Expertise
and insight
for the future

Mai Vu

Assemblin - Smart Building Control Systems

Helsinki Metropolia University of Applied Sciences
Information Technology

Work Placement

Final Report

18 December 2020

Contents

List of Abbreviations

AI	Artificial Intelligence.
ANN	Artificial Neural Network.
API	Application Programming Interface.
FMI	Finnish Meteorological Institute.
IT	Information Technology.
ML	Machine Learning.
MSE	Mean Squared Error.
NaN	Not A Number.
REST	Representational State Transfer.

1 Introduction

Assemblin project is a cooperation between Digi-Salama, Assemblin company, and Finnish Meteorological Institute (FMI). It is one of the twenty-five projects of Digi-Salama that focus on applying Artificial Intelligence (AI) and Machine Learning (ML) in automation for real estate. The primary task is to upgrade the heating and cooling control system from the Assemblin company's building using data from FMI. The ultimate objective is to optimize the use of energy.

As Finland is a cold country most time of the year, the energy consumed for heating is considerable. In particular, space heating accounted for nearly a quarter of end energy use in 2005. It is essential to minimize the unnecessary usage of heating equipment. [1, pp. 2 & 8]. Nowadays, most buildings can automatically control the heating system by turning it on and off when the indoor temperature is not in the desired range. Assemblin project aims to further improve this current system by employing weather data from FMI. Indeed, the weather can greatly affect the building. Therefore, weather information such as temperature, wind speed, sun intensity, etc., is beneficial and valuable for a smart, intelligent control system.

Assemblin combines both data from the Assemblin company's building and FMI weather data to find their correlation. Information from both sides is carefully analyzed to estimate the most impacting factors affecting the building temperature. Building ML models is the next critical step. Models learn from this merged dataset to generate and give commands to the building's heating system. With these AI algorithms, the Assemblin control system can adjust the indoor temperature according to the upcoming weather event. This approach is expected to optimize power usage and helps save energy.

This report's goal is to present promising models for this particular problem. The report mainly shows the Information Technology (IT) side of the Assemblin project, including the data analysis process, models' results, and further development. The following chapter shows the general information about the companies and the Mechanical and the IT part of the project.

2 Background Information

2.1 Companies and Human Resource

- **Assemblin**: Operating in Sweden, Norway, and Finland, Assemblin is an end-to-end installation and service partner, specialized in electrical engineering, heating, cooling, sanitation, ventilation, cooling, etc. The company focuses on designing, installing, and maintaining technical systems and automation. Assemblin also actively pursues sustainable development, of which efficiently using energy is one of thirteen sustainability aspects the company wants to perform. More information about Assemblin can be found on the company's website [2].

  On behalf of Assemblin company, Jarkko Turunen is the instructor and monitor, providing the building material and ensuring the outcome of this project is what the company needs.

- **Digi-Salama**: Digi-Salama project is a collaboration between Metropolia University of Applied Sciences and the City of Vantaa. Digi-Salama carries out 25 projects. Each project concentrates on diverse areas using different technologies. More information about Digi-Salama can be found on the project's website [3].

  Digi-Salama also wants to explore whether new advanced technologies can have a considerable impact on energy consumption. Assemblin project serves this purpose. It focuses on applying ML in automation, enhancing the current building system to optimize energy consumption.

  Antti Liljaniemi, Aarne Klemetti, and Lauri Ristolainen are supervisors of this Assemblin project on the Digi-Salama side. They are professional and experienced tutors, providing students with complete and enthusiastic supports. They also help push the project forward and set short-term, long-term objectives through different project stages.

A group consisting of 6 students works on this project, divided into 2 smaller teams: IT and Mechanical. The Mechanical team includes Melina Friman, Markko Hyvönen, and Heikki Lomu. The IT team has Mai Vu, Zoltán Gere, and Antti Pasanen. Mai Vu, who is the writer of this report, and Zoltán Gere are responsible for the data analysis process and the AI algorithms. The team organization is shown in figure 1.

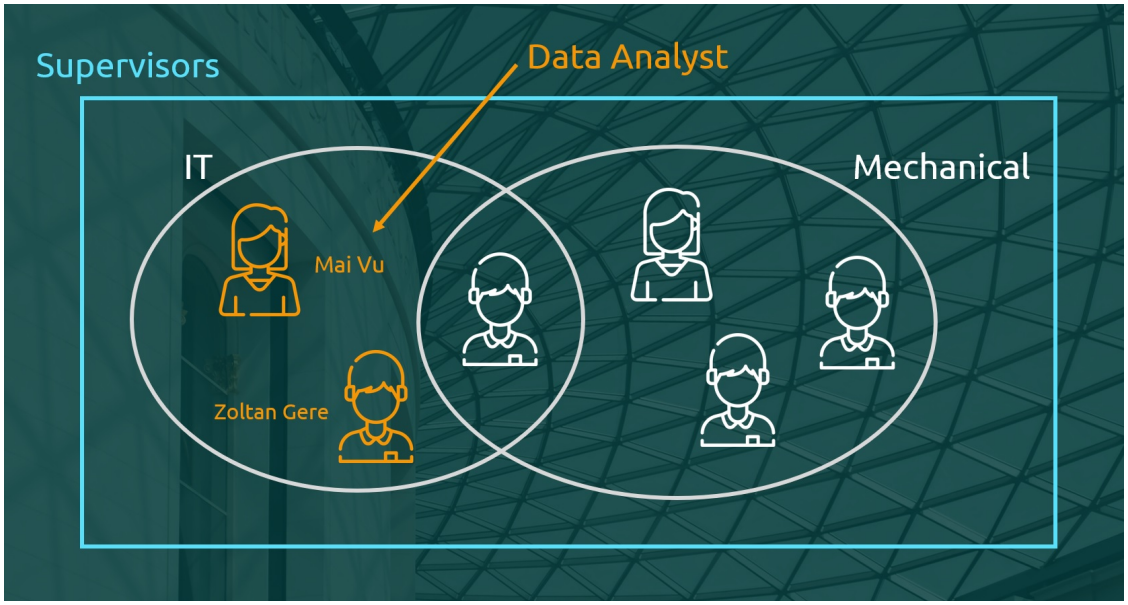

Figure 1: Assemblin team organization and Data Analyst role.

- **FMI**: The Finnish Meteorological Institute is an administrative division of the Ministry of Transport and Communications. The institute provides observation and researches on weather-related subjects, such as atmosphere, weather, air quality, etc., to ensure citizens' quality of life. More information about FMI can be found on the FMI website [4].

  Anders Lindfors is a research professor from FMI. He helps the project on the weather analysis side. Also, he wants to test a new feature called the sun intensity to see if the feature affects the building temperature.

2.2 Mechanical Engineering

The Mechanical team's tasks are to explore the building system from Assemblin, control the testing building's heater and air conditioner, and integrate the model into the real

building loop control system. Due to the lack of time, only the first part of the project is finished. The Mechanical team achieves the exploration of the building system and builds a small simulator room control system. In the next stage of the project, the team will use this simulator room to test the models and later integrate models into the system.

2.3 Information Technology

The IT team's job can be split into 2 parallel parts: building the simulator control software and analyzing data. The former goes to Antti Pasanen, and the latter belongs to Mai Vu and Zoltán Gere. While the control system is mostly completed and might only need some minor changes, the data analysis and model's developing process only reaches halfway due to the lack of time. Various models are created; a few of them get tested in the simulator environment. The results are analyzed, but no time left for further improvement. The model testing in the Mechanical's room and the real building will be performed in the next stages of the Assemblin project.

2.3.1 Control Software

The control software is written in Python and runs on Raspberry Pi. It consists of 4 threads running simultaneously. The first and second ones are for reading weather data from FMI Open Data service, the sun intensity data from pre-downloaded files from another FMI service called Ilmanet, and for transferring it to the simulator through REST API. Another thread is to apply AI algorithms to calculate the setpoint and send it to the simulator. The last thread is simply for the simulator to read and update the data. Each thread runs its own loop to do the task and sleeps until its next cycle. The length of the thread's cycle can be modified as well.

The data collection for building the models is done by this control software by recording the simulator each minute. The team lets the simulator runs and collects data for days to build a sufficiently large dataset for AI algorithms. The control software also has a testing mode where the software can read the data from files, arranging the same situation to examine different models.

2.3.2 Data Science

The Data Science part of the Assemblin project in the first stage involves 2 steps: data analysis and model development, and is separated into 2 approaches: temperature based (in charge by Mai Vu) and sun intensity based. The expected outcome in this stage is 2 AI models focusing on 2 above features separately. In the next stage of the project, these 2 features will be combined to create more accurate models.

For the temperature based model's development process, Python and Jupyter Notebook are used. They are chosen because of the straightforward syntax, easy debugging interface, and open libraries for ML algorithms development.

The data analysis process is a critical step, as it is essential to understand the provided data. The process mostly utilizes Pandas features to preprocessing and explore the data. Pandas is a powerful software library for data manipulation and analysis, written for Python programming language [5]. Some Pandas features used in this process are DateTime format, missing data handling, data column insertion/deletion, data set merging, etc.

The second step is to build the ML models result in the wanted outputs as a benchmark to turn on or off the heater and the air conditioner. The models in this project are developed using Scikit-learn, an open, simple, and efficient ML library [6]. Scikit-learn metrics functions are also used to evaluate model performance.

The Data Science part is documented in the below section. It starts with the information of the datasets and the descriptions of features. After that, some important observations on the datasets are mentioned. Following are the details of the potential model for the temperature based approach.

3 Data Analysis Process

3.1 Overview

The project used 2 type of datasets: building data from Assemblin's simulator and weather forecast from FMI services.

3.1.1 Building Data

An original dataset contains the recorded data of the simulator provided by the Assemblin company, including a total of 13 columns. The one described here is the most extended available dataset, collecting from 24.11 at 15:45 to 26.11 at 12:46. The spreadsheet is first read. Missing values are detected by the value -999 and are replaced with Numpy Not A Number (NaN) - a numeric data type. The *Date* and *Time* are combined, changed to a Datetime format, and set to be the index column afterward for easier manipulation. After briefly preparing the spreadsheet, the new dataset is formed, and its total number of columns is 12. Samples of it are shown in figure 2.

Datetime	Room Temperature	Room Setpoint	Heating Demand	Heating Power	Cooling Demand	Cooling Power	Current Outside Temperature	Outside Temperature External	Current Solar Power	Solar Power External	Supply Air Temp
2020-11-24 15:45:43	21.645634	21.0	0.0	0.0	88.592758	730.760681	1.8	1.5	41.639999	32.959999	18.947823
2020-11-24 15:46:43	21.571623	21.0	0.0	0.0	91.331947	738.987122	1.5	1.5	32.959999	32.959999	19.019285
2020-11-24 15:47:43	21.520304	21.0	0.0	0.0	94.145180	760.433594	1.5	1.5	32.959999	32.959999	18.985868
2020-11-24 15:48:43	21.650057	21.0	0.0	0.0	23.693481	193.855194	1.5	1.5	32.959999	32.959999	19.039131
2020-11-24 15:49:43	21.856585	21.0	0.0	0.0	32.320637	272.257507	1.5	1.5	32.959999	32.959999	19.044643

Figure 2: Samples of the recorded data from the building simulator.

Because the dataset is the recorded data from the simulator, it is initially sorted by the *Datetime* column. As mentioned earlier, the data is written once each minute. However, the simulator sometimes crashes, so there might be some missing time.

The *Room Temperature* indicates the current temperature inside the room, while the *Room Setpoint* is the desired temperature for the room to be at. Assemblin company has some setting rules for the *Room Setpoint*: it changes daily to 21 degrees at 8 AM and 17 degrees at 9 PM; it stays stable at 17 degrees during the weekend. *Room Setpoint* is the most critical criteria that controls the on/off of the heating/cooling system. In addition, there is another variable called *Comfort Error*. If the *Room Temperature* is in the range, the cooling and heating are off. The comfort range in the morning is ±0.5 degree and at night is ±1.0 degree of the *Room Setpoint*. Thus, when the *Room Temperature* is below the *Room Setpoint* - *Comfort Error*, the heating is on and vice versa.

The *Heating/Cooling Demand* shows the percentage of capacity of the running heating/cooling system, while the *Heating/Cooling Power* presents total power consumed for *Heating/Cooling Demand* (in W).

Outside Temperature External and *Solar Power External* are the data from FMI and also the inputs of the simulator. The *Current Outside Temperature* and *Current Solar Power Outdoor* are the simulator's variables to save the *External* values. Thus, the *Outside Temperature* columns are the same, as well as the *Solar Power* columns. Lastly, the *Supply Air Temp* is simply the temperature of the supply air.

3.1.2 Weather Forecast Data

The weather dataset is collected from FMI simultaneously with the building data; thus, it has the same recorded period as the above spreadsheet. However, instead of taping each minute, this data is updated each hour.

The spreadsheet contains 6 columns: *Date*, *Time*, *Current Temperature*, *Forecast Temperature in 1 hour*, *Forecast Temperature in 2 hour*, and *Forecast Temperature in 4 hour*. Same as the building dataset, the *Date* and *Time* are merged into *Datetime* column and set as the index. Samples of the dataset is in the figure 3.

Datetime	Current Temp	Temp in 1hour	Temp in 2hour	Temp in 4hour
2020-11-24 15:00:00	1.5	1.2	1.0	-0.1
2020-11-24 16:00:00	1.2	1.0	0.3	-0.2
2020-11-24 17:00:00	1.0	0.4	0.3	1.1
2020-11-24 18:00:00	0.4	0.3	0.8	1.4
2020-11-24 19:00:00	0.3	0.8	1.1	1.6

Figure 3: Samples of the recorded weather data from FMI.

3.2 Data Exploration

Both data sets are separately examined first. Some aspects to consider are the descriptive statistics, correlation matrix (figure 4), and time series plots. The correlation matrix shows the relationship between different components by grading from -1.0 to 1.0. Each variable has the highest correlation with itself; thus, the matrix's diagonal has all 1.0 values. The higher the positive correlation, the more likely one is in direct proportion with another. And vice versa, the lower the negative value, the correlation indicates inversely proportional.

Correlation value 0 shows that 2 variables do not relate. In figure 4, it is hard to conclude if there is a relationship between the heating/cooling demand and outside temperature. The correlation matrix shows nothing much since it plots the relationship between variables, not with the Datetime. Thus, the time series plots are more exciting.

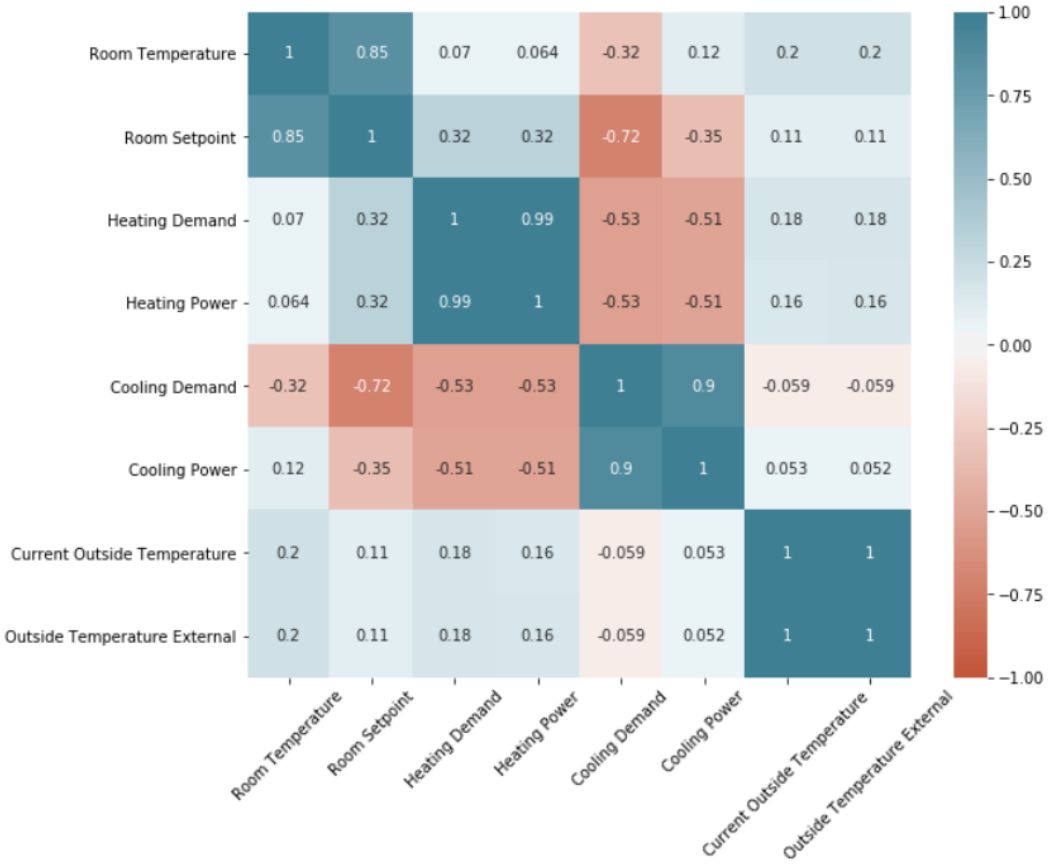

Figure 4: Correlation matrix between building data features.

The first plot (figure 5) includes *Room Temperature*, *Room Setpoint*, and *Current Outside Temperature*. One of observations is that at night, the *Room Temperature* could not get low enough to reach the *Room Setpoint*, indicating the cooling system might have worked all night long.

The second plot shown in figure 6 contains *Room Temperature*, *Room Setpoint*, and *Heating/Cooling Demand*. The *Heating/Cooling Demand* is scaled down from range [0, 100] to range [0, 10] for better visualization. Indeed, the air conditioner was running during night time. Additionally, the transitions when the *Room Setpoint* changed were rough. For example, just before 8 AM on November 25, the cooling demand was 100%. At exactly 8 AM, the *Room Setpoint* increased to 21 degrees; the heating demand was now 100%. An assumption is that if the transitions are smooth, less energy is needed.

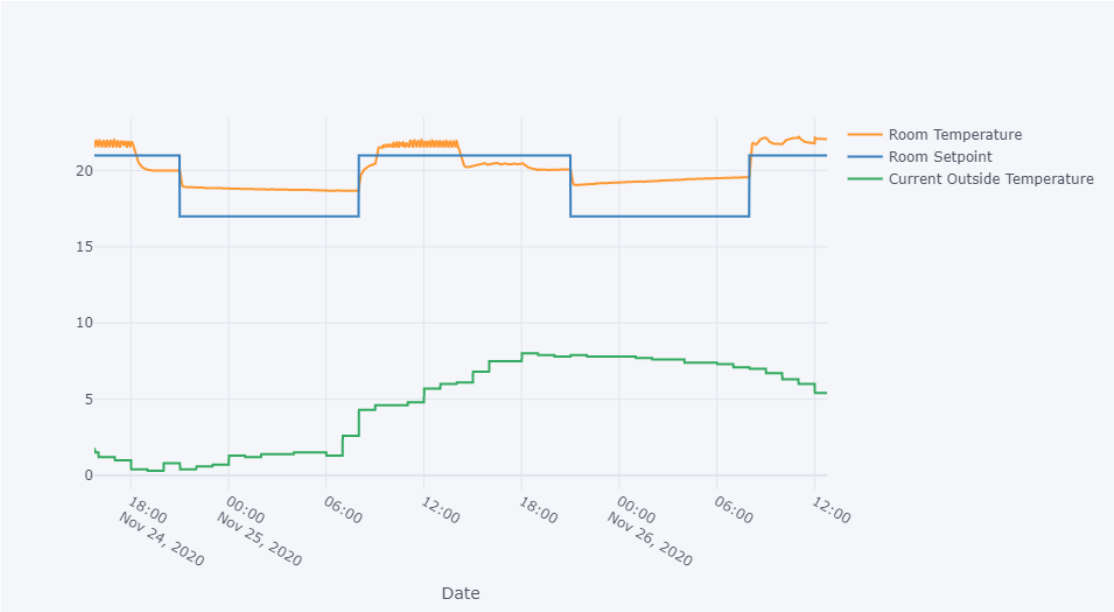

Figure 5: Time series line graph with the *Room Temperature*, *Room Setpoint*, and *Current Outside Temperature*.

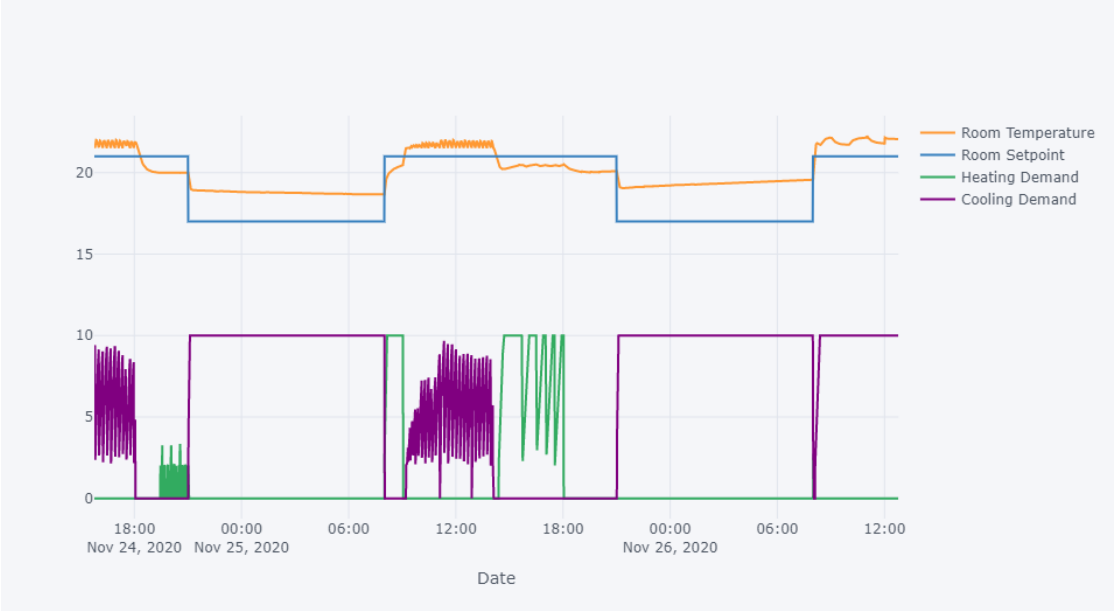

Figure 6: Time series line graph with the *Room Temperature*, *Room Setpoint*, and *Heating/Cooling Demand*.

3.3 Potential Models

3.3.1 Model Without Weather Forecasts

One straightforward option is instead dropping/increasing the *Room Setpoint* suddenly, *Room Setpoint* can be changed gradually. Simple linear formulas are applied to make this change. As in figure 7, the *Room Setpoint* shifts before and after the adjusting time

30 minutes. This setting reduces the total time that the new *Room Setpoint* violated the original comfort range. This option is not using the weather forecast data; only the *Datetime* is wanted.

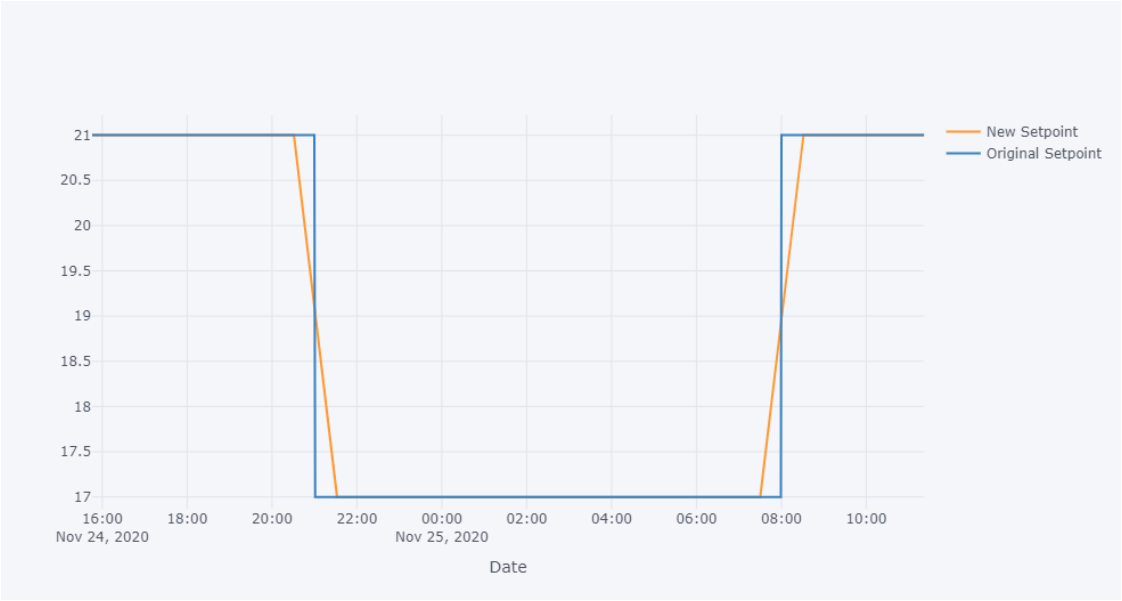

Figure 7: Time series line graph with the original and new *Room Setpoint*.

3.3.2 Temperature Based Model

The goal is to create a ML model to predict the correct *Room Setpoint* based on the weather forecast to optimize the energy consumption but still ensures the room comfort temperature range. Considering the available building and weather forecast data, it is hard to pick the right ML methods for this problem.

Assuming the problem is treated as supervised learning, the label would be the *Room Setpoint* that only equals 21 or 17. Firstly, there is no need for other variables to calculate this *Room Setpoint*. Secondly, the goal is to reduce power consumption, so producing the similar *Room Setpoint* will not do the job. Thirdly, even if models are created, even they might have good results on the evaluation metrics, they will not perform well since the closer the models to this label, the more alike the predictions equal 21 or 17. Still, the objective of supervised learning is for a model to have higher accuracy or lower error. In a word, supervised learning using the original setpoints seems not a good option.

It might be an unsupervised learning problem since there is not true/correct data of the *Room Setpoint* serving the initial goal. However, the appropriation appears unclear too.

The purpose of unsupervised learning is to categorize samples based on the similarity between them. The Assemblin problem is simply not this type.

Indeed, the Assemblin project should be handled as a supervised learning problem, but with different setpoints as the label. Correct setpoints that optimize the heating/cooling power should be the proper label. However, it is expensive and impossible to collect this kind of data. It might also need human involvement to give wise commands for the system as a part of the data collection process. Therefore, the goal changes a little bit: rather than fully optimizing the power consumption, a decrease in it is good enough. The plan is to trial and error different setpoints to search for the ones that lower the heating/cooling power, use it as ground truth, and build the ML models.

Due to the lack of time, very few settings and models got tests. The setpoint linear formula seems to be promising. Thus, it is used to build the models as the labels. From there, various supervised learning methods for a regression problem can be used. Advanced one such as Artificial Neural Network (ANN) can be trained. However, as the plan is trial and error and as ANN is expensive and takes a longer time to run, the simple linear regression is a more suitable option. The polynomial regression method is trained too, but the models are overfitting even with the degree equals 2.

The linear regression model is built using the Scikit-learn library. The building data and weather forecast data are merged. Some columns such as *Datetime*, *Heating/Cooling Demand*, *Heating/Cooling Power*, *Solar Power* are deleted. The final dataset for building models is in figure 8. It shows that the model takes 6 inputs: *Room Temperature*, *Room Original Setpoint*, *Current Outside Temperature*, and *Temperature Forecast in 1/2/4 hour(s)*. And the label for this model is the *New Setpoint*. The dataset is then standardized using Scikit-learn preprocessing library and divided into the train and test set with a ratio of 80:20 without shuffle.

	Room Temperature	Room Setpoint	Current Temp	Temp in 1hour	Temp in 2hour	Temp in 4hour	Setpoint
1106	21.827171	21.0	4.6	4.8	5.5	6.2	21.000000
940	18.672634	17.0	2.6	4.3	4.7	4.9	18.466667
1135	22.022980	21.0	4.8	5.5	6.0	6.5	21.000000
333	18.915287	17.0	0.4	0.6	0.7	0.7	17.266667
953	19.496689	21.0	4.3	4.7	4.9	5.5	19.333333

Figure 8: The dataset for the linear regression model.

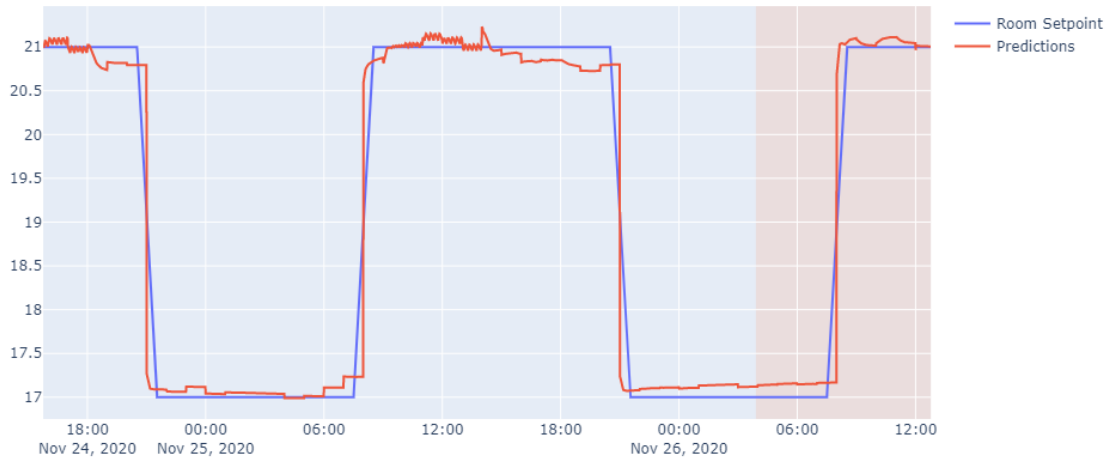

Figure 9: Time series line graph with the new *Room Setpoint* and the linear model predictions.

Figure 9 plots the label and the linear regression model predictions. The blue section is the train data, while the red one is the test data. The linear regression is very similar to the labels. The metrics show that the model is in good shape: Mean Squared Error (MSE) is approximately 0.1, and the R2 score is above 0.95 for both the train set and test set. The MSE indicates that the predictions error compared with the true label is around ±0.1 degree. The metrics are relatively the same for the train and test set; therefore, the model is not underfitting or overfitting.

4 Model Testing Results

The models, including the setpoint's linear formula and the default setting, are employed inside the thread of the control software. The simulator runs each case with the same temperature and solar power data to compare the results. The test is 21 hours long, starting from 4 PM to 1 PM the next day. After running, the spreadsheet for each model is analyzed. The tasks are to check whether the building condition is normal, the room temperature does not vary much from the original setpoint and comfort error, and the power consumption is less than the default setting.

4.1 Model Without Weather Forecasts

First to be mention is the setpoint linear formula since it is the principle for the temperature based ML model. The building seems to function normally. The *Room Temperature* did

decrease/increase gradually like expected. The total time that the heating and cooling system work is reduced compared to the default setting. However, the total amount of power is larger than the default. This might be because when the heating and cooling system was working, it mostly worked at its full productivity (figure 10 and 11).

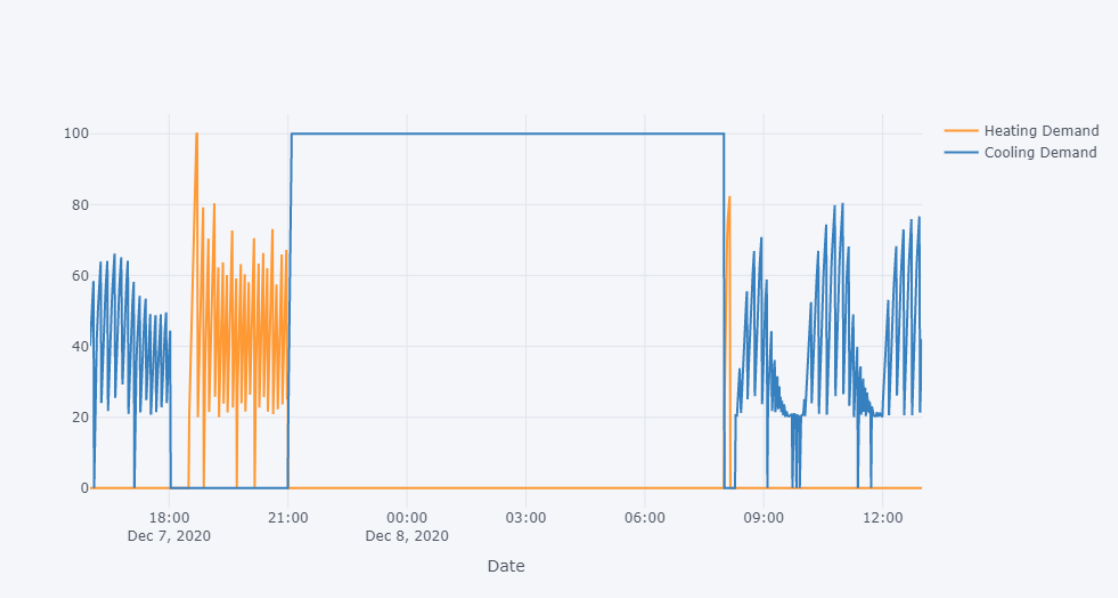

Figure 10: Time series line graph with the heating/cooling demand of the default setting.

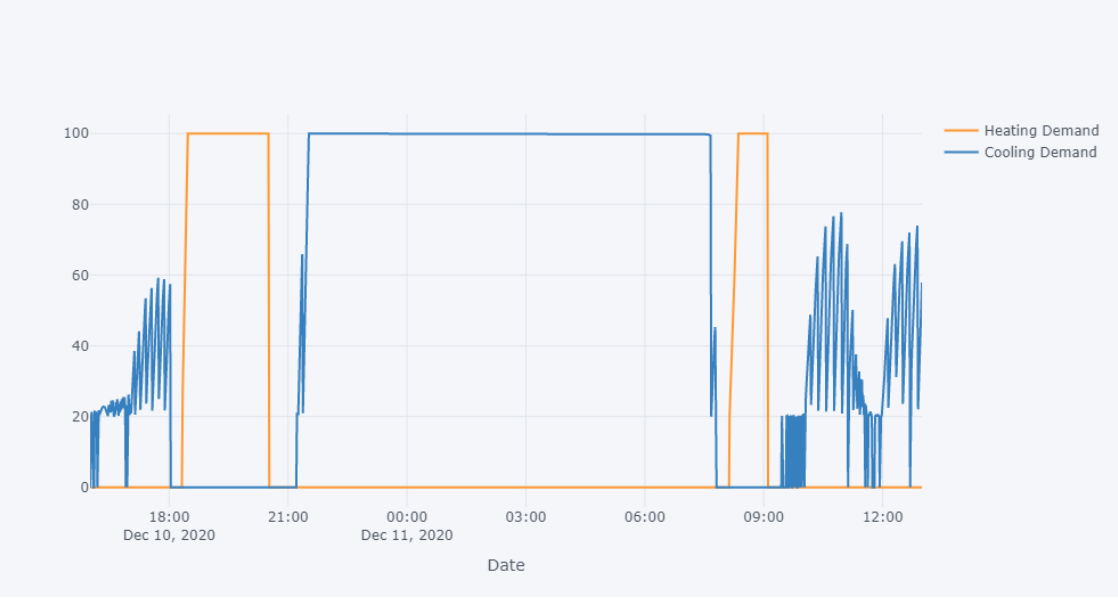

Figure 11: Time series line graph with the heating/cooling demand of the setpoint fomular setting.

4.2 Temperature Based Model

The linear regression shows nothing better than the default setting. Still, the good side is that nothing abnormal happened when the model was running. The predicted *Room*

Setpoint varies from 16.9 degrees to a little bit higher than 21 degrees, not too different from the original *Room Setpoint*. It decreased/increased instantly at the setpoint changing time (figure 12), unlike the labels. The total working time of the heater and air conditioner is longer than the default one. This results in much bigger power consumption.

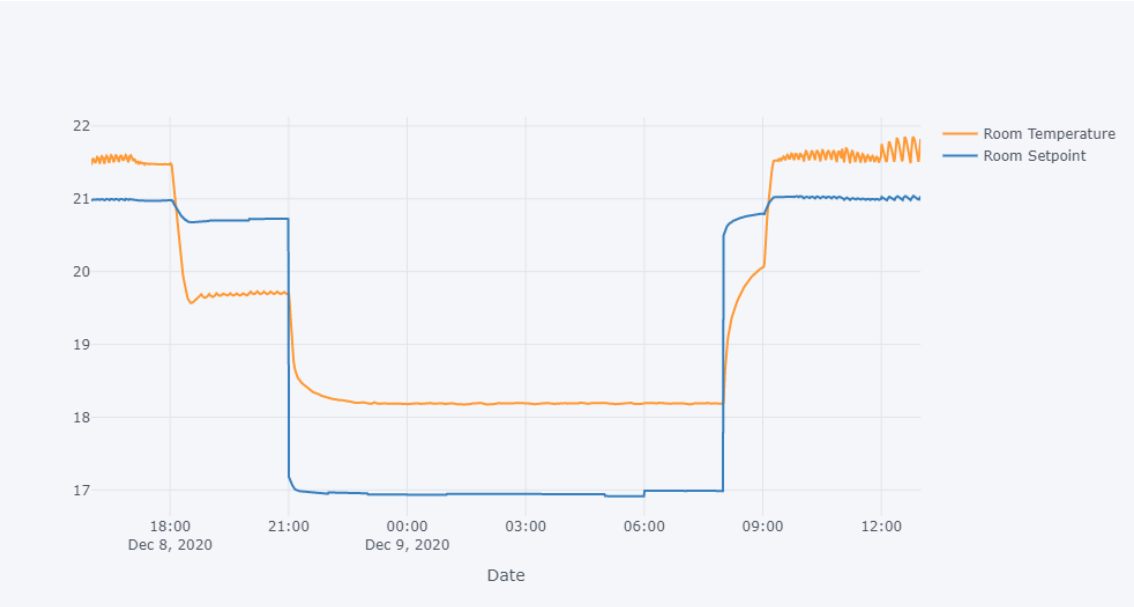

Figure 12: Time series line graph with the *Room Temperature* and the predicted *Room Setpoint* of the linear regression model.

5 Further Development

The linear regression model above did not acknowledge a *Room Setpoint* rule to keep the setpoint stays 17 degrees during the weekend. The *Datetime* column should be added to recognize this setting. Some ML methods do not accept Datetime format in the dataset. Thus, the *Datetime* column might be extracted to get, for example, date, hour, and minute columns for the algorithms to run.

The central problem of not having a good model might be in the *Room Setpoint* formula setting itself as it is the base for the ML model. Better ground truth is needed to be found in the future. This can be done by finding a better *Room Setpoint* formula or manually controlling the *Room Setpoint* that foresees the future temperature. Also, the *Supply Air Temp* should be modified to reach the desired setpoint.

With better labels, other ML methods can be considered to train the models. Time series

models such as ARIMA can be investigated. Another choice is ANN. ANN gives better results when it needs to approximate non-linearity functions.

Lastly, reinforcement learning might be an option since there are no valuable truth labels, and it is expensive to collect them. Reinforcement learning is trained and runs at the same time, not pre-trained like supervised learning. The model learns and updates itself by taking feedback from developers.

6 Conclusion

The project itself moved slowly in the beginning due to various reasons. Mostly because the problem is too hard to approach, and the objective was not clear before. Thus, the analysis process of the building data and the model testing started a little bit late. However, the process was smooth in the end. Supervisors are very supportive, friendly, and enthusiastic. I personally also received constructive comments from my Innovation Project teacher. The team stayed together and helped each other lot. We are happy and grateful since we can learn more, can make mistakes, and can grow on our expertise.

The report presents the data analysis process and the development of 2 models: without weather forecast and temperature based. The models are expected to utilize the weather data to control the building system wisely and efficiently use energy. In the end, the problem has not been solved since both 2 models did not successfully save more power.

The results are not perfect at the moment. On the IT side, especially the Data Science area of the project, more research and trial on ML is needed. More data can be collected so that there are more data to work with. There are other aspects to explore. Hopefully, the initial objective can be achieved in the very near future.

Bibliography

1 Energy efficiency in Finland - a Competive Approach. Motiva Oy; 2006. Available from: https://www.motiva.fi/files/8005/Energy_Efficiency_in_Finland_A_Competitive_Approach.pdf [cited October 18 2020].

2 Assemblin;. Available from: https://www.assemblin.com/ [cited November 29 2020].

3 Digi-Salama;. Available from: https://digisalama.metropolia.fi/ [cited November 30 2020].

4 Finnish Meteorological Institute;. Available from: https://en.ilmatieteenlaitos.fi/ [cited November 30 2020].

5 Pandas Documentation; 2020. Available from: https://pandas.pydata.org/pandas-docs/stable/index.html [cited December 05 2020].

6 Scikit-learn - Machine Learning in Python;. Available from: https://scikit-learn.org/stable/# [cited December 08 2020].