Dynamic Setpoint Control in HVAC systems to Minimize Electricity in Non-residential Buildings

Using a Multi-Variable Model and Predictive Modeling

Britt Hale (2612168)

Thesis Master Business Analytics

Master thesis Business Analytics

Dynamic Setpoint Control in HVAC systems to Minimize Electricity in Non-residential Buildings

Using a Multi-Variable Model and Predictive Modeling

Author: Britt Hale (2612168)

First Supervisor Vrije Universiteit:
Rianne de Heide

Second Supervisor Vrije Universiteit:
Sandjai Bhulai

Vrije Universiteit Amsterdam
Faculty of Science
De Boelelaan 1081a
1081 HV Amsterdam

Supervisors Heroes B.V.:
Joshua Touati
Sinit Tafla
Marc Cooper

Heroes B.V.
Data Science Department
Burgemeester Verderlaan 15F
3544 AD Utrecht

March 5, 2023

Preface

This thesis is part of the requirements for completing the Master of Science in Business Analytics at Vrije Universiteit Amsterdam. This report describes optimizing electricity usage in a non-residential building while keeping the building comfortable for its occupants. Multiple Machine Learning algorithms have been applied; the best one is chosen and applied in a simulation. This research is conducted in the Data Science department at Heroes B.V.

Acknowledgments

My sincere thanks go to all who have contributed to the completion of this thesis. A special thank you to my Heroes B.V. supervisors, Joshua Touati, Sinit Tafla, and Marc Cooper, for their helpful feedback and constructive criticism. Recognition is also extended to my supervisor from the university Rianne de Heide for her guidance and mentorship. Countless hours of work and resources were made possible by their help and the Vrije Universiteit. Combining all these efforts led to a smooth process. Over the course of this project, it was clear that I had to combine all my learned knowledge about Data Science. One key takeaway from this process has been the importance of perseverance, even when the process gets complicated. Personally and professionally, I have learned a lot throughout this process. Evidently, this research project has expanded my knowledge and skills in Data Science. Reflecting on this process, I am grateful for the experience and look forward to applying the skills gained in the next challenge.

Abstract

The heating, ventilation, and air conditioning (HVAC) system accounts for 50% of all the energy used by buildings. To protect the environment from the negative impact of greenhouse gasses and reduce the costs of energy usage in buildings, an energy-efficient HVAC system is crucial. This HVAC system is often used in buildings to manage CO2, energy, and air levels. The HVAC has different settings called setpoints. Depending on factors such as the outside weather, it is necessary to change these HVAC setpoints throughout the day. This research will investigate how to optimize this system throughout the day to decrease energy usage in a building while retaining comfortable temperature and CO2 levels.

The research is divided into three phases: pre-processing, prediction, and optimization. First, the data needs to be processed. This is done by removing outliers, dealing with missing values, and ensuring that the dataset contains consistent values (e.g. every hour). Moreover, feature engineering is done by adding temporal values. The data consists of the HVAC setpoints, external weather variables, historical data, and added temporal values.

The Random Forest, XGBoost, LightGBM, and Lasso algorithms have been applied using three methods: default settings, a grid search, and a grid search combined with Principal Component Analysis. For Random Forest, the default settings perform best. For XGBoost, LightGBM, and Lasso an extensive grid search over the most important features performed best. In all algorithms, the temperature in the upcoming hour scored an R^2 of near 1. A reason for this could be a lack of variety in the data: there is a pattern and the temperature does not differ more than ±0.5°C compared to the previous hour. All algorithms appear not to be overfitting since the training and test evaluation metrics score around the same value. For XGBoost and LightGBM the extra parameter 'subsample' was added to prevent overfitting. In all algorithms, PCA showed a decrease in performance. A reason for this could be that features have a non-linear relationship. The model that performed best was XGBoost since it has relatively the best evaluation metrics and follows the peaks in the data. Lasso did not follow the actual predictions accurately. LightGBM predicted peaks that were not in the actual data. Random Forest also predicted well on the evaluation metrics, but slightly worse than XGBoost.

A real-time HVAC system is simulated by going over the test set hour by hour. For each hour, the current levels in the building are combined with all combinations of the setpoints. For each combination, a prediction is made for the CO2, temperature, and electricity in the upcoming hour. The combination with the lowest electricity usage, while staying within comfort range is chosen. The comfort range is chosen by doing literature research and looking at the data. The optimal setpoint combination is passed on to the HVAC system, whereas the levels in the building will adjust to that. The test set contained a total of 166 days. The original test set used 7603.08 kWh, while the simulation used 6204.92 kWh. This is a decrease of 18.39%. Taking the mean price of electricity in the Netherlands in February 2023, this results in (7603.08-6204.92) * €0.69 = €964.73 saved in 166 days. When training on the training set, but doing the simulation over the whole dataset, a total of €3294.19 was saved in 552 days. The electricity usage was reduced with 17.62%. The summer months used the least electricity. Throughout the year there were peaks of saving, but also some peaks where no electricity was saved. A zoom-in in a week shows that the electricity usage in the simulation is more stable compared to the original. A reason could be that having a less fluctuating HVAC setpoint uses less electricity, compared to changing the setpoint often. Further research could focus on getting a better dataset with more information on the HVAC settings, and how the levels in a room change if the setpoints change. Extra variables could be added such as occupancy, air quality, humidity, and information on the windows (open/closed). Moreover, multiple types of buildings can be added to the research.

Contents

1 Introduction

This section will explain the context of the research and the company where the research is conducted will be elaborated. Moreover, the problem statement and the research questions will be explained. Lastly, the thesis outline will be provided.

1.1 Context

In the past 130 years, the earth's average temperature has risen by 1 degree and the sea level by 20 centimeters. In the Netherlands, the temperature even rose by 1.7 degrees. The earth will get warmer if greenhouse gas emissions continue to grow. This will have dangerous effects on people, nature, and the environment. The consequence could be that the season for growing and blooming already begins earlier than before. Animals' and plants' habitats change due to factors like rising temperatures. Consequently, an increasing number of animal and plant species are going extinct. Moreover, extreme weather will occur more often, such as heavy rainfall. Because a large portion of the Netherlands is below sea level, it is extra vulnerable to flooding. To take preventive measurements, the dikes need to be higher, which is highly expensive to build. Lastly, the effects of climate change could be harmful to human health. Air pollution and allergies will increase. A shortage of food and drink can occur, which could lead to people being required to leave their city or nation. Subsequently, this can have negative consequences for the Dutch global economy.

Numerous factors, including the rise of greenhouse gases in the atmosphere, contribute to climate change. The leading cause of this is the release of greenhouse gases like CO2 and methane. The earth's temperature rises due to greenhouse gases' ability to keep heat in the atmosphere. The earth would be freezing without greenhouse gases, but it will become too hot if there are too many of them [1]. The Netherlands must emit 55% fewer greenhouse gases in 2030 compared to the number of emissions emitted in 1990. The target is even 60%. The European Union member states have agreed that they will be climate neutral by 2050. This means that greenhouse gas emissions in 2050 will not exceed what has been recorded, so the net emissions will be zero [2].

The use of energy in buildings contributes significantly to greenhouse gas emissions. 7.85 Gt of carbon dioxide (CO2) emissions from energy use in the building industry were the cause of 33% of all energy-related emissions worldwide in 2002. Global warming will directly impact buildings. Although heating energy usage will decrease, the demand for cooling will increase [3]. According to numerous studies, heating, ventilation, and air conditioning (HVAC) account for about 50% of all the energy used by buildings [4]. Finding a way to bring this number down is crucial to slow down global warming. Besides this important aspect, there is also an economic aspect for companies. Due to the government's efforts to achieve climate neutrality, businesses are now aware that they must undertake structural adjustments to lower their CO2 emissions. When the electricity usage of companies is decreased, the costs will also decrease. This topic is a current issue. Many economies were still trying to recover from the effects of the pandemic. The Russian invasion of Ukraine and the following war worsened the energy crisis. Due to this invasion, a blockade and sanctions were imposed on Russia. Since the European Union imports a large amount of energy from Russia (about 40% of natural gas, 25% of oil, and 50% of coal imported into the EU in 2019), this war has increased the region's energy pricing crisis. The World Bank predicts energy costs to rise by more than 50% in 2022 [5]. Moreover, carbon pricing might become a reality in the future. This means that companies that pollute more should pay more for their carbon emissions [6].

Since the HVAC system accounts for around half of the energy usage in buildings, it is a primary concern to take this down. The goal of this system is to maintain a comfortable temperature and indoor air quality for building occupants, regardless of the outdoor weather. This is done by using the system their heating, ventilating, and air-conditioning units. There are two types of indoor comfort: thermal and indoor air quality. When it is cold outside, heating devices must be turned on to provide warmth and a comfortable indoor temperature. Similarly, when it is warm outside, the heat should decrease. To maintain stable thermal comfort, the heat balance that determines the indoor temperature needs to be controlled by the HVAC system [7]. There are different factors to take into account. Firstly, external weather conditions, usage patterns, and building characteristics - many of which vary stochastically - all significantly impact energy savings. Secondly, preferences for thermal comfort vary according to the occupants' health, ages, genders, and living spaces. Many present-day HVAC systems use local control (i.e., two-position on and off) or proportional, integral, and derivative (PID) control. However, they have limited potential regarding energy savings and thermal comfort. It is primarily intended to ensure that indoor climates fulfill the expected setpoints, such as indoor temperature, relative humidity, and CO2 concentration. Creating intelligent HVAC control using prediction methods to provide comfortable indoor climates while using less energy usage has obtained promising results [8].

1.2 Problem Statement

As mentioned before, 33% of the worldwide energy-related emissions in 2002 were from energy use in the building industry. The heating, ventilation, and air conditioning (HVAC) system accounts for 50% of all the energy used by buildings. A building's ventilation system impacts (either positively or adversely) the levels of indoor pollutants as well as the temperatures and relative humidity, which affect how comfortable it is inside. Proper ventilation brings in fresh air from the outside, lowers indoor pollutant levels, improves indoor air quality, and subsequently increases the occupants' productivity. Inadequate ventilation and poor air quality could even harm students' performance, capacity for learning, and productivity. In extreme situations, when pollutant concentrations reach specific thresholds, it can cause throat and nose conditions that cause absenteeism [9]. Moreover, consumers want to prevent paying unnecessary electricity costs. These reasons combined with protecting the environment from the negative impact of greenhouse gasses, an energy-efficient HVAC system is crucial.

There are different factors to take into account when optimizing energy usage. For example, the outside weather could influence the thermal comfort in the building. Moreover, the temperature, CO2 levels, and electricity usage vary throughout the day. The HVAC has different settings called setpoints (e.g. the temperature setpoint can be set to 20 degrees). Depending on factors such as the outside weather, it is necessary to change these HVAC setpoints throughout the day. This research will investigate how to optimize this system throughout the day to decrease energy usage in a building while maintaining thermal comfort.

1.3 Research goal

The goal of this research will be to deliver a report which goes into detail about HVAC optimization based on Machine Learning. This report will elaborate on which algorithm performs best, how to get the best results, and the possibilities of further research on this topic. This research can be regarded as a success when clear answers can be found in the results around my research questions.

1.4 Research question

To summarize, the research aims to answer the following questions:

> "How can the electricity usage of HVAC systems in non-residential buildings be decreased compared to its current performance while maintaining a comfortable indoor climate? Moreover, which optimization algorithm performs best to optimize the HVAC system's electricity usage?"

1.5 Organization

This research is conducted as a collaboration between The Vrije Universiteit and Heroes B.V. Heroes B.V. is a Dutch company that is an expert in the field of Data, AI, and IoT. They have around 20 employees spread over different departments. Sustainability is one of their core values. To live up to this fundamental, they ensure they take on clients to make the world a better place. For example, to make European cities more sustainable, they took part in a European procurement process called AI4Cities. This research will help them to extend their solution for this competition.

1.6 Thesis outline

This report is structured as follows: first, the HVAC system will be shortly explained and literature research is done to see what similar research has already been performed. In Section 3, the methodology used for this research is explained. Section 4 explains which data was available for this research and elaborates on the data quality. Subsequently, Section 5 shows the results of this research. The conclusion of this research is drawn in Section 6. The limitations and recommendations for further research are also included in the conclusion.

2 Literature

This section will cover the HVAC system, how comfort is measured, and which related research has already been performed.

2.1 What is the Heating, ventilation, and air conditioning system?

Buildings can be heated, cooled, and ventilated using mechanical equipment called a heating, ventilation, and air conditioning (HVAC) system. The air is distributed by, for example, fans or blowers that can be controlled automatically or manually by the users of the building. These settings are called setpoints. The heating part is used to raise or lower the temperature of water/air. The ventilation part provides the volume of fresh outside air. Lastly, air conditioning keeps the building cool by eliminating heat from the air. Temperature, humidity, fresh outside air supply for ventilation, filtration of airborne particles, and air movement are all regulated by this system.

HVAC systems are multi-input multi-output systems. The heating and cooling components of this system will be elaborated briefly. The heating element can be employed in various ways. It can be used by directly emitting radiation into the area and/or allowing free convection. Moreover, transferring electricity/heated water to devices to heat the air directly or with forced circulation is also possible. The most popular form of warming up a space is to place warm air into a room and diffuse it there to mix it with the cooler air already present. An equal air volume is removed simultaneously to transport away some potential pollutants. The cooling element comes next. Most modern structures cool themselves to make their occupants more comfortable, especially in warm weather. To balance out the energy that the space is gaining, cooling is used to flow air into that space. The energy in the room increases most frequently because of sunlight, sources in the building (e.g. people, lighting, or machinery), or from its warmer surroundings [10].

2.2 How to measure comfort in a building?

Comfort levels are subjective and differ per person. This report is focused on CO2, temperature levels, and energy usage. The latter aims to be as low as possible to save costs and emit as few as possible greenhouse gasses. For the first two, there are some guidelines to use.

The indoor CO2 levels are mainly influenced by the occupant's breathing process, the total number of occupants, the length of the occupation period, the size of the room, and outdoor CO2 levels. When CO2 levels do not match the requirements, it indicates insufficient ventilation. There are different opinions on the optimal levels of CO2 in part per million (ppm). According to the World Health Organization (WHO) and American Society of Heating Ventilation and Air-Conditioning Engineers (ASHRAE) Standard 62.1–2016, indoor CO2 levels should be below 1000 ppm to prevent the harmful effects of poor air quality. However, the Representatives of the European Heating and Ventilation Associations (REHVA) Guidebook requires CO2 levels during full room occupation below 1500 ppm [9].

During the mid-20th century, the number of lightweight buildings increased. This caused buildings to overheat during summer, even in nations like the United Kingdom, which are not known for their hot summers. UK Health and Safety Executive (HSE) states that questions about how to prevent workplace overheating during summer are the most common questions they receive from businesses. According to the UK Health and Safety Executive guidance publication, Thermal Comfort in the Workplace, an acceptable temperature range for most people in the UK is between 13°C and 30°C. The lower temperature is meant for more rough work, and the higher temperature is for physically inactive work [11]. According to the Centers for Disease Control and Prevention, thermal comfort is influenced by one's body temperature, physiological adjustments, and heat generation from metabolism. Temperature, humidity, air movement, and clothing affect how much heat is transferred from the body to the environment. The impression of thermal comfort is influenced by one's body temperature, heat generation from metabolism, heat transfer to the surroundings, and physiological changes. Temperature, humidity, air movement, individual activities, and clothing affect how much heat is transferred from the body to the atmosphere. ASHRAE advises the temperature to vary from around 20.3°C to 23.9°C during winter and from 23.9°C to 26.9°C during summer. These numbers assume slow air movement and 50 % indoor relative humidity. The choice of clothes greatly influences the temperature fluctuations between the seasons [12].

2.3 Related work

This section describes the work of already conducted research related to this topic.

2.3.1 Optimizing the HVAC system

Because of the high amount of electricity usage by HVAC systems, related research has been conducted. Selamat et al. state that there are two general goals: single-objective and multi-objective optimization. The first one focuses on minimizing electricity usage while keeping a minimum comfort level. The latter optimizes electricity usage while maximizing comfort level.

The authors state that there are three approaches to reaching these goals. The first one is operational parameters optimization, including electricity reduction and prediction of the HVAC system. The parameters of the HVAC system can be optimized to prevent unnecessary electricity usage. The predictive approach in HVAC systems can be implemented in real-time applications. The parameters of the HVAC system have a relationship with the output parameters, such as temperature, energy consumption, and humidity in a building. These values can be predicted and anticipated on that. An example to model the single-objective approach is to find the relationship between the desired output (y), the present and past values of the controllable parameters (x), and the uncontrolled parameters (v). Examples of these x variables in the HVAC system are 'supply air static setpoint', and 'supply air temperature setpoint'. An example of the uncontrollable parameter v is the outside air temperature. The multi-objective is similar to this, but it also takes into account e.g. the indoor air quality (thermal comfort). This approach makes a model that minimizes electricity usage, violation of temperature, violation of humidity, and violation of CO2 concentration. This technique allows for a 12.4% reduction in electricity usage without significantly breaking the thermal restrictions. A disadvantage of this approach is that to predict accurately, the HVAC system should be inspected and understood well. This could take a large amount of time. The second method is focused on reducing unnecessary energy waste. It takes into account the time that is needed to adjust the settings in the HVAC system. An example is that when the 'supply air temperature' setpoint is adjusted, it will take time to reach the desired level in the building. There can be an overshoot of supply air temperature, which means that there has been unnecessary electricity usage. By reducing the tracking inaccuracy of the system reaction time, this can be minimized. The third approach uses the design of the building to minimize electricity usage. They do this by choosing the correct type of HVAC system and the right construction materials [13].

Considering the background of this research is Data Science, the first approach is the most topic related. The combination of hourly prediction using Machine Learning techniques, and optimization based on multiple variables (electricity and thermal comfort), while also using hourly dynamic setpoint adjustment has not been studied extensively. Some studies use a fixed setpoint for the HVAC system, an on/off approach, or make the system rely on occupancy schedules. Having real-time proactive adjustable HVAC setpoints could decrease electricity usage in a building, while staying within comfort levels.

2.3.2 Random Forest

Wang et al. use the Random Forest (RF) algorithm to predict the hourly building energy usage of two educational buildings in Florida. Random forest is an ensemble prediction model which combines the predictions from two or more models. It is made up of a variety of distinct regression trees called Classification and Regression Trees (CART). CART is a set of questions that splits the learning sample into smaller parts. CART asks only yes/no questions and searches for all possible variables/values to find the best split [14]. The recursive partitioning method used for tree construction in RF is the same as that used for tree development in CART. However, CART is regarded as an unstable learner because even a minor change in the training data could alter the choice of the initial cutpoint, which could change the entire tree structure. By making predictions using multiple trees instead of just one, RF solves the instability problem with CART. CART is an unbiased predictor but provides the correct prediction on average. Thus, combining these trees of high diversity would make the model stable.

Figure 1 shows the steps of this algorithm. The training data is given as input to the RF, which is first randomly sampled with replacement to create multiple new training data sets. These subsets have the same size as the initial one, but sampling with replacements leads to some repeated data points. Each subset is given to a different decision tree, each given a specific outcome. Since this is a regression problem, the average of all outcomes of the trees is taken. This combination of all these trees is the outcome.

Wang et al. state that there are three user-defined parameters. The first one is the node size, which is the maximum depth of each tree. This decides when to stop the tree from splitting. They set this value to 5 because past studies have shown that this is a generally recommended value for regression problems. The second parameter is the number of produced trees in the forest. This number should not be too high due to computation time. It should also not be too low, since this could lead to decreased accuracy. They found that 300 was optimal. The last parameter is the amount of randomly selected features to grow the tree. It introduces randomness by randomly selecting an x number of features from the input and choosing the optimal split. The authors tried all options and used k-fold cross-validation to verify the outcome. This divides the data into k subsets, which makes it possible to validate the RF k times. Each of the k subsets is used one time for validation. They have 11 variables and used six as optimal input for this parameter.

Their input consists of one calendar year of meteorological data, occupancy, and time-related data (i.e., time of day). They compare RF with Regression Tree (RT) and Support Vector Regression (SVR) algorithms. RF outperformed both by 14-25% and 5-5.5% respectively. Thus, RF was the best prediction model [15].

2.3.3 XGBoost

Bassi et al. state that the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) held a Kaggle competition ‘Great Energy Predictor III’ where innovative work on applying Machine Learning models for predicting building energy usage was applied. The top solutions all utilized various ensembles with combinations of "Light Gradient Boosting Machine”

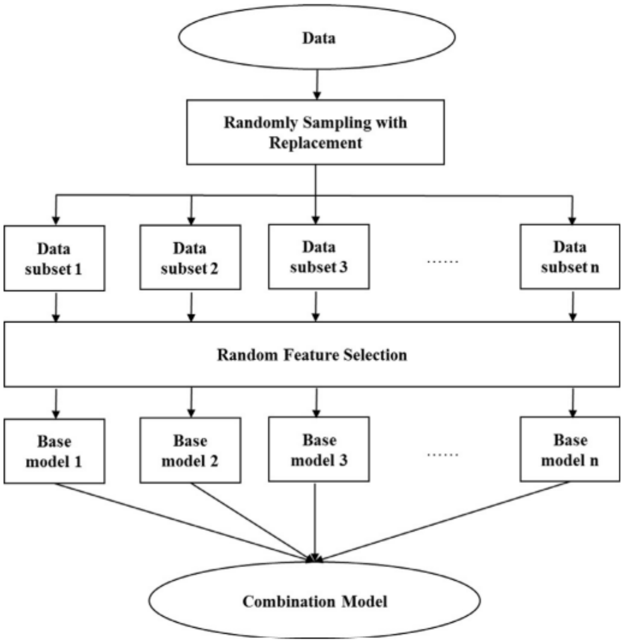


Figure 1: Model development procedure of RF [15].

(LightGBM), "Categorical Boosting" (CatBoost), and "Extreme Gradient Boosting" (XGBoost). To the best of the authors' knowledge, no thorough evaluation of these models' ability to estimate building energy usage has been done. The authors focused on these three algorithms. The two that scored best, XGBoost and LightGBM, will be elaborated. They aim to forecast building energy consumption by using historical building energy consumption data.

Random Forest and Gradient Boosting are ensemble methods since they combine outputs from individual trees. The main difference is that RF uses trees in parallel and independently of each other. Gradient Boosting builds a new tree based on an already-built one. XGBoost and LightGBM are both gradient boosting algorithms. Therefore, gradient boosting in general will first be explained. Algorithm 1 shows the pseudocode of Gradient Boosting. First, a tree is initialized and its loss function is computed. The optimal value of γ is searched to minimize this loss function. For the new tree created, the pseudo residuals (i.e., the difference between the actual value and predicted value) are calculated by taking the derivative of the loss function w.r.t. the previous prediction. Now, the tree is trained, and a terminal node R_{jm} is created. Here, J is the total number of leaves. After that, γ_{jm} is searched for that minimizes the loss function on each terminal node j. Lastly, the prediction of the combined model F_m is updated. This is repeated for a specified number of trees M.

Although gradient boosting generally works well for regression and classification problems, it is sensitive to overfitting. In particular, if a model fully matches the pseudo residuals the next iteration decreases the model by the value of these residuals, therefore producing pseudo residuals that are zero. This would terminate the gradient boosting process before optimization can be done. Some hyperparameters of a boosting algorithm are:

- The learning rate v, which is a value between $0 < v \leq 1$. It controls the size of each step during gradient descent. Mostly, $v \leq 0.1$ is optimal to reduce the generalization error. It

Algorithm 1 Gradient Boosting

```
Initialize: $F_0(\mathbf{x}) = \arg\min_\gamma \sum_{i=1}^{N} L\left(y_i, \gamma\right)$
for $m = 1$ to M do
    Compute pseudo residuals: $F_{m-1}$: $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for $i = 1, \ldots, n$
    Train tree and create terminal node $R_{jm}$ for $j = 1, \ldots, J_m$.
    $\gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{jm}} L\left(y_i, F_{m-1}(x_i) + \gamma\right)$ for $j = 1, \ldots J_m$
    Update the model: $F_m(x) = F_{m-1}(x) + \upsilon \sum_{j=1}^{J_m} \gamma_{jm} 1\left(x \in R_{jm}\right)$
Output: $F_m(x)$
```

should not be too high to avoid exploring too far in a direction. It also should not be too low since it might take too long to converge.

- The maximum depth of decision trees is mostly between 3-5.
- The subsample rate. Usually, random subsampling without replacement is employed. Due to the randomness introduced by this, the base learner estimates' variance has increased.
- The maximum number of features to consider is similar to the RF methods.
- The minimum number (N) of samples to split an internal node. Due to this, trees can only grow to a maximum depth of training data size $-N$ [16].

XGBoost (Extreme Gradient Boosting) is a gradient boosting algorithm for scalability. In literature, the XGBoost model has been highly effective. In 11 out of the 15 articles featuring XGBoost that were reviewed by Bassi et al., XGBoost outperformed competing models. XGBoost can perform parallel processing, which could be the cause of its effectiveness. The data is sorted in a block structure before training. It then uses the structure in subsequent iterations. This saves time and increases efficiency by significantly lowering computation. Another characteristic of XGBoost is to choose and store important features. External feature selection techniques are not always required. It could occasionally even worsen XGBoost's performance. Moreover, it uses level-wise (horizontal) tree growth.

XGBoost uses a variety of techniques to increase effectiveness. Since sorting data is often the most time-consuming step when working with trees, the algorithm stores data in blocks that are saved in compressed sparse columns (CSC) format. Each column is sorted by feature. Only at the start of the procedure sorting takes place. To determine the best split at each node, a split search algorithm executes a linear scan to gather statistics of all nodes. Moreover, XGBoost can automatically manage missing values, which is useful when dealing with sparse datasets.

Bassi et al. used a dataset of an office building in Chicago, containing four years of data. They used time-related data (i.e., day of the week), weather data, and electricity HVAC, heating gas, and electricity. They trained the XGBoost model first by using the XGBoost initial parameters. Using a grid search, they optimized the values of depth, learning rate, and the number of iterations. Using 5-fold cross-validation resulted in an XGBoost result of $R^2 = 0.9819$ with a variance of 0.0039 [16].

2.3.4 LightGBM

LightGBM contains numerous randomization methods, including column randomization and bootstrap subsampling, as well as an extensive range of learning hyperparameters. LightGBM introduces two new techniques compared to XGBoost to speed up the training process:

1. Gradient-based one-sided sampling (GOSS). This is used to filter data instances to find a split. LightGBM randomly discards data instances with smaller gradients during down-sampling since they do not contribute as much to decisions with bigger gradients.

2. Exclusive Feature Bundling (EFB). The algorithm combines features that are mutually exclusive together. The difficulty of organizing the dataset into a histogram at the start of the process will be significantly reduced since there are fewer features (because they are combined). This makes the complexity go from $\mathcal{O}$(#data * #feature) to $\mathcal{O}$(#data * #combinations) [16].

Another difference with XGBoost is that instead of level-wise (horizontal) tree growth, LightGBM uses leaf-wise (vertical) tree growth. Found benefits of using LightGBM are increased efficiency, faster training speeds, lower memory usage, greater accuracy, and the capacity to handle big amounts of data compared to other decision tree-based models.

As stated before, Bassi et al. used 5-fold cross-validation, resulting in a LightGBM result of R^2 = 0.9864 with a variance of 0.0040 [16].

2.3.5 Lasso regression

Now that Random Forest and Gradient Boosting methods are elaborated, one last regression algorithm will be elaborated. This will provide an overview of how different models react to the data used in this research. Lasso (Least Absolute Shrinkage and Selection Operator) regression is a type of linear regression that uses L1 regularization. A couple of benefits are that it can perform feature selection, handle multicollinearity, and reduce overfitting by decreasing the complexity of the model.

Multicollinearity happens when multiple predictors are highly correlated with each other. It can also occur when multiple variables are dependent on one other. When multicollinearity occurs, it is hard to find estimates of individual coefficients for the target variables, which results in incorrect conclusions about the relationship between the input and target variables. Therefore, multicollinearity in the data must be eliminated.

The standard multiple linear regression model is shown in Equation 1. Here, y is the target vector variable, x is a matrix of input variables, β a vector of unknown constants, and ε a vector of random errors that is Normally and Independently Distributed $(0, \sigma^2)$. When multicollinearity would be present in the data, it would inflate the variances of the parameter estimates, which leads to the significance of individual predictor variables. It can give vital problems when estimating and interpreting β. Moreover, multicollinearity could lead to overfitting.

To combat these problems, regularization will be used. This is designed to help generalize models with complex relationships. This method adds a penalty to model parameters (except for the intercept). This will enable the model to generalize the data instead of overfitting it.

$$y = x\beta + \varepsilon \tag{1}$$

The equation used in Lasso is shown in Equation 2. The first part is the residual sum of squares, which is the sum of square errors between the predicted and actual values. Added to this is the sum of the absolute value of coefficients. λ=0 would lead to ordinary least squares estimations, while a large λ would make the coefficients approach zero and could lead to underfitting. It aims to shrink the coefficient of less essential features to zero, which would eliminate the features. Thus, it can perform feature selection and by making the model less complex, it also prevents overfitting [17].

$$\sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{p} X_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} \left|\beta_j\right| \tag{2}$$

3 Methodology

This section will describe the plan of action for the research. Moreover, it describes the steps that need to be taken to answer the research question.

3.1 Research action plan

This research will simulate a building it's indoor comfort over a period of time while reducing electricity usage and remaining within comfort levels. Figure 2 shows the general outline. The building contains a sensor, which collects data on e.g. the CO2, temperature, and electricity levels for every hour. These levels are used to predict the CO2, temperature, and electricity usage for the next hour. A wide range of setpoints in combination with the current levels in the building is used for prediction. The setpoint that has the lowest electricity usage in the upcoming hour while remaining within comfort levels is chosen and sent to the HVAC system. The comfort levels depend on the temperature and CO2 levels within the building. The HVAC system will adjust its settings to optimize electricity usage in the next hour.

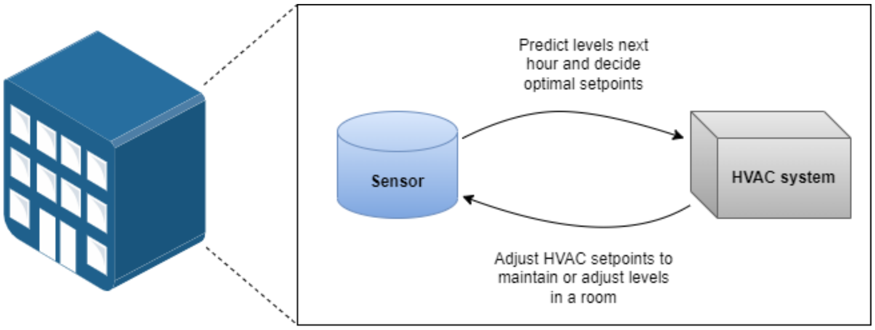


Figure 2: General outline of this research.

This research can be split into 3 phases:

1. Pre-processing
2. Prediction
3. Optimization

First, pre-processing the incoming data from the sensor needs to be done. The data quality needs to be ensured to enhance performance. For example, the data should be checked for consistency, missing data, and outliers. Secondly, the prediction phase is executed. Multiple algorithms are analyzed to find the most accurate method to predict the temperature, CO2, and electricity levels for the upcoming hour. The last phase will combine these 3 steps. A simulation is used to optimize electricity while staying within comfort levels. Each hour, the current levels in combination with a range of combinations of setpoints are used to predict the CO2, temperature, and electricity levels for the next hour. The best algorithm from phase 2 is used for this. The optimal setpoints are chosen and sent to the HVAC system. The following sections will elaborate on these phases.

3.2 Phase 1 and 2

This section will describe the steps taken in this research to execute phases 1 and 2.

3.2.1 Experimental set-up

Figure 3 shows the procedure for phases 1 and 2. First, the dataset is pre-processed by removing outliers and dealing with missing values. Moreover, it is ensured that the data has collected consistent measurements (e.g., every 10 minutes or hour). Secondly, feature engineering is done by adding temporal values. After this, the data is split chronologically into a training and test set for each model. Thus, there are three different models: CO2, temperature, and electricity for the upcoming hour. The first 70% of the data is used as training data and the latter 30% as the test set. This is done to prevent data leakage, which means that the data used to train the algorithm contains information that the model is trying to predict. Thus, it has an unfair head start. This could lead to unreliable and bad prediction outcomes. For example, if the temperature of 23-11-2022 at 11.00 would be predicted in the test set, but the values of 10.00 and 12.00 are known from training, the prediction would not be valid.

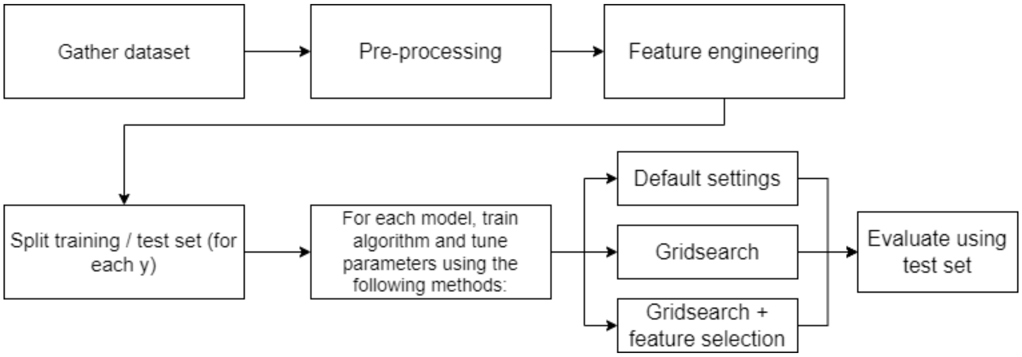

Gather dataset
Pre-processing
Feature engineering
Split training / test set (for each y)
For each model, train algorithm and tune parameters using the following methods:
Default settings
Gridsearch
Gridsearch + feature selection
Evaluate using test set

Figure 3: Procedure of executing phases 1 and 2.

When the data is prepared and split into a training and test set, Random Forest, XGBoost, LightGBM, and Lasso Regression will be performed using three different methods:

1. With the algorithm's default settings.
2. By doing a grid search to find the optimal parameters.
3. By doing a grid search with feature selection.

The second and third methods use a grid search. This is a hyperparameter tuning technique to try different combinations of parameters. It combines all combinations of an input grid and evaluates the performance with R^2. It returns the optimal parameter combination that gives the highest R^2. This metric measures how well the model fits the data, the higher the better. The data is a time series, therefore the grid search uses a time series split. The method is visualized in Figure 4. This method is similar to k-fold cross-validation, but instead of randomly splitting

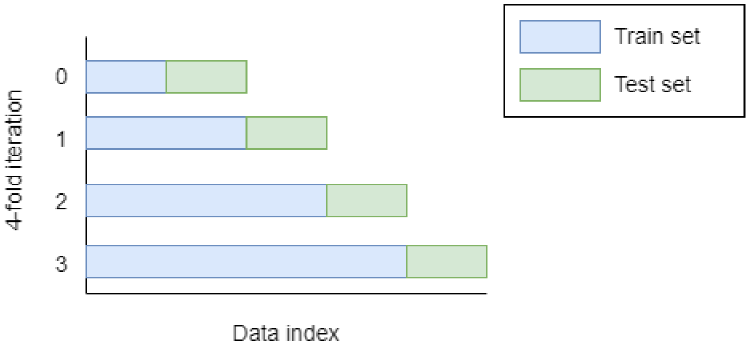


Figure 4: Global outline of k-fold time series split.

the data, it is now split based on time. This prevents the model from seeing future data during training.

The third method uses feature selection to enhance performance. It selects the most important variables and removes redundant or irrelevant features. It can be done in multiple ways. This research will use Principal Component Analysis to reduce dimensionality. Lastly, using the test set, predictions are made and evaluated. The following sections will elaborate on this process.

3.2.2 Pre-processing

Outlier removal

The first step is to remove outliers. Inconsistent or incorrectly identified models can result from outliers in the training data. In statistical analysis, outliers in the data can significantly affect the estimation of sample data's mean and/or standard deviation. It can even lead to unreliable or wrongly identified models. This can result in either over or under-estimated values. Therefore, outliers are unwanted in a dataset. A data point that significantly differs from other data points within a particular dataset is referred to as an outlier. To identify and remove outliers, the interquartile range (IQR) method will be used. It measures the variability and is based on dividing the dataset into quartiles. An observation outside the range is what IQR refers to as an outlier, as shown in Equation 3. The constant k can be set to 1.5, as found in the literature [18].

$$\mathrm{IQR} = [Q1 - k(Q3 - Q1), Q3 + k(Q3 - Q1)] \tag{3}$$

Dealing with missing values

The time-series data gathered by the sensors could be inconsistent due to sensor failures and network disconnections. This produces noisy data with missing values that restricts performance analysis. Furthermore, the length of the data gap or the number of consecutive missing values could be up to several weeks because it takes time to calibrate, reinstall, and recover a sensor. Moreover, values could also be missing if it was labeled as an outlier and thus removed while preprocessing the data. Imputation, which replaces these missing values with the expected ones, is an effective approach. When imputing up to 8 consecutive values, linear interpolation works best. The K-nearest neighbors (KNN) approach works better for wider gaps of up to 48 consecutive missing data. For even larger gaps, computationally intensive methods, such as matrix factorization should be used [19]. The latter method is not preferred and should therefore be prevented by selecting a period with less than 48 consecutive data points.

Linear interpolation builds a curve that best fits a set of data points using first-degree polynomials. To create new data points or estimate the missing data, only two data points are needed. However, as the duration of continuously missing data lengthens the estimation quality declines. Equation 4 shows the linear interpolation method. Here, A_k represents the missing data point. A_n is the last data point before the missing data at time K_n. The next data point after the missing value is A_{n+1} at time K_{n+1}. As a result, the missing value A_k can be estimated by the equation.

$$\frac{A_n - A_k}{A_n - A_{n+i}} = \frac{K_n - K_k}{K_n - K_{n+i}} \tag{4}$$

KNN is a supervised learning method for classification and regression imputation. First, the distance between the empty value and the rest of the data is calculated. This is typically done by using the Euclidean distance shown in Equation 5. The square root is taken of the sum of the squared differences between the empty value x and another (non-empty) value in the dataset y. n is the number of features in the dataset. Secondly, the closest k data points are selected based on the distance. Lastly, the empty value is imputed by taking the average of these closest data points. This is repeated for every empty value in the dataset [20]. The true value of these empty values is not known, therefore the accuracy cannot be calculated. Here, k is set to the default value of five.

$$\text{KNN} = \sqrt{\sum_{i=1}^{n} \left(x_i - y_i\right)^2} \tag{5}$$

3.2.3 Feature engineering

Correlation

The features can be divided into X (e.g. the HVAC setpoints that can be adjusted) and Y (the variables that will be predicted). Literature research indicates a strong correlation between the time of day and building occupants' hourly, daily, and weekly working patterns. Therefore, several temporal values should be added. For commercial buildings, the usage pattern is regular. For example, the usage pattern of a Sunday is similar to other Sundays [15, 21]. All the features that will be added to X are:

- Time of day (0-23), day of the week (Mon=0, Sun=6), month (1-12), and season (1-4) of all timestamps.
- Electricity, temperature, and CO2 value of the previous hour, the value 24 hours and 48 hours before.
- The difference in electricity usage, temperature, and CO2 value compared to the previous hour (Δ).
- The difference in °C between the external and inside temperature.

Thus, the X features contain the HVAC setpoints, the external weather variables, historical data, and these added features. To make predictions on the data, three extra columns are added: the value of electricity, temperature, and CO2 in the upcoming hour compared to the previous hour ($t+1$). These are the Y variables.

To decide which variables to select, the linear correlation between two variables X and Y is taken into account. The Pearson Correlation Coefficient (PCC), shown in Equation 6, is used for this. Here, COV(X, Y) is the covariance between the two variables. σ_X and σ_Y are the deviations of X and Y. The PCC ranges from -1 to +1. -1 means that X and Y have a negative linear correlation, and 1 means that they are positively linearly correlated. 0 means that X and Y are not linearly correlated. When $\rho_{X,Y} \geq 0.6$, it can be said that X and Y have a strong linear correlation. When $\rho_{X,Y} \geq 0.8$, there is an extremely strong linear correlation [22].

$$\rho_{X,Y} = \frac{\mathrm{COV}(X, Y)}{\sigma_X \sigma_Y} \tag{6}$$

As mentioned before, this research can be divided into X and Y variables, where Y needs to be predicted based on X. Therefore, it is desired that they have a high correlation. X variables with an extremely high correlation with a Y variable could indicate data leakage since it could result in the algorithm knowing too much during training.

Principal Component Analysis

Now that the correlation between the features is known, it is clear that some features are more important than others. A common dimensionality reduction method is Principal Component Analysis (PCA). It preserves as much information as possible while reducing the number of variables. It is commonly used when there are many features, and it is difficult to choose the dominating features manually [21]. The PCA method creates multiple Principal Components. This is done by making an orthogonal transformation of the data of possibly correlated variables into a new set of linearly uncorrelated features. The number of components is always less or equal to the number of original features since the goal is to reduce the dimensionality. The new variables (the principal components) are uncorrelated and aim to contain as much original data as possible. The first component has the largest possible variance, the maximum remaining information in the second, and so on. The new components are linear combinations of the original features and therefore less interpretable. The PCA process can be divided into four steps:

1. Standardization
2. Create the covariance matrix
3. Calculate eigenvectors and eigenvalues based on the previous step
4. Construct the components

Each step will be elaborated on. Since the data available in this research uses different units (e.g., °C, ppm, kWh), the features need to be normalized to make an equal contribution. When there are differences in units, the features with larger values will dominate over the small values while they are equally important. Equation 7 shows that for each feature n the normalization is done.

$$Z_i = \frac{x - \mu_i}{\sigma_i}, \qquad \forall i \in n \tag{7}$$

Since all features now have the same scale, the covariance matrix will be computed. The goal is to investigate whether the features are correlated with each other. The matrix is an nxn matrix (where n is again the number of features), as seen in Equation 8. More information about correlation can be found in the previous section.

$$C^{n \times n} = \left(c_{i,j}, c_{i,j} = \mathrm{cov}\left(\mathrm{Dim}_i, \mathrm{Dim}_j\right)\right) \tag{8}$$

The third step is calculating the eigenvectors and eigenvalues based on the covariance matrix. This is possible since this matrix is square (nxn). Equation 9 shows how to find the Eigenvalue λ of the covariance matrix C, where I is an identity matrix.

$$\det(C - \lambda I) \tag{9}$$

When λ is known, Equation 10 is used to find the Eigenvector. Here, V is the Eigenvector of A and λ again the Eigenvalue. There will be n different λ's and thus n corresponding Eigenvectors. Now that these values are known, they should be ordered in descending order. As explained before, the first principal component has the largest variance (i.e., highest Eigenvalue). The second component is uncorrelated (i.e., perpendicular) to the first component and has the second highest variance, and so on.

$$AV = \lambda V \tag{10}$$

The last step is to construct the components. There are now n principal components that might not all be important enough to keep. It is possible to simply discard the components with the lowest Eigenvalue. Information will be lost by doing so, but if the value is not high it will not be much data. This is necessary because the goal is to reduce dimensionality compared to the original features. The remaining components form a matrix of the corresponding Eigenvectors, called a 'Feature vector.'

$$\text{Final dataset} = \text{Feature vector}^{\ T} * \text{Standardized original dataset}^{\ T} \tag{11}$$

The final part is shown in Equation 11, where the transpose of the feature vector is multiplied by the transpose of the original input data [23, 24].

3.2.4 Evaluation metrics

Multiple performance evaluations will be used to evaluate an algorithm's performance. This section will elaborate on these metrics. The first metric is R^2, which evaluates how well a model fits the data. A high R^2 score means that the predicted and observed values fit very well. The R^2 is shown in Equation 12, where n is the sample size, x_t is the predicted value, y_t is the observed value, and $\bar{y}$ the mean of y_t [15]. R^2 ranges from 0 to 1, where 1 means the predictions are identical to the observed values.

$$R^2 = 1 - \frac{\sum_{t=1}^{n} \left(y_t - x_t\right)^2}{\sum_{t=1}^{n} \left(y_t - \bar{y}\right)^2} \tag{12}$$

The second metric is the Root Mean Square Error (RMSE) shows the sample standard deviation of the residuals between predicted and observed values and is used to quantify significant errors. Moreover, it demonstrates how the model response varies in terms of variation. Large errors are highly penalized since it geometrically multiplies the error. The RMSE is shown in Equation 13. The variable input explanation is the same in Equation 12 [15]. The lower the value of the RMSE, the better the model is.

$$\text{RMSE} = \sqrt{\frac{1}{n} \times \sum_{t=1}^{n} \left(x_t - y_t\right)^2} \tag{13}$$

Lastly, a statistical measure called MAPE compares the residuals with the observed values to describe how accurate the prediction was. It shows the average percent difference between forecast and actual prediction. The absolute value is taken because this measure states it is irrelevant whether the forecast was too high or too low. MAPE typically represents accuracy in percentages. Equation 14 shows the MAPE, where the same input variables are used as in Equation 12 [15]. The lower the value for MAPE, the better the algorithm predicts values.

$$\text{MAPE} = \frac{1}{n} \times \sum_{t=1}^{n} \left| \frac{x_t - y_t}{y_t} \right| \times 100\% \tag{14}$$

3.3 Phase 3

This section will elaborate on the steps taken to execute the last phase in this research.

3.3.1 Experimental set-up

When the data is pre-processed and the best algorithm to predict CO2, temperature, and electricity levels for the upcoming hour is known, the simulation can be done. Figure 5 shows the general action plan for this.

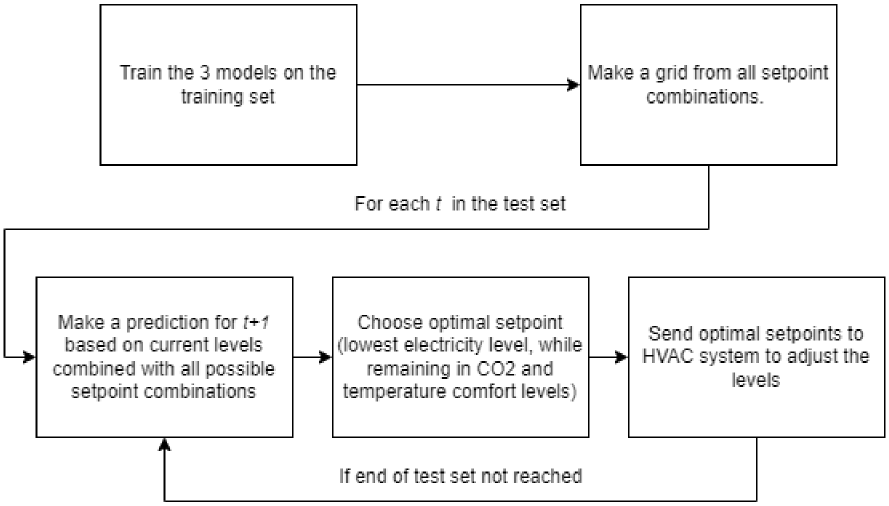

Train the 3 models on the training set
Make a grid from all setpoint combinations.
For each *t* in the test set
Make a prediction for *t+1* based on current levels combined with all possible setpoint combinations
Choose optimal setpoint (lowest electricity level, while remaining in CO2 and temperature comfort levels)
Send optimal setpoints to HVAC system to adjust the levels
If end of test set not reached

Figure 5: General action plan of phase 3.

First, the three models are trained on the training set. Moreover, a grid from all possible combinations of the HVAC setpoints is made. Before the simulation starts, a couple of assumptions are made:

- The predictions for $t+1$ at time t are taken as the truth. This is because there is no information available on what happens when changing the HVAC setpoints.
- The weather prediction is unknown, only the current weather at time t is known.

The test set contains data with values for every hour t in the building. The simulation involves recreating a "real-time" process by going over every hour t in chronological order. At time t, the simulation knows the current levels of all features. These levels are combined with all the setpoints. For each combination, the three models are run and a prediction is made as shown in Table 1. Here, x represents a feature, and s is an HVAC setpoint. The table shows a couple of possible combinations of the setpoints. For each combination (row), a prediction is made.

The next step is to choose the optimal setpoint. This is the row that minimizes electricity usage while keeping the CO2 and temperature levels within the comfort range. The exact values of these comfort levels will be based on the literature research and looking at the data. If multiple setpoint combinations give the same lowest electricity usage for $t+1$, the following formula is used to pick the optimal combination. Here, i is the setpoint with a total of n setpoints. It sums the difference between the current setpoint and the possible optimal setpoint. This is done since it costs the least amount of energy to adjust the setpoints to a setpoint close to the current values.

$$\text{Optimal combination} = \min \Sigma_{i=1}^{n} \mid s_{\text{current}i} - s_{\text{option}i} \mid \tag{15}$$

The optimal combination of setpoints for hour t is sent to the HVAC system and it will adjust the settings to reach the predicted values. This is done for every hour in the test set to simulate a real life environment.

Table 1: Example of the options during the simulation. These numbers are fictive and are only meant as an example.

	x1	**x2**	**x3**	**s1**	**s2**	**s3**	**CO2 +1h**	**Temp. +1h**	**Elec. +1h**
t	100	150	80	18	200	5	400	22	1
t	100	150	80	18	220	10	500	23	2
t	100	150	80	20	200	5	500	22.5	3
t	100	150	80	20	220	10	450	24	4

The simulation can be regarded as successful when the electricity usage of the building in the test set is less than the electricity usage used in the original test set.

4 Data

This chapter describes the provided data by Heroes B.V., and all the steps required before predictions could be made. Lastly, an exploration of the data is performed.

4.1 Data availability

Nuuka Solutions is a company that optimizes building performances. They collaborate with the municipality of Helsinki and ensure that the provided dataset is available. Heroes B.V. has a connection to the API from *Nuuka Solutions*. The data is transferred to Heroes B.V.'s database on a daily base. I have gotten access to this Azure database, which contains the following tables: 'Building', 'Data categories', and 'Data points'. An overview of the tables in the database and how they are related to each other is given in Figure 6. The relationship between the tables is given by the arrow, with either a '1' or 'n' at both ends. A '1' to 'n' relationship (one-to-many) means that one row in table A could be linked to many rows in table B. For example, a building with ID 1 can be linked to many data categories. A '1' to '1' (one-to-one) relationship means that each row in table A is linked to another specific row in table B. Moreover, the arrows between the table point to the feature where they are linked.

Each table will be shortly elaborated. First, the table 'Building' collects the available information about the building, such as in which city it is placed, the exact location, and the building type. It is an educational building located in Helsinki, Finland. The exact location is (long, lat) = (60.173, 24.9479). The building was built in 1924 and has a surface of 5141 m^2, of which 1681 m^2 is heated. For example, parking spots do not have to be heated. Secondly, the table 'Data categories' contains information about which data the sensors are gathering in that specific building. 'Id' is a unique identifier representing a single sensor that is connected to that specific building. A name, description, unit, and category name are also given. 77 Unique categories are being measured, reaching from the amount of electricity used to exhaust fan frequency converter power. In total, there are 668 different sensors. For example, the temperature is tracked by 42 different sensors. Moreover, the 'Data points' table contains all the values each sensor has measured. To combine the sensors and categories with the measured values at each timestamp, an inner join is performed within the database using the unique 'Id' (from 'Data categories') and 'Data_categories_id' (from 'Data points') keys. The data covers a period from 01-01-2020 to 07-11-2022.

As stated before, 77 different categories are measured by one or multiple sensors. Throughout the conversation with the domain expert of Heroes B.V., it became clear that only a few categories are interesting to analyze. For example, a category such as 'supply duct static pressure relative control deviation' is irrelevant to the system's energy consumption and therefore deleted. The categories that remain are each category that contains the word 'setpoint' since this is a setting that can be changed throughout the day. Moreover, variables that are interesting to predict based on these settings are chosen. Table 2 shows the 13 remaining categories and the 167 sensors that track these in descending order.

Lastly, external weather data is gathered using a python package called *Meteostat*. The 'Building' table contains the exact location, which is related to the 'City' table. This means that the local weather data could be collected. Since the sensor data ranges from 01-01-2020 to 07-11-2022, the same period is chosen for the weather data. The collected categories are shown in Table 3.

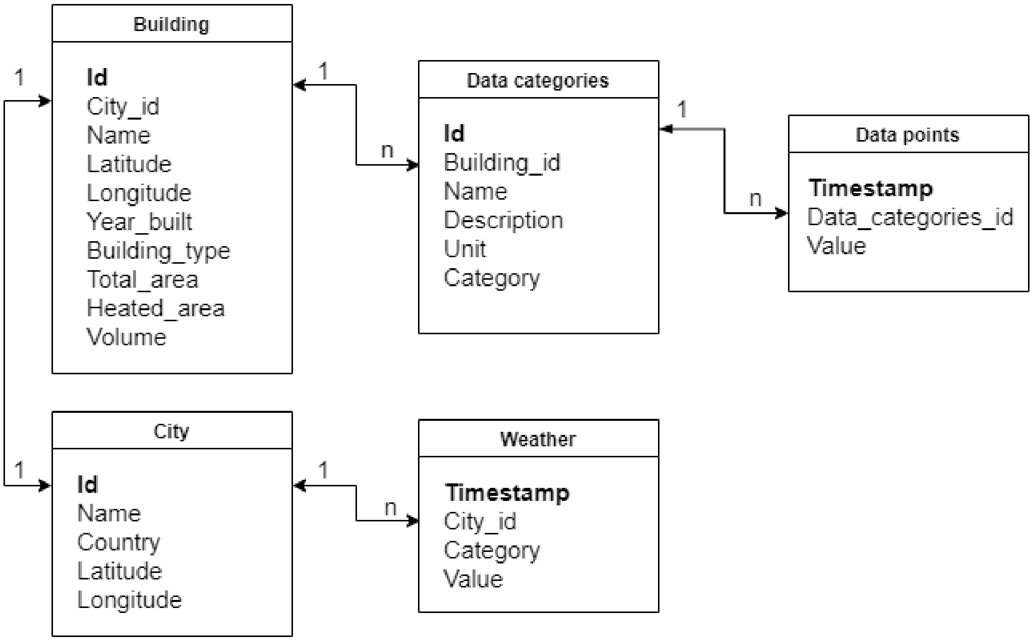


Figure 6: Overview of the data model in the database.

4.2 Pre-processing

As stated before, the data comes from two different sources: the *Nuuka Solutions* sensor data and the *Meteostat* weather data. They will be investigated separately and later merged.

Sensor data

Looking at the sensor data, it is clear that a few remarkable things are happening. First, some sensors measure differently: every 10 minutes, every hour, or very inconsistently. Secondly, some sensors only provide a few data entries. This means that they were only working some of the time. Since not every category has the same amount of sensors and the before-mentioned inconsistencies, the mean is taken for every category across all its sensors per hour. When diving deeper into the data, it turns out that there are big gaps of missing data. Instead of empty values in the dataset, there are no entries of these missing dates due to a broken pipeline at the time. These problems are solved by doing an aggregation. This means that per category, there is one value per hour which is the hourly mean across all sensors for that category. The data is provided where all categories are in one single column. A pivot is performed so there are 13 category columns, each representing a category. The index is the timestamp of when each value is measured. This results in a dataset with a length of 25.008.

After taking a closer look at the categories, it appears that some have a constant value throughout the dataset. The categories, followed by their constant value, are: 'ventilation network pressure difference setpoint' (30 Pa), 'radiator heating network pressure difference setpoint' (43 Pa), 'domestic hot water network supply temperature setpoint' (58 °C), and 'CO2 setpoint' (700 ppm). A constant value means that it will not add value and is thus deleted from the model. 'Temperature setpoint' has a constant value of 22 °C for most of the dataset. During a short period, it is suddenly set to 38 °C. It is also chosen to delete this column, resulting in 8 categories.

Table 2: Information on the available sensor categories and the number of sensors.

Category	Unit	Amount of sensors
Temperature	°C	42
CO2	ppm	38
CO2 setpoint	ppm	25
Temperature setpoint	°C	25
Discharge temperature setpoint	°C	9
Exhaust duct static pressure setpoint	Pa	9
Supply duct static pressure setpoint	Pa	9
Electricity	kWh	5
Domestic hot water network supply temperature setpoint	°C	1
Radiator heating network pressure difference setpoint	Pa	1
Radiator heating network supply temperature setpoint	°C	1
Ventilation network pressure difference setpoint	Pa	1
Ventilation network supply temperature setpoint	°C	1
		167

Table 3: Information on the available weather categories.

Category	Unit
External air temperature	°C
External average wind speed	km/h
External dew point	°C
External total precipitation	mm
External relative humidity	%
External sea-level air pressure	hPa
External snow depth	mm
External total sunshine duration	Minutes
External weather condition code	1-27
External wind direction	Degrees
External wind peak gust	km/h

Weather data
When collecting the weather data, it comes in the desired format (index as timestamp and one category per column). The following chapters will dive deeper into the missing values and outliers.

4.2.1 Missing values

Sensor data
The data aggregation has led to the dataset containing consistent hourly measurements, and empty values where there were no measurements. There are 27.453 empty values, which is around 14% of the dataset. Figure 7 provides a clear overview of when each category has missing values. The categories are shown on the x-axis, and the time (in hours) on the y-axis. White means no value is present at that timestamp, and black means there is. It is very noticeable that there is a precise moment when the pipeline broke, and no data was sent to the database any-

more. The largest white missing stroke lasts from 01-11-2020 to 03-03-2021, which is around four months. Lastly, it is noticeable that all categories stopped working for 135 consecutive values on 08-09-2022, except for the categories 'CO2', 'Electricity,' and 'Temperature'. To avoid these gaps, it is chosen to cut the dataset. The cut dataset starts on 03-03-2021 and ends on 08-09-2022, which results in 13.295 rows.

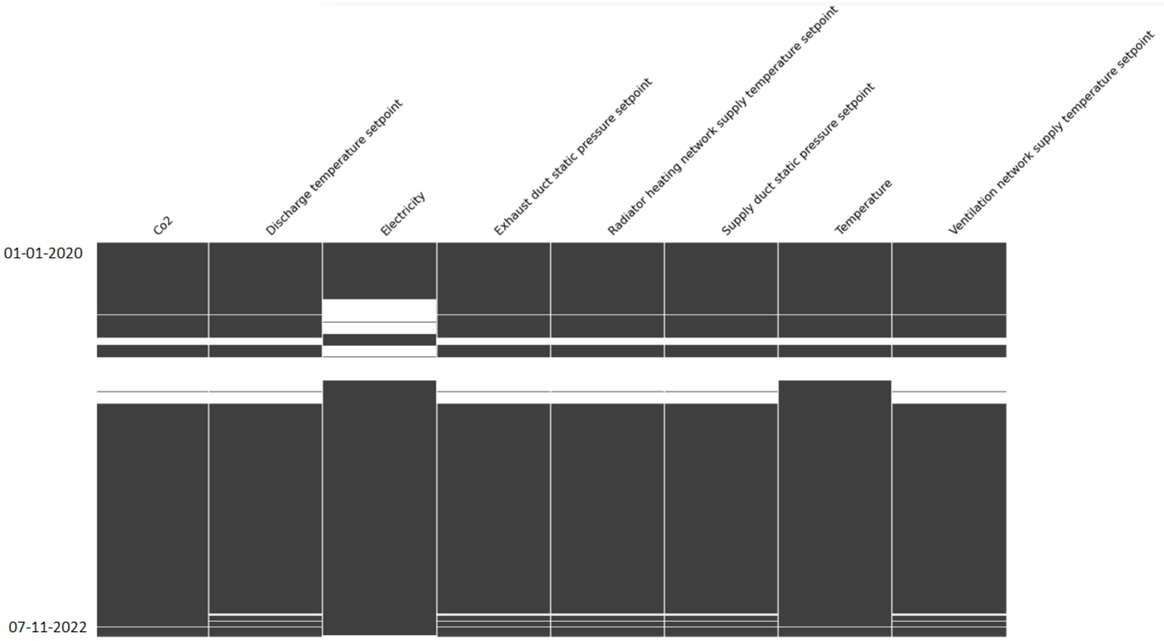


Figure 7: Overview of missing sensor values over time per category.

Weather data

The weather data is also cut down from 03-03-2021 to 08-09-2022. There are 25.792 missing values, which is around 18 % of the dataset. After taking a closer look at the dataset, it becomes clear that the categories 'External Snow Depth' and 'External total Sunshine Duration' are respectively around 92 % and 100 % empty. Moreover, the category 'External weather condition code' consists of 153 consecutive empty values. It is undesired to cut the dataset more. It is therefore chosen to delete this category and the two mostly empty categories. This results in less than 1% missing data.

4.2.2 Outliers

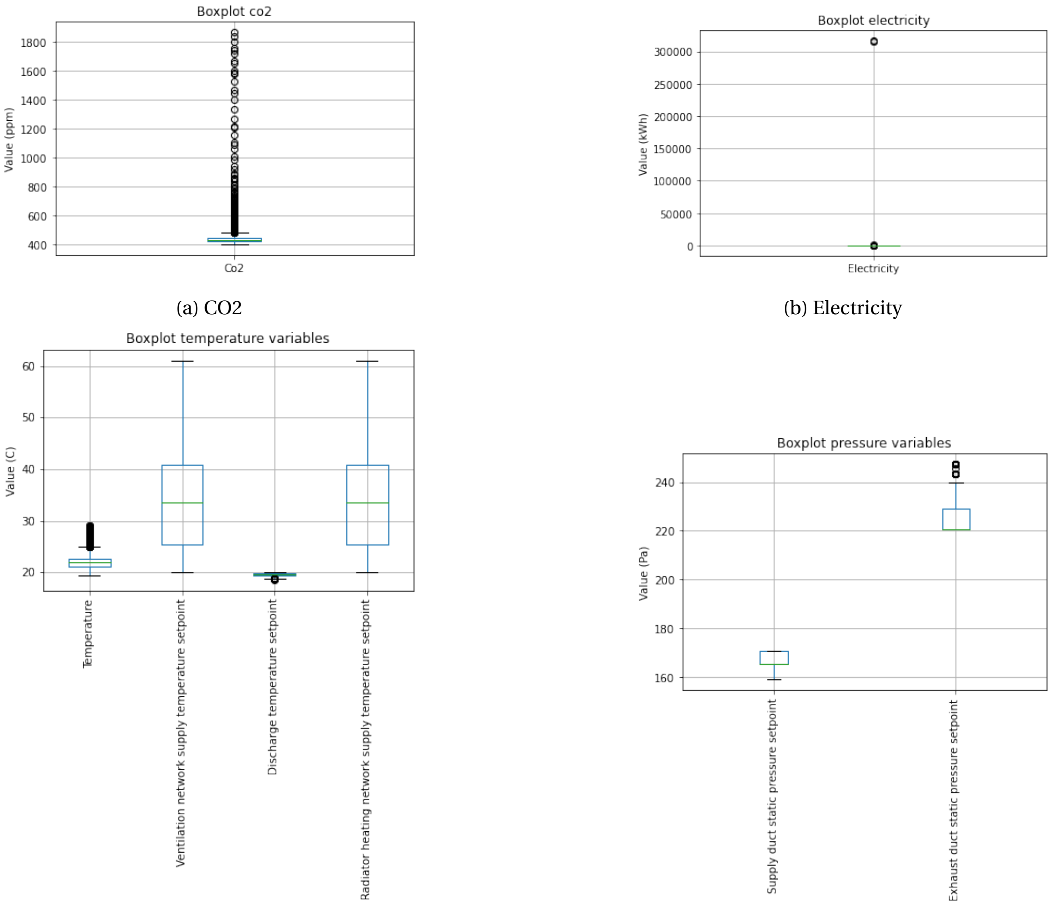
As explained in Section 3.2.2, it is necessary to remove outliers to enhance performance. The IQR method is used for this.

Sensor data

To visualize outliers, a boxplot is made and shown in Figure 8. The categories are sorted based on their units. Figure 8a shows that the category 'CO2' appears right-skewed. Most values are around 450 ppm (in the left tail), while there are also values on the long right side of the tail. An outlier is an observation significantly different from the rest of the data, shown as dots outside the whiskers of the box plot. However, these are not regarded as outliers in this case. The first quantile is at 420,05 ppm, and the third is at 446,80 ppm. According to the IQR method, everything below 379.93 ppm and above 486.93 ppm will be seen as an outlier and should thus be removed from the dataset. An explanation could be that only around 10% of the data is higher than 550 ppm. Thus the box plot whisker does not include these values. According to Section 2.2, the CO2 levels should either be below 1000 ppm or 1500 ppm according to different organizations. In the boxplot, it can be seen that some values reach 1800 ppm. These values are mostly during working hours, which could mean that there were many people in the building. Talking to the domain experts of Heroes B.V., these levels are not recommended, but they could occur. The mean/median of CO2 is 451/427 ppm. Most values are between 400 to 800 ppm. Even though it is unwanted to remove real levels, looking at the more common values and the boxplot, the IQR method is used to remove outliers. This is done to improve the accuracy of the algorithms. The k is set relatively high to 10, to only remove the largest outliers. This means that values below 167.028 or above 697.355 are removed from the dataset.

Figure 8b shows that 'Electricity' has clear outliers in the data. While most values are between 0 and 6 kWh, some values are around 300.000 kWh. This is a clear measurement error and is thus deleted using the IQR method. Here, k is set to 3. Setting it to $k = 1.5$ as recommended in the literature would remove real data. Therefore it is chosen to set k higher. Setting $k = 3$ ensures that all values below -5.159 or higher than 8.618 are removed.

Figure 8c shows the boxplot of all temperature-related variables. According to Section 2.2, the 'Temperature' levels should be between around 20 and 27 °C. Again, the boxplot whiskers show a higher value which is not unlikely to occur. 'Ventilation network supply temperature setpoint' and 'Radiator heating network supply temperature setpoint' show that the median is in the middle of the box. This could indicate a normal distribution. It is again not chosen to remove any outliers. Lastly, the pressure boxplot shows a few outliers for the 'Exhaust duct static pressure point'. However, these outliers are not that big, and there could be a fluctuation. Therefore, also no outliers were removed.



(a) CO2

(b) Electricity

(c) Temperature

(d) Pressure

Figure 8: Boxplot of the sensor variables.

Weather data

For the weather data, Figure 9 shows outliers in the dataset. However, these values are not unlikely to happen. Weather can change drastically during the year. There are no measurement errors. Therefore, it is chosen not to delete any weather data.

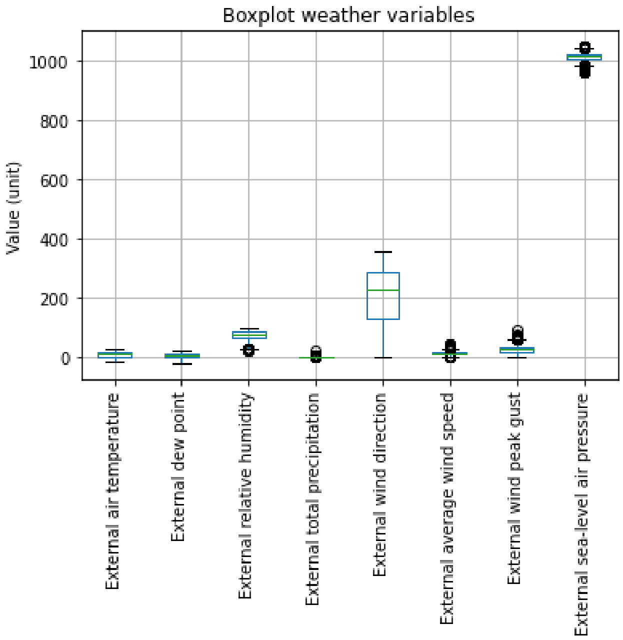


Figure 9: Boxplot of the weather variables.

4.2.3 Imputing empty values

The dataset contains empty values. Since the outliers were removed, this led to more empty values. Using Section 3.2.2, the empty values were imputed. As described, up to 8 consecutive values will be imputed using linear interpolation. For gaps up to 48 consecutive values, K-nearest neighbors will be applied. The longest consecutive missing value is 10 hours. After using both methods for imputation, the dataset now contains zero empty values.

4.3 Feature engineering

As described in Section 3.2.3, extra features are added to the dataset. First, the sensor and weather data are merged into one dataset, after which the features are added. This results in the following features, where each variable $x \in X$ and $y \in Y$. This leads to a total of 33 X and 3 Y variables, shown below:

$$X = \left\{ \begin{array}{c} \text{CO2} \\ \text{CO2 -1 hour} \\ \text{CO2 -24 hour} \\ \text{CO2 -48 hour} \\ \text{CO2 }\Delta\text{-1 hour} \\ \text{Dayofweek} \\ \text{Discharge temperature setpoint} \\ \text{Electricity} \\ \text{Electricity -1 hour} \\ \text{Electricity -24 hour} \\ \text{Electricity -48 hour} \\ \text{Electricity }\Delta\text{-1 hour} \\ \text{Exhaust duct static pressure setpoint} \\ \text{External air temperature} \\ \text{External average wind speed} \\ \text{External dew point} \\ \text{External relative humidity} \\ \text{External sea-level air pressure} \\ \text{External total precipitation} \\ \text{External wind direction} \\ \text{External wind peak gust} \\ \text{Hour} \\ \text{Month} \\ \text{Radiator heating network supply temperature setpoint} \\ \text{Season} \\ \text{Supply duct static pressure setpoint} \\ \text{Temperature} \\ \text{Temperature -1 hour} \\ \text{Temperature -24 hour} \\ \text{Temperature -48 hour} \\ \text{Temperature }\Delta\text{-1 hour} \\ \Delta\text{ External - inside temperature} \\ \text{Ventilation network supply temperature setpoint} \end{array} \right\}$$

$$Y = \left\{ \begin{array}{c} \text{CO2 +1 hour} \\ \text{Electricity +1 hour} \\ \text{Temperature +1 hour} \end{array} \right\}$$

4.3.1 Correlation

To see if the features influence each other, the correlations are computed and shown in a heatmap, as explained in 3.2.3. Figure 10 shows that ’Hour’, ’Month’, and ’Season’ barely correlate with the other features. ’Season’ does correlate with ’Month’, as one can expect. It was expected that these features would influence temperature (summer versus winter) or electricity usage (needing more electricity to warm up the space). Moreover, the correlation plot shows multi-collinearity between multiple features. This means that one independent variable can be linearly predicted from the variable it correlates with. This can lead to difficulties for the prediction algorithm. Lastly, it can be seen that the Y variables have an extremely high correlation with some features.

For example, ’Temperature +1 hour’ has ten features with extreme correlation. This could indicate data leakage, which is unwanted, as explained before.

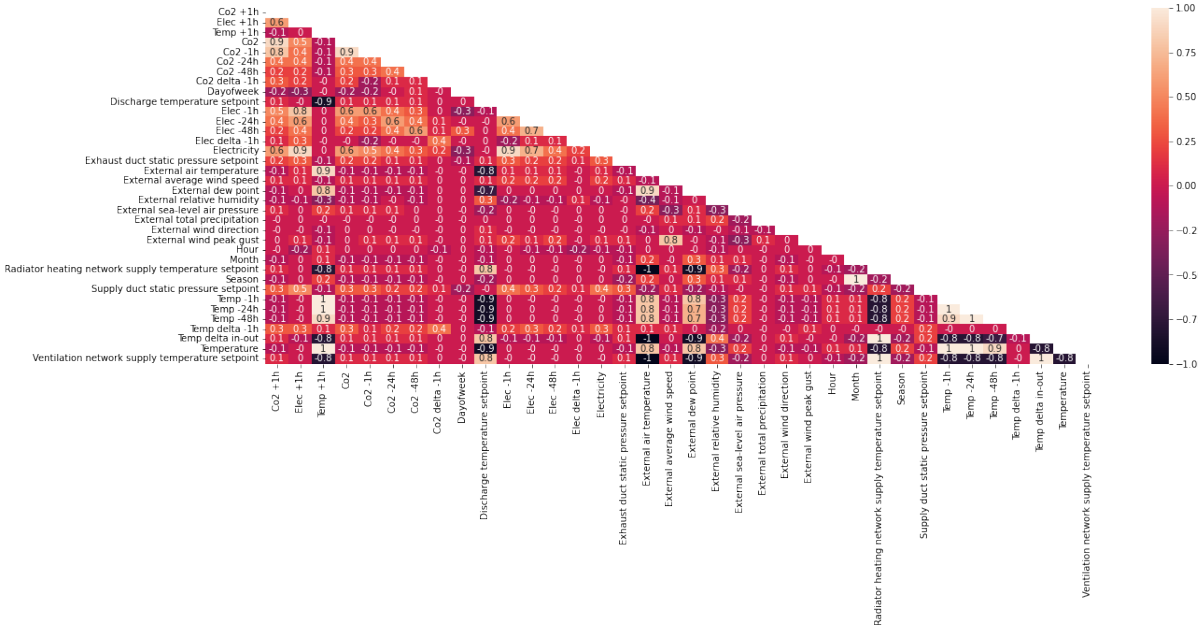


Figure 10: Heatmap showing the correlation of the variables with each other.

4.3.2 Principal Component Analysis

It is clear that there are many features, and dimensionality reduction is desired. PCA is useful when multi-colinearity exists between the features, as in this case. By transforming the features into new features that are uncorrelated from each other, multi-collinearity could be solved. As explained in Section 3.2.3, the first step is normalizing the data. After this, the algorithm is executed.

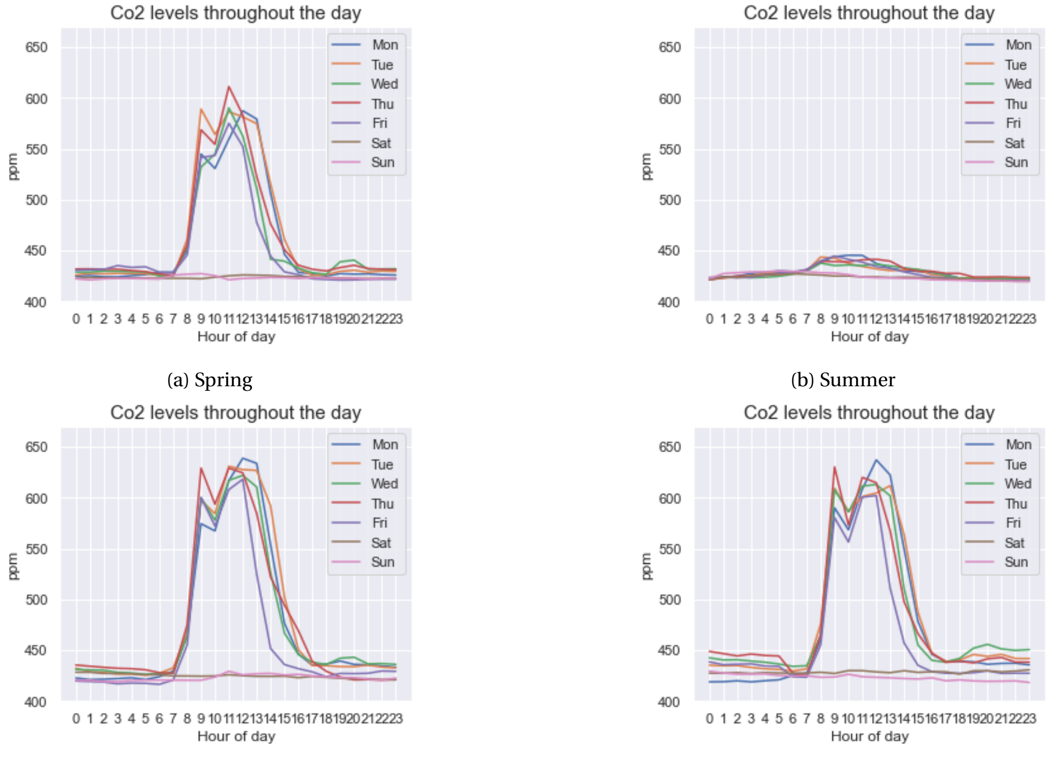
The algorithm can decide how many components are necessary for the incoming data. A scree plot is used to analyze the components. It shows the principal components created and each component’s variance. As explained before, the first component contains the most variation, the second component the second most, etc.

4.4 Exploration

During the data exploration, a couple of insights were obtained. The values of the Y variables (CO2, temperature, and electricity) were investigated throughout the day. The dataset is split into four seasons (spring, summer, autumn, and winter) and investigated separately. As can be seen in Figure 8, the Y variables appear skewed. Therefore, the median is taken instead of the mean of all days in that season. Figure 11 shows the CO2 value of each hour in the day during each season. During summer, the CO2 levels differ much from the rest of the year. The ppm does not exceed 450, while during the rest of the year, 600 ppm is mainly reached. During the whole year, it appears that the building is not used on Saturday and Sunday. The ppm remains relatively low compared to the other days. The CO2 peak is between 8h-15h. The researched building is an educational building, which can indicate that during that time there were the most people in the building.

The temperature values throughout the day per season can be found in the appendix. During summer, the temperature is the highest. This can naturally be explained since summer is the warmest period of the year. Moreover, the building increases by around 1.5 °C during the day. Winter has the coldest period, as seen in the plot. Saturday and Sunday do not differ as much from the rest of the days compared to CO2 levels. It is also remarkable that during summer and spring, the temperature levels on Saturday and Sunday are higher than on the other days of the week. A reason for this could be that the HVAC system is saving energy by not cooling the building.

Lastly, the electricity values can also be found in the appendix. The electricity usage is highest during the Autumn and Winter period, which can be explained by the cold weather and the HVAC system has to warm up the building. There is a peak in the middle of the day during the whole year. It is remarkable that on Sundays, there is a peak around 10.00. An explanation could be cleaners that come in to clean during the weekend.



(a) Spring

(b) Summer

(c) Autumn

(d) Winter

Figure 11: CO2 levels throughout the day per season.

5 Results

This section elaborates on the results of the research. The first section will elaborate on the prediction, and the second will focus on the simulation.

5.1 Phase 2: Prediction

This section will elaborate on the outcome of predicting the CO2, temperature, and electricity values for the upcoming hour. Three methods were used for this: Random Forest, Gradient Boosting (LightGBM and XGBoost), and Lasso Regression.

5.1.1 Random Forest

A grid search combining all of the most important parameters in Random Forest is applied. Table 4 shows the optimal combination of settings according to the grid search and its evaluation metrics using the Random Forest algorithm.

Starting with the prediction for CO2 in the upcoming hour, the table shows that the default parameters perform best. It gives the highest R^2 and a relatively low RMSE / MAPE. The average difference between the predicted versus actual values is according to the RMSE 13.727. According to the MAPE, the actual values differ only around 1% from the predicted values. Moreover, the table shows that PCA performs less well than using all the input features. PCA reduced the number of features from 33 to 17 remaining components. Figure 12 shows the scree plot. The eigenvalues are shown on the y-axis, and the number of components is on the x-axis. As explained before, the first component contains the most information. The second component contains the second most information and so on. After the third component, the amount of information per component decreases. However, lots of information would be lost if the number of components was reduced from 17 to 3.

Table 4: Performance evaluation for each model based on Random Forest. The metrics are an evaluation score on the test set.

		Hyperparameters			*Metrics*		
		Max depth	**# Estimators**	**Max features**	R^2	***RMSE***	***MAPE***
CO2 +1h	Default	None	100	33	**0.915**	**13.727**	**1.087**
	Grid search	9	100	33	0.914	13.740	1.096
	Grid search & PCA	8	10	17	0.709	25.339	3.164
Temp. +1h	Default	None	100	33	**0.995**	**0.092**	**0.268**
	Grid search	7	10	33	0.994	0.101	0.301
	Grid search & PCA	10	10	17	0.964	0.242	0.822
Elec. +1h	Default	None	100	33	**0.920**	**0.458**	**2.513**
	Grid search	9	10	33	0.913	0.478	8.121
	Grid search & PCA	10	50	14	0.693	0.899	20.166

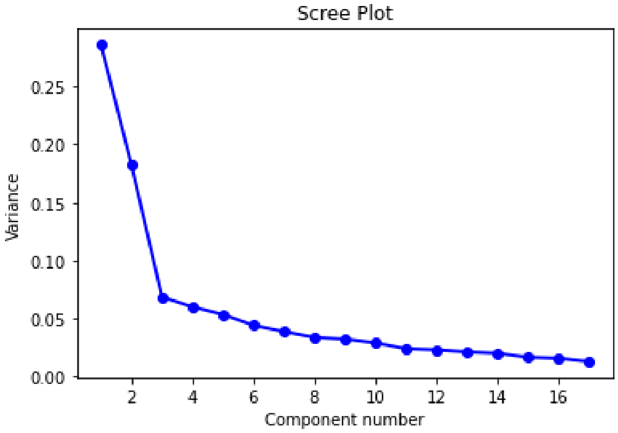


Figure 12: Scree plot of Random Forest on CO2, using grid search and PCA method.

RF performs best on CO2 prediction using the default setting approach. Each tree has an unlimited maximum depth using the default settings. It will build 100 decision trees and consider all features when determining a split. The model appears not to be overfitting since the evaluation metrics in the training and test set do not differ much. This means that the model generalizes well during training. Another feature of this algorithm to prevent overfitting is random feature selection. At each node, a random subset of features is chosen to split on. Moreover, the grid search also took the parameter 'Maximum amount of samples' into account to increase performance on the test set. The default setting uses all samples, while the grid search and grid search + PCA method respectively set this to 50% and 30%.

To visualize the performance compared to the actual prediction, Figure 13 is shown. The left figure shows the first 70% of the data (y train) and, after that, the last 30% of the data (y test versus y prediction). The right figure shows a closeup of the performance on week 26 of 2022. It can be seen that the prediction (in orange) follows the actual values (in blue) quite accurately, including peaks.

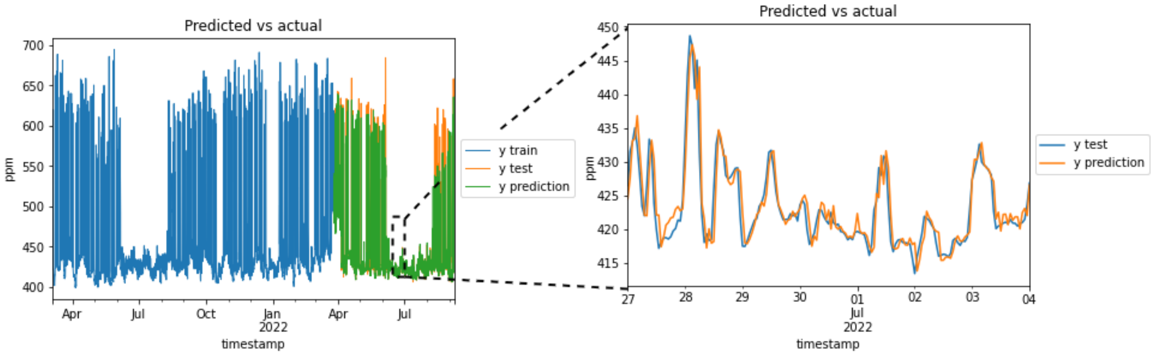
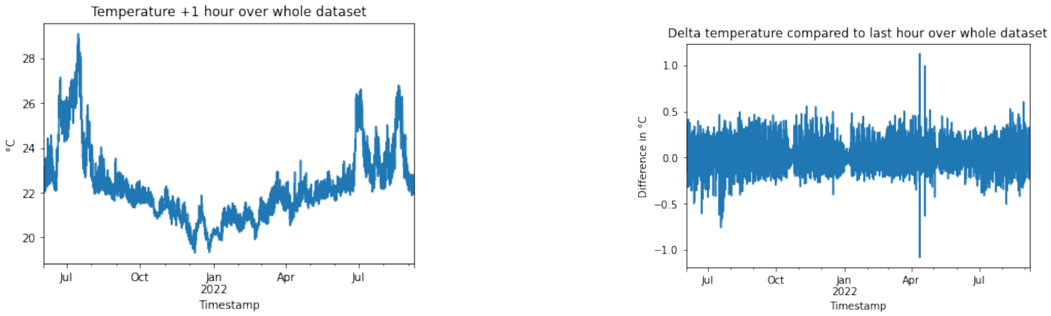


Figure 13: Actual versus predicted CO2 levels using Random Forest with default settings. The zoom-in shows week 26 of 2022.

The temperature prediction scores really well. The model does not appear to overfit, since the training and test evaluation metrics are around the same value. An explanation for the good performance can be seen in Figure 14. The left figure shows the temperature over time, showing a clear pattern. The right figure shows that the difference in temperature compared to the last hour is barely larger than ±0.5 °C. The data does not vary much, which can explain the result. For both electricity and temperature, the default settings perform best. PCA again does not improve the performance.



(a) Temperature over time.

(b) Delta compared to previous hour.

Figure 14: Temperature over time and the delta compared to previous hour.

5.1.2 XGBoost

The results of using XGBoost to predict CO2, temperature, and electricity in the upcoming hour are shown in Table 5. All 3 models found that the grid search model performs best. The grid search found that for CO2 and temperature, using more, but smaller trees (700 instead of 100) compared to the default settings improved performance. Moreover, adding a subsampling rate also improved performance on the test set and prevents overfitting at the same time. This indicates the percentage of data used per tree building, and when to set lower, the fewer data is looked at during each tree. PCA again worsens performance. Lastly, the model does not appear to be overfitting, since the train and test evaluation metrics are around the same values.

Table 5: Performance evaluation for each Y variable based on XGBoost. The metrics are based on the test set.

		Hyperparameters				*Metrics*		
		Max depth	**# Estimators**	v	**Subsample**	R^2	**RMSE**	**MAPE**
CO2 +1h	Default	6	100	0.3	1	0.923	13.072	1.075
	Grid search	4	700	0.01	50%	**0.925**	**12.884**	**1.064**
	Grid search & PCA	10	700	0.05	40%	0.779	22.077	2.563
Temp. +1h	Default	6	100	0.3	1	**0.995**	0.092	0.275
	Grid search	2	700	0.1	70%	**0.995**	**0.087**	**0.260**
	Grid search & PCA	1	700	0.2	40%	0.972	0.213	0.724
Elec. +1h	Default	6	100	0.3	1	0.925	0.443	**4.112**
	Grid search	7	100	0.05	60%	**0.929**	**0.433**	6.038
	Grid search & PCA	7	700	0.05	30%	0.746	0.818	8.315

To visualize this method, Figure 15 is shown. As elaborated before, the left figure shows the y train and thereafter the y test versus y prediction. The right figure shows a close-up of the performance on week 26 of 2022. The close-up shows that the predictions follow the actual values quite closely. It catches the peaks and predicts according to that.

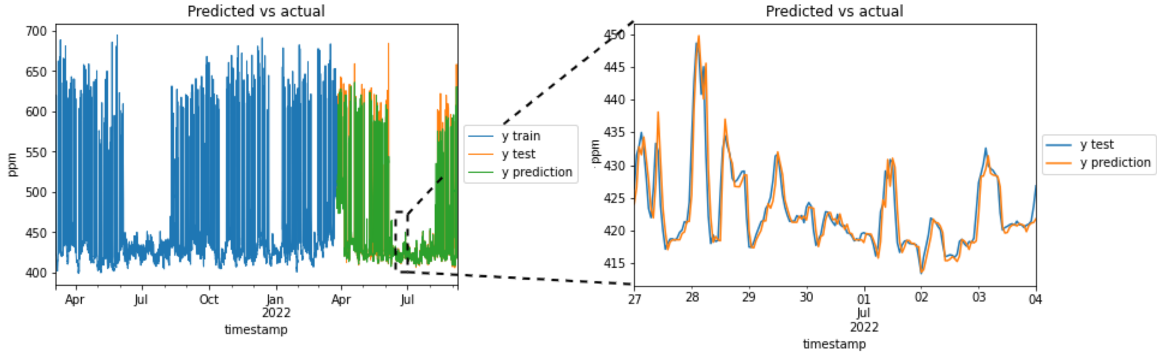


Figure 15: Actual versus predicted CO2 levels using XGBoost while using a grid search. The zoom-in shows week 26 of 2022.

5.1.3 LightGBM

Table 6 shows the results of predicting CO2, temperature, and electricity in the upcoming hour using LightGBM. It can be seen that for all three models, the grid search method performs best. The model does not appear to overfit, as the training and test evaluation metrics are around the same values. The same conclusions can be drawn from this model as from XGBoost in the previous section.

Table 6: Performance evaluation for each model based on LightGBM. The metrics shown are based on the test set.

		Hyperparameters				*Metrics*		
		Max depth	**# Estimators**	v	**Subsample**	R^2	**RMSE**	**MAPE**
CO2 +1h	Default	None	100	0.1	All	0.921	13.236	**1.058**
	Grid search	4	300	0.05	40%	**0.927**	**12.670**	1.078
	Grid search + PCA	2	700	0.2	40%	0.790	21.546	2.702
Temp. +1h	Default	None	100	0.1	All	**0.996**	**0.083**	**0.245**
	Grid search	3	400	0.2	40%	**0.996**	0.084	0.256
	Grid search + PCA	1	700	0.2	40%	0.970	0.221	0.752
Elec. +1h	Default	None	100	0.1	All	**0.926**	**0.441**	6.562
	Grid search	6	700	0.05	40%	**0.926**	0.442	**4.675**
	Grid search + PCA	8	700	0.05	40%	0.742	0.823	12.111

Even though XGBoost and LightGBM show similar results, there is a difference. This is shown in Figure 16. Here, the left figure shows the y train and y test data, and the right figure shows a close-up of week 26 in 2022. While XGBoost was able to predict accurately, it can be seen that LightGBM predicts peaks (in orange) that are not in the test data (blue line).

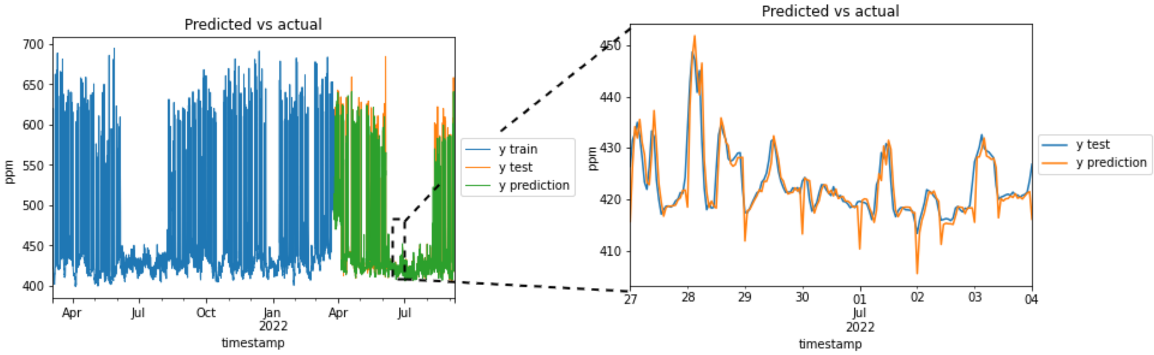


Figure 16: Actual versus predicted CO2 levels using LightGBM with a grid search. The zoom-in shows week 26 of 2022.

5.1.4 Lasso Regression

To prevent one feature from having too much influence over λ, it is necessary to first standardize the data and have all features on the same scale. This will help the algorithm estimate the coefficients. Table 7 shows the results of using Lasso to predict CO2, temperature, and electricity. It can be seen that grid search performs best for all three models. The grid search is done to find the optimal parameter λ. CO2 has a relatively high optimal λ, while temperature and electricity have a relatively low optimal λ. A low λ means less shrinkage occurred, while a high value means more shrinkage occurred. This can lead to a model with lower coefficients and reduced variance.

Table 7: Performance evaluation for each model based on Lasso regression. The metrics are based on the test set.

		Hyperparameters	*Metrics*		
		λ	R^2	**RMSE**	**MAPE**
CO2 +1h	Default	1	0.793	21.348	2.776
	Grid search	0.99	**0.796**	**21.229**	**2.716**
	Grid search & PCA	0.31	0.739	24.003	3.375
Temp. +1h	Default	1	0.591	0.819	2.664
	Grid search	0.01	**0.993**	**0.104**	**0.308**
	Grid search & PCA	0.01	0.956	0.268	0.905
Elec. +1h	Default	1	0.642	1.113	61.216
	Grid search	0.06	**0.782**	**0.757**	**25.55**
	Grid search & PCA	0.06	0.607	1.017	43.617

Figure 17 shows the actual versus predicted values regarding CO2 in the upcoming hour. It can be seen that the prediction (orange line) did catch a pattern, but it does not match the actual values (blue line) well.

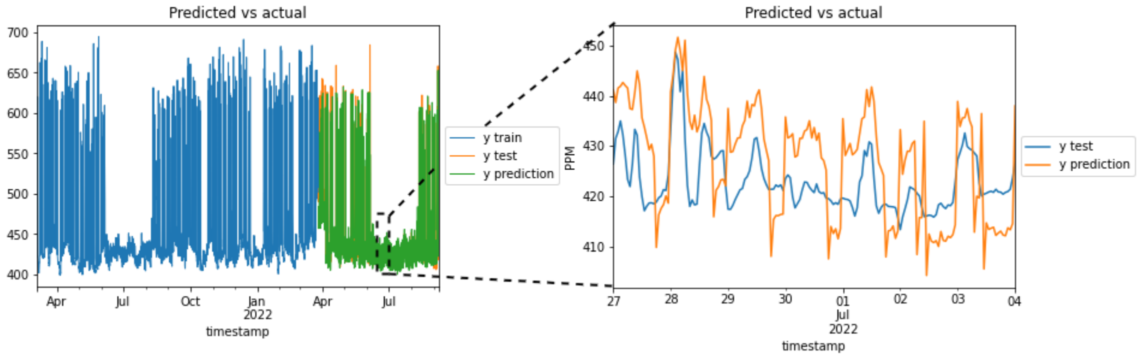


Figure 17: Actual versus predicted CO2 levels using Lasso with a grid search. The zoom-in shows week 26 of 2022.

5.2 Phase 3: Optimization

The previous section described all four algorithms using three different methods. It was seen that Lasso showed the worst results, and LightGBM also did not follow the actual values well. For the simulation, XGBoost will be used, since it performs slightly better compared to Random Forest. The XGBoost scores a bit better on CO2 and electricity, and it scores the around same on temperature.

When the three models (CO2, electricity, and temperature) are trained on XGBoost, a grid is made of all setpoint combinations. Table 8 shows the five setpoints and the minimum and maximum values within the dataset. Since we are optimizing the setpoints, it is desired to explore a wider range of options. The simulation range is shown on the right.

When running the algorithm, it soon became clear that using a grid of all options combined was computationally too expensive. If this simulation would be in real life, the optimal setpoints should be there in minutes, not hours (there is no point in predicting the next hour if the calculation would take hours). The grid was computationally expensive since there are 5 setpoints, and it would exponentially grow. For example, if they all had 10 options, the grid would be size 10*10*10*10*10. To optimize the speed, five separate grids are made. The first grid contains all the options of the first setpoint 'Discharge temperature setpoint'. For all other setpoints, the median is taken. The median is chosen since it gets closer to the setpoint that is chosen most often compared to the mean. As explained in Section 3.3.1, a prediction is made for all these combinations. The best setpoint is chosen and saved for 'Discharge temperature setpoint'. Now, the same method is used to find the optimal setpoint for that hour for 'Exhaust duct static pressure setpoint.' This is repeatedly done for all setpoints. Lastly, these optimal setpoints are combined and a final prediction is made for CO2, electricity, and temperature in the upcoming hour. This way is computationally far less expensive.

The simulation chooses the optimal setpoints that minimize electricity usage and where CO2 and temperature levels stay within thermal comfort. Table 9 shows the optimal levels of CO2 and temperature during all seasons. Since the analyzed building is an educational building, and looking at Figures 11 and 21, it is assumed that office hours are between 7.00-17.00 on a weekday (Monday to Friday). Outside these hours, the comfort range is taken wider since there are no occupants and thus minimizing electricity can be seen as more important than thermal comfort.

Literature research shows that CO2 levels should be below 1000 ppm (WHO and ASHRAE) or

Table 8: The five setpoints in the simulation and the range the data contains and the range the simulation contains.

	Data		Simulation	
	Min	**Max**	**Min**	**Max**
Discharge temperature setpoint	18.6	19.9	16	23
Exhaust duct static pressure setpoint	220.4	229.2	215	235
Radiator heating network supply temperature setpoint	20	47.7	15	50
Supply duct static pressure setpoint	165.3	170.8	160	175
Ventilation network supply temperature setpoint	20	47.6	15	50

below 1500 ppm (REHVA). However, when looking at the data, it shows that the levels are mostly between 400-700 ppm. It is therefore chosen to set the thermal CO2 levels lower than 1000/1500 ppm. Temperature levels should be between 20.3°C to 23.9°C during winter and 23.9°C to 26.9°C during summer according to ASHRAE. Outside office hours, the temperature is allowed to be higher than recommended, since the HVAC system will save energy by not cooling the building. During spring and summer, it is again looked at the data and seen that there are often peaks to 27 °C. It is therefore chosen to set the values as shown in the table.

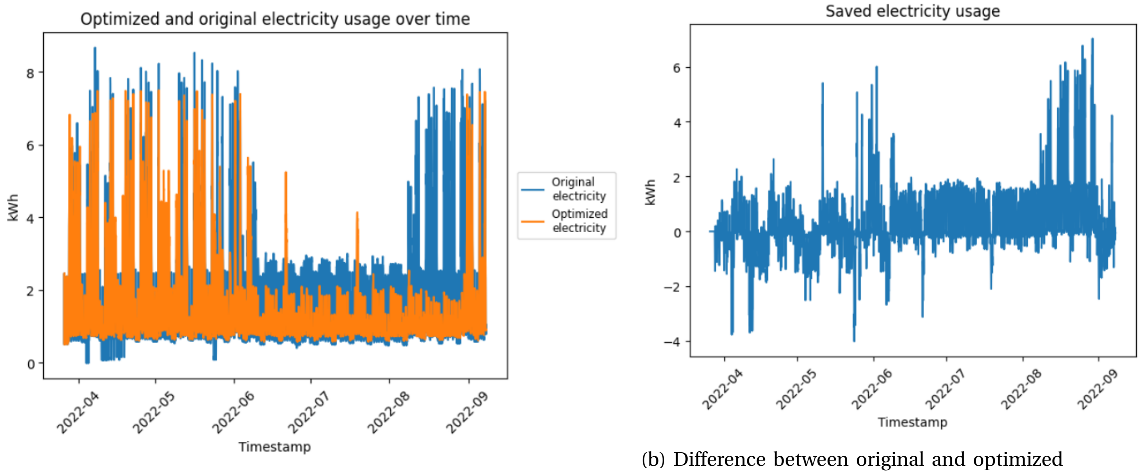
Table 9: Comfort levels regarding CO2 (shown in ppm) and temperature (shown in °C).

		Season			
		Winter	**Spring**	**Summer**	**Autumn**
CO2	Office hours	350-700	350-700	350-700	350-700
	Else	200-700	200-700	200-700	200-700
Temperature	Office hours	20-24	21-27	21-27	20-24
	Else	18-27	18-27	18-27	18-27

The test set contains a total of 166 days. The original test set used 7603.08 kWh, while the optimized set uses only 6204.92 kWh over the same period. This is a decrease of 18.39%. Using the mean costs of electricity in the Netherlands in February 2023, this results in (7603.08-6204.92) * €0.69 = €964.73 saved in 166 days [25]. To visualize this performance, several plots are made. Figure 18 shows on the left both the original and optimized electricity usage over the test set. The right figure shows the saved energy usage, which is computed as:

$$\text{Saved electricity usage}_t = \text{Original usage}_t - \text{Optimized usage}_t \tag{16}$$

Here, t stands for timestamp t. When the line is above zero, there was a saving. When the line is below zero, the simulation used more electricity than the original dataset. One should note that the lines cannot be compared exactly one-to-one, since the simulation predicts based on previous predictions. The left figure shows that during April and May, there were peaks in usage and the simulation followed those peaks. During June and July, the original usage was more stable. The optimized electricity usage during those months was more stable, using around 1.5 kWh instead of 2.5 kWh. There were also peaks in the original dataset during August, but the simulation remained stable. The right figure shows that during August, the most electricity was saved. Months such as April and May showed sometimes a peak where the original dataset used less energy than the simulation.



(a) Original and optimized electricity usage.

(b) Difference between original and optimized electricity usage.

Figure 18: Original and optimized electricity usage over the test set.

To see how the electricity usage is during the week, a close-up of week 34 is shown in Figure 19. It shows that the peaks up to 7.5 kWh in the original dataset are not in the optimized electricity usage. The optimized electricity usage shows a more stable line. The last two days (27 and 28 August) are Saturday and Sunday. It can be assumed that the educational building is closed and therefore uses less electricity.

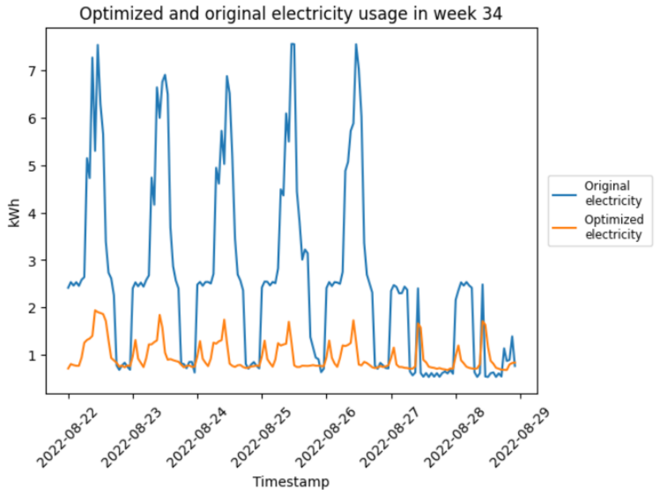
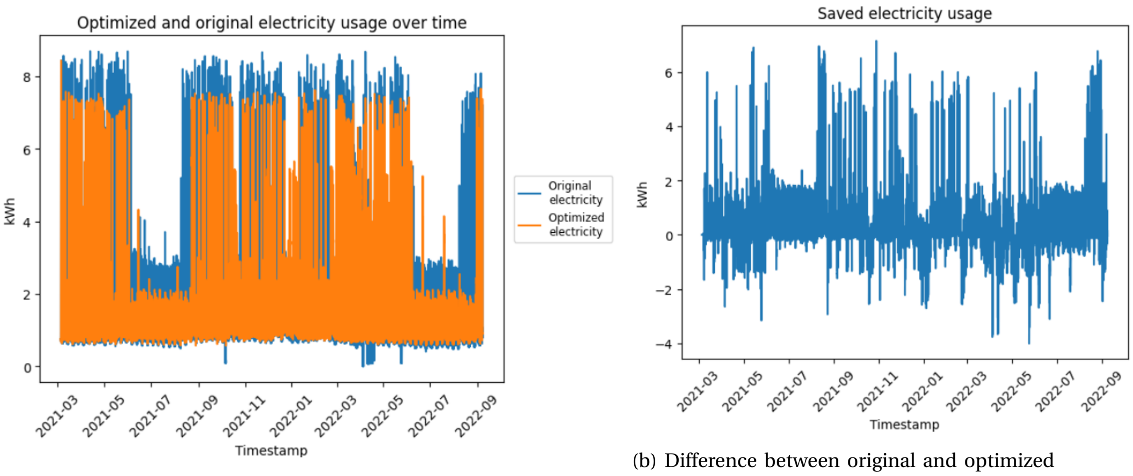


Figure 19: Actual versus predicted electricity values in Week 34 of 2022 (test set).

The dataset contains 166 days between April and September, which are relatively warm months. It could be that the simulation behaves differently during colder months. To analyze this, the simulation is also done over the whole dataset. The algorithm is trained on the first 70% of the data like before, but the simulation starts on day one of the full dataset instead of only the test set. It now simulates 552 days, with more seasons included. The original dataset uses 27100.44

kWh, while the simulation only uses 22326.25 kWh. This is a reduction of 17.62%. Again using the mean costs of electricity in the Netherlands in February 2023, this resulted in (27100.44-22326.25) * €0,69 = €3294.19 saved in 552 days. Figure 20 shows the same concept as before, but now the simulation has been applied to the whole dataset. It must again be noted that the lines cannot be compared exactly, since the simulation makes predictions based on previous predictions. It can be seen that during the summer period (July and August) the least amount of energy usage is used. During the rest of the year, there are both peaks in savings, but also that the original usage used less than the simulation.



(a) Original and optimized electricity usage.

(b) Difference between original and optimized electricity usage.

Figure 20: Original and optimized electricity usage over the full dataset.

6 Conclusion

This research consisted of three phases: pre-processing, prediction, and optimization. It became clear that the data needed to be cleaned since the sensors provided inconsistent data with large missing gaps of data. Extra features were added such as temporal values and external weather data.

When the dataset was finished, the prediction phase was executed. Four algorithms (Random Forest, XGBoost, LightGBM, and Lasso Regression) were used to predict the temperature, CO2, and electricity usage in the upcoming hour. Three methods were used for each algorithm: the default settings, a grid search, and PCA combined with a grid search. The default settings worked best for the Random Forest algorithm. XGBoost and Lasso performed best by using a grid search. LightGBM worked best for CO2 using a grid search, but for temperature and electricity the default settings. In each algorithm, the last method (PCA combined with a grid search) did not increase performance. During the pre-processing phase, it was found that there was multi-collinearity between the features. PCA can help with this problem by reducing the dimensionality and transforming the features into a new set of uncorrelated variables. It turned out that PCA did not improve the performance. A reason for that could be that PCA relies on linear relationships, and there can be non-linear relationships between the features. Another explanation could be that the number of components chosen by the algorithm could have lost vital information to analyze the correlation between variables. It turns out that using all the input features is more effective than dimensionality reduction. The algorithms can handle the correlated features themselves. Random Forest does this with feature randomness and the structure of the trees. Gradient Boosting does this with feature subsampling and regularization. Lasso does this by selecting by using built-in feature selection and selecting one feature from the correlated features. It is also noticeable that when PCA is used, a smaller model is chosen (e.g. less number of estimators). This is logical since the input features are less than the other methods. Overall the model that performed best was XGBoost, since it has relatively the best evaluation metrics and follows the peaks in the data. Random Forest also predicted well on the evaluation metrics, but slightly worse compared to XGBoost. Lasso did not follow the actual predictions accurately. LightGBM predicted peaks that were not in the actual data. Lastly, the algorithms do not appear to be overfitting, since the evaluation score on the training and test set score around the same values.

The last step was to apply the best algorithm in a simulation. XGBoost was used to predict CO2, temperature, and electricity usage in the next hour. It became clear that a full grid search of all setpoint combinations was computationally too heavy. Therefore, the setpoints were optimized individually. Based on plots of the data per season and literature research, the optimal thermal comfort settings were established. The simulation showed that the optimized electricity usage is more stable compared to the original electricity usage. A reason for this could be that changing the setpoints often costs more electricity, compared to having a constant level of setpoints. This is because the HVAC system needs to work to change the setpoints and thus the levels in a room, which costs electricity. The simulation shows that there is potential in this research. The original test set used 7603.08 kWh, while the simulation uses 6204.92 kWh. This is a decrease of 18.39%. Taking the mean electricity price in the Netherlands in February 2023, this results in (7603.08-6204.92) * €0.69 = €964.73 saved in 166 days. It must be noted that these numbers cannot be compared one-to-one, since the simulation makes predictions based on predictions. The test set contains the months of April to August, which are relatively warm months. To see how the simulation does in colder months, the models are trained on the training set, and the simulation is started on the full dataset. The electricity has decreased with 17.62% using the action plan of this research. This led to (27100.44- 22326.25) * €0.69 = €3294.19 saved in 552 days. During the months of July and August the least energy was used. The researched building is an educational building, thus it can be assumed there was a summer holiday. Throughout the year there were peaks where up to 6.5 kWh was saved, while there were also timestamps when the simulation did not save energy.

This research extended current literature in various ways. This approach was inspired by the literature, as described in Section 2. As Wang et al. described, the three most important parameters for Random Forest were analyzed by using k-fold cross-validation extensively. They have access to similar data, but also information on occupancy in the building. They compared Random Forest to Support Vector Regression, where Random Forest was the best algorithm. The current research applied Random Forest in an HVAC simulation, which was outperformed by Gradient Boosting (XGBoost). The authors did not apply their findings to hourly adjustable setpoints in the HVAC system as in the current research. They solely focused on the best algorithm for hourly prediction and did not apply any simulation to the HVAC system. Moreover, Bassi et al. compared XGBoost and LightGBM to estimate building energy usage. They also used an extensive grid search to find the optimal parameters, combined with using k-fold cross-validation. XGBoost scored R^2=0.9819 and LightGBM an R^2=0.9864. The current research shows similar results on this evaluation metric. Their research focuses only on comparing Gradient Boosting models to get the most accurate building energy prediction. Again, there is no application to the HVAC system, while this research does. This research also tests a regression model in an HVAC simulation, but it turned out not to be suitable for the data. Selamat et al. used a multi-objective optimization model. Here, violation of thermal comfort is taken into account, while minimizing electricity usage. By using this method, 12.4% of the energy usage is reduced without compromising extremely on the thermal comfort constraints. The difference with the current research is that the current research does not allow thermal comfort violations. The current research minimizes electricity usage, while also staying within the comfort zone. This could lead to higher electricity usage, but it does stay within the comfort zone of the building. It is up to the building owner to state what they think is important: lower costs, but occasionally violating thermal constraints, or relatively higher costs but staying within comfort range.

At the start of this research, the following research question was posed: “How can the electricity usage of HVAC systems in non-residential buildings be decreased compared to its current performance while maintaining a comfortable indoor climate? Moreover, which optimization algorithm performs best to optimize the HVAC system’s electricity usage?" It is found that implementing an hourly simulation of a real-time HVAC optimization algorithm is the answer to this question, which resulted in a positive outcome that promises good results during further research. The optimal algorithm for prediction is either XGBoost (Gradient Boosting) or Random Forest. The Lasso regression model resulted in a relatively worse outcome. It was also found that using all features as input performed better than feature selection using PCA.

Heroes B.V. can use this research to further extend their platform to save energy. Clients could reduce their CO2 footprint and save money. As explained before, the Dutch government has set goals to be climate neutral in the future. A carbon tax could become reality. Together with the Russian invasion of Ukraine which led to an increase in electricity prices, more companies realize they need to reduce their CO2 footprint. This research shows the potential of a dynamic multi-variable HVAC system, which will save electricity and therefore money for clients. The following section will elaborate on how Heroes B.V. can extend this research.

6.1 Limitations and further research

There were some limitations during this research. There was no information available on the type of HVAC system or any information in general on the used system. It was unknown what the building was already doing to optimize electricity usage. Moreover, the data was limited in regards that there was no information available on why the building owners chose those setpoints. Information on a wider range of setpoints in combination with more data would be valuable information. Moreover, no testing could be done on a real life HVAC system. This would tell more about how long it takes for the room to process the new setpoints.

Regarding future research, the sensors in the building could be programmed so that they would also collect values regarding air quality, humidity, and occupancy. It would also be useful to have information on whether the windows in the building are open or closed since this could quickly change the values in a room. These could be added to the model for higher prediction accuracy and better regulation of the HVAC system. Moreover, this research now focuses on a whole building since the data per room was inconsistent. When the sensors are working for each room, an effective room optimization model could be implemented. Together with the occupancy rate, this could save energy usage more if empty rooms could be left out of the HVAC optimization. Another research idea is to implement daily optimization instead of hourly optimization. When more data is gathered, it could become possible to pick a setpoint during office hours that could be effective. Lastly, different types of algorithms can be investigated. The current research focuses on tree algorithms, gradient boosting, and regression. Models such as deep learning or reinforcement learning could be applied to see if it would improve performance.

There are also ways to extend this research. The current research is applied to one building, but it could be scaled to multiple buildings. The buildings should be clustered by their building characteristics and energy usage patterns. The buildings that are similar to each other can be optimized in the same type of way. It would be important to ensure that the data of each building is clean, the sensors are working and the data is consistent. Moreover, different building types can be researched. The current research analyzes an educational building, but it can be extended to residential, commercial, or industrial buildings. Each has a different HVAC system design and could also have different goals regarding electricity usage. The final goal would be to implement a real-time implementation to different types of buildings to dynamically adjust the HVAC system.

References

[1] Rijksoverheid. Klimaatverandering en gevolgen. https://www.rijksoverheid.nl/onderwerpen/klimaatverandering/gevolgen-klimaatverandering. visited 28-11-2022.

[2] Rijksoverheid. Voortgang klimaatdoelen. https://www.rijksoverheid.nl/onderwerpen/klimaatverandering/voortgang-klimaatdoelen. visited 28-11-2022.

[3] Diana Ürge-Vorsatz, LD Danny Harvey, Sevastianos Mirasgedis, and Mark D Levine. Mitigating co2 emissions from energy use in the world's buildings. *Building Research & Information*, 35(4):379–398, 2007.

[4] Chang-Ming Lin, Sheng-Fuu Lin, Hsin-Yu Liu, and Ko-Ying Tseng. Applying the naïve bayes classifier to hvac energy prediction using hourly data. *Microsystem Technologies*, pages 1–15, 2019.

[5] Behnam Zakeri, Katsia Paulavets, Leonardo Barreto-Gomez, Luis Gomez Echeverri, Shonali Pachauri, Benigna Boza-Kiss, Caroline Zimm, Joeri Rogelj, Felix Creutzig, Diana Ürge-Vorsatz, et al. Pandemic, war, and global energy transitions. *Energies*, 15(17):6114, 2022.

[6] De Nederlansche Bank. Better carbon emissions pricing. https://www.dnb.nl/en/green-economy/carbon-pricing/. visited 29-11-2022.

[7] Rui Yang and Lingfeng Wang. Optimal control strategy for hvac system in building energy management. In *PES T&D 2012*, pages 1–8. IEEE, 2012.

[8] Yuzhen Peng, Adam Rysanek, Zoltán Nagy, and Arno Schlüter. Case study review: Prediction techniques in intelligent hvac control systems. In *9th International Conference on Indoor Air Quality Ventilation and Energy Conservation in Buildings (IAQVEC 2016)*, 2016.

[9] Ayesha Asif and Muhammad Zeeshan. Indoor temperature, relative humidity and co2 monitoring and air exchange rates simulation utilizing system dynamics tools for naturally ventilated classrooms. *Building and Environment*, 180:106980, 2020.

[10] Faye C McQuiston, Jerald D Parker, and Jeffrey D Spitler. *Heating, ventilating, and air conditioning: analysis and design*. John Wiley & Sons, 2004.

[11] Fergus Nicol and Michael Humphreys. Maximum temperatures in european office buildings to avoid heat discomfort. *Solar Energy*, 81(3):295–304, 2007.

[12] Centers for Disease Control and Prevention. Indoor environmental quality: Hvac management. https://www.cdc.gov/niosh/topics/indoorenv/hvac.html. visited 13-01-2023.

[13] Hazlina Selamat, Mohamad Fadzli Haniff, Zainon Mat Sharif, Seyed Mohammad Attaran, Fadhilah Mohd Sakri, and Muhammad Al'Hapis Bin Abdul Razak. Review on hvac system optimization towards energy saving building operation. *International Energy Journal*, 20(3), 2020.

[14] Roman Timofeev. Classification and regression trees (cart) theory and applications. *Humboldt University, Berlin*, 54, 2004.

[15] Zeyu Wang, Yueren Wang, Ruochen Zeng, Ravi S Srinivasan, and Sherry Ahrentzen. Random forest based hourly building energy prediction. *Energy and Buildings*, 171:11–25, 2018.

[16] Abnash Bassi, Anika Shenoy, Arjun Sharma, Hanna Sigurdson, Connor Glossop, and Jonathan H Chan. Building energy consumption forecasting: A comparison of gradient boosting models. In *The 12th International Conference on Advances in Information Technology*, pages 1–9, 2021.

[17] Deanna N Schreiber-Gregory. Ridge regression and multicollinearity: An in-depth review. *Model Assisted Statistics and Applications*, 13(4):359–365, 2018.

[18] Husein Perez and Joseph HM Tah. Improving the accuracy of convolutional neural networks by identifying and removing outlier images in datasets using t-sne. *Mathematics*, 8(5):662, 2020.

[19] Brian Cho, Teresa Dayrit, Yuan Gao, Zhe Wang, Tianzhen Hong, Alex Sim, and Kesheng Wu. Effective missing value imputation methods for building monitoring data. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 2866–2875. IEEE, 2020.

[20] Ming-Chang Wang, Chih-Fong Tsai, and Wei-Chao Lin. Towards missing electric power data imputation for energy management systems. *Expert Systems with Applications*, 174:114743, 2021.

[21] Huajing Sha, Peng Xu, Chonghe Hu, Zhiling Li, Yongbao Chen, and Zhe Chen. A simplified hvac energy prediction method based on degree-day. *Sustainable Cities and Society*, 51:101698, 2019.

[22] Yaqing Liu, Yong Mu, Keyu Chen, Yiming Li, and Jinghuan Guo. Daily activity feature selection in smart homes based on pearson correlation coefficient. *Neural Processing Letters*, 51(2):1771–1787, 2020.

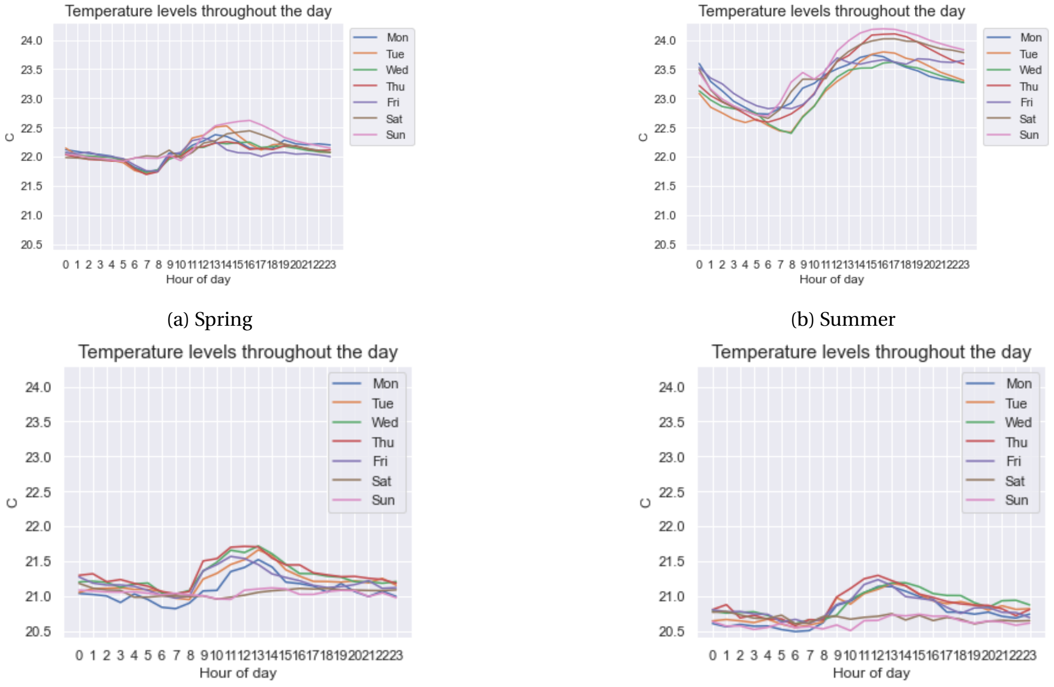
[23] Sheetal Girase, Debajyoti Mukhopadhyay, et al. An item-based collaborative filtering using dimensionality reduction techniques on mahout framework. *arXiv preprint arXiv:1503.06562*, 2015.

[24] Lindsay I Smith. A tutorial on principal components analysis. 2002.

[25] Overstappen.nl B.V. Stroomprijs per kwh. `https://www.overstappen.nl/energie/stroomprijs/`. visited 28-02-2023.

Appendices

A Y Values throughout the day



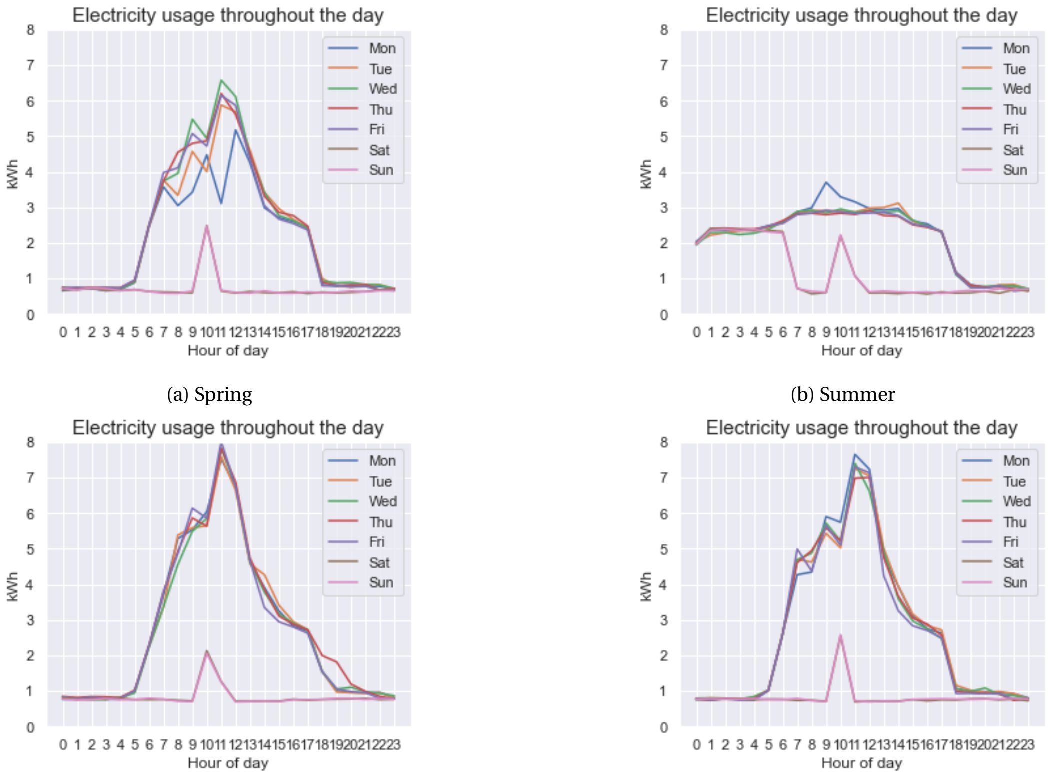
(a) Spring

(b) Summer

(c) Autumn

(d) Winter

Figure 21: Temperature levels throughout the day per season.



(a) Spring

(b) Summer

(c) Autumn

(d) Winter

Figure 22: Electricity throughout the day per season.