that would be able to forecast with certain accuracy the occupancy.
In the last years most methods related to forecasting were statistical methods like ARIMA or neural networks like LSTM. In order to properly choose our prediction method we first had a look into the data and the available features.

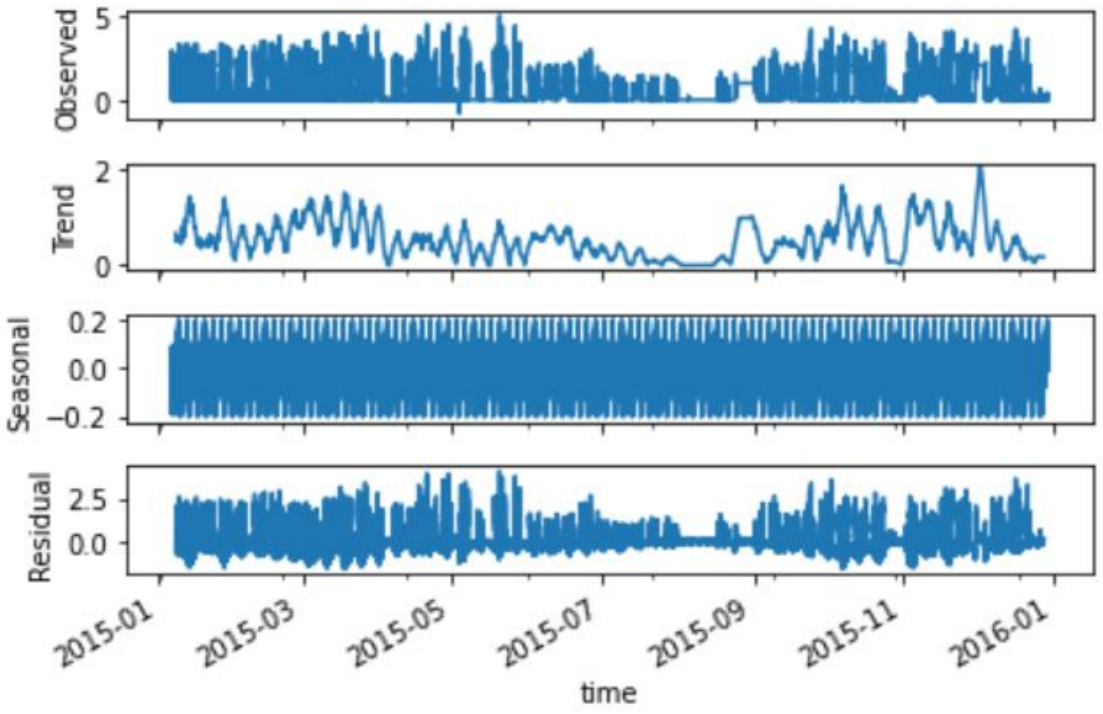
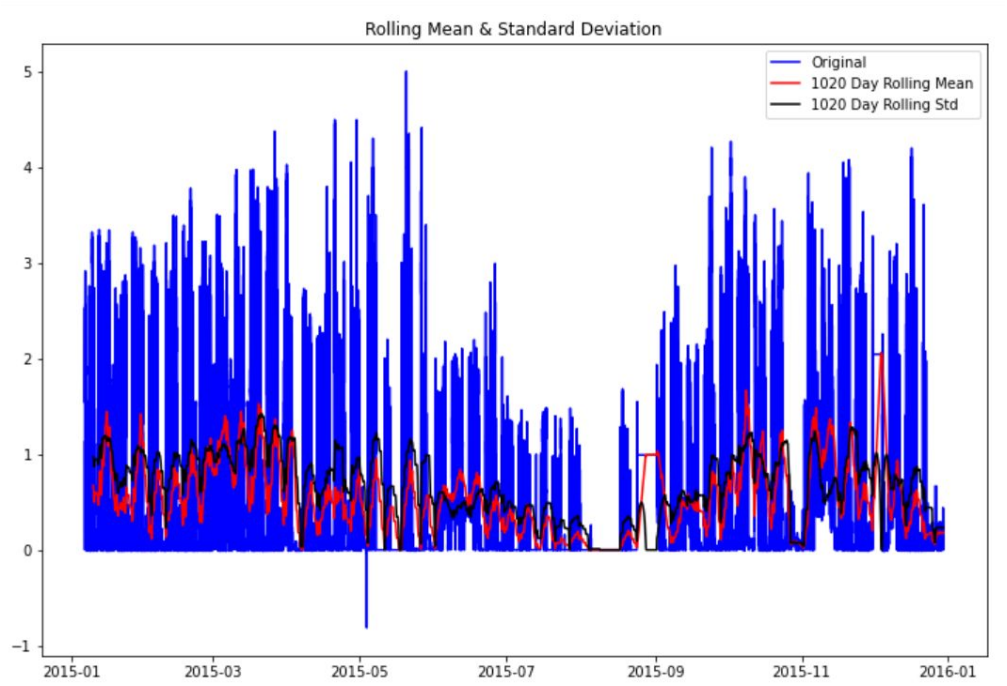


Figure 12: Dataset and statistical features

A time series is a series of data points in time order that can be described by the trend (component that doesn't repeat), seasonality (component that repeats) and residual (the rest). The python statsmodel allowsto clearly decompose our time series into these three components. Understanding these charts is an important step to decide what model could be used. Except for the weekly pattern there is no clear conclusion that can be made from this data since it is a small office varying from 1 to 5 persons it is not as easy to detect a pattern as if it was a big company. ARIMA is the most suited statistical forecasting method since it is more flexible (it combines two statistical models).
To use ARIMA the time series must be stationary meaning it should not depend on time, so no matter where you look the point should be the same. So it looks good with our data.




```
Results of Dickey-Fuller Test:
p-value = 0.0000. The series is likely stationary.
Test Statistic                  -24.308637
p-value                           0.000000
#Lags Used                       15.000000
Number of Observations Used    102914.000000
Critical Value (1%)              -3.430414
Critical Value (5%)              -2.861568
Critical Value (10%)             -2.566785
dtype: float64
```

Figure 13: The Dickey-Fuller test applied to our dataset

The Dickey-Fuller test is here to confirm the stationarity of our dataset by studying the rolling mean and standard deviation over each hour. According to this analysis it is worthy to try ARIMA and maybe some other statistical method.

According to last year's research there are definitely some methods that are praised to deal with forecasting problems, the statistical ones such as ARIMA and the machine learning neural networks such as LSTM and CNN-LSTM (see in the load forecasting paper). We were also interested in checking new and open source algorithms that had a lot of success in recent years forecasting Kaggle competitions (with awards around 25 000 dollars!!). Among these competitions there were recurrent ones reappearing such as XGBDT: gradient boosting decision tree.

Before going in depth into details let's have a look into the input features that could be used in our model and how.

a) Historical data

The most important way to predict the future is to study the past data, depending on the methods used we will see that we deal differently to put it as an input. Either we use sequences of historical data, let's say we use the two last days to predict the next one. Or either we use statistical features of the past data.

b) Calendar feature

The calendar feature is first of all creating input vectors to characterize the time step we are trying to predict. Which day of the week is it, what hour of the day, what day in the month?

time	year	dayofmonth	dayofweek	hour
2015-01-08 17:00:00	2015	8	3	17
2015-01-08 17:05:00	2015	8	3	17
2015-01-08 17:10:00	2015	8	3	17
2015-01-08 17:15:00	2015	8	3	17
2015-01-08 17:20:00	2015	8	3	17

Figure 14: Calendar feature table

Another available and interesting component is the gmail calendar of the main office occupant. Which is an event name with a starting and ending time. To use it on python needs some manipulation starting with the Calendar python library. Requests are written in a function to decode the calendar information and then introduce them in a dataframe that has the same structure as the Occupancy dataframe to be able to make the link.. Then we could have a clear look on the event repartition:

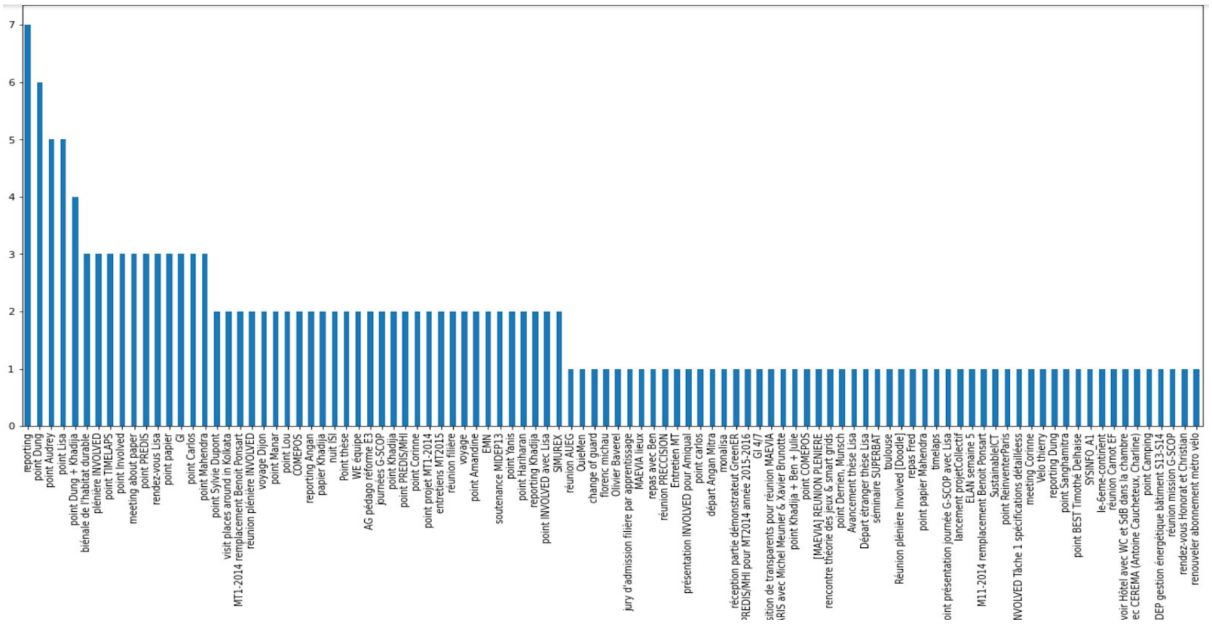


Figure 15: Event repartition of the calendar

What can be observed on that chart it's that the events are all different, the maximum occurence we get is 7 for the event called "reporting" and then most recurrent events are the ones with "point" in their names, besides these events were coinciding with strong occupancy in the building.. Since for the model we used a string input was not accepted we

had to convert them into numbers. So according to the calendar analysis all the events with “point” or “reporting” were converted to a “2”. All the other events were converted to a “1” and when there is no event it is reported to a “0”.
When comparing this calendar transformation to the real occupancy data it seems like a good approach. Indeed the events in the calendar match strongly with the occupancy, for example in august we clearly see that there is no event in the calendar and also no occupancy.

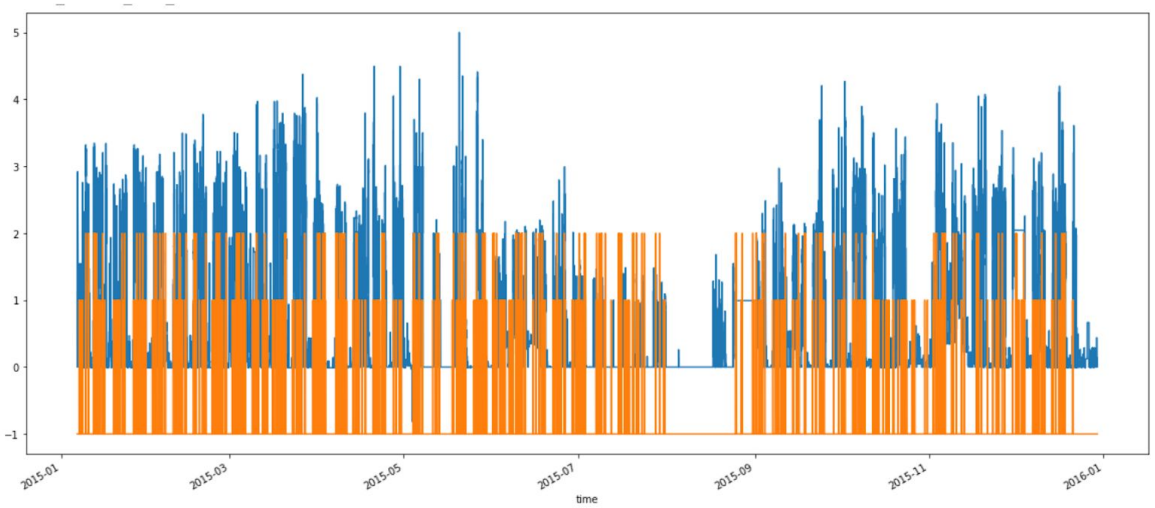


Figure 16: Occupancy data based on the calendar events

In order to use the calendar as a feature the past calendar data and the future calendar data will be used.
Finally it is frequent in time series forecasting to use a holiday feature but since we clearly since when are the holidays in the occupants calendar it seems unnecessary to do so.

c) Other potential features

In order to improve occupancy forecast there are some other methods for example connecting the code to the phone GPS location to complete/correct the calendar but this approach would be more “politically correct” in a house and intimate context rather than in the professional since it is totally an intrusive approach.

Overview of different methods

1. MLP,LSTM, CNN, CNN-LSTM

MLP

These different neural networks were used, there are two types. The ANN, artificial neural network Artificial use pattern-recognition methodology for machine learning. It is composed of many interconnected neurons divided into different layers, each of the neurons are linked by a weight that suits the model to the problem. The non-linearity is what makes it a good solution in many cases. So we give an input pattern to the model and it gives the best corresponding output pattern according to how it was trained.
The problem with MLP is that it doesn’t see the input as a sequence so it is losing on pattern recognition.

CNN

This is a convolutional neural network, that is mostly used today in image recognition. It has the advantage through convolutional layers that can treat the features at different levels.

LSTM

LSTM is a RNN, recurrent neural network that is one type of artificial neural network, this technique called Long and Short Term Memory is mostly used in predicting future sequences based on directly given input past sequence still keeping in its long term memory the most important sequences to recognize.

CNN-LSTM

Multiple forecasting research papers have proved the efficiency of the hybrid model CNN-LSTM, first CNN models are used first to apply historic data and then extracted features from the CNN layers are used as input in the LSTM model.

2. ARIMA

For ARIMA we use the pdmarima library, we were confronted to some difficulties since even if there is a auto arima function on python that can calculate the best parameters for the model it needs a lot of RAM, so for example it was difficult to obtain good results with the 5 min data. That is why we decided to resample our dataset to one hour to get better and faster results. It took two hours and half to run the model to forecast only one hour...

3. LightGBM

LightGBM is a gradient boosting opensource framework developed these last years n=by Microsoft that uses very powerful techniques but with fast resolution compared to the other traditional method. To explain briefly the method a gradient boosting algorithm is a model that uses each iteration of defining a model in order to learn from the errors.We used the LightGBM regressor decision tree inspired by the M5 Forecasting kaggle competition 3% leader algorithm. The GridSearch function is used to hypertune the model, meaning that we are trying to find out which are the best model parameters based on a cross validation technique. Cross validation is a method that enables us to use different training and testing sets so that our model gets more robust and not only adequate for only one type of training set. Another interesting tip in the code we used is that it used recursive prediction. In this algorithm the output is only one value not a sequence of values compared to the other algorithms we used, meaning that if we wish to predict the next 1 hours, with a sampletime of 5 min we run 12 times our model to get the 1 hour prediction everytime using the last time prediction. Furthermore in this model deeper feature engineering was made, instead of giving all the last 2 days as input we directly give statistical features resulting from the past observations, like the average.

2. General comments and results analysis

We compared the different models over a one year data set of 5min time interval, testing different input features for different forecasting times:

	1 hour	12 hours	24 hours
Naive	-0.602	-0.602	-0.602
Regression	0.665	0.201	0.170
MLP_24	0.774	0.293	0.207
MLP_48	0.740	0.265	0.203
MLP_24+Calendar	0.773	0.403	0.337 (0.236 for 12)
MLP_48+Calendar	0.759	0.423	0.330
LSTM_24	0.741	0.267	0.172
LSTM_48	0.693	0.244	0.157
LSTM_48+Date	0.689	0.1703	0.098
LSTM_24+Calendar	0.685	0.241	0.125
LSTM_48+Calendar	0.708	0.263	0.143
CNN-LSTM_24	0.780	0.202	-0.038
CNN-LSTM_48(10min)	0.732	0.1804	-0.07
CNN-LSTM_24+Date(20min)	0.767	0.32	0.189
CNN-LSTM_24+Calendar	0.755	0.296	0.156
CNN-LSTM_48+Calendar	0.767	0.275	0.142
LightGBM+Cal	0.911	0.843	0.710
ARIMA	...	...	...

Results by resampling the dataset over 1hour by taking the mean:

	24 hours
Naive_1	-0.288
Regression_1	0.32
MLP_24_1_Cal	0.377
MLP_24_1	0.380
LSTM_24_1_Cal	-0.5

Final model:

The most complex models like neural networks are not necessarily the solution to obtain an accurate model. Also models relying too much on statistics are complex to manage, they need more data analysis to study the trend and seasonality, the functions that can solve the problem need a lot of computational power. We tried with a sample time of 5min, 5 min of sample time was too much for the model that could not run properly but also 1 hour the model didn't seem to capture the trend. To my mind ARIMA is a model difficult to master since it needs a lot of statistical knowledge and from the moment you have a trend that is not in the "book" it is difficult to find information on how to perform it well.

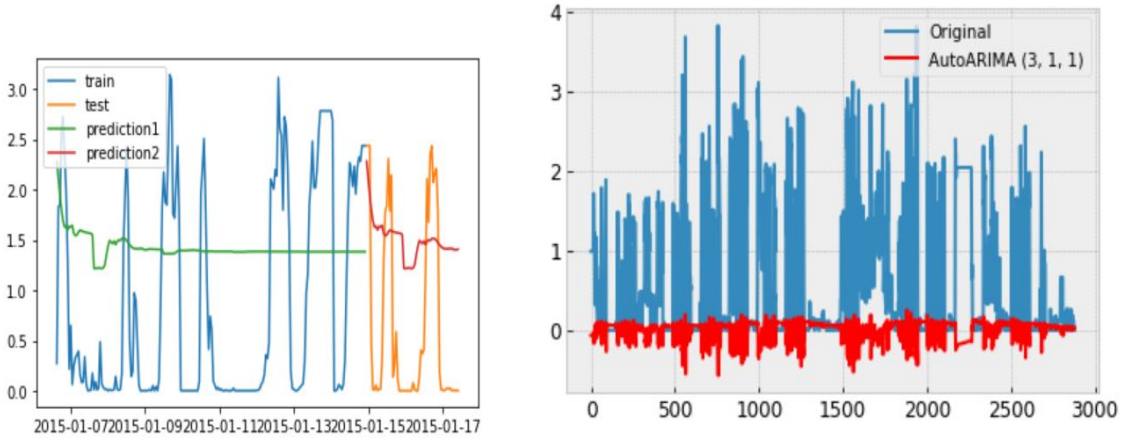


Figure 17: ARIMA model prediction

As we can see on the results table most of the algorithms we used have relatively close results but anyone outperforms the others maybe MLP more than the rest. What made the difference wasn't either the fact of trying to use or invent new possible features. The robust and accurate model that got the best model is the one we spent more time analysing the data, performing some feature engineering to show explicitly some features that the other models couldn't see by just having the sequences as input. The advantage of LightGBM is that it can get accurate with fast training. We also used hypertuning with 3 fold cross

validation that enabled us to find the best LightGBM model. To conclude on the forecasting part: the importance in forecasting is not the model but more how you use the data. Rather than spending too much time trying new models what makes the real difference is how you use the data. Some things that could have improved the forecasting results would have been perhaps to make some ensemble model which seem to be recently the most successful ways to resolve forecasting problems. We also saw that making a less frequent sample time enabled us to get more accurate results which makes sense since there is less data to forecast and less fluctuation.
Below we can see the most important features used by the LightGBM model to get the most accurate results. It makes sense that the last known data step is the one giving the most useful information. But it looks like the trend of the last week is also meaningful (standard deviation of r2016,5712, rolling mean shifted by 1 week over 12 hours).
Finally in the estimated dataset some occupancy trends are bizarre for example in some days we have for all day long an occupancy of 2 for 24 hours. Some outlier removal analysis should be made but since it concerns patterns it is a bit more complicated as if it was only one specific value at a specific time. The outlier analysis was performed manually but it could have been interesting to find a technique that removes the pattern outliers.

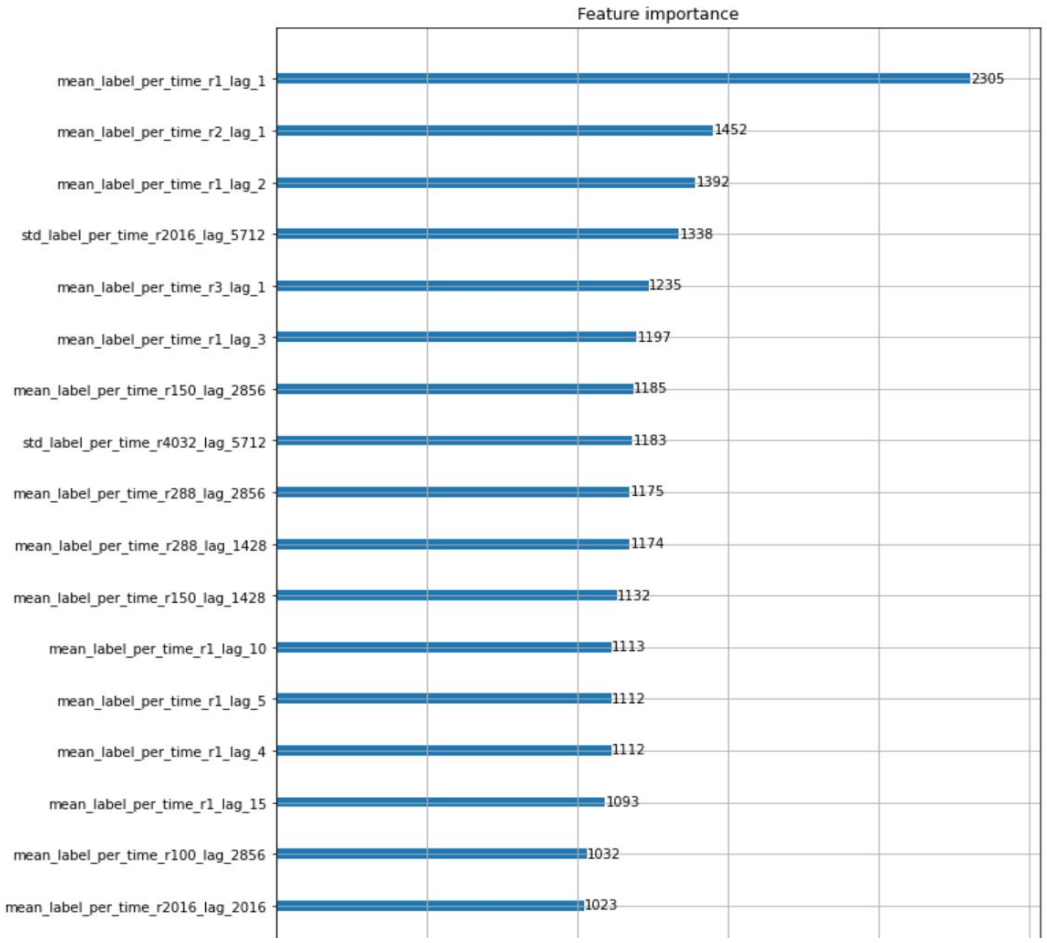
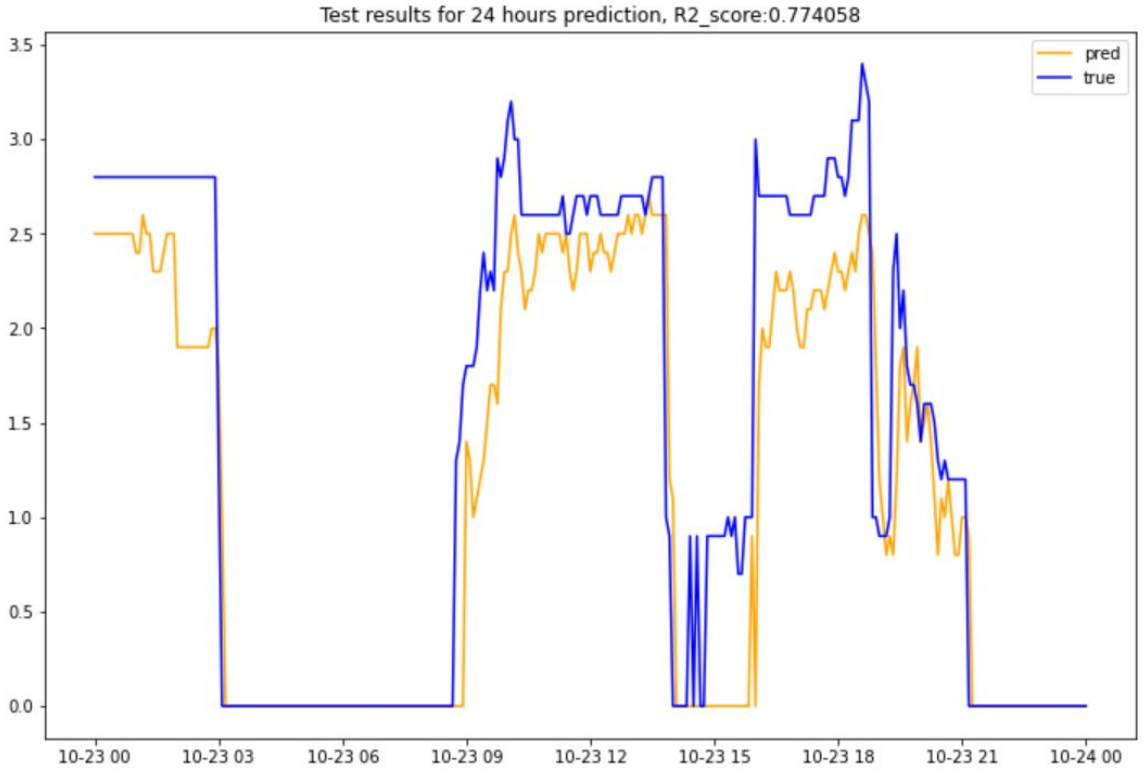


Figure 18: Feature's importance used by the LightGBM model

Below we can see for example the obtained results obtained for 24 hours. The part where the accuracy decreases the most is the transition between holidays and work, the model

seems to have difficulties to integrate the transition pattern, in order to solve this we should have used more data to capture more holidays transitions.



Application example: Office Energy Savings

Now that we are able to predict with a certain accuracy score how many people there will be in the office, how could we use it.We wanted to have a look into how energy savings could be done. In this office in Grenoble we are mainly interested in heating. Let's suppose for example that 21 degrees is the optimal comfort temperature (more or less 2 degrees) when the office is occupied, but when the office is unoccupied the room could reach a setback temperature of 15 degrees. The issue is that to heat the room depending on the heating device it can take hours to reach the right temperature, so to prevent discomfort to the occupants when arriving we have to try the right time to turn on the heater.
So the problem is the following: what is the best heating strategy knowing 12/24 hours ahead when and how many people will come in and out of the room? And for example with the new sanitary recommendation when is the optimal time to open 30 min the windows?

a) Office modelling

To test our idea, it is necessary to have a thermal model of the office. An approach could be to estimate the building's envelope properties following for example the resistance method proposed by the french RT2012 reglementation building framework. Since the three main values we are interested in calculating are the indoor temperature, heating power and indoor CO2 level since we have measured this data for over one year with some other features, a "numeric" model can be guessed. This machine learning model can be a simple linear regression. We obtained a model with an accuracy of:

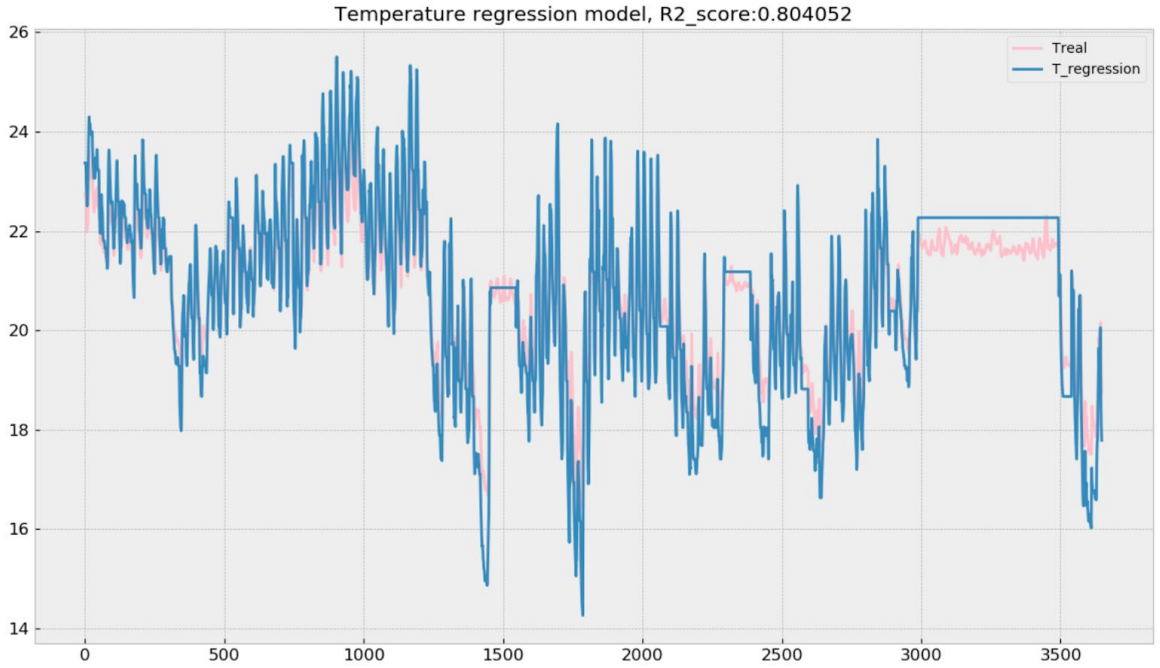


Figure 19: Temperature regression model vs the real temperature

b) Predictive control modelling

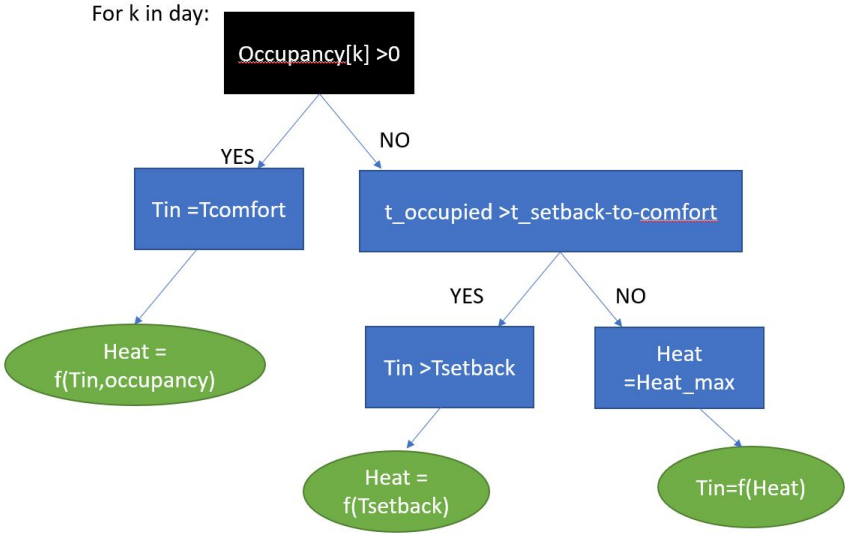
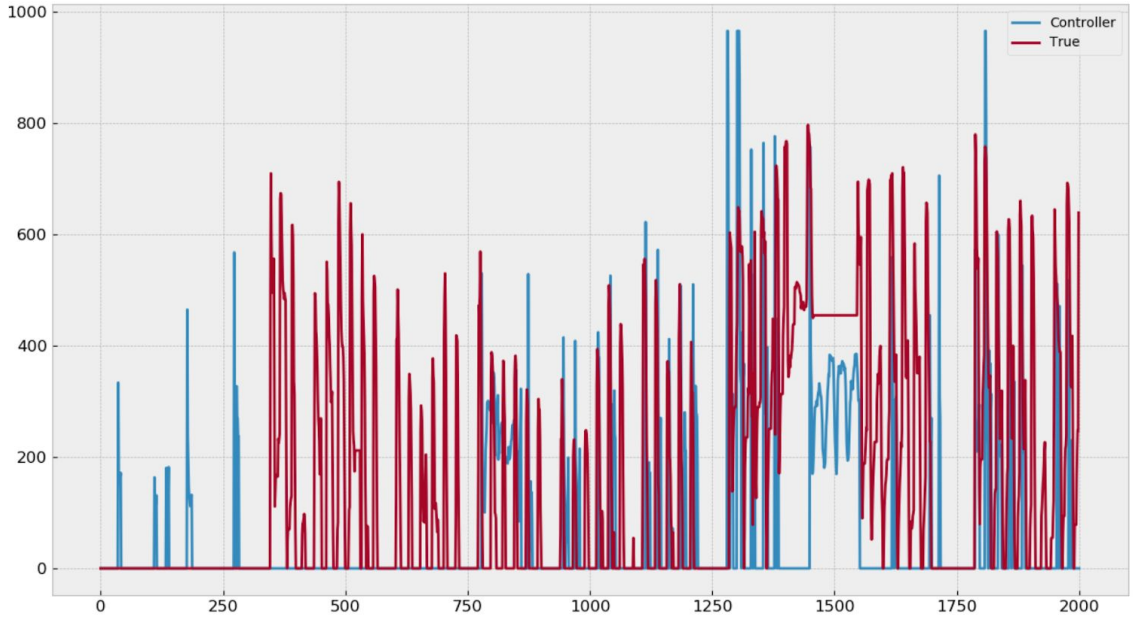


Figure 20: Simple model of the predictive control implementation

In order to plan for 24 hours how to control the heating an algorithm with naive hypothesis was written on python using the regressive models discussed in the previous part. This is a really simple theoretical model but a nice way to use it could be by implementing a Raspberry Pi controller that costs between 20 and 30 euros.
So the technique was then tested within 10 days, below we can observe the results of the real heating consumption versus the one with the predictive controller...



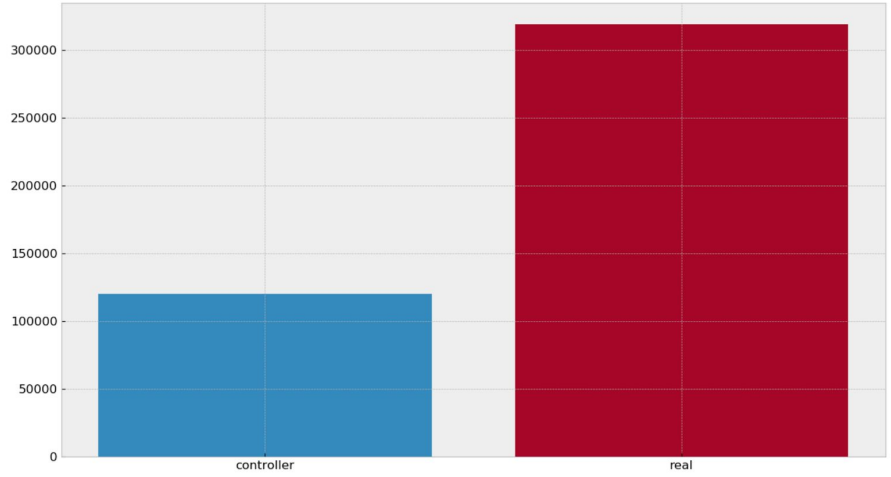


Figure 21: Energy saving using a predictive control modelling within 10 days

c) Making a simple and interactive Kivy App

This part is a thought on how to make efficient the use of the occupancy forecasts for the occupants. Maybe this could be a future operative project on implementing a Raspberry Pi since it works Kivy GUI. Kivy GUI is a user interface that works on python and that also works on your phone. The app gives you the planned 24 hours optimised heating programme of the office, and since it is additional cost instead of installing and resources usage it tells you when is the optimal time for you to open the window for the indoor air quality. Additionally by using the button update it enables the occupant to correct the prediction if he feels it is necessary (that could help improve the model and be a less intrusive way to train machine learning algorithms).

Figure 22: User interface of an energy management using the predictive model (Kivy)

Conclusion and perspective

Overall, this project has been a good and interesting introduction for us to machine learning, and its applications, specially in our field of study. We have respected the timeline set for our project, as well as the goals we wanted to achieve, we've built and tested various prediction models, and compared between them, we have introduced different features, and evaluated them, determined which make more sense and add accuracy to our models. We did sensitivity studies throughout the models, specifically with sample time and features.

We wanted at the end of project to concretize what we did in our project, and introduced some example applications, that can use our predictive model to save energy, using the occupancy forecasting, and it can even suggest some basic comfort and air quality improvements, such as decreasing the CO_2 concentration in the office by opening the window at a certain time in the day to not waste energy. We are certain that this project has definitely helped us set the first foot in machine learning, and that it will be useful in our future endeavors, especially with the newly introduced 2020 thermic regulation (rt 2020) which focuses on, among other things, smart building management that some big companies are working on using machine learning and predictive models.

References:

- Manar Amayri, Stéphane Ploix, Sanghamitra Bandyopadhyay. Estimating Occupancy in an Office Setting. The First International Symposium on Sustainable Human-Building Ecosystems (ISSHBE),Oct 2015, pittsburgh, United States. 10.1061/9780784479681.008. hal-01246159
- Shadan Golestan, Sepehr Kazemian, and Omid Ardakanian. 2018. Data-Driven Models for Building Occupancy Estimation. In e-Energy '18: International Conference on Future Energy Systems, June 12–15, 2018, Karlsruhe, Germany.ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3208903.3208940
- Lam, K. P., Hoynck, M., Dong, B., Andrews, B., shang Chiou, Y., Benitez, D., and Choi, J. 2009. Occupancy detection through an extensive environmental sensor network in an open-plan office building. In Proc. of Building Simulation 09, an IBPSA Conférence.
- Different_Time_Series_Modeling_Approaches_for_Short_Term_Load_Forecasting.
- https://github.com/minaxixi/Kaggle-M5-Forecasting-Accuracy/blob/master/model_recursive.ipynb
- Bojer, Meldgaard, Learnings from Kaggle's forecasting competitions,2020