HIGH SCHOOL OF COMMUNICATIONS OF TUNIS

P2M PROJECT

Intelligent air conditioner management based on machine learning

Author:
Ben Slimen Aya
Mlika Eya

Supervisor:
Sami Tabbene
Erij Sioud

A report submitted in fulfillment of the requirements for the P2M project at sup'com

with

SFM Technologies

May 10, 2022

Acknowledgements

We would first like to express our gratitude to Mr. Sami Tabbene, who proposed the subject, accepted to supervise us and made us benefit from his experience.

This work was carried out in collaboration with SFM technologies, during a period of approximately 4 months. We would like to thank all the team members, interns, administrative and technical staff for their help during this period. We particularly thank Mrs. Erij Sioud and Mrs. Sirine Saihi for the time they devoted to us and the precious advice they gave us.

Many people whose names are not mentioned helped us and gave us a hand. We hope one day to be able to show them our deepest gratitude.

Contents

Chapter 1: Introduction

This document is a reports on all of our activities realized during our P2M project at Sup'com. Our mission was to develop a smart layer to an existing Iot solution "cool@-C". In order to make better decisions and save energy. The work consists of applying machine learning algorithms to predict the right temperature and to detect abnormal cases related to the activities of the air conditioner, consequentially managing the energy costs of an air conditioner .

This project had has as a goal to exploit our knowledge acquired in class and make us gain more in depth practical training. This project allowed us to become familiar with the machine learning algorithms and to discover Time series algorithms, learn a new web development framework and get acquainted with a NoSql SGBD.

We have structured our report in 6 parts, starting with an introduction and the definition of the state of the art. Then we delve deeper into the project analysis, the methodologies and technological choices we made in development. In the model implementation section,which follows the CRISP-DM methodology for data mining, we talk more about the data and machine learning algorithms we used. Followed by the web application section where we talk in depth of how we developed and exploited the results found in the previous section.

Finally, we will end our report with a conclusion, in which we will summarize all that has been achieved. We will highlight the added values that we have brought to this project. This project can be subject to evolutionary extensions, in the framework of continuous improvement; this will be the subject of the perspectives.

Chapter 2: State of the art

2.1 Introduction

The use of individual air conditioners in administrations or other public spaces generates important costs due to the negligence of the users. It is consumes almost half of the energy of buildings and 20 % of overall national energy consumption Therefore it is essential for a building manager to control the consumption of air conditioners through tools that automate the turning on and off of air conditioners and the setting of authorized temperatures. This is where the SFM team developed an Iot solution for the management of the functions of the air conditioners.

2.2 Critic of the existing system

The existing system is a set of inter connected Iot devices called Cool-@C that detects the temperature and movement. As it is the system lacks the autonomous decision making and smart control over energy consumption. The solution is a box made in SFM which contains a controller and temperature and movement sensor this box collects the data from the temperature and movement sensor and send it to the server.

But this solution still lacked a smart layer to further improve the level of automation.

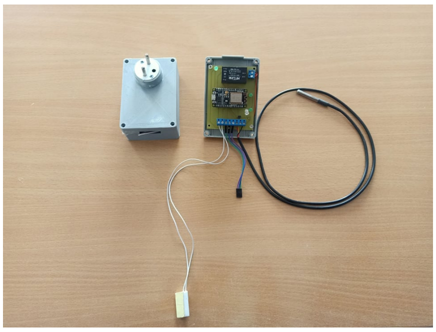


FIGURE 2.1: The Cool@C device

2.3 The solution

As a solution, we intended to develop a smart layer based on machine learning algorithms. The objective of the module is to predict and provide data driven decisions of the most adequate temperature to the user and the least power hungry plan. This module will work on the data gathered by the Iot and extract the best features to make decisions. Then through an API load the decisions on the equipment: to adjust the temperature.

Chapter 3: Project analysis and methodology

3.1 Project objectives

As stated before this project has for objective to develop a mart layer that can:

- Predict the best temperature wanted by the user
- Add to existing the system the energy consumption control
- Add to existing the system the detection of anomalies

3.2 Work environment

To achieve the before mentioned objectives we will need a set the technologies and tools to get the work done.
For the data engineering and the predictions we used:

- **Python** as a programming language because it has a very large choice of libraries and open-source projects for machine learning.
- **Google Colab** which is a a free working environment that provides us with powerful resources that allows us to meet with time series algorithms requirements. Added to that Colab allows us to share notebooks and is easily accessible

For the web application development we used:

- **Django :** is an web development framework. It is built on Python and performs a ML operation on it.
- **Mongodb :** We used Mongo DB to store massive amounts of data and manage the data system.

For the version control we used:

- **GitHub :** is a web service for hosting and managing software development. We used to facilitate collaboration on project.

3.3 Project architecture:

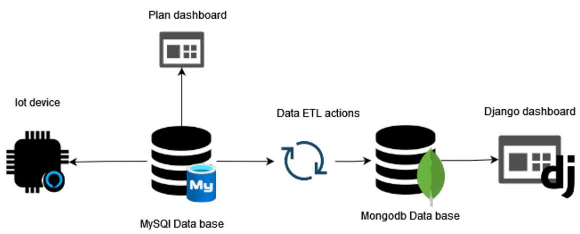


FIGURE 3.1: Project architecture

Chapter 4: Model Implementation

4.1 Objectives:

In order to apply our solution, the information must be identified and extracted from their original locations. These are heterogeneous data sources that may include internal company data, stored in the production databases of the various departments or external sources, retrieved via remote services and web services.

4.2 Data scraping :

One of the techniques we used in our project is Web scraping is a technique that consists in reading a web page and detecting the different elements that make up its content, allowing to locate and classify the information that is available.
This technique allowed us to scrap from the internet past exterior temperatures As simple as the scraping process is we were met with difficulties, especially when searching for the best data source. After thorough search we settled with tutiempo.net website for it provides temperature every hour.

FIGURE 4.1: Data scraping source

And thanks to this process we acquired the following data that we saved in a CSV file.

TimeStamp	TempExt
2021-08-01 00:00:00	30
2021-08-01 01:00:00	30
2021-08-01 02:00:00	26

FIGURE 4.2: *Scrapped Data*

4.3 Data engineering

4.3.1 Data comprehension

This phase aims to determine precisely the data to be analyzed and to make the link between the data and its meaning from a business point of view. To start, we selected a set of temperature and movement data that were collected from sensors in the SFM apartments over a period of 6 months. Once the data is extracted in the preparation area, we apply several transformation steps such as the selection of the data necessary for the implementation of the model (we removed the ID-box, Device-name...). To merge our 2 datasets, we found out that we don't have the same time scales, the outdoor temperatures are collected every hour, but the temperature sensor captures the values every 4 seconds and the moouvement sensor works when someone moves in the office. Also the time-stamp format is not the same for the different data sources. So we convert the time scale to a standard form and then merge these scales.

Mvt	Temp	Date	Time	TempExt
0	29	2021-08-10	00:00:04	36
1	30	2021-08-10	00:00:08	36
0	30	2021-08-10	00:00:12	36

FIGURE 4.3: *Clean Data example*

4.3.2 Data cleaning

In this phase we were faced with problems like:

- **Missing values:** After merging the data sets, we got several missing values .

To address this problem, two methods were used:

- Categorical imputation: the movement sensor takes a value of 1when someone moves in the office. Missing cells are usually filled with zeros.
- Numerical imputation: the external temperatures scraped from the web are collated every hour, while the internal data are collated every four seconds. So the empty lines in the time stamps for one hour are filled with the average value.

- **Outliers :**

By visualizing the data we detected the outliers which are extreme values that stand out greatly from the overall pattern of values in a data set or graph. So we removed the outliers which are temperatures below 0°c

- **Normalization :**

To reduce the complexity of the models, we test the two normalization algorithms: Min Max Scaling: This process involves scaling all values of a feature in the range 0 to 1 Standardization: This is the process of scaling the data values so that they acquire the properties of the standard normal distribution.

we can see that the standardization gives a clearer distribution.A good feature engineering will allow to eliminate irrelevant variables and keep only those that improve the performance of the model and avoid overlearning.

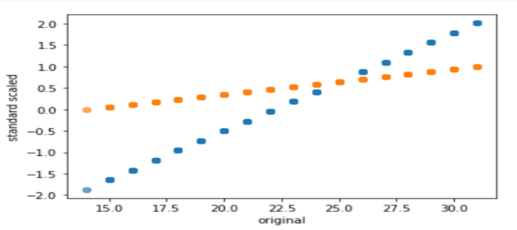


FIGURE 4.4: Data scraping source

4.4 Data visualization

In order to provide understandable information about the date we visualized the data using the **Plotly**. Which is an open source Python module that is used for data visualization.Here is an example of the graphs:

FIGURE 4.5: Mean Temperature by date

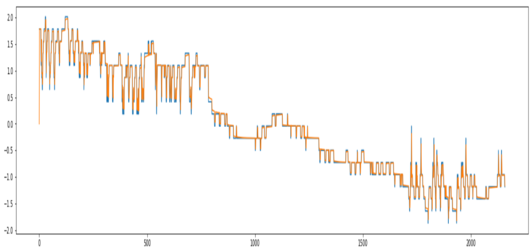
4.5 Model choice

To predict the appropriate future temperature and the state of the air conditioner. We identified the problem as a time series analysis one because of the cyclicity and seasonality of the data.

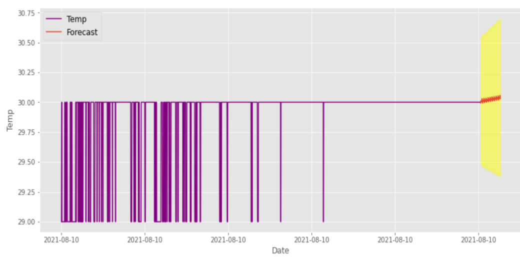
4.5.1 ARIMA/SARIMA

ARIMA is a statistical model designed for the analysis and prediction of data from a time series. It helps determine a transformation or change perspective. The machine learning model that uses Arima supports specific algorithms and calculations. These algorithms are required to express a prediction based on previous values, the time scale used and their fluctuations.
Since we are trying to predict the temperature in the future based on the values recorded previously, ARIMA is an algorithm that meets our needs.
Since we are trying to predict the temperature in the future based on the values recorded previously, SARIMA is an algorithm that can meet our needs.

Simulation :



(a) ARIMA forecasting



(b) ARIMA forecasting

4.5.2 LSTM: Long short-term memory

LSTM was used since the time lags are unknown between important events in a time series.

Simulation :

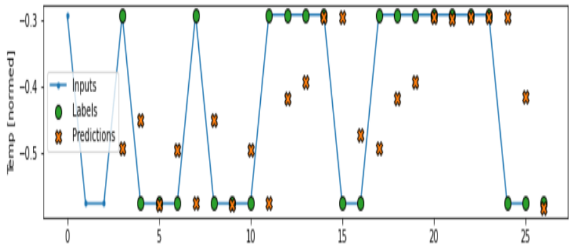


FIGURE 4.6: LSTM forecasting

4.5.3 Model comparison for temperature prediction :

Model	R2 Score
ARIMA	0.98
SARIMA	0.7
LSTM	0.132

-> After seeing these results, we notice that the ARIMA model has the best R2 score So ARIMA is the best model for predicting temperature.We worked with RNN but it takes more than 17 hours to predict the temperature.

4.5.4 Time Series K-mean :

In this part we try to predict the state of the air conditioner, so we chose this model because it helps us to find the state of the air conditioner (functional/non-functional) according to its temporal change.
Simulation :

DateTime	Cluster
2021-08-10 00:59:59	Cluster 0
2021-10-15 17:59:59	Cluster 0
2021-10-16 10:59:59	Cluster 0
2021-10-16 11:59:58	Cluster 0
2021-10-16 12:59:59	Cluster 0
...	...
2021-10-25 03:59:59	Cluster 1
2021-10-25 02:59:58	Cluster 1

FIGURE 4.7: Cluster forecasting

Chapter 5: Web Application Implementation

5.1 Requirements specification

1-Functional requirements	2-Nonfunctional requirements
• Recommend the appropriate temperature to the user by viewing the data. • Allow you to authenticate and manage your profile and data • Display an alert message to the user in case of an anomaly	• Ergonomic • Highly available • Reliable and secure • Scalable

5.2 Database Conception

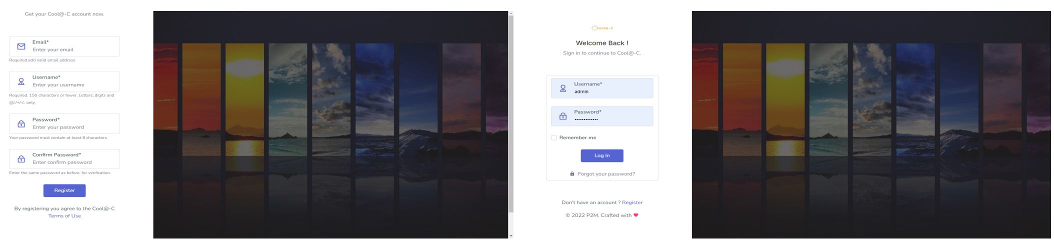
The advantage of using a No SQL database like MongoDB is the fact that we can define the schema in any way needed. So for our we application we defined a basic authentication schema with a user that requires an email, password, and a username. We also defined some roles for the application like the superuser for the administrator dashboard. As for the measures of temperature and movement we defined the following schema

```
{"id": 0,
"Mvt": 0,
"Temp": 29,
"Date": "2021-08-10",
"Time": "00:00:04",
"TempExt": 36
}
```

FIGURE 5.1: measure json schema

5.3 Implementation

As any web application we start by the implementation by the first security layer which is the login and registration page:



(a) Register page

(b) Login page

FIGURE 5.2: authentication pages

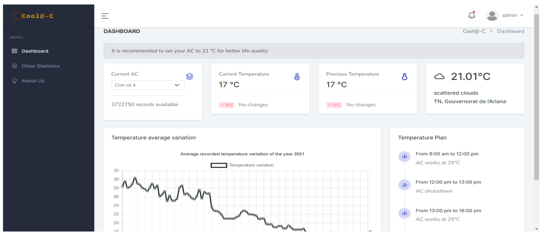


FIGURE 5.3: dashboard page

The user is then welcomed by a dashboard that contains the stats of the current recorded temperature and the previous one with a notification of the state of the AC utilisation. The user can also find a real-time current weather information.

On another page the user can find more detailed information about the weather and the recorded temperatures and movement traffic inside the office by selecting the date he wants to check.

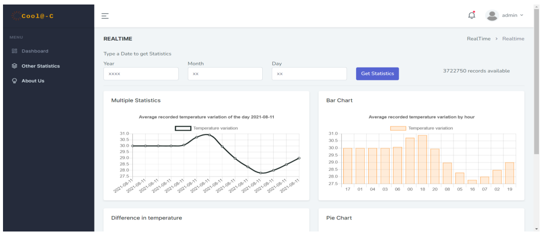


FIGURE 5.4: Other statistics page

The user can select the date check the average temperature variation on that day. See the statistics of the recorded temperatures by hour. Check the differences between the maximum recorded temperature the minimum recorded temperature and the average by hour on that day. And finally check the traffic recorded by the movement sensor.

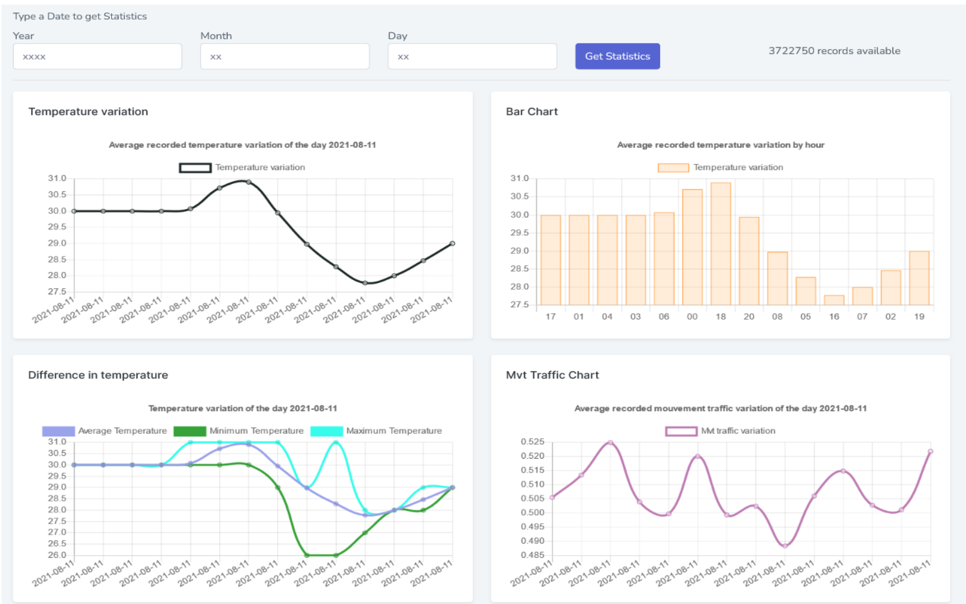


FIGURE 5.5: A close up of possible visualisations found on the app day 2021-08-11

Chapter 6: Conclusion

The project launched by SFM technologies, which is the result of a purely strategic desire, aims to evolve its organization and make better decisions. This project is an ecologically driven project aimed to lower power consumption and improve AC systems.

During this project, we were given records of temperature measures and movement. In order to find the best temperature to set the AC to. So we started with research where we had to look for the best usages of Air conditioners, we had to familiarize our selves with the Iot device and some basic notions of the Iot world. Then in the data engineering phase we followed the CRISP-DM method for the best results. From the data we had we could apply the many time series models. In the end, we settled with the ARIMA times series algorithm because it had the best performance.

And to finalize our work with a concrete product we developed an API that provides the data, result of the data engineering phase. The API is then consumed by an analytical dashboard that contains useful information and statistics with a notifications feature to inform the user if there was a problem with the system.

This project is still open for improvement. With the right data we can make even better data driven decision. We can use reinforcement learning models to make predictions according to the user preferences, the commands given to the AC and the previous data of temperature. This way the decision will be more user friendly and adapts to the user needs. And if we can get data on the power usage of the AC, we can understand and make power consumption lower according to the exterior temperatures.

In this world of data, the quality of the collected records is what can make us reach the best decisions.