Effects of temporal resolution on data mining and machine learning algorithms in the built environment

Liao, Roy

Award date:
2021

Technische Universiteit
Eindhoven
University of Technology

Department of the Built Environment
Building Services

Effects of Temporal Resolution on Data Mining and Machine Learning Algorithms in the Built Environment

Master Thesis

Juo Yu Liao (1573640)

Supervisors:
Prof. Ir. W. (Wim) Zeiler
Ir. W. (Waqas) Khan
Eindhoven, August 2021

Declaration concerning the TU/e Code of Scientific Conduct for the Master's thesis

I have read the TU/e Code of Scientific Conduct[i].

I hereby declare that my Master's thesis has been carried out in accordance with the rules of the TU/e Code of Scientific Conduct

Date

Name

ID-number

Signature 

Insert this document in your Master Thesis report (2nd page) and submit it on Sharepoint

[i] See: http://www.tue.nl/en/university/about-the-university/integrity/scientific-integrity/
The Netherlands Code of Conduct for Academic Practice of the VSNU can be found here also.
More information about scientific integrity is published on the websites of TU/e and VSNU

Acknowledgements

This project is part of NWO 647.003.001 about using small data and big data on neighborhood energy and data management integration system.

Abstract

This thesis project presents an approach to apply data mining and machine learning for building energy consumption data. Specifically, it analyses the influence that different temporal resolutions of time series have in both pattern identification and energy consumption forecast in the domains of built environments. Various techniques were chosen due to their profound performance and suitability including, Spearman rank-order correlation, breakout detection, k-means clustering, and Long Short-Term Memory (LSTM) neural networks. The architecture of LSTM is also affected because changes in temporal resolution provoke an increment or decrement in the quantity of data resulting in necessary adaption for architecture. Previous researches in energy consumption prediction and load pattern analysis have mainly focused on model performance improvement rather than defining the number of data samples during a specific range of time. These studies attempt to investigate the relations of different machine learning algorithms, such as Support Vector Machines and Convolutional Neural Networks, and the impact of temporal resolutions. However, the influence of temporal resolution has not yet been analyzed using comprehensive and high-resolution (low frequency) data sets. This study aims to improve the consistency of the modeling techniques and define a best-unified resolution of data for different applications in the built environment to improve the energy efficiency of the power system. The results show that the temporal resolution of the data significantly affects data mining and machine learning outcomes in the built environment, and this effect is positive when the time series captures the pattern of the predicted frequencies. Data that are lower than the hourly resolution display more significant load patterns. In contrast, the 15-minute resolution data performs the best results that present typical commercial building energy load curves and make energy consumption predictions in the balance with accuracy and processing time.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Buildings represent a large portion of energy consumption and environmental emissions in urban areas. According to the International Energy Agency (IEA), buildings account for approximately 30% of total final energy consumption [1]. More than 70% of electricity generation comes from non-renewable resources [2]. Many countries experience energy shortages due to the increasing energy generation cost and rapid depletion of non-renewable resources. The need for energy efficiency in all sectors has raised concern from government authorities worldwide [3]. Energy efficiency is one of the crucial issues that the world is facing, and government authorities are gaining awareness of this need.

The advent of smart meters enables energy consumption to be measured and collected at all levels from the grid to the building's main circuit. Smart meters are a component of an advanced metering infrastructure, which consists of a communication network, a data management system, and an optional gateway [4]. Meter reading data is collected in the distribution network at different collection frequencies, which means that a large amount of data at different levels is influx into the system. In particular, advances in SCADA systems allow energy consumption data to be sampled at high frequency (1Hz), enabling meters to measure, store and transmit high-quality data at the high temporal resolution, where the amount of data can grow exponentially [4]. One of the key issues for energy efficiency is handling large amounts of data when balancing processing time and analysis quality.

As electricity cannot be stored and conserved efficiently and easily, forecasting and characterization of energy consumption are effective approaches to improve energy efficiency. Energy consumption forecasting is to explores the connection between the demand patterns and the supply availability of electricity. Forecasting and characterizing the electricity demand of a building or a neighborhood level is important for implementing urban energy management and efficient power-system planning [5]. Demand load patterns and supply control can be identified by accurate predictions of electricity consumption, which can be achieved with load profile data analysis. In this case, the 6-minute resolution data is usually used for building performance decisions on the demand side. At the same time, the grid operator makes decisions after every 15 minutes, indicating the conflicting interests of the demand and supply sides. However, in demand-side management, the resolution of data and signals should be aligned from both sides in terms of better implementation efficiency. Therefore, further study needs to investigate the optimal temporal resolution data for both the supply and demand sides.

In general, the impact of temporal resolutions on machine learning or data mining of energy forecasting has not been explored in detail in the existing literature. The operational energy behavior of buildings is highly dependent on various non-linear variables, including building physics [6], functional characteristics, household information, and meteorological, as well as temporal properties [7] [8]. The past researches show that energy consumption load metering in the built

environment has been shown at a low temporal resolution. Energy consumption load profiles are generally gathered for different dwelling types at different sampling frequencies from 1 to 30 minutes [9]. It is still an open question to forecast electricity demand at high-resolution monitoring's relative utility.

The Recent studies [7, 10, 11, 12, 13, 14] have extensively compared the performances of different techniques. Still, a minor focus has been put on studying the effects of temporal resolution resampling on data mining and machine learning applications. Ushakova and Mikhaylov [15] emphasize the importance of considering both broad time scales and temporal resolutions to describe load profile features and household consumption forecasting accurately. The impact of temporal resolution can be significant as the consumption profile fluctuates at a high temporal resolution [16]. The information of the electricity load profile is lost at low temporal resolution. Hence, the load profile dynamics become increasingly biased when a lower temporal resolution is envisaged. Thus, the load profile should be sampled at a more fine-grained level to present its behavior more accurately.

The primary focus of this thesis project is to analyze the effects of temporal resolution on data mining and machine learning in the built environment. The research aims to understand the influence that each resolution has on the actions or assessments of machine learning and data mining models by varying the aggregation granularity of the dataset across several temporal resolution scales. This study aims to improve the consistency of the modeling techniques and define a best-unified resolution of data for different applications in the built environment to improve the energy efficiency of the power system.

1.1.1 Data-driven model in building energy performance forecasting

Building demand estimation has been performed using traditional engineering software packages (e.g., EnergyPlus) embedded with structural, geometric, and material building properties [17]. Nowadays, advancements in smart metering technology lead to the increasing number of smart meters (SM) installations in houses. The increment of smart meters has enabled researchers to develop a data-based approach to forecasting building energy consumption with near real-time data streams from each building [18]. Recent studies by Zhou et al. [19] focus on describing the challenges and opportunities related to SMs data for more intelligent energy management. Traditional engineering-based energy forecasting relied on data simulation, whereas sensor-based approaches leverage real-world data from the built environment.

Compared to traditional engineering-based methods, sensor-based data-driven approaches require fewer input data and low complexity, which shows that the new approach is superior to the conventional physics-based methods [7]. Data-driven modeling techniques can provide fast and highly accurate forecasting [5]. Data-driven modeling can give accurate energy forecasting as long as model selection and parameter setting are suitable for the applications and assessments.

Data-driven modeling works on data acquired from smart energy meters, building management systems, and weather stations. The acquired data is used in a machine-learning algorithm to extrapolate the relations between energy load consumption and variables impacts such as temperature, occupancy, and time effects [13]. Standard techniques used by researchers include statistical methods and machine learning models [20]. Autoregressive Integrated Moving Average (ARIMA) models are one of the statistical methodologies that are considered the basic and the most general form of time series forecasting technique [21]. Wang and Meng [13] used the ARIMA model to forecast energy consumption for the entire Hebei province in China. The ARIMA models are suitable and preferable for short-term forecast of time series forecasting due to the simplicity of their structure [20]. However, the main disadvantage of ARIMA models is that ARIMA models fail to capture the time series' nonlinear patterns while performing long-term predictions [22]. Recently, machine learning methods such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Long Short Term Memories (LSTM) have gained popularity in energy demand forecasting. The significant advantage of machine learning models is their flexible nonlinear modeling capability [7]. The models can adapt based on the data features and detect nonlinear patterns of

Temporal resolution	References
1-sec	[17], [23]
2-sec	[23]
5-sec	[23]
10-sec	[23]
15-sec	[23]
30-sec	[23]
1-min	[22], [24], [25], [23]
2-min	[23]
5-min	[23]
10-min	[5], [23], [26], [27], [28]
15-min	[18], [23], [29], [30], [31], [32]
30-min	[23], [26], [29], [31], [33], [34], [35], [36]
1-hour	[5], [24], [25], [26], [29], [30], [31], [33], [37], [38], [39], [40], [41]
2-hour	[26], [31], [33]
6-hour	[31], [42]
12-hour	[31]
Daily	[5], [24], [29], [42], [37], [41], [43], [44]
Weekly	[24], [42], [37], [43]
Monthly	[3], [42], [41], [45]

Table 1.1: Literature published related to different temporal resolution

the data [23]. This advantage leads to a better time series forecasting performance for building energy consumption using machine learning and deep learning models.

1.1.2 The impact of temporal resolution on machine learning and data mining

The performance of machine learning and data mining varies according to diverse settings. The characteristics of building datasets selected for different applications in the built environment have demonstrated important implications on the adopted model performance. The parameter selection, the temporal resolution of the data, and inherited errors during the data collection phase are likely to influence the mining and learning outcome. The temporal resolution of data affects uncertainty in all kinds of machine learning models. Dynamical resampling is frequently used to investigate the relation between input and output variables. Various forecasting resolutions have been studied with different temporal resolutions by multiple researchers, as shown in Table 1.1.

Machine learning has recently been broadly used in energy demand forecasting on account of energy efficiency improvement in the built environment [3, 5, 24, 42, 41, 43]. Studies have been conducted at varying temporal resolution scales. Energy consumption forecasting is conducted in [31] from using monthly data, weekly data to daily data. As more energy consumption data is accessible at a finer temporal resolution, researchers could perform modelling on resolution of 12-hourly [31], 6-hourly [31, 42], 2-hourly [26, 31, 33], and up to hourly scale [5, 24, 25, 26, 29, 30, 31, 33, 37, 38, 39, 40, 41]. Furthermore, former studies have conducted and compared data on a scale of 30 minutes [23, 26, 29, 31, 33, 34, 35, 36] , 15 minutes [18, 23, 29, 30, 31, 32], and even in intervals of 10 minutes [5, 23, 26, 27, 28]. Heghedus et al. (2018) [17] demonstrated short-term forecasting for in-home energy consumption at a resolution of only 1 second. While the trend has been moving towards utilizing finer resolution and more temporally granular data in forecasting, the impact of temporal resolution has yet to be analyzed using a comprehensive and systematic resolution energy consumption dataset.

Some studies attempt to investigate the relations of different modeling performances and the impact of temporal resolutions. Various machine learning and data mining methods are presented

in previous studies. Jain et al. (2014) [5] perform a Support Vector Regression (SVR) model to investigate the impact of temporal resolution on performance accuracy for residential buildings. They study the different temporal resolutions from 10-minutely, hourly, to daily. The results indicate that the most effective models are built with hourly consumption for SVR models, with a coefficient of variation (CV) of 2.16% [39]. Amarasinghe et al. (2017) [39] also used hourly electricity load data to compare machine learning models' performances. The results are obtained from the diverse methodology, including Convolutional Neural Networks (CNN), LSTM sequence-to-sequence (LSTM S2S), Factored Restricted Boltzmann Machines (FCRBM), ANN, and SVM. Experimental results show that CNN outperformed SVR while producing comparable results to the ANN and other deep learning methodologies at hourly data resolution. Marino et al. (2016) [25] compare the standard LSTM modeling performance using hourly and 1-minutely data resolution. The LSTM algorithm produces accurate results with hourly data but fails to perform well with 1-minutely resolution data. Kim and Cho (2019) [46] also compare the performance of LSTM and proposed CNN-LSTM modeling at hourly and 1-minutely resolution data. The lower the resolution, the lower the modeling error rate obtained. Luesis et al. (2017) [33] prove that the forecasting error is reduced with coarsening temporal resolution since the load fluctuation is smoothed at high resolution. For example, a power-intensive electric oven may be used for 30 minutes in total each morning between 7 AM and 9 AM in a household. At a 30-minutely resolution, the time and magnitude of the peak will be different for each 30-minutely interval. If the 120-minutely resolution is used instead, the model will capture and learn to predict the observed peak load. With a 30-minutely resolution, the peak load may occur in any of the adjacent half-hour intervals from day to day, hindering the predictability of the peak load. Forecast techniques will act conservatively to minimize the forecast error at a 30-minutely resolution. Hence, the forecast models will issue a forecast value close to the average during the forecast period. Peak load information is lost at coarse temporal resolution. On the other hand, Cao et al. (2020) [43] conclude that electrical load prediction is more likely to achieve higher accuracy at finer temporal resolution. Zhou et al. (2017) [26] carry out model forecasting with resolutions of 10-minutely, 30-minutely, 1-hourly, and 2-hourly, and the results show that a low temporal resolution has low prediction accuracy most time in a day

Data analysis of load profile features offers information regarding the impact of temporal resolution on the change in the magnitude of the peak and trough load. Hernandez et al. (2020) [23] find the sample mean of the consumption load decreased by 17.67% at the coarser temporal resolution, ranging from 1 second to 30 minutes. A higher peak reduction of 59.09% happens at coarser resolutions. Kristensen et al. (2017) [42] discover that information about important thermodynamic processes seems to be leveled out or even lost with decreasing the temporal resolution of the training data. Bassamzadeh and Chanem (2017) [30] use the Bayesian network (BN) based method to discover dependency relations between contributing variables of electricity demand at various temporal resolutions. The results show that the learned BN structures for demand modeling at hourly and 15-minutely data resolution are different. Variables (e.g., temperature) present a lower dependency on electricity demand in BN structure at 15-minutely data resolution. This result can be explained that aggregating data from 15-minutely to 1-hourly may cause loss of fine-scale variations and thus discover new dependency structures in the dataset. Previous studies have implemented data analysis and model forecasting at different temporal resolutions. However, it is still needed to conduct a thorough investigation on the impact of temporal resolution on data mining and machine learning due to the lack of comprehensive study at various temporal scales of data sets.

Although the current trend is to use more temporally granular data sets in the built environment, the influence of temporal resolution has not yet been analyzed using comprehensive and high-resolution (low frequency) data sets. Most of the studies mentioned in Table 1.1 use a daily or even an hourly or 30-minutely time resolution, but they do not contain sufficient information to specific actions or assessments for further applications accurately. Quite a few papers [29, 30, 37, 39, 40, 41] evaluate the error due to the use of coarse-grained data but never investigated temporal resolutions lower than 5-minutely. Moreover, a shortcoming is that most studies cover a short-term time horizon which involves a reduced period chosen to characterize critical

features of calendar effects. Examples are covering weekdays and weekends, different times of the day, and different seasons. The time horizon envisaged in Table 1.1 is typically restricted to minutes, a few hours, or a few days, which gives inaccurate results since it does not take seasonality into account. The systematic review of discussion regarding the comparison of various modeling techniques and relations among the use of input data and energy performance forecasting in the built environment can be found in [7, 11, 47, 12, 13, 14, 16]. However, further study is still needed due to the discrete temporal resolution of input data sets and the lack of investigation regarding the impact of temporal resolution on data mining and machine learning in the built environment.

1.2 Aim and Objectives

This thesis aims to study the effects of multiple time resolution frequencies on machine learning models in the built environment. The data is kindly provided by Kropman Installatietechniek, a building construction consultant company in the Netherlands. The case study building, Kropman Office in Breda, is situated in the west of the city Breda, a town in the southern part of the Netherlands. Building energy consumption data is collected with different time resolutions (1-minutely, 15-minutely, 1-hourly, etc.) for information extraction pattern identification and forecasting. The results will be compared among the performance indicators to evaluate the outcome differences. The best setting among all the demonstrated models will be proposed to the Kropman company. The specific objectives of this research are to:

1. Identify the effects of various time resolution frequencies on data mining and machine learning in the built environment.
2. Specify use cases on different time resolutions for Kropman building data.
3. Analyze the effects of resampling (Interpolation) due to data-imbalanced conditions on modeling.
4. Formulate and verify a general estimate of uncertainty due to resampling.

1.3 Problem Statement

This thesis will discuss how the influence of different temporal resolutions in the domains of the built environment impacts the results of both data mining outcomes and machine learning performance. Also, it will be studied the relation with univariate and multivariate features between building energy consumption and related building data such as weather conditions.

Therefore, based on the previous statements and having identified a knowledge gap, the study will follow the research question following:

Main question:

How the choice of temporal resolution frequency influences data mining and machine learning modeling in the built environment?

To answer the main research question, the question is divided into the following sub-questions.

Sub-question:

1. What time resolution is important for building performance control and should be used for analysis?
2. Which data mining techniques are suitable for extracting knowledge of pattern information?
3. Which machine learning algorithms are the best way to build good predictions of building performance?

4. What is a good approach to compare results at different temporal resolutions?

1.4 Thesis Outline

The remainder of this thesis is structured as follows:

- Chapter 2 - Theoretical Background: This chapter gives a brief introduction to the techniques that will be used in this study.
- Chapter 3 - Methodology: This chapter provides an overview of the designed experiment.
- Chapter 4 - Results and Analysis: This chapter presents the investigation among data mining, machine learning, and temporal resolution.
- Chapter 5 - Discussion: This chapter discusses and evaluates the limitation of the study and highlights opportunities for future work.
- Chapter 6 - Conclusion: This chapter summarises the research.

Chapter 2

Theoretical Background

2.1 Data Mining

Data mining (DM) is the process of extracting and discovering hidden knowledge in large data sets and involves approaches at connecting machine learning, statistics, and database systems. To explore the data set, the temporal features extraction is based on the most simplified statistics calculation, including the normalization model and Spearman rank-order correlation (ROC) coefficient. The normalization effect allows a more appropriate comparison of the electricity consumption magnitude of buildings at different temporal resolutions [48]. Normalizing the energy consumption time series makes the data fit the interval [0-1]. This process makes it possible to identify the time series with equivalent consumption patterns instead of identical consumption volumes. Normalization metric is intended to provide a basis for comparison between temporal resolutions and is used as a key indicator in numerous benchmarking and performance analysis techniques. As the focus of this paper is data mining and machine learning on energy consumption data, we normalize by:

$$Normalization = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{2.1}$$

The other set of feature extraction is calculation related to how much influence outside weather has on the consumption of a building. As recommended in the work of Miller and Meggers, a process of utilizing the Spearman Rank Order Correlation (ROC) coefficient is applied to approximate the correlation between outside weather conditions and the energy consumption of a building [48]. The ROC essentially ranks the items in two different lists (X and Y), and the ratio quantifies whether these lists are positively or negatively correlated. In this study, the two variables are weather data, i.e., outdoor temperature, solar radiation, air pressure, and relative humidity, and energy consumption. The coefficient ranges is *-1* (highly negative correlation) and *+1* (highly positive correlation). For example, when the correlation is positive (when ROC is positive and close to *+1*), where energy consumption is cooling sensitive and consumption increases with increasing temperature, and when the correlation is negative (when ROC is negative and close to *-1*), where energy consumption in the time range is sensitive to heating, as consumption increases with decreasing temperature.

Spearman Rank Order Correlation (ROC) coefficient

$$r_s = \rho_{rg_X, rg_Y} = \frac{cov(rg_X, rg_Y)}{\sigma_{rg_X} \sigma_{rg_Y}} \tag{2.2}$$

where:
- ρ = the usual Pearson correlation coefficient, but applied to the rank variables
- $cov(rg_X, rg_Y)$ = covariance of the rank variables
- σ_{rg_X} = standard deviation of the rank variable X
- σ_{rg_Y} = standard deviation of the rank variable Y

The Pearson correlation coefficient is implemented and compared to Spearman's rank-order correlation coefficient as a control group.

Pearson correlation coefficient

$$\rho X, Y = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \tag{2.3}$$

where: $cov(X, Y)$ = covariance of X and Y
σ_X = standard deviation of X
σ_Y = standard devistion of Y

2.2 Signal Decomposition

A valuable technique for exploring energy consumption data is seasonality and trend decomposition. In forecasting and temporal data mining, temporal data from different periods that are influenced by seasonal factors such as the month of the year or the day of the week tend to exhibit similar types of behavior, i.e., seasonal patterns. Building energy data belongs to this category can be applied to the same feature extraction techniques commonly used in financial or social science analysis.

These techniques typically attempt to decompose time series data into several components that represent the underlying nature of the data [49]. For example, energy consumption data for commercial buildings tend to be cyclical in their weekly schedules, where the occupants are typically white-collar professionals who go to work at specific times during the week and return home at particular times. Weekends are usually free time with little or no activity. These user behavior characteristics develop relatively predictable patterns, called seasonality in time series analysis. Seasonality is a fixed and known consistent cycle and is a feature that is often extracted before creating a predictive model. In addition, trends are another common feature of temporal data. A trend is an increase or decrease in data over time, usually without following a specific pattern. Trends are generally caused by less systematic factors than seasonality and are often due to external influences. For building energy consumption, trends are expressed as gradual variations in consumption over long periods of time, such as weeks or months [48].

The classical decomposition method of time series originated in the 1920s with the work of Frederick R. Macaulay of the National Bureau of Economic Research on the ratio to moving average approach [50]. The details of the internal algorithm of the classical decomposition procedure are described by [51]. There are two methods of classical decomposition: an additive decomposition and a multiplicative decomposition. In this study, the additive decomposition is used to avoid any complications with very low load values at various temporal resolution. The additive decomposition is written as :

Time Series Decomposition

$$Y_t = f(S_t, T_t, E_t) = S_t + T_t + E_t \tag{2.4}$$

where: Y_t = data at period t
S_t = seasonal component at period t
T_t = trend-cycle component at period t
E_t = remainder (or irregular or error) component at period t

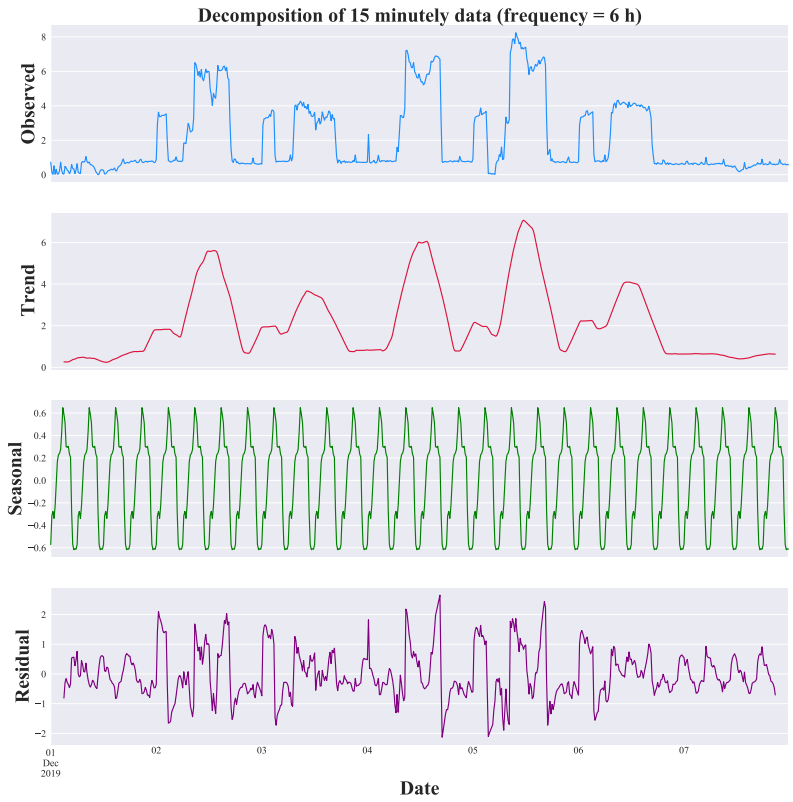

Figure 2.1: Example of decomposition in the energy load time series data.

2.3 Breakout Detection

A class of feature extraction is associated with capturing typical and atypical usage patterns from building energy consumption data. These features aim to quantify whether buildings have daily or weekly consistency, such as whether certain building types have specific types of usage patterns. The concept of pattern consistency is related to the level of fluctuations in building energy consumption over a long period of time (e.g., a year). In general, the magnitude of fluctuations is related to seasonal variations in the building's operating schedule. For example, the overall energy consumption of commercial buildings is typically more consistent over a year than that of schools or universities, resulting in less fluctuation in building energy consumption patterns.

In this case study, pattern consistency was determined by forking a library of breakout detections from the R programming package [52]. The package was developed by the social media company Twitter to detect breaks in time series data. The breakout function decomposes a time series into segments of three types, namely Steady state, Mean shift, and Ramp up/down.

- Steady state: The time series follows a fixed mean;
- Mean shift: The time series jumps directly from one steady state to another;
- Ramp up/down: A gradual increase or decrease in the value of the time series from one steady state to another over a fixed period.

A break, also known as a mean shift or ramp up/down in time series data, indicates a significant change in the mean value of the time series data over a sustained period, as shown in Figure 2.2.

The goal of breakout detection is to identify the point of change when the probability distribution of the time series changes. Breakout detection is a univariate statistical analysis technique that can identify unexpected changes in building performance by using a single time series that does not require parameter tuning. Some of the benefits of breakout detection are that it does not require the same amount of data as machine learning algorithms to run effectively, nor does it require a combination of techniques to detect building performance problems accurately. This capability can be used to prevent non-critical alarms in building management systems and improve building performance.

Breakout detection is based on the E-Divisive with Medians (EDM) [52] calculation. EDM uses an enhanced variant of energy statistics that is more flexible to anomalies by using robust statistics (i.e., median). The idea of energy statistics is to compare the distance between the mean values of two random variables contained in a continuous larger time series. However, the presence of anomalies can limit the validity of using the mean in this process since a single anomaly can have a significant effect on the mean of the time series. For this reason, the EDM technique is based on a more robust median. Based on the robust median, the EDM technique exists as a way to create a computationally tractable process for determining whether a new block of time series data is significantly different from the previous one by using advanced distance statistics that are exceptionally robust. Details of the mathematical background used in the package can be found in the study by James et al. [53].

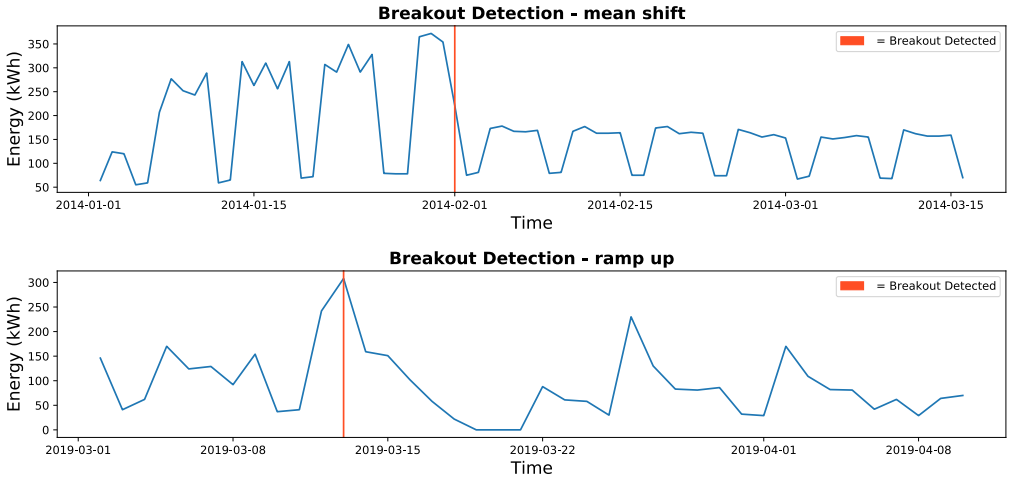

Figure 2.2: Examples of breakout detection by evaluating changepoint in the time series data.

2.4 K-means Clustering

As mentioned in [54] and [55], the K-means clustering algorithm is the most prevalent technique for generating building energy consumption patterns. K-means is a simple and robust algorithm for partitioning n observations into k clusters. It is done by evaluating the similarity between n observations and cluster centroids using the squared Euclidean distance as the clustering criterion. K-means is initialized with a random cluster centroids. Each observation is assigned to the nearest cluster centroid, which is updated to the value obtained by averaging all objects assigned to that cluster. This process is repeated until the algorithm converges. The convergence of the algorithm to the optimal global solutions depends on the initial partitioning. Therefore, the algorithm must be run several times with different initialization [56]. A flow chart of the K-means clustering algorithm is shown in Figure 2.3, and the figure is adapted from [57].

In unsupervised learning, there is no natural quantification of the discrepancy between model and truth because the true clusters are unknown. Robust cross-clustering validation is then carried

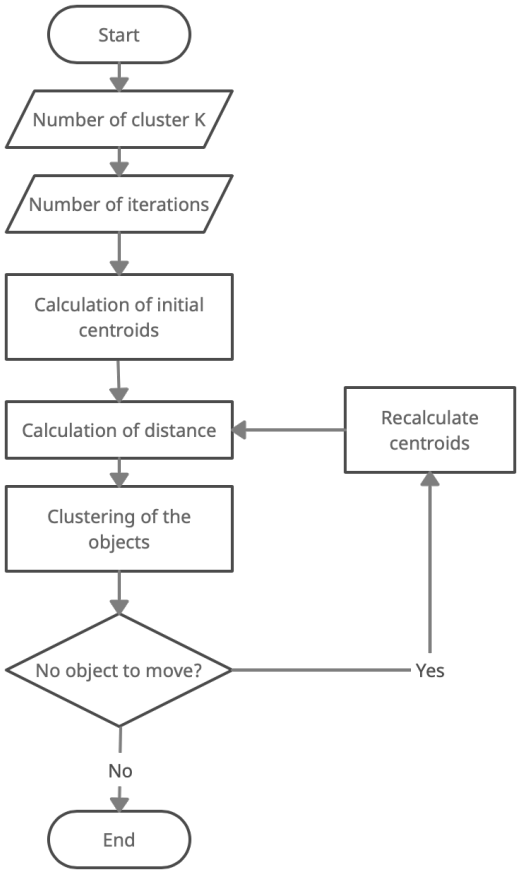

Figure 2.3: Flow chart of the K-means algorithm.

out, where the selection of optimal cluster number is first realized from the computation of cluster evaluation indices [55]. Based on the findings of Chicco [58], Tureczek and Nielsen [55], the study selected three indices to assess the different attributes of clusters, namely the Mean Index Adequacy (MIA), the Cluster Dispersion Index (CDI), and the Davies-Bouldin Index (DBI). The three selected evaluation indexes are also most commonly used for building energy clustering [54]. Although none of the indices identify the true underlying structure, their values for different cluster counts provide an indication of how many clusters are retained in the final clusters.

By plotting the progression of the indices as a function of visual inspection for cluster number, it is possible to identify abrupt changes or fluctuating patterns that can help select the number of clusters within the data set. We evaluate several indices jointly, as the combination can be applied to strengthen the argument for the selection of a specific number of clusters.

Mean Index Adequacy (MIA)

The *MIA* index calculates the square of the average distance from each member of a cluster to the cluster centroid and scales it by the number of classes K:

$$MIA = \sqrt{\frac{1}{K}\sum_{k=1}^{K} d^2(C_k)} \tag{2.5}$$

where: C_k = calculated center of cluster k
$d^2(C_k)$ = squared average Euclidean distance within cluster k

The *MIA* index measures the within-cluster dispersion. A high *MIA* value corresponds to large distances within the clusters, indicating a poor fit.

Clustering dispersion indicator (CDI)

The *CDI* index consists of MIA scaled by the average distance between any two clusters $d(C)$. The *CDI* prefers large inter-cluster distances and small intra-cluster distances. Smaller values of *CDI* indicate better clustering [59].

$$CDI = \frac{1}{d(C)} \sqrt{\frac{1}{K} \sum_{k=1}^{K} d^2(C_k)} \tag{2.6}$$

Daviese-Bouldin Index (DBI)

The *DBI* index evaluates the overlap between clusters. It is quantified by evaluating the average intra-cluster distance, given by $diam(C_i)$, of all cluster i and subsequently comparing all pairs of clusters divided by their centroid distance $d(C_i, C_j)$ and before selecting the maximum distance for each class. Smaller values of *DBI* indicated that the K-means clustering algorithm classifies the data set properly [60].

$$DBI = \frac{1}{K} \sum_{i=1}^{K} \max_{j \neq 1} \frac{diam(C_i) + diam(C_j)}{d(C_i, C_j)} \tag{2.7}$$

2.5 t-SNE

A dimensionality reduction method can be used and then visualized in the two-dimensional plane for verification to strengthen the arguments of clustering results. A popular algorithm used for this purpose is called t-distributed stochastic neighbor embedding (t-SNE), introduced by Maaten and Hinton in 2008[61]. The t-SNE model is an unsupervised dimensionality reduction. The model first converts Euclidean distances into conditional probabilities to express the similarity between points and then maps the data points onto probability distributions by an affine transformation consisting of two main steps.

1. The t-SNE model constructs a probability distribution among high-dimensional objects. Similar objects have a higher probability of being chosen, and different objects have a lower probability of being selected.

2. The t-SNE model constructs the probability distribution of these points in the low-dimensional space so that the two probability distributions are as similar as possible to each other.

In addition, t-SNE is a nonlinear dimensionality reduction algorithm that is very suitable for downscaling high-dimensional data to 2 or 3 dimensions [61], i.e., visualizing large real-world datasets with hundreds or even thousands of dimensions. In this study, the visualization results of the t-SNE technique are compared with those of the K-means clustering algorithm for validation. Figure 2.4 shows the process of t-SNE, and the figure is adapted from [62].

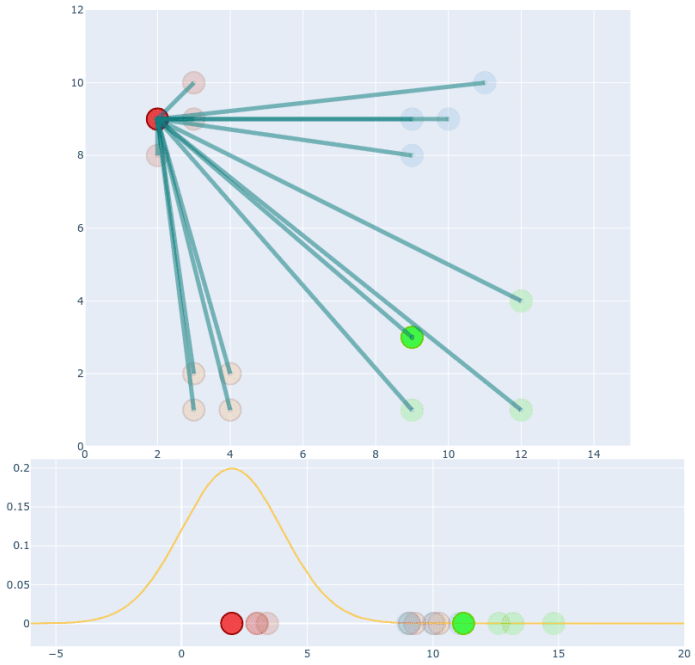

Figure 2.4: Example of t-SNE process

2.6 LSTM

2.6.1 Deep Learning - Neural Networks

The predecessors of modern deep learning were simple linear models; however, linear models have many limitations. Since neuroscience is considered an important source of inspiration for deep learning research, neuroscientists have found that most mammalian brains can use a single algorithm to solve most of the different tasks that their brains can solve. Neuroscience gives us a reason to rely on a single deep learning algorithm to solve many various tasks [63]. With the investment in research, deep learning has developed multiple uses, and the following describes the categories and applications of deep learning.

Deep learning is a branch of machine learning based on artificial neural networks that enable computational models to learn representations of data. These methods aim to discover complex structures in large datasets by using backpropagation algorithms to indicate how the machine should change its internal parameters. The structures are used to calculate the representation in each layer based on the representation of the previous layer. With enough of this combination of automatic learning and feature representation transformations, complex functions can be learned. In other words, by mapping the original input directly from the data to the output, the mechanism allows the system to learn complex functions. Crucially, these feature layers are learned from the data using a general-purpose learning process that does not rely on the design of human engineers [64]. Deep learning has become widely employed for various domains of science, business, and government because of its capability to discover intricate structures in high-dimensional data. It is considered as opposed to the conventional machine-learning techniques, which has beaten other machine-learning techniques in speech recognition, natural language processing, brain circuits reconstruction, social network filtering, machine translation, prediction of drug molecules, bioinformatics, and other fields. The ability of deep learning to solve supervised or unsupervised

problems is a critical feature that increases the applicability of the method. In addition, another fact that has contributed to the popularity of deep learning is the ability to process large amounts of data [65].

Based on the properties of the neural network, it may be categorized into three different deep learning architectures, such as Feed-Forward Neural Network (FFNN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN). Different domains require different depth architectures that suit their specific types of inputs, outputs, and questions. In this study, the dataset contains temporal information, which means that each value will be predicted based on the history of previous values. RNNs can handle time-series data because the activation of the hidden cyclic state of each step depends on the hidden state of the previous time step. At the same time, traditional neural networks transfer knowledge to the next layer exclusive of reference to the previous step. Therefore, RNNs are theoretically suitable for predicting time-based feature sequences. However, due to the inherent limitations of RNNs, i.e., gradient explosion and vanishing, training sequences with long time steps is challenging [3]. The Long Short-Term Memory neural network solves this time series problem. LSTM is an evolutionary model of RNN, which is more efficient than traditional RNN. This aspect will be further elaborated on in the next section.

2.6.2 LSTM Networks

Recurrent Neural Networks (RNNs) are neural networks that employ recurrence, which basically uses a feedforward pass information over the neural networks. RNNs have been successful applied to prediction problems where the input data are in the form of a sequence.

The decision in a recurrent layer takes at time step t will be affected by the decision made at time step t-1. The recurrent networks have two inputs, the current time step input X_t and the hidden state of past h_{t-1}. The weighted input and hidden state are combinedly compressed by a logistic sigmoid function or hyperbolic tangent (tanh) to make gradients workable for backpropagation. The backpropagation algorithm propagates the final layer errors backward from the output layer to the inputs of each hidden layer. The weights will be updated based on those weights assigned earlier by calculation of their partial derivatives. The recurrent networks need to backpropagate the error through all the previous time steps, which is not feasible and may cause the vanishing and exploding gradient problem, i.e., the gradient of the weights becomes too small or too large [22]. By using logic functions, it is possible to remove obstacles by squashing too large gradients in the case of gradient explosion. However, the main difficulty is gradient vanishing, as the gradient becomes too small to propagate and reflect any changes in the parameters to be learned. A variant of the existing RNN with long short-term memory units (LSTM) introduced by Stepp Hochreiter and Jürgen Schmidhuber in 1997 solves the gradient-related problem [66]. The main feature of Long Short Term Memory networks (LSTMs) is their ability to preserve the error that can be backpropagated through layers. The responsible element that provides memory is called a memory block, and the information is contained in the gated cell. In the following lines, the internal mechanism of LSTMs is explained.

The internal mechanism is shown as Figure 2.5, each line transmits a whole vector from the output of a node to the inputs of other nodes. The red circles perform as element-wise operations, e.g., vector multiplication and addition, while the yellow boxes represent neural network layers. (The figure is adapted from [67]) The top horizontal line is running through the diagram controls the cell state. Each LSTM unit receives three sources of information, two from the information of the previous unit and the other from the current input. LSTMs contain information in the gated cell. The cell makes choices about what to store, discard, and when to read via the gates that open and close. The step-by-step explanations are following.

The first step in LSTMs is to decide what information to store or throw away from the cell state using a sigmoid layer. It is also called the forget gate layer. It looks at outputs of the previous block and the new input vector, h_{t-1} and X_t, then outputs a number between zero and one for each number in the cell state that contains information from the previous memory unit, C_{t-1}. A one represents keeping the information completely while a zero represents discarding the

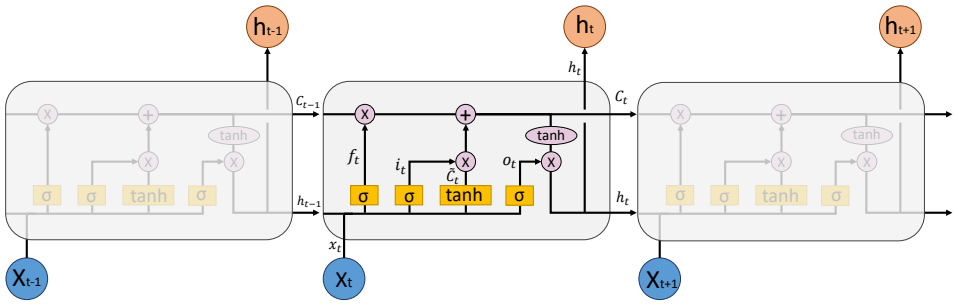

Figure 2.5: Example of LSTM structure

information completely. The mathematical function is:

$$f_t = \sigma(W_t \cdot [h_{t-1}, X_t] + b_f) \tag{2.8}$$

The second step will decide which new information is going to be stored in the cell state. This operation is performed in two parts, a sigmoid layer which decides the new values and a tanh layer which creates a vector of new candidate values $\tilde{C}_t$.

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i)h = 6 \tag{2.9}$$

$$\tilde{C} = tanh(W_C \cdot [h_t - 1, X_t] + b_C) \tag{2.10}$$

In the next step, the previous cell state $\tilde{C}_{t-1}$ will be updated into the new cell state $\tilde{C}_t$ with two operations. It will multiply the previous state by f_t to forget the useless information which was decided earlier. Then a combination is made to add $i_t * \tilde{C}_t$. These values become the new candidate values, scaled by how much we decided to update each state value.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{2.11}$$

Finally, it will determine the output. This output will be based on one cell state but will be a filtered version. First, a sigmoid layer is run to select what parts of the cell state are going to be the output. Then, the cell state is put through tanh and multiply it by the output of the sigmoid gate, so that only the output parts are decided.

$$o_t = \sigma(W_o * [h_{t-1} - 1, X_t] + b_o) \tag{2.12}$$

$$h_t = o_t * tanh(C_t) \tag{2.13}$$

To this point, it has summarized and explained the advantages of the methods used for data mining and learning algorithms. The next scenario will investigate how temporal frequency affects the output in multi-step data analysis models.

Chapter 3

Methodology

The main aim of this project is not to compare the performance of the algorithms but to compare data mining and machine learning results when data resolution is varied. Thus, we have used one algorithm with a different input setting. The structure of the study is presented in Figure 3.1, and the intermediate steps are illustrated in each of the phases. The first phase is to define the problem and form supporting research questions; this is done by looking into relevant literature (chapter 1). The second phase is to develop a robust approach to answer the research questions.

In the methodology of the second phase, a two-step process is proposed as a means to extract knowledge from the entire building meter. The first step is the creation and exploration of a description of the phenomena occurring in the original data at various temporal resolutions. This operation aims to transform the data into a human interpretable format and visualize the patterns in the data. The data is extracted and preprocessed using a library of data mining techniques to distinguish between types at various temporal resolutions. These features are visualized using a heat map format to evaluate the correlation between different temporal resolutions and weather features. The load patterns at the selected temporal resolution are compared with the designed metrics accordingly.

The second step of the methodology focuses on the learning mechanism, extracting interpretable insights for the load pattern clustering and predictive learning application. Only one of each supervised and unsupervised learning mechanism is implemented due to time limitations. This process allows the analyst to understand the impact of each temporal feature on each per-

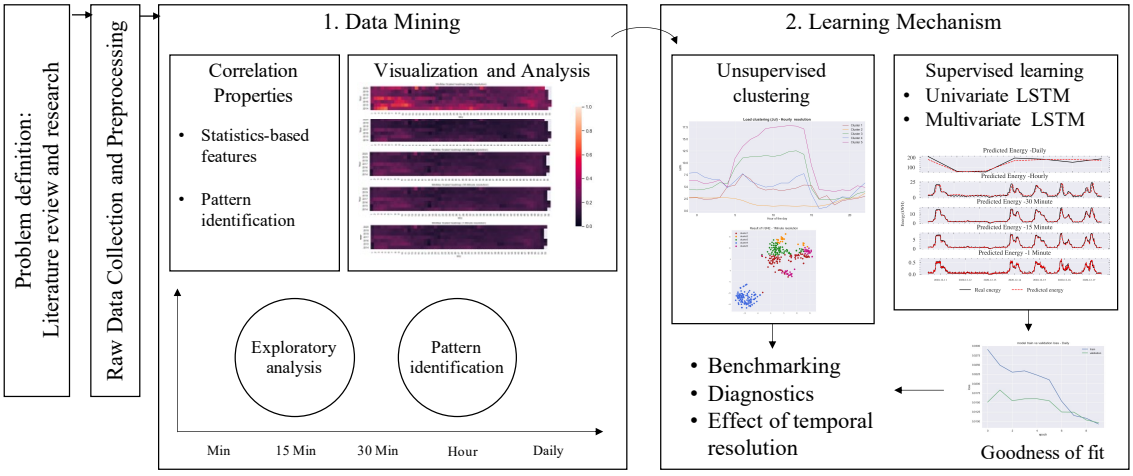

Figure 3.1: Overview of study framework

formance separately. Three benchmarks are implemented accordingly in this study: accuracy, error metrics, and cross-validation. One of the main results of the learning process is a discussion of what input time resolution is most important in data mining and machine learning for building energy consumption. This approach provides exploratory insight into the effect of temporal frequency on various features of commercial buildings. Detailed information on each step is presented in the following subsections.

3.1 Literature Review

The first phase in this research was to define the problem and relevance of the topic; hence a literature review was conducted. The literature review starts with an extensive overview of data-driven models on energy consumption forecasting for buildings. It proceeds to review distinct aspects of mining and machine learning time series data. In this review, the usage of data frequencies and their application in modeling techniques was investigated.

To obtain all the relevant literature, the search terms were based on the four elements, i.e., one focused problem and three extended interventions, see Table 3.1. The hits per search category can be found in the Appendix A.

Table 3.1: Search terms used to find relevant literature on temporal resolution

Problem	**Intervention**	**Intervention**	**Intervention**
Temporal	Building	Energy consumption	Forecasting
Resampling	Household	Energy demand	Accuracy
Sampling frequency	Buildings/household	Energy	Prediction

3.2 Data Preparation

The second phase in this research starts with data preprocessing. Data preprocessing is vital for any data-driven approach. Real-world data is often imperfect and containing inconsistencies and redundancies, which is not directly applicable for starting a data mining process and resulting in unsatisfactory results. [68] Preprocessing realizes two main tasks, data cleaning and data transformation, both targeting enhancing data quality. The procedure involves data consistency (coherent matching of datasets), data completeness (no missing values), and accuracy (outlier removal). Preprocessing is an essential step in the data mining process to obtain optimal results out of the envisioned analytics and preparing the data into an appropriate format for mining. The data and method to be applied for preprocessing are introduced in the following sections.

3.2.1 Data and Case Study Building

This section introduces the smart meter electricity consumption data that will be analyzed in this paper. The data is kindly provided by Kropman Installatietechniek, a building construction consultant company in the Netherlands.

The case study building, Kropman Office in Breda, is situated in the west of the city Breda which is a town in the southern part of the Netherlands. The building is a three-story high traditional office building. Like most office buildings in the Netherlands, the case study building is connected to a dedicated mid-voltage transformer station. A schematic overview of the installed electrical meters is indicated with I and II in Figure 3.2 where the arrows show the possible energy flows interaction between the systems and power grid. Figure 3.2 illustrates the installed electrical power capacities for all connections of both major electricity load groups at the Kropman Breda Building. The installed electrical power capacity for lighting is about 11% of the total. At the

same time, the other power connection covers about 89% of the installed capacity, including air handling unit, humidifier, chiller, BESS, and PV power generation system.

The entire building's electrical consumption was monitored for seven years from 2014 to 2020 in daily, hourly, and 1-minutely resolution, resulting in 2556 available daily energy profiles. Associated climate data were extracted with identical hourly resolutions from the Royal Netherlands Meteorological Institute (KNMI) weather station Gilze-Rijen located 10 kilometers away.

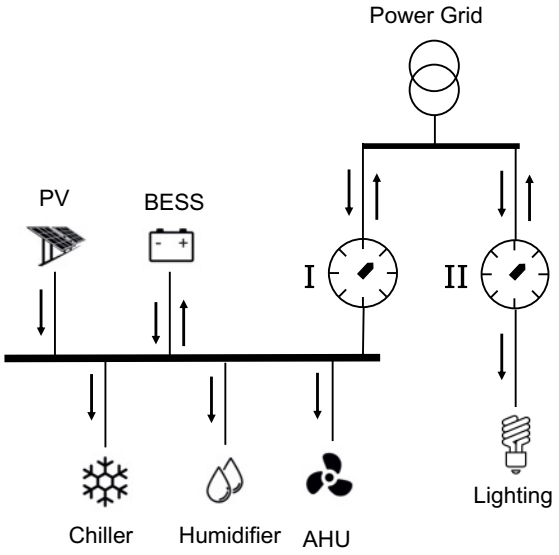

Figure 3.2: Schematic overview of the Kropman Breda office installed load system and power grid connection

The precise number and types of smart meter data employed in this paper are described in Table 3.2.

Table 3.2: Initial data description

Data Description	Value
Country	Netherlands
Region	North Brabant postal code: 4813 AC (City of Breda)
Supplier	Kropman Installatietechniek
Recording Temporal Resolution	Daily, Hourly, 1-minutely
Start	1 January 2014 00:00:00
End	31 December 2020 23:59:00
Length	2556 observations (daily readings), 61359 observations (hourly readings), 3681561 observations (1-minutely readings)
Type	Commercial building

3.2.2 Data Cleaning

Data collected through electrical sensors are usually noisy and often incomplete. Before analysis, the data were preprocessed to remove missing values and unreliable data. Data cleaning involves two parts of process, i.e., detecting the inaccurate and noisy parts of the data and correcting (completing, replacing, and modifying) the incomplete or irrelevant data. Even though most reviewed analytics from the literature perform missing values filling before outlier detection,

this study first considers outlier detection to avoid filling missing values from the interfered data set. This process could prevent identifying a more significant and more difficult portion of outliers in the next step. We consider outliers to be the data above the 99.999th percentile. As recommended in the work of Cho et al. [69], missing value filling is later performed with a linear interpolation since the missing gap is smaller than eight observations. Consecutive missing values of more than eight observations were treated separately such as replaced with identical time intervals that are resampled from other temporal resolution data.

3.2.3 Data Transformation

Data transformation involves partitioning the time series data into a format suitable for later data mining steps and normalizing the data to ensure fair similarity comparisons between load profiles during machine learning. In the clustering analysis, the data is divided into daily load profiles to identify typical load patterns in situations where individual demand is highly volatile due to user behavior. The daily profiles are then normalized using a Min-Max deflator to show the minimum misclassification errors. In the LSTM network, the datasets are divided into training and test sets and normalized using the Min-Max scaler to remove the correlation between all sampled energy data with the same distribution and to remove data features.

3.3 Temporal Resolution

The time resolution or granularity is the data sampling rate, which may be equal to or smaller than the acquisition rate of the meter. The maximum acquisition rate is determined by the technical parameters of the meter, such as its capacity for storing average information. In practice, the high resolution is a data measurement every few seconds to 30 minutes. Current SCADA systems can sample energy data at high frequency (1HZ), but the standard practice is to store the average value for 1 minute, or longer [23]. Low-resolution profiles present data over a longer period of time, e.g., several hours. The key is to choose a solution that weighs the level of detail that represents the basic consumption behavior of the user against the need to store and process the data.

3.3.1 Resampling

The temporal resolution is the sampling frequency of the data, in other words, the number of identical repetitive events per unit time. A change in temporal resolution is a change in the granularity of the time series, i.e., resampling. There are two types of resampling:

- Upsampling: the time series is modified, and the number of observations per time increases
- Downsampling: the frequency of the time series is decreased

In the case study, the raw energy consumption dataset was measured at a 1-minute frequency and recorded in the database as 1-minute, hourly, and daily resolution data. In a building environment, the control frequency is typically used for building performance control using 6-minute data, while the grid makes control decisions every 15 minutes. Therefore, 30-minute, 15-minute, and 6-minute data were downsampled from the 1-minute data for a more comprehensive study of the different time scales.

In Figure 3.3, the left time series plot is displayed in the original resolution of 1 minute. The right plot represents the aggregated downsampling time series with a resolution of 15 minutes. By comparing these two figures, the downsampled time series dataset has fewer outliers and stronger regular patterns. Downsampling reduces the number of observations per unit time, so it requires less memory size, and the model can run faster. This technique normalizes the data, thus reducing the number of outliers, but the model will predict worse irregularities due to unrepresented outliers. In short, there is a trade-off in obtaining better results. Therefore, the purpose of this study is

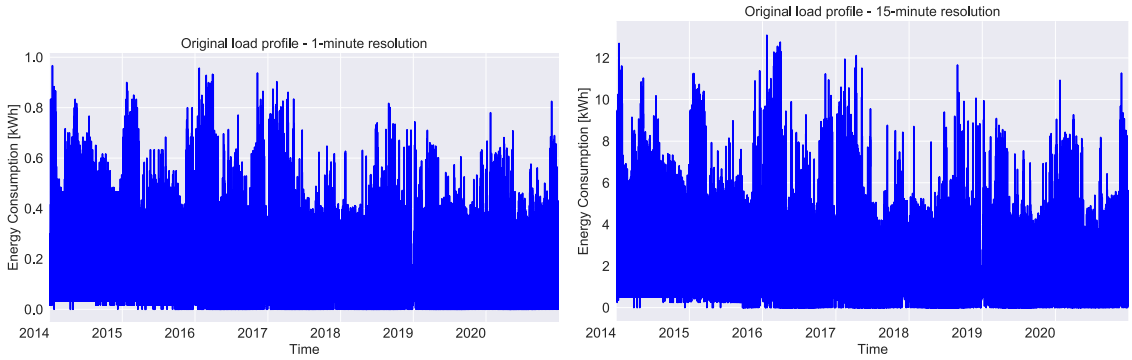

Figure 3.3: Representation of the time series of the Kropman building energy consumption data and resampled data

to find a balance between temporal resolution and model performance. The other original energy consumption load profiles are presented in Figure 3.4. The following section describes in detail the common techniques that have been applied to building performance analysis. In addition, each variable used in the machine learning model will be set according to previous studies, leaving only frequency as a variable to investigate its impact in the field of data mining and machine learning in the built environment.

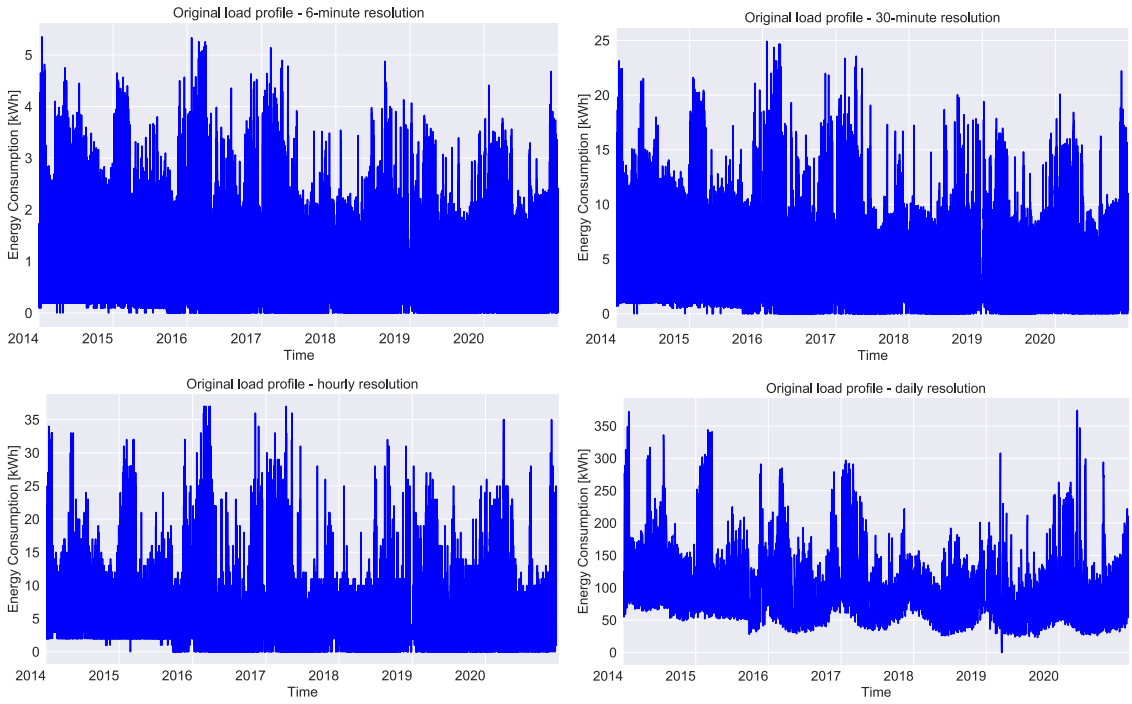

Figure 3.4: Representation of the time series data of the Kropman building energy consumption

3.4 Data Mining

Data mining (DM) is the procedure of extracting and discovering hidden knowledge in large data sets and contains methods at the intersection of machine learning, statistics, and database

systems. In the domain of built environment, DM is a powerful emerging technique to extract both building features and temporal features, i.e., physical characteristics, use of the building, pattern consistency, and weather dependency [48]. Temporal features are an aggregation of the behaviors exhibited in time-series data. They are features that aggregate sensor information, inform analysts through visualization, or are used as training data in predictive models. The process is designed to quantify qualitative behavior.

3.4.1 Exploratory Data Analysis

Statistically based temporal feature extraction is the most simplified data mining technique. Among them, ratio-based normalization is the most common method to compare energy consumption magnitudes at different temporal resolution data properly. As described in Section 2.1, all datasets are normalized by the MinMax deflator in the interval [0,1] and visualized in heatmap format. To obtain a clearer and more significant visualization, each dataset is grouped by week and compared to different temporal resolution scales year by year. The normalized data become more regular and easier to adapt to machine learning models for prediction. In addition, the correlation between energy consumption and weather scenarios is calculated in the interval [-1,1] by Spearman Rank Order Coefficient (ROC) and explored visually by heat maps in the same format. The ROC is implemented in statistical functions of Scipy version 1.6.2 and Python version 3.9.1. The statistical correlation results are then applied to a multivariate LSTM network prediction model to explore the relationship between energy and weather further.

3.4.2 Pattern Identification

Breakout detection is a high-performing technique that has recently been applied to building energy consumption time series to capture usage patterns and quantify consistency over time. Breakout detection runs using a single time series without excessive hyperparameter tuning and can effectively and accurately identify the change points of user behavior. The breakout detection methodology consists of two phases:

1. Evaluation of the seasonal presence in the data.

2. Implementation of EDM Breakout algorithm on the data.

breakthrough The classical additive decomposition is applied to evaluate the presence of seasonality in the data. In this study, the decomposition is implemented in Statsmodels version 0.12.2 and python 3.8.0. The frequencies of decomposition are set according to different temporal resolutions. Once a clear seasonal component is found, the seasonal characteristics of the data set are removed for further detection of breaks and thus prevent potential false positives.

BreakoutDection was cloned from a package on GitHub forked by Roland Hochmuth. The original breakout detection is an open-source R package developed by Twitter [52]. The model was built in Python version 3.9.1. There are only two parameters that need to be adjusted in breakout detection, namely the minimum threshold and the penalty constraint. The minimum threshold is periodic, so once a breakout is detected within the bounding period, it is impossible to test another mean shift within the minimum unit time. The penalty constraint is to control the amount of penalization. In this study, 30 days was chosen as the minimum threshold. The penalty constraint (beta) was set to 0.0001 to 0.000001 through multiple tests.

3.5 Machine Learning

Machine learning (ML) is research related to programming computers to automate the process of transforming observed data into outputs learned from the input data [70]. The input data of an algorithm is called the training data and consists of a set of features that are used as output prediction variables. If the features in the training data are labeled as output variables, they can

be used to guide the learning process, i.e., supervised learning. In contrast, if the training data contains only unlabeled feature variables, the learning process is called unsupervised learning. Due to the emergence of smart meters, ML has been widely used in building energy load prediction [71].

3.5.1 Cluster Analysis

In cluster analysis, time series datasets are transformed into the form of daily load curves to eliminate autocorrelation features and reduce computational costs. The results are averaged over three hundred iterations of the default maximum set of iterations with the same configuration (k number). Random initialization is applied to the center of mass of the cluster. The Euclidean distance is used to calculate the distance between the contour data and the centroid points. The k-means algorithm uses clusters ranging from 2 to 10. The k-means construct used in this study is implemented in Scikit-learn version 0.24.2 and Python version 3.9.1. To study the seasonal variation of energy load patterns in a typical commercial building environment based on user behavior, the dataset was then divided into four seasons to implement seasonal clustering.

To validate the clustering results, the manifold class in Sklearn version 0.24.2 implements the dimensionality reduction technique t-SNE to convert the time series into a two-dimensional distribution. The perplexity is set to the maximum value of 50 in the range (5-50) suggested by van der Maaten and Hinton to balance the attention between local and global aspects of the data and iterates 1000 times to achieve a stable configuration [61]. Random state initialization is applied to obtain a more globally stable state.

3.5.2 LSTM Neural Network

In LSTM models, a simple model selection method is to randomly divide the data set into three parts, i.e., the training set, the validation set, and the test set, if the given samples are sufficient. The training set is a set of observations used for model training, and the model learns from these samples to fit the parameters of the prediction model. The validation set is the set of parameters used to control the complexity of the model during training. The testing set is a set of observations used to evaluate the actual performance of the selected model, i.e., to evaluate the generalization ability of the model. In this study, all the data sets in the LSTM model are split into 60% training set, 20% validation set, and 20% testing set, as shown in Figure 3.5.

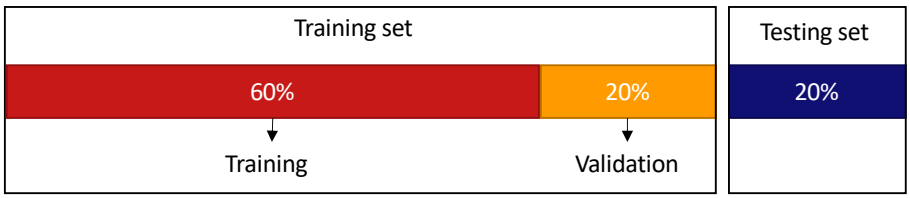

Figure 3.5: Train test split

Constructing neural networks is one of the most challenging and time-consuming tasks for machine learning researchers. Several variations need to be tuned, and there are thousands of different combinations. Once the network architecture is determined, the next step is to find the optimal hyperparameter settings. There are defined methods for these variables that can help achieve the best performance of neural networks. Hyperparameter optimization is performed using SKlearn's gridsearchCV tool, and the model is wrapped in a Keras Regressor to perform grid search. Table 3.3 shows the common search values selected in this study. In addition to the parameters, the number of time steps, i.e., the lag, is another critical decision in the LSTM architecture. LSTM models have limited ability to remember from memory blocks and transfer knowledge between networks. In this study, the number of time steps is defined by going back and forth trying to use the parameters between the minimum value of the cycle and the maximum

Table 3.3: Hyperparameter seach

Hyperparameter	**Search value**
Batch size	1, 8, 16, 32, 64
Learning rate	0.001, 0.01, 0.1
Dropout rate	0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
Number of neurons	1, 5, 10, 16, 32, 64, 128

value determined by the feedforward method. The time step for time series with different temporal resolutions is calculated as the equivalent fraction of the frequency, calculating its inverse and multiplying it by the proposed lag of the method. For example, the time step for day-resolution data is 7, and the time step for hour-resolution is 168. The LSTM network model was constructed using Keras version 2.4.3 and Tensorflow version 2.4.1 and implemented in Python version 3.8.0.

3.5.3 Neural Network Architecture

One of the critical decisions in the field of neural networks is to build the network structure. In addition to the difficulty of creating neural networks, the project had to deal with additional complexity. Although the data were, in fact, modeling the same problem, i.e., the prediction of energy consumption in commercial buildings, the project used datasets with different temporal resolutions and different frequencies, requiring different neural network model architectures and combinations of hyperparameters. To address these difficulties, we use downscaled data to reduce the amount of data and the structure required to capture the data patterns. Thus, architecture is defined, and the architecture is extrapolated to other temporal resolutions. The best results are usually obtained using lower resolution data, so the architecture defines the daily resolution method of the model that should be extrapolated. This approach utilizes a model with fewer data and requires less processing time.

The process of reaching the optimal neural network architecture configuration is essentially an incremental approach. In order to keep the study simple and to focus on the effect of temporal resolution, each layer of the LSTM added is a memory block. The first step is to create a neural network with only one memory block and then test several regularization techniques to check which technique can better avoid overfitting and obtain better performance. Overfitting is a common problem in deep learning; when the model trained from the training data is too close to the training data itself (with minimal error), but instead deviates from the generalized objective and fails to successfully represent the data other than the training data, this phenomenon is called overfitting. To avoid this, neural networks have some methods, such as L1/ L2 regularization, and dropout. Dropout regularization directly remove parts of neurons during training, including their inputs and outputs, and this removal can be done simply based on a probability p. The unnecessary information or error will be prevented to carry to the next layer of further computation, and the processing time will effectively reduced.

Once the experiment has determined which regularization technique is the best, the next step is to increase the size of the architecture by adding more layers internally. Thus, the starting point is to have a structure of memory blocks and their corresponding regularization methods, and from that point on the increase, the number of memory blocks until the structure achieves the best performance. The metrics will be defined in the following subsection.

3.5.4 Metrics

Choosing the appropriate metric is a task that should be considered while building a machine learning model. In the study, the values of two metrics were used to evaluate the prediction results, including the mean absolute percentage error (MAPE) and the normalized root mean square error (NRMSE). The use of these performance measures represents a variety of approaches to evaluating the model. MAPE and NRMSE are size-independent relative performance measures

that represent the relative standard of prediction error between actual and predicted values. Unlike the mean absolute error (MAE) and root mean square error (RMSE), these two metrics are absolute performance metrics commonly found in regression problems and allow us to compare the actual level of error between the observed and predicted values. In this study, the performance metrics used are calculated as follows.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{3.1}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{y_i - x_i}{y_i}| * 100 \tag{3.2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{3.3}$$

$$NRMSE = \frac{RMSE}{y_(max) - y_(min)} \tag{3.4}$$

where n is the number of validation points and refers to the forecasted values, y_i refers to the actual values.

3.6 Experiment Environment

The primary purpose of this study is not to compare the performance of the algorithms but to compare the results when the data resolution varies. Therefore, the experiments follow the general setup commonly seen in most research cases in architectural environments. Based on the general settings, the results would be more applicable to similar research and developed into more specific purposes, such as applying building performance management at the best temporal resolution and improving the energy efficiency in the system. The selected data mining and machine learning methods were applied using Python 3.9.1 and 3.8.0 on an i5-1038NG7 core CPU computer with 2.3 GHz and 8 GB of RAM.

Chapter 4

Results and Analysis

4.1 Correlation Heatmap

To study the effect of temporal resolution on data mining and machine learning in the built environment, it is necessary to create experiments with the same or similar settings. Configuring experiments with the same characteristics will allow us to observe the effects of different frequencies individually. Firstly, it is critical to select the appropriate temporal resolution sequence for the comparative study. The smart meters installed in this project were measured using a 1-minutely resolution and recorded in a database with 1-minutely, hourly, and daily resolution data sets. Therefore, the three original datasets were directly selected for the experiment while maintaining the integrity of the database. On the other hand, one of the project's goals is to make a week-long prediction, so the most common approach is to use the same or higher frequency of input data. According to the findings of [42], commercial utility meter data with high temporal resolution (less than 1 hour) will support more accurate model calibration and parameter inference, thus achieving more optimized predictive performance. In order to thoroughly investigate the effect of temporal frequency, 1-minutely data was sampled as 15-minutely data and 30-minutely data to provide continuity of time resolution. Many previous studies have also chosen 15-minutely and 30-minutely temporal resolutions, as shown in Table 1.1. In addition, the 6-minutely resolution is an important frequency for building performance control; thus, the 6-minutely data resampled from the 1-minutely data are included in this study. For the above reasons, the final temporal resolution of the data was chosen as follows:

1. Daily
2. Hourly
3. 30 minutely
4. 15 minutely
5. 6 minutely
6. 1 minutely

The first step in data mining leads to simply start with the exploration of statistical features. The major category of statistical features is ratio-based features. These features often have a normalizing effect in which datasets can be more appropriately compared to each other. The first extracted metric of this type is the consumption magnitude of energy normalized by the time period, which is one of the most commonly calculated for building performance analysis. To better compare the differences between various temporal resolution datasets, the raw data are normalized by MinMax scaler and visualized in heat map format as shown in the Figure 4.1. From the daily-resolution data to the hourly-resolution data, the fluctuations significantly level out and then become smoother as the frequency of the data increases. In the daily resolution data, each square represents a weekly average energy consumption. The heat map shows much higher energy consumption in the first few weeks of the year and much lower energy consumption in the middle of the year. The same finding is not seen in the other resolution data, i.e., hourly, 30-minutely, 15-minutely, 6-minutely, and 1-minutely, where the energy consumption has leveled off over the same period at the high resolution. The metric is intended to provide a basis for comparison between various temporal resolution datasets.

Another useful statistical indicator is related to how much weather affects energy consumption in buildings since changes in energy consumption are mainly influenced by changes in weather. As introduced in subsection 3.4.1, the Spearman rank-order correlation (ROC) coefficient is suitable for approximating the correlation between weather conditions and building energy consumption. The correlation coefficient is visualized in Figure 4.2. The correlation values are calculated individually each week. This process results in fifty-two to fifty-three calculations each year using 7, 168, 336, 672, 1680, or 10080 samples. In this case study building, consumption in week 2 to week 21 is more heating sensitive, which can be observed clearly from the one-dimensional heat map. Week 28 to week 40 is more cooling sensitive. Week 1 and week 53 should be excluded because the sample size is smaller, resulting in less representative results. The lower resolution data shows a clearer pattern of correlation between energy and weather scenarios. The reason for this is related to the fact that user behavior does not change much in a continuous hour, and changes in user behavior usually occur after a much longer period of time, such as an hour or half a day, on the other hand, the weather usually does not change significantly within an hour.

The Spearman rank-order correlation coefficient (ROC) is based on the usual Pearson correlation coefficient but applies to rank variables. To verify the utility of ROC, the results were compared with Pearson's correlation coefficients calculated by the identical approach. Based on the comparison between ROC and Pearson correlation, the visualization heat map presents similar

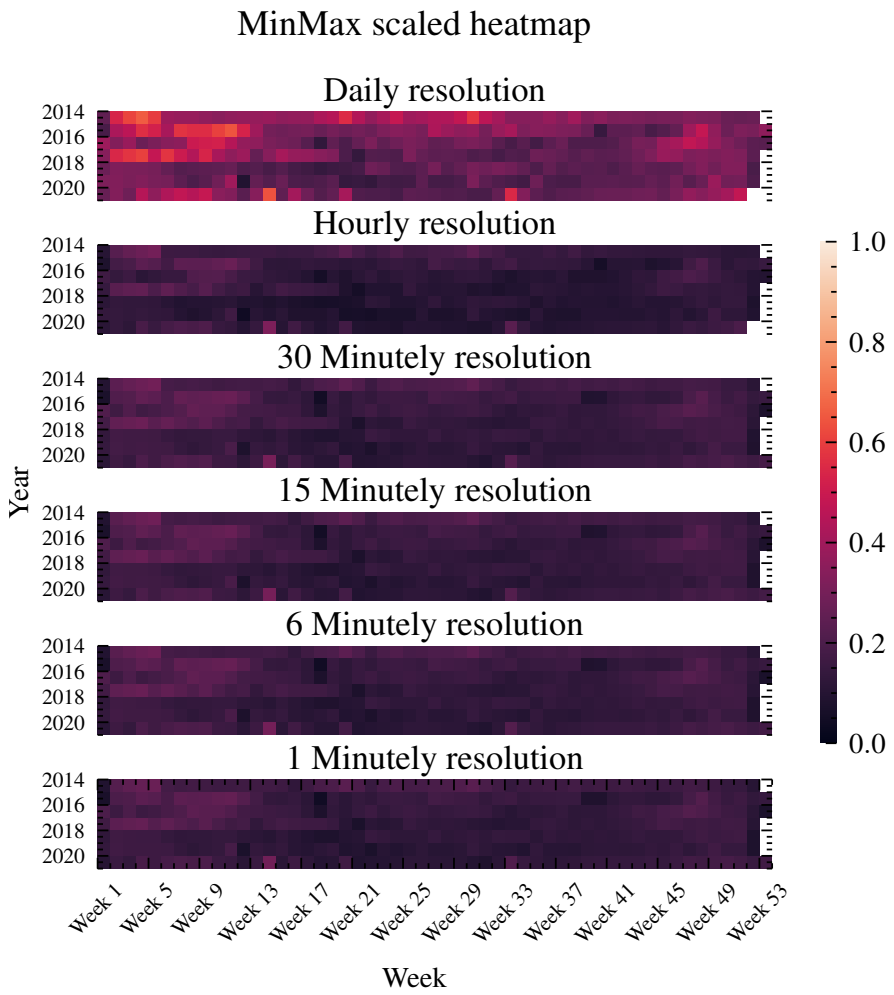

Figure 4.1: MinMax scaled heat map

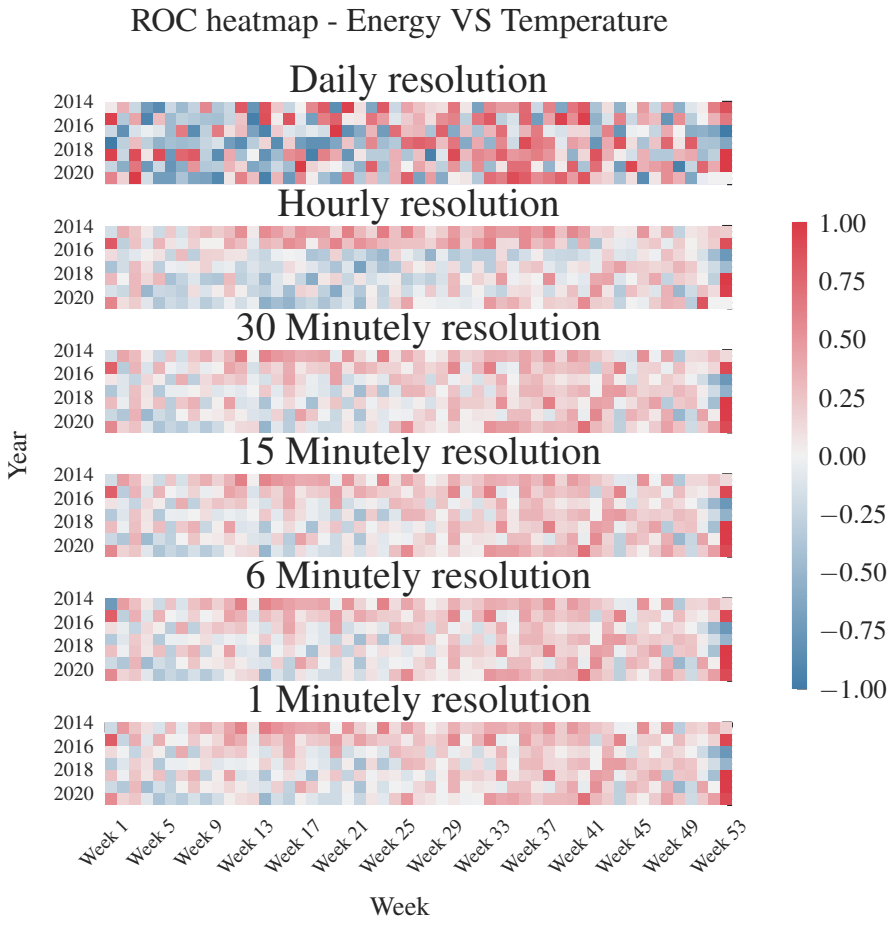

Figure 4.2: Spearman rank-order correlation heat map energy v.s. temperature

results, but the ROC results show a more noticeable visualization effect. The heat map of Pearson correlation can be founded in Figure B.1. The ROC coefficient is not a perfect indicator of energy consumption, as it only detects correlation. However, the ROC coefficient can be calculated quickly and makes it easy to calculate and observe weather dependence.

In addition to the outdoor temperature, other weather conditions, such as relative humidity, solar radiation, and air pressure, were studied in relation to building energy consumption, as shown in Figure 4.3. The results for air pressure can be seen in the Figure B.2, which shows a random variation, which may be related to the randomness of the wind. The relative results for humidity and solar radiation show an ordered variation. Interestingly, the results for humidity and radiation show opposite correlation values, i.e., building energy consumption is almost positively correlated with solar radiation and almost negatively correlated with air humidity. As the frequency of the data decreases, the correlation between energy consumption and humidity becomes more pronounced. This may be related to the fact that solar radiation and air humidity vary more significantly over a longer period of time, e.g., longer than an hour. In addition, the Kropman office installed a photovoltaic system in 2015, and significant changes can be observed from the hourly resolution data.

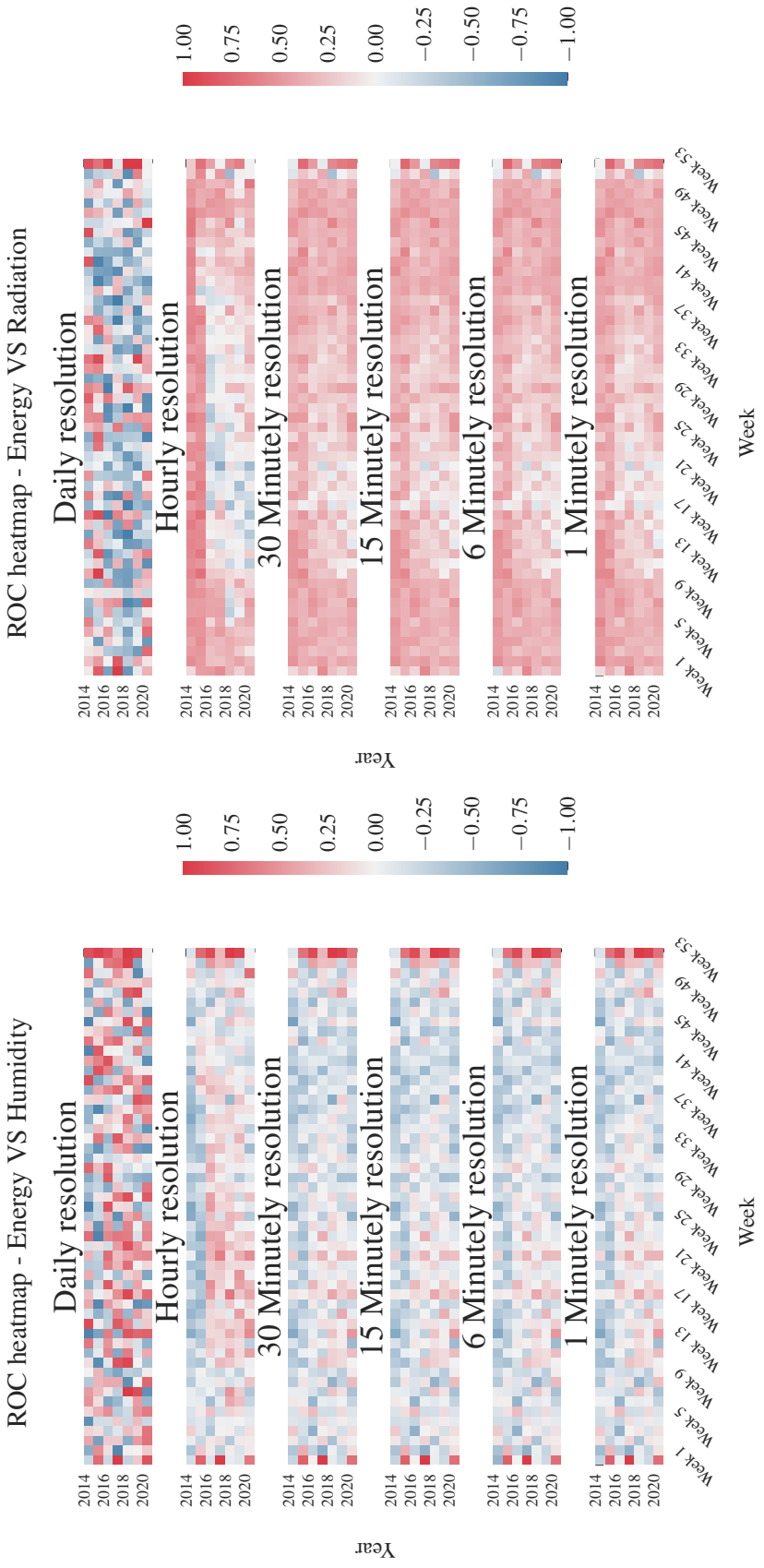

Figure 4.3: Spearman rank-order correlation heat map energy v.s. humidity / energy v.s. radiation

4.2 Breakout Detection

The second step of data mining developed in this study is related to capturing typical and atypical usage patterns from building energy consumption data. As described in subsection 3.4.1, the goal of these functions is to quantify whether buildings have daily or weekly consistency, such as whether certain building types have specific types of usage patterns. To improve the completeness of the study, a process of energy consumption time series decomposition was used to extract trend, seasonality, and residual information from the dataset, followed by breakout detection to quantify variations in the steady-state of building energy consumption over the time horizon.

The classical additive decomposition method introduced in subsection 3.4.2 is applied to time series data with a frequency of 1 week. The additive decomposition simplifies the calculation and extracts the trend components from the original time series data using moving averages (see Figure 4.4, Figure 4.5, Figure 4.6). The visualization results of one month clearly show a similar trend calculated for the six temporal resolution data. Only the trend component of the daily data is like a low-pixel picture compared to the trend component of the hourly data and others. On the other hand, the extracted seasonal components are plotted for the same periodic period within the same time frame. In this respect, only the daily time-resolution data indicate strong seasonal features. The observed variation of the daily load profile is significantly lower than that of the hourly load profile, suggesting that the former estimate is more accurate and preferable to the hourly load profile.

However, periodicity still can be seen in the seasonal component of the hourly data and others, indicating a higher frequency of the hourly and others load curve. Other periodicity frequencies are also tested, such as 24-hour, 12-hour, 6-hour, 3-hour, 1-hour, and 15-minute listed in Table 4.1. When the frequency of periodicity is higher, the data with higher resolution show stronger seasonal characteristics (see Appendix C). After several attempts, the periodic frequency of 24 hours already shows a significant seasonal component for time series decomposition at different temporal resolutions as shown in Figure C.1, Figure C.2, and Figure C.3. The seasonal components are removed accordingly for further breakout detection analysis in the following section.

Table 4.1: Periodic frequency settings for time series decomposition at different temporal resolution

Resolution	**Periodic Frequency**
Daily	1 week
Hourly	24 hours
30-Minutely	12 hours
15-Minutely	6 hours
6-Minutely	1 hours
1-Minutely	15 minutes

In this case study, pattern consistency was determined by splitting a library of breakout detection from the R programming package. This package was developed by the social media company Twitter to process their time-series data. As described in subsection 3.4.2, this function is used to identify mean shifts in the dataset associated with abrupt jumps in time series data. The identified seasonal components are extracted out to eliminate the effect of autocorrelation. (Figure 4.7, Figure 4.8, Figure 4.9) illustrate the breakout detection process for a full year of data from the case study building. The dataset includes six different temporal resolutions, and in this study, 30 days was chosen as the minimum threshold. The minimum threshold is periodically bounded. Once a breakout is detected within the boundary period, it is not possible to test another mean shift within the minimum unit time.

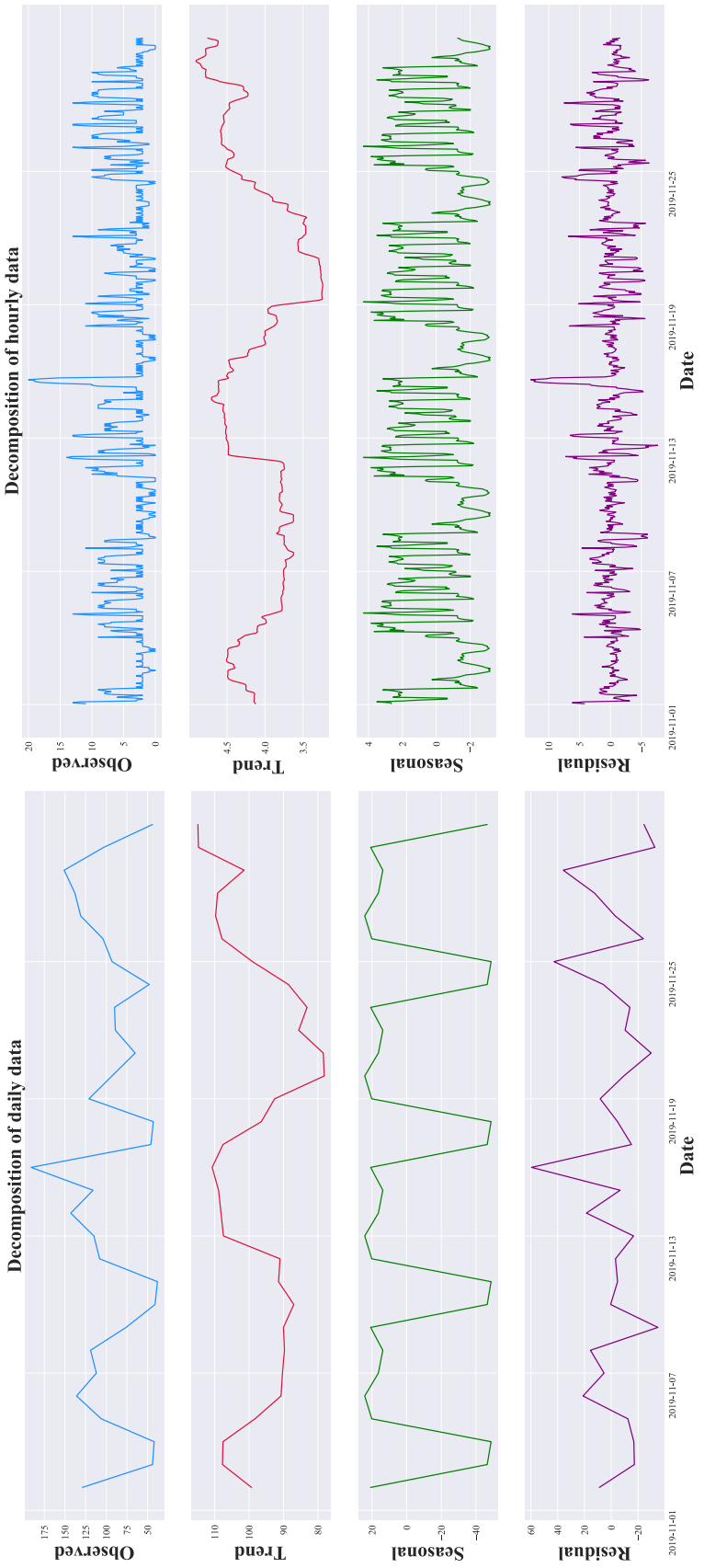

Figure 4.4: Decomposition of daily and hourly data

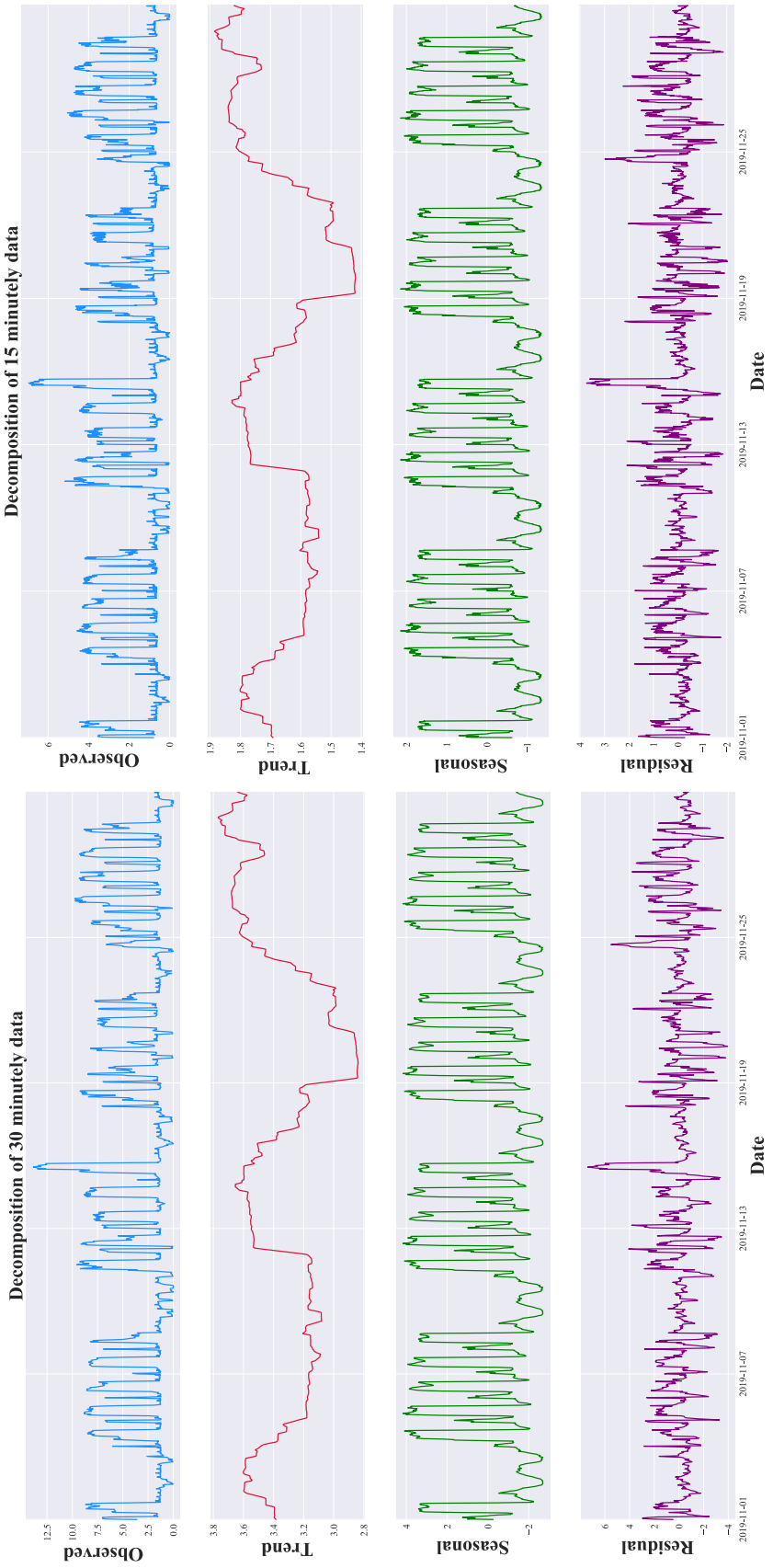

Figure 4.5: Decomposition of 30-minutely and 15-minutely data

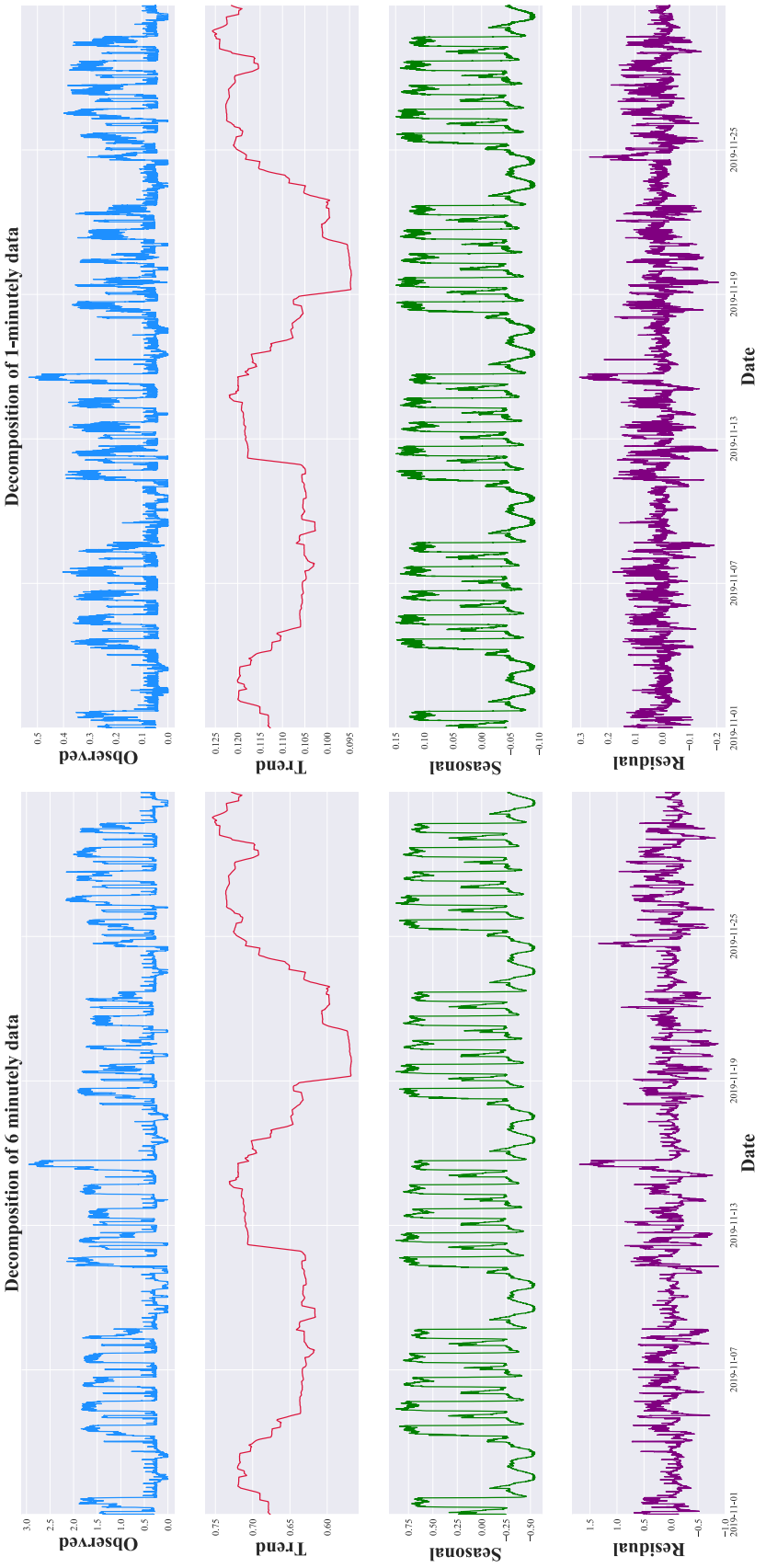

Figure 4.6: Decomposition of 6-minutely and 1-minutely data

The number of detected breaks gradually decreases to near zero based on daily data to 1-minute data observations. This is a fact related to the smoothing effect of the high-resolution data. In the daily resolution data, nine breaks can be detected from a whole year of data, almost one break per month. When the threshold size decreases, the number of detected breakouts gradually increases due to the large fluctuations in the daily resolution data.

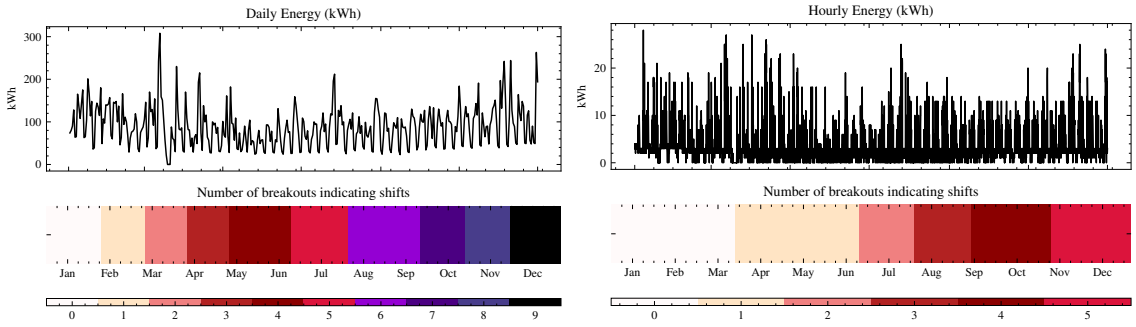

Figure 4.7: Breakout detection to test for long-term volatility at daily (left) and hourly (right) temporal resolution

In addition, a total of five breakouts are detected from hourly to 15-minute resolution data, but the detected breakouts show different time points in between. A break is detected for July at the hourly resolution, but the same breakpoint is not visible in the 30-minute and 15-minute data. The point of the breakout was shifted to August. This difference can be attributed to another control parameter: the threshold penalty, which reduces the size of the mean shift of July in the hourly data and enlarges the mean shift or jump of August in the 30-minute and 15-minute data. The detected breaks of September and October for hourly data are then shifted to October and November at 30-minutely and 15-minutely resolutions due to the minimum threshold of 30 days.

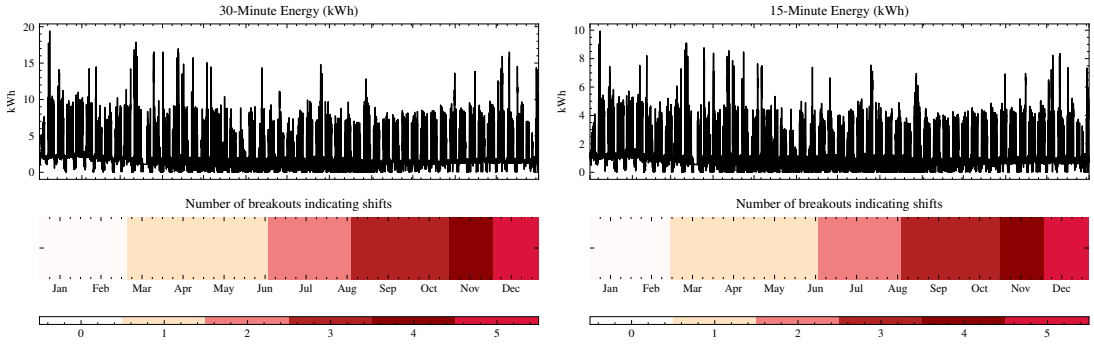

Figure 4.8: Breakout detection to test for long-term volatility at 30-minutely (left) and 15-minutely (right) temporal resolution

The penalty constraint parameters are all 0.00001 for hourly, 30-minutely, and 15-minutely, while only 0.0001 for daily resolution. In 6-minute data, the penalty constraint shrinks to 0.000001 since more effort is needed to detect the breakout. The processing time gets significantly longer when temporal resolution gets higher. A total of 3 breaks are identified in the 6-minute data, showing disruption in March, July, and September, but missing the interruption in the wintertime. In the 1-minute data, only one breakout during July to August is identified, even with the penalty constraint of 0.000001. Overall, the breakout point is most pronounced in the summer months, indicating that most people who work in the Kroppman Building spend their vacations in the summer.

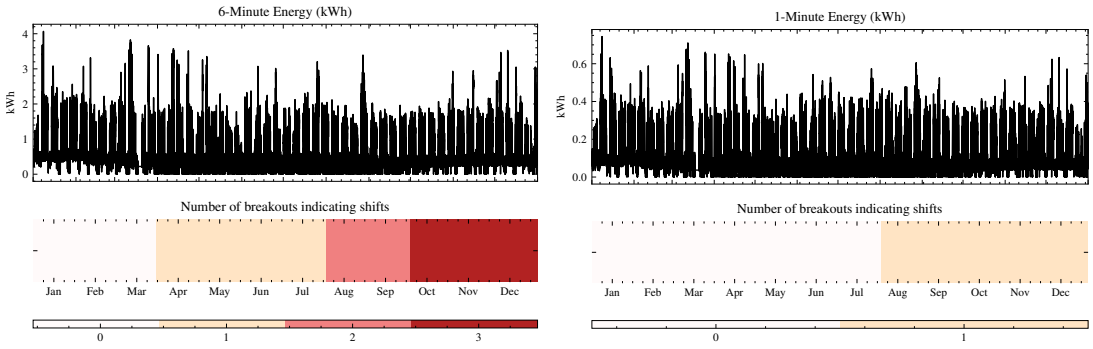

Figure 4.9: Breakout detection to test for long-term volatility at 6-minutely (left) and 1-minutely (right) temporal resolution

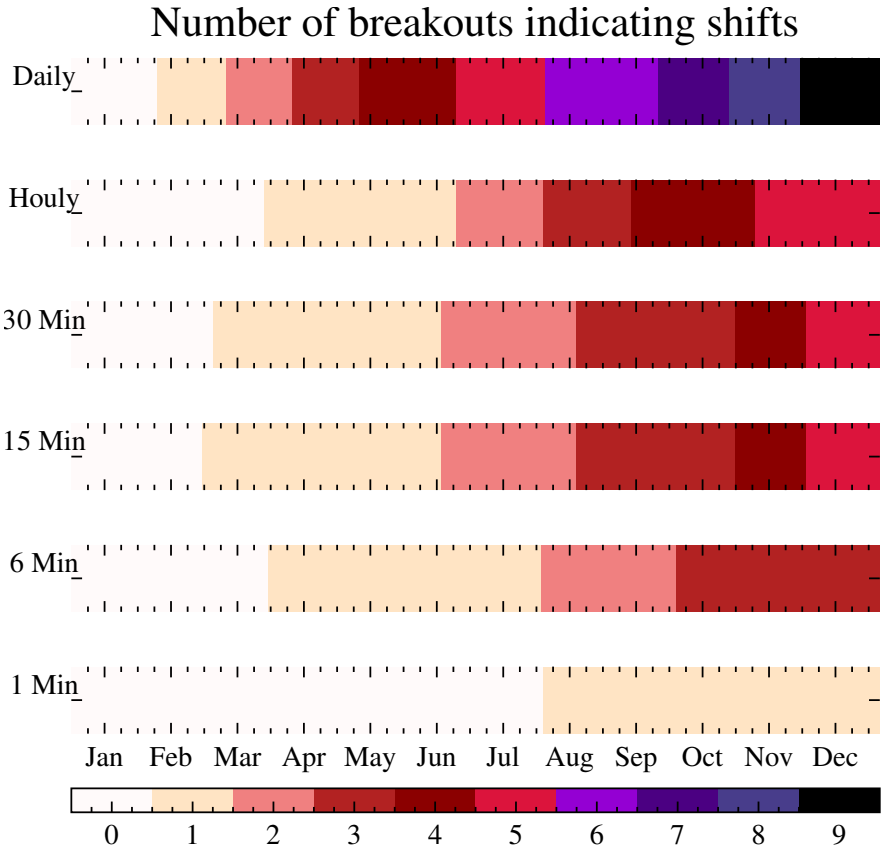

Figure 4.10: Breakout detection variation

4.3 Clustering

The obtained knowledge from feature extraction and pattern identification are effectively conducted on unsupervised learning, i.e., cluster analysis, as discussed in subsection 3.5.1. Due to the autocorrelation feature in the time series data, the data set is transformed into a load curve with a time step of 24 hours. After that, the data are processed using min-max normalization to identify the time series with equivalent energy consumption patterns.

As described in subsection 3.5.1, we look for elbow breaks in the index development, which suggests that more clusters do not improve the clustering results. We calculate three selected validation indices for each number of clusters from 2 to 10; Figure 4.11 shows the index development. The MIA index shows almost no variation at low-resolution data, indicating stability and almost immediate flattening, giving no indication of cluster selection. In contrast, the DBI index exhibits more significant variability, and elbow points can be observed at the number of 4 clusters in addition to hourly resolution data. The CDI index shows considerable variation and jagged horizontal development, indicating no specific number of clusters. The above results indicate that cluster analysis shows a more significant variation on higher resolution data, which yielded better scores, with the worst results when resolutions lower than 30 minutes.

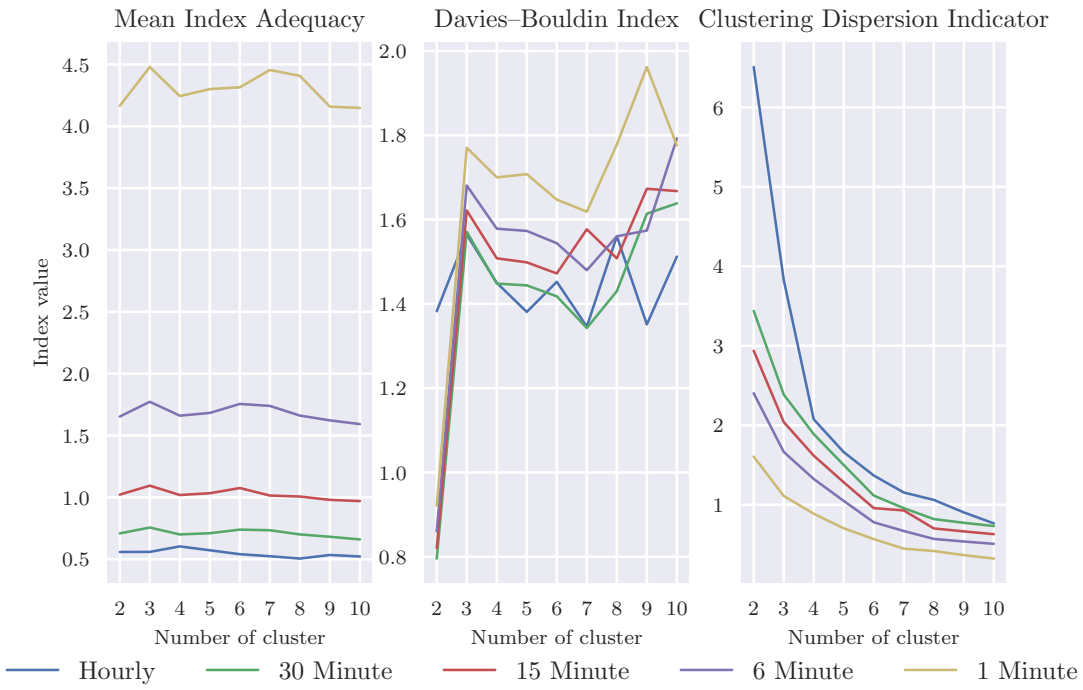

Figure 4.11: Clustering index values

Better clustering results are visible at resolutions higher than 30 minutes, while the DBI index values for 15-minute and 30-minute data are much lower than those for 6-minute and 1-minute data. Figure 4.12 and Figure 4.13 show the corresponding plots of the mean values of the different clusters in the case of four clusters at 15-minutely and 30-minutely resolutions. The typical commercial energy consumption load curve can be clearly observed, where the energy consumption increases from 8 am and decreases from 5 pm on weekdays. A clear pattern of energy consumption during the weekend is also shown, indicating almost zero user activity. Besides, the nighttime energy consumption may be related to the HVAC system running at midnight in the summertime among weekday load profiles. Thus, a seasonal analysis is implemented to further study the relation of clustering and seasonal patterns described in the following subsection.

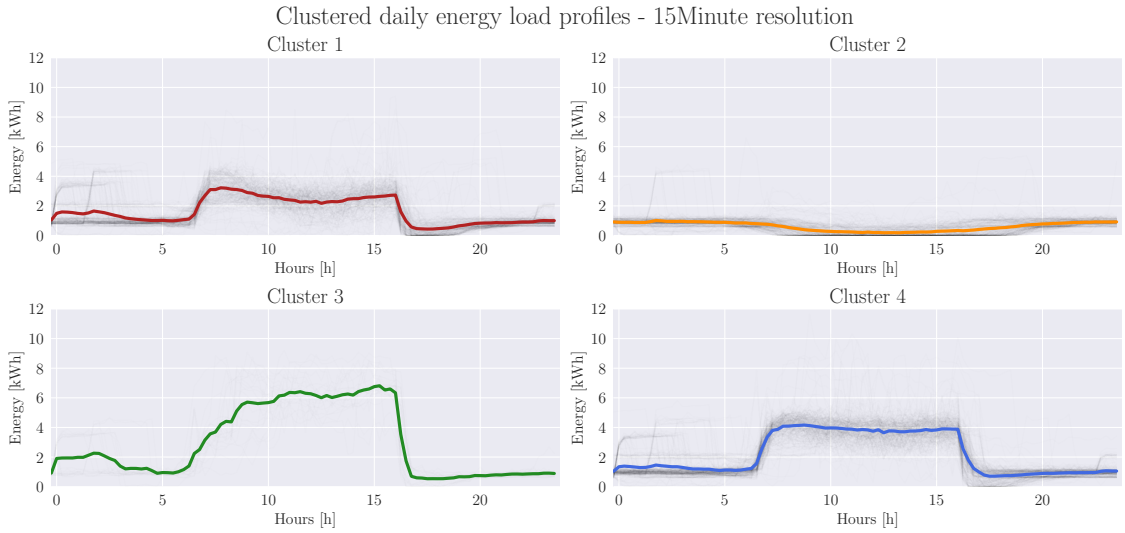

Figure 4.12: K-means clustering of 15 minutely data

Since the differences between models and facts cannot be quantified naturally, we validate the clustering results by the t-distributed stochastic neighbor embedding (t-SNE) technique. The dimensionality reduction capability of t-SNE is particularly suitable for the visualization of high-dimensional data sets. Figure 4.14 shows the visualization results of the reduced building energy consumption data on a two-dimensional plane. A total of four clusters can be seen, and the 15-minute resolution dataset shows more robust clustering than the 30-minute data, indicating a clearer clustering commercial building energy pattern at 15-minutely resolution.

In addition, we combine seasonal features with cluster analysis and apply the same clustering approach to the four seasonal datasets at 15-minutely resolution, as shown in Figure 4.15. Due to the high weather variability in both summer and winter, three clusters were found, while two clusters were found in the spring and fall. The year's highest temperature occurs in summer, and the lowest temperature occurs in winter, which results in higher energy demand for cooling and heating systems, respectively. From the summer dataset, the significant energy consumption of the midnight load appears, which is related to the fact that the HAVC system runs longer in summer. On the other hand, a significant increase in the heating system's energy consumption can be seen in load profiles of the winter season. In conclusion, the number of clusters verifies the best clustering results for the 15-minute resolution data, showing the typical commercial energy consumption load curve for different seasons.

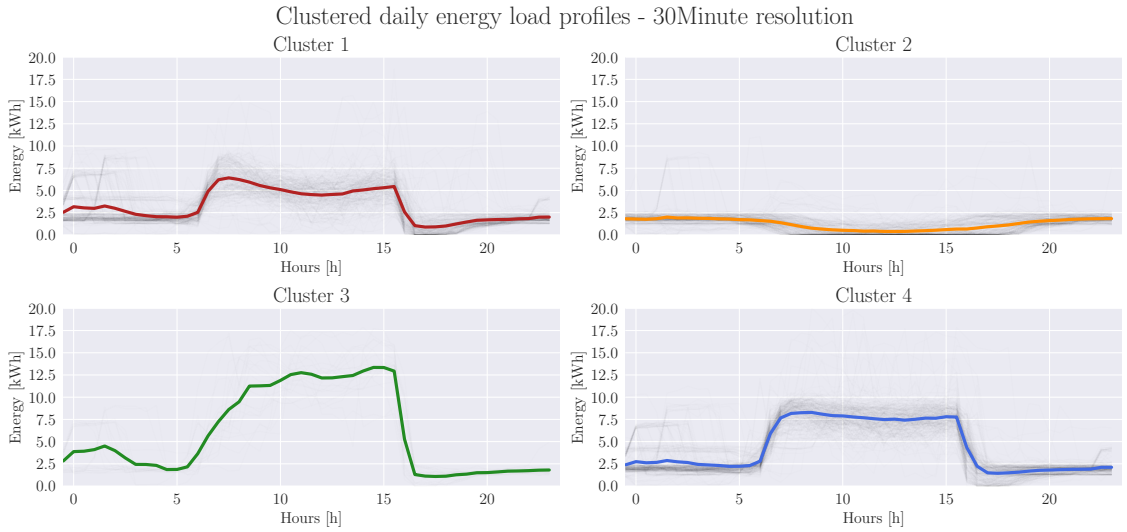

Figure 4.13: K-means clustering of 30 minutely data

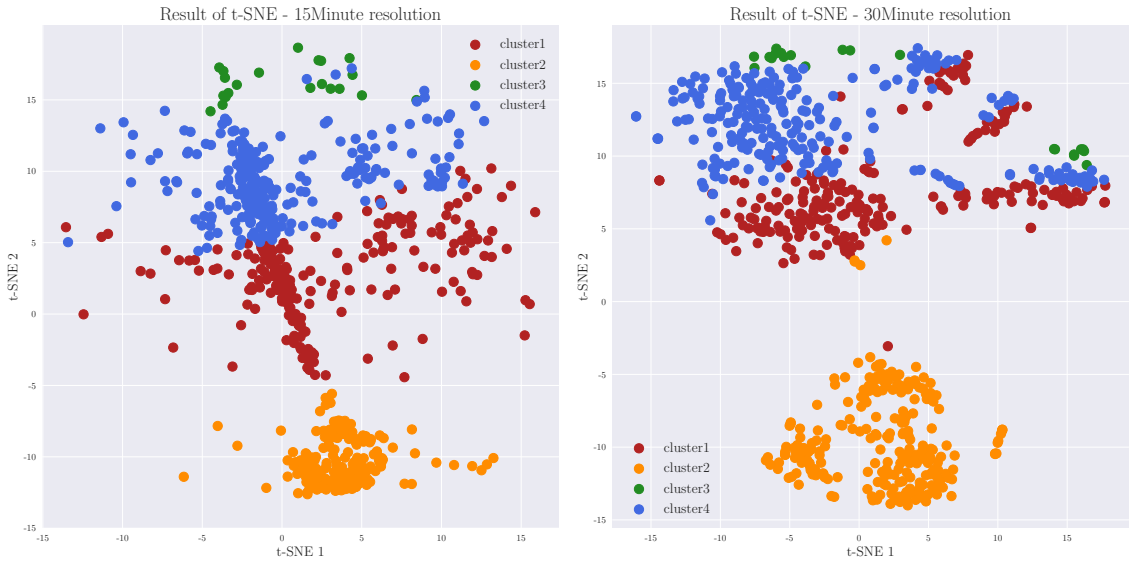

Figure 4.14: t-SNE validation of clustering

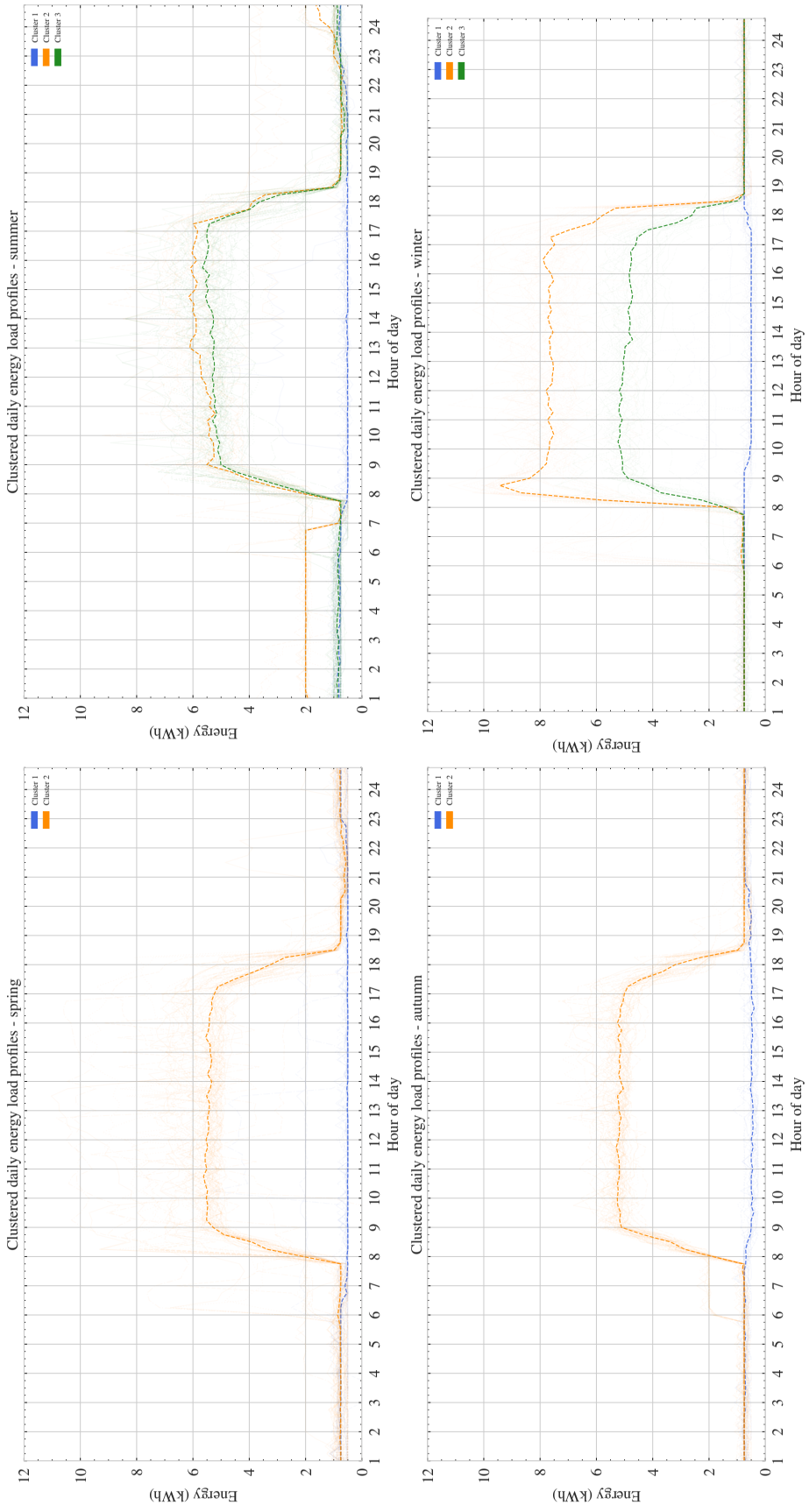

Figure 4.15: Seasonal clustering

4.4 Forecasting

To investigate the effect of temporal resolution on forecasting, a variant of RNN, namely LSTM networks, is applied to compare the predictive performance of data with different temporal resolutions. As discussed in the previous chapter, the best approach is configuring the daily resolution as the standard dataset for creating neural networks. The goal of creating architecture is not to achieve the best prediction but to create a network that is easily adaptable to other frequencies. Therefore, prediction performance is important but not a critical factor in defining a neural network architecture. After testing several different architectures, the most efficient and least complex model is illustrated in Figure 4.16, which is the simplest architecture model that was found more adaptable to other temporal resolutions.

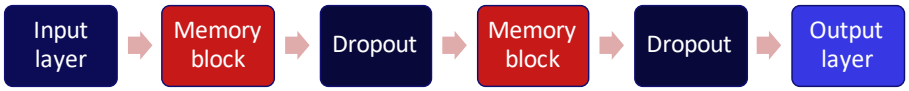

Figure 4.16: Standard architecture of the LSTM models defined

Since the focus is on testing the relationship between temporal resolution and predictive performance, the model structure simply consists of two memory blocks as the representative layer of the LSTM model in this study. In addition, a dropout block is constructed after each memory block as a regularization layer to remove incorrect information and avoid overfitting. Other types of structures can improve the performance of the network, but they may generate some noise in the analysis that interferes with the results. On the other hand, the memory block can be optionally hyperparameterized to obtain better results. The defined neural network is tested using all the hyperparameters, and the optimal configuration for different architectures is shown in Table 4.2.

Table 4.2: Hyperparameter seach

Resolution	**Units**	**Batch size**	**Learning rate**	**Dropout rate**
Daily	(16,10)	32	0.1	0.2
Hourly	(16,10)	32	0.01	0.2
30-Minutely	(16,10)	32	0.001	0.1
15-Minutely	(32,10)	16	0.001	0.1
6-Minutely	(32,10)	16	0.001	0.0
1-Minutely	(32,10)	16	0.001	0.0

First, it is observed that the higher temporal resolution (1-30 minutely) needs more time for understanding the structure of the data since their learning rate needs to be smaller. On the other hand, the higher temporal resolution also uses more units where a larger dimension of the output matrix is required, indicating the finer classification during the computation. Also, it can be appreciated that higher temporal resolution requires a smaller batch size for each epoch because of the more units. These results can be explained based on the fact that the same forecasting period is one week in terms of the balance between dimensionality and information size. Otherwise, the lower resolution requires much fewer units and batch size to improve computational efficiency. Finally, it can be connected that the higher resolution needs less value in the dropout. This can be defended based on the fact that having more data makes the prediction understand the better structure and then not learning from a specific structure avoiding the overfitting.

We have conducted experiments by aggregating the total energy consumption by 1 minute, 6 minutes, 15 minutes, 30 minutes, hourly, and daily. The resolution decreases as changing from minutely to daily. We used LSTM models to compare the results of the experiments for the next

one week of prediction. For the performance evaluation, the mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE), and normalized root mean square error (NRMSE) were used.

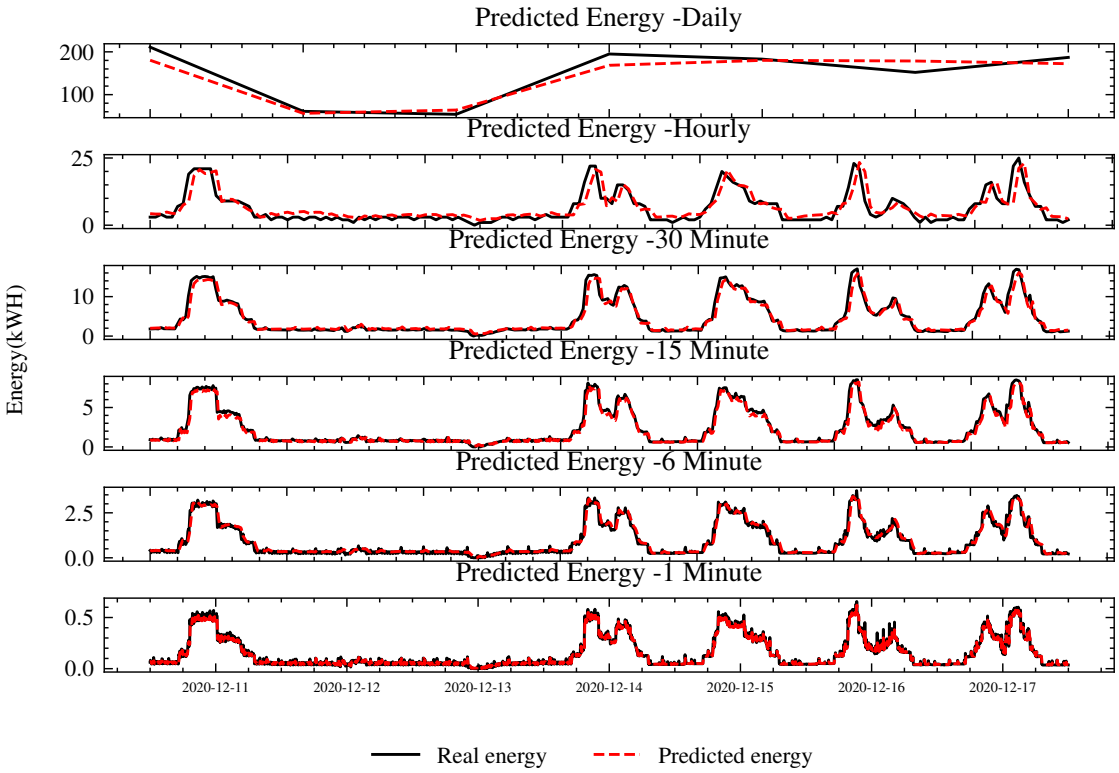

Figure 4.17: Energy load forecasting by univariate LSTM

Figure 4.17 illustrates the prediction results of each LSTM model according to the temporal resolution. It clearly shows better accuracy and reduced false predictions at finer resolutions, but the processing time increases rapidly (see Table 4.3). It is fair that the data sample size increases simultaneously with the time frequency. For the same reason, due to the computational limitations of the computers used in the experiments, the 1-minute data set was trained using only a 2-month sample, which is much less than the 6.9-year training sample used in the rest of data sets. However, this also demonstrates that the 1-minute dataset can be used to predict building energy consumption with a relatively small number of samples.

The predictive performance of the LSTM is compared between univariate and multivariate attributes using an appropriate configuration framework to investigate the relationship between temporal resolution and weather data. LSTM neural networks are able to solve forecasting problems with multiple input variables simply. In general, multivariate attributes provide more information for model training in recurrent neural networks to obtain better predictions.

Figure 4.18 shows that the prediction performance of the multivariate LSTM model is not much improved compared to the univariate LSTM model. From Table 4.4, we can see that the accurate evaluation of the multivariate LSTM model is actually slightly lower than that of the single-variable LSTM model. Figure 4.19 shows a comparison of univariate LSTM and multivariate LSTM. The processing time increases rapidly while accuracy only improves slightly from the univariate LSTM model to the multivariate LSTM model. Therefore, we have shown that it can achieve a robust commercial building energy forecast with only univariate inputs.

Table 4.3: Performance metrics of univariate LSTM network

Resolution	**MAPE (%)**	**MAE (kWh)**	**NRMSE (%)**	**RMSE (kWh)**	**Process Time (second)**
Daily	2.367	16.568	0.125	19.610	12.89
Hourly	0.012	1.973	0.118	2.955	486.93
30-Minutely	0.002	0.692	0.068	1.156	1782.66
15-Minutely	4.29 e-04	0.288	0.056	0.480	7220.89
6-Minutely	5.9 e-05	0.099	0.039	0.146	11751.21
1-Minutely	9.72 e-07	0.010	0.028	0.019	1620.22 (estimated: 67077.11)

Table 4.4: Performance metrics of multivariate LSTM network

Resolution	**MAPE (%)**	**MAE (kWh)**	**NRMSE (%)**	**RMSE (kWh)**	**Process Time (second)**
Daily	3.272	22.901	0.166	26.051	11.82
Hourly	0.010	1.722	0.115	2.863	251.05
30-Minutely	0.003	0.786	0.065	1.103	3103.60
15-Minutely	3.84 e-04	0.258	0.050	0.423	12653.64
6-Minutely	6.1 e-05	0.102	0.044	0.166	23795.55
1-Minutely	9 e-06.	0.014	0.045	0.025	667.62 (estimated: 110557.87)

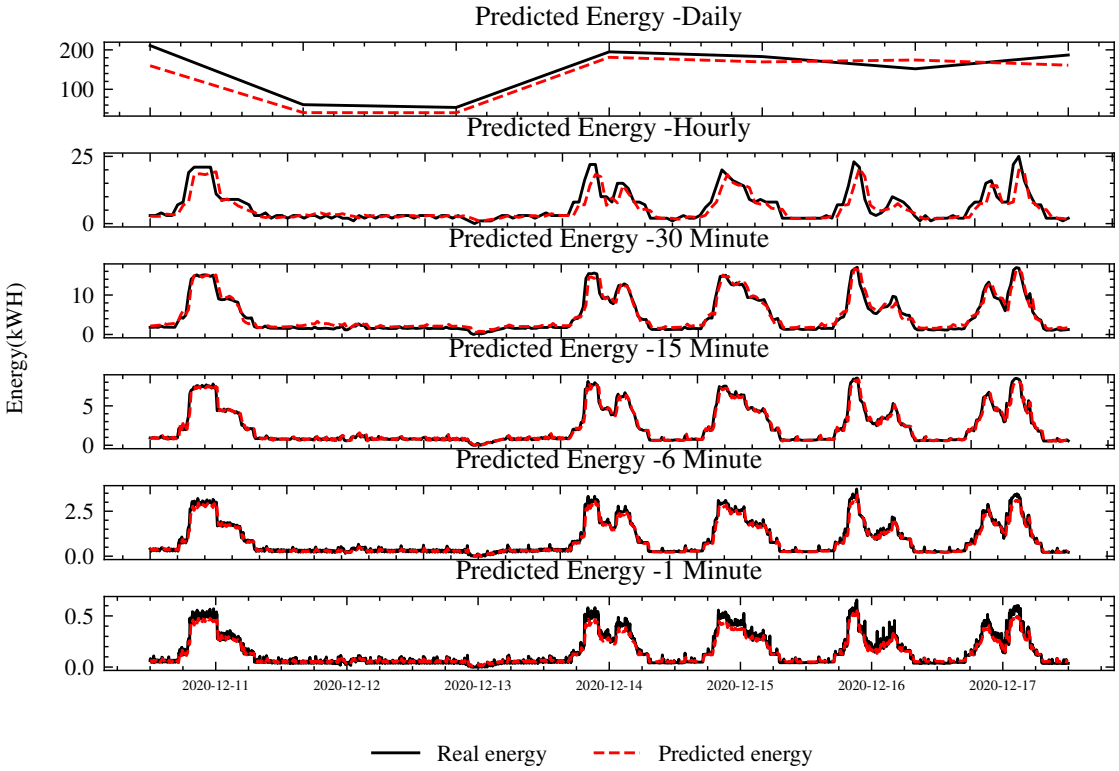

Figure 4.18: Energy load forecasting by multivariate LSTM

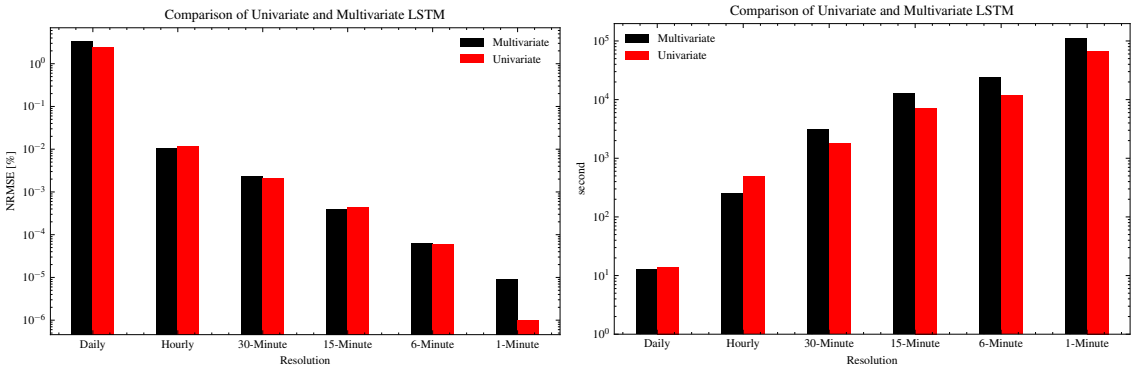

Figure 4.19: Comparison of univariate LSTM and multivariate LSTM

4.5 Summary

This work analyzes the impact of temporal resolution in the mining and learning of building electrical load profiles. Several algorithms have been systematically tested by changing the resolution of the input data (actual building energy consumption). The results are evaluated with benchmark metrics and compared through visualization tools.

Data mining of load curve data is achieved by several techniques. The time-series data are normalized using a min-max scaler to observe the variation within the load curve of building energy consumption using a heat map format. In addition, the correlation between energy and weather scenarios is investigated on the Spearman rank-order coefficient for extracting time features. In addition to temporal features, another important knowledge of pattern feature identification is analyzed based on breakout detection, which evaluates the presence of seasonality in the data. Data resolutions lower than the hourly resolution show more significant load patterns.

The quality of the learning algorithms is measured using various internal evaluators. The reliability of k-means cluster members is based on MIA, CDI, and DBI evaluators. The 15-minute resolution data show better performance and are able to show typical commercial building energy load curves. On the other hand, the 15-minute resolution data performs best results through the LSTM network and balances accuracy and processing time.

Chapter 5

Discussion

5.1 Key Findings

In this thesis dissertation, data mining performs better with low-resolution datasets (e.g., hourly and daily resolution). The results of the energy and weather correlation analysis can provide implications for the selection of model input features, and the breakout detection is suitable for checking the abnormal activity of the building performance system. Low temporal datasets bring the benefit of smaller datasets and result in shorter runtime, thus allowing analysts to extract data knowledge faster. Fast processing time also reduces computational costs, resulting in less complex hardware systems and the use of cheaper computers. In addition, fast extraction of data knowledge allows analysts to apply the extracted information to machine learning later, leading to the appropriate selection of relevant features and the appropriate setting of hyperparameters. The rapid creation of data mining results allows the analyst or controller to react to any abnormal activity at an early stage.

In machine learning, 15-minute resolution data is used to create a clear cluster of commercial load patterns through k-means clustering algorithms, providing a useful basis for distribution network operators and building performance system controllers. The 15-minute resolution data also shows the best results through the LSTM network, balancing accuracy and processing time, and the 15-minute data is also commonly used for decisions from grid control. Consistency in the temporal resolution of the grid-side and building-side data sets can provide a more efficient communication system for demand-side management and building energy management systems. Energy consumption data is extracted from a single commercial building, and the quality of the dataset has a significant impact on model performance, especially for missing data. From the energy consumption prediction results of the LSTM neural network, the neural network model is able to predict future energy consumption based on short-term datasets with the high temporal resolution, thus reducing the impact of missing data.

5.2 Implication of the Study

The primary implication of this study is the development and selection of temporal data resolution for commercial energy consumers and the application with optimal configuration to investigate data mining and construct machine learning models. Where most existing research has been conducted in the built environment over low temporal resolution data sets and limited resolution scale, this work tests the performance of Spearman rank-rank order correlation, breakout detection, clustering techniques, and LSTM neural networks used in the commercial energy sector at various temporal resolutions over a seven-year period. Best-practice approaches are validated by providing a robust comparison of commonly used temporal resolution from the grid side and building side and high-frequency data set.

The temporal resolution of data sets can cause a significant difference when processing big data

in terms of accuracy and runtime. Significantly, the errors generated at one moment are likely inherited to following learning that emphasizes the demand for usable structure and fast computation. Based on the proper data resolution selection, the earlier reaction from the breakout can be done in online learning with a shorter runtime, resulting in a favorable systematic environment for both controller and consumers. A helpful basis will also be created for distribution network operators in energy load clustering.

Unlike most previous studies that select one or two temporal resolution data for machine learning in the built environment, this study develops a qualitative evaluation framework that compares six resolutions. By pointing out the appropriate data resolution for forecasting with qualitative evaluation measures, an LSTM structure is selected that is more adaptable, representative, and simple than what would have otherwise been the case. This has the advantage of reducing the uncertainty on electrical system control when selecting the best temporal resolution.

This work will be of particular interest to energy generation and utility companies that are seeking to develop and maintain their management system.

5.3 Limitation of the Case Study

This thesis project is a limited scenario with a distinct scope, and not all possible experiments can be completed due to several variables such as time, data type, and even computational capability. The case study was selected from the load curve of the Kropman office building, which contains only one commercial building and represents a small percentage of the whole building type. Other dwelling types such as residential buildings, schools, and universities were excluded from this study. In addition, spatial factors were excluded because of potentially unreliable instrumentation between each office.

The data available for this case study was 85 percent, and all data from June to July 2013 were missing and could not be interpolated due to the large time interval. Therefore, only seven years of data from 2014 to 2020 were used in this study.

On the other hand, the limited CPU and RAM size of the experimental computer and the lack of GPU prevented the use of larger neural network architecture to process large data sets. The computers themselves are different from those used in industry, which may lead to different results, such as processing time.

5.4 Limitation of the Used Methods

In this project, the dataset used is based on a specific time series phenomenon: the energy consumption of a single commercial building in the Netherlands. Due to the domain of the study, it was difficult to find other datasets to test the findings and results of the paper. Therefore, the exact mathematical relations and modifications of the neural network architecture cannot be directly applied to other problems and cannot be used to change the temporal resolution of the data. However, the general idea of reducing the learning rate by adding layers on higher temporal resolution data should be successfully applied. The problem will be more related to tuning the model parameters rather than studying the relationship between the results and the various temporal resolutions, as this work is done in this thesis project.

The breakout detection inspects for abrupt changes in the time series data under the constraint level given from an initial time threshold. The break is determined based on the variation in the time series data, so it is challenging to apply it to very high-resolution data. Breakout detection results for 1-minute resolution data have been very difficult to detect any break within a year.

The applied unsupervised machine learning clustering algorithm is K-means clustering. The starting prime is determined randomly, and the number of clusters is chosen based on their MIA, DBI, and CDI scores and the generated profiles. However, these choices in determining the optimal number of clusters are not fixed. The optimal value depends on the results and knowledge of the data, so it is difficult to reproduce it on different data samples.

5.5 Comparison with Other Research

The results of this study are compared with other recent studies that have examined the effects of temporal resolution. The research areas have all investigated the performance of machine learning algorithms in the built environments. Granell et al. showed that the k-means algorithm is robust to data resolution effects in the 4-60 minute time resolution range [72]. The k-means algorithm is faster and is suitable for this type of research area. Better results are obtained at a frequency of at least 30 minutes; ideally, 15 minutes or more, which helps electricity retailers to identify differences between consumers' time-series electricity usage [72]. On the other hand, data collected over 30 minutes will provide a valuable basis for distribution network operators, which is consistent with the results of this thesis project.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This work analyzes the impact of temporal resolution on data mining and machine learning in the built environment. Several algorithms are systematically tested by varying the resolution of the input data (actual building consumption). The results are evaluated by visualization format and performance benchmarking metrics.

Different temporal resolutions affect the results of the Spearman Rank Order Coefficient correlation analysis. Lower than hourly resolutions provide a better indication of significant changes in correlations between energy consumption load and weather scenarios, especially between energy and temperature, humidity, and solar radiation. Breakout detection performs the optimal results at a similar temporal resolution, where daily data are detected with the highest number of breaks throughout the year. The greater the fluctuations in low-resolution data, the more significant the changes in user behavior, ensuring that building managers are more likely to be aware of potential abnormal activity. In contrast, higher resolution data such as 15 minutes achieves a good balance between accuracy and runtime when clustering and predicting through the simple k-means algorithm and LSTM neural network, respectively.

The proposed work implies that a frequency of at least 15 minutes is required to help the building managers and grid controllers to identify the differences between time-series energy usage data. The consistency of the grid-side and building-side temporal resolution allows the analyst to implement the control or analysis model more efficiently. We conclude that it is the best strategy for extracting data mining knowledge from low-resolution data (e.g., daily) and for load pattern clustering and energy consumption prediction using higher resolution data (e.g., 15 minutes).

6.2 Review of the Goals

At the beginning of this project, several goals were pronounced and define the scope of the thesis project. These objectives will be analyzed along with the final statement of these previous considerations.

1. Identify the effects of various time resolution frequencies on data mining and machine learning in the built environment.
The project finds a way to examine the impact of various temporal resolution frequencies on data mining and machine learning, including the process of exporting multi-step LSTM models from specific temporal resolutions to different temporal resolutions based on the increase of layers and in the decreasing of learning rates. In addition, defined clustering index values are established to perform robust k-means clustering analysis for different temporal resolutions. Temporal features and pattern recognition tools are also developed as well as appropriate visualization tools and

parameter settings.

2. Specify use cases on different time resolutions for Kropman building data.
The results show that temporal resolution is an essential factor in mining and learning Kropman building data. Lower resolution data (e.g., daily and hourly data) show more significant energy consumption patterns than other time-frequency data. The higher resolution data allows the model to capture the hidden structure and patterns of the time series sequences, which leads to better results.

3. Analyze the effects of resampling (Interpolation) due to data-imbalanced conditions on modeling.
Detection of data-imbalanced conditions is incomplete in this study due to the modeling results reach nearly 100% accuracy. The false-positive and false-negative may exist in the clustering evaluation where a small number of unique load patterns are visible at both 15-minutely and 6-minutely resolution. The other evaluator should be applied, such as F-score. Besides that, the best approach for changing the resolution of a time series data in this case study is to apply down sampling. Down sampling is grouping the time series that dataset becomes more normalized and avoiding the outliers. At the same time, the down sampled data still conserve the features and properties of the original data.

4. Formulate and verify a general estimate of uncertainty due to resampling.
The metrics found are CDI, MIA, and DBI for cluster analysis; MAPE and NRMSE for LSTM network models. these metrics show the error in relative performance measurements independent of size and allow us to compare the actual level of error between actual and predicted values since the error can be compared to other time-series data with various time frequencies and measurements.

6.3 Future Work

During the development of the master's thesis project, several ideas have been on my mind, some of which could not be implemented because they were not related to the thesis objectives or there was not enough time to implement them. All these ideas will be presented below to indicate future areas of research.

1. Studying deeper of the machine learning for creating a structure that improves the current results. More rigorous analysis of techniques, including the k-means with different numbers of clusters and LSTM neural networks with different architecture, is warranted.

2. Create a method to find the best hyperparameter settings based on some steps automatically. Epochs will be affected by the batch size, depending on the time steps and resolution. Online learning shows an advantage with updating weights after each training instead of modifying the batch size, which can learn faster and simplifying the process.

3. Embedding spatial analysis in research to enhance spatio-temporal features and break the limits of building type. The current study is limited to commercial buildings in the southern Netherlands. However, the combination of spatial analysis would be a useful adaption in a subsequent study, such as different dwelling types and locations. Multiple applications would be extended to fit in more circumstances.

Bibliography

[1] IEA. World energy outlook 2019. *IEA, Paris*, 2019.

[2] United States Energy Information Administration (EIA). International energy outlook 2017. 2017.

[3] Hyojoo Son and Changwan Kim. A deep learning approach to forecasting monthly demand for residential-sector electricity. *Sustainability (Switzerland)*, 12:3103, 4 2020.

[4] Baran Yildiz, Jose I Bilbao, Jonathon Dore, and Alistair B Sproul. Recent advances in the analysis of residential electricity consumption and applications of smart meter data. *Applied Energy*, 208:402–427, 2017.

[5] Rishee K. Jain, Kevin M. Smith, Patricia J. Culligan, and John E. Taylor. Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy. *Applied Energy*, 123:168–178, 6 2014.

[6] Yibo Chen, Hongwei Tan, and Umberto Berardi. Day-ahead prediction of hourly electric demand in non-stationary operated commercial buildings: A clustering-based hybrid approach. *Energy and Buildings*, 148:228–237, 8 2017.

[7] Soheil Fathi, Ravi Srinivasan, Andriel Fenner, and Sahand Fathi. Machine learning applications in urban building energy performance forecasting: A systematic review. *Renewable and Sustainable Energy Reviews*, 133, 11 2020.

[8] Jacopo Torriti. A review of time use models of residential electricity demand. *Renewable and Sustainable Energy Reviews*, 37:265–272, 2014.

[9] Elena Gonzalez, Bruce Stephen, David Infield, and Julio J. Melero. Using high-frequency scada data for wind turbine performance monitoring: A sensitivity study. *Renewable Energy*, 131:841–853, 2 2019.

[10] Nan Wei, Changjun Li, Xiaolong Peng, Fanhua Zeng, and Xinqian Lu. Conventional models and artificial intelligence-based models for energy consumption forecasting: A review. *Journal of Petroleum Science and Engineering*, 181:106187, 2019.

[11] Tanveer Ahmad, Hongcai Zhang, and Biao Yan. A review on renewable energy and electricity requirement forecasting models for smart grid and buildings. *Sustainable Cities and Society*, 55, 4 2020.

[12] Kadir Amasyali and Nora M El-Gohary. A review of data-driven building energy consumption prediction studies. *Renewable and Sustainable Energy Reviews*, 81:1192–1205, 2018.

[13] Chirag Deb, Fan Zhang, Junjing Yang, Siew Eang Lee, and Kwok Wei Shah. A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74:902–924, 2017.

[14] Mathieu Bourdeau, Xiao qiang Zhai, Elyes Nefzaoui, Xiaofeng Guo, and Patrice Chatellier. Modeling and forecasting building energy consumption: A review of data-driven techniques. *Sustainable Cities and Society*, 48:101533, 2019.

[15] Anastasia Ushakova and Slava Jankin Mikhaylov. Big data to the rescue? challenges in analysing granular household electricity consumption in the united kingdom. *Energy Research & Social Science*, 64:101428, 2020.

[16] Hai-xiang Zhao and Frédéric Magoulès. A review on the prediction of building energy consumption. *Renewable and Sustainable Energy Reviews*, 16(6):3586–3592, 2012.

[17] Cristina Heghedus, Antorweep Chakravorty, and Chunming Rong. Energy load forecasting using deep learning. pages 146–151. Institute of Electrical and Electronics Engineers Inc., 7 2018.

[18] Kaile Zhou and Shanlin Yang. Understanding household energy consumption behavior: The contribution of energy big data analytics. *Renewable and Sustainable Energy Reviews*, 56:810–819, 2016.

[19] Kaile Zhou, Chao Fu, and Shanlin Yang. Big data driven smart energy management: From big data to big insights. *Renewable and Sustainable Energy Reviews*, 56:215–225, 2016.

[20] Xiping Wang and Ming Meng. A hybrid neural network and arima model for energy consumption forcasting. *J. Comput.*, 7(5):1184–1190, 2012.

[21] Harveen Kaur and Sachin Ahuja. Time series analysis and prediction of electricity consumption of health care institution using arima model. In *Proceedings of Sixth International Conference on Soft Computing for Problem Solving*, pages 347–358. Springer, 2017.

[22] Sumit Kumar, Lasani Hussain, Sekhar Banarjee, and Motahar Reza. Energy load forecasting using deep learning approach-lstm and gru in spark cluster. In *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, pages 1–4. IEEE, 2018.

[23] Jesus-Casa Hernandez, F Sanchez-Sutil, A Cano-Ortega, and Carlos R Baier. Influence of data sampling frequency on household consumption load profile features: A case study in spain. *Sensors*, 20(21):6034, 2020.

[24] Tae-Young Kim and Sung-Bae Cho. Predicting residential energy consumption using cnn-lstm neural networks. *Energy*, 182:72–81, 2019.

[25] Daniel L Marino, Kasun Amarasinghe, and Milos Manic. Building energy load forecasting using deep neural networks. In *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 7046–7051. IEEE, 2016.

[26] Yaping Zhou, Zhun Jerry Yu, Jun Li, Yujian Huang, and Guoqiang Zhang. The effect of temporal resolution on the accuracy of predicting building occupant behaviour based on markov chain models. *Procedia Engineering*, 205:1698–1704, 2017.

[27] Arun Sukumaran Nair, Tareq Hossen, Mitch Campion, and Prakash Ranganathan. Optimal operation of residential evs using dnn and clustering based energy forecast. In *2018 North American Power Symposium (NAPS)*, pages 1–6. IEEE, 2018.

[28] A Sancho-Tomás, M Sumner, and Darren Robinson. A generalised model of electrical energy demand from small household appliances. *Energy and Buildings*, 135:350–366, 2017.

[29] Zhaoxuan Li and Bing Dong. A new modeling approach for short-term prediction of occupancy in residential buildings. *Building and Environment*, 121:277–290, 2017.

[30] Nastaran Bassamzadeh and Roger Ghanem. Multiscale stochastic prediction of electricity demand in smart grids using bayesian networks. *Applied energy*, 193:369–380, 2017.

[31] Kwonsik Song, Kyle Anderson, SangHyun Lee, Kaitlin T Raimi, and P Hart. Non-invasive behavioral reference group categorization considering temporal granularity and aggregation level of energy use data. *Energies*, 13(14):3678, 2020.

[32] R Sendra-Arranz and A Gutiérrez. A long short-term memory artificial neural network to predict daily hvac consumption in buildings. *Energy and Buildings*, 216:109952, 2020.

[33] Peter Lusis, Kaveh Rajab Khalilpour, Lachlan Andrew, and Ariel Liebman. Short-term residential load forecasting: Impact of calendar effects and forecast granularity. *Applied energy*, 205:654–669, 2017.

[34] Widyaning Chandramitasari, Bobby Kurniawan, and Shigeru Fujimura. Building deep neural network model for short term electricity consumption forecasting. In *2018 International Symposium on Advanced Intelligent Informatics (SAIN)*, pages 43–48. IEEE, 2018.

[35] A Jayanth Balaji, DS Harish Ram, and Binoy B Nair. A deep learning approach to electric energy consumption modeling. *Journal of Intelligent & Fuzzy Systems*, 36(5):4049–4055, 2019.

[36] Shailendra Singh and Abdulsalam Yassine. Big data mining of energy time series for behavioral analytics and energy consumption forecasting. *Energies*, 11(2):452, 2018.

[37] Israr Ullah, Rashid Ahmad, and DoHyeun Kim. A prediction mechanism of energy consumption in residential buildings using hidden markov model. *Energies*, 11(2):358, 2018.

[38] Cheng Fan, Fu Xiao, and Shengwei Wang. Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques. *Applied Energy*, 127:1–10, 2014.

[39] Kasun Amarasinghe, Daniel L Marino, and Milos Manic. Deep neural networks for energy load forecasting. In *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pages 1483–1488. IEEE, 2017.

[40] Yibo Chen, Hongwei Tan, and Umberto Berardi. Day-ahead prediction of hourly electric demand in non-stationary operated commercial buildings: A clustering-based hybrid approach. *Energy and Buildings*, 148:228–237, 2017.

[41] Alex Nutkiewicz, Zheng Yang, and Rishee K Jain. Data-driven urban energy simulation (duels): A framework for integrating engineering simulation and machine learning methods in a multi-scale urban energy modeling workflow. *Applied energy*, 225:1176–1189, 2018.

[42] Martin Heine Kristensen, Ruchi Choudhary, and Steffen Petersen. Bayesian calibration of building energy models: Comparison of predictive accuracy using metered utility data of different temporal resolution. *Energy Procedia*, 122:277–282, 2017.

[43] Lingyan Cao, Yongkui Li, Jiansong Zhang, Yi Jiang, Yilong Han, and Jianjun Wei. Electrical load prediction of healthcare buildings through single and ensemble learning. *Energy Reports*, 6:2751–2767, 2020.

[44] Jatin Bedi and Durga Toshniwal. Deep learning framework to forecast electricity demand. *Applied energy*, 238:1312–1326, 2019.

[45] Rodrigo F Berriel, Andre Teixeira Lopes, Alexandre Rodrigues, Flavio Miguel Varejao, and Thiago Oliveira-Santos. Monthly energy consumption forecast: A deep learning approach. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4283–4290. IEEE, 2017.

[46] Tae Young Kim and Sung Bae Cho. Predicting residential energy consumption using cnn-lstm neural networks. *Energy*, 182:72–81, 9 2019.

[47] Tanveer Ahmad, Huanxin Chen, Yabin Guo, and Jiangyu Wang. A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand: A review. *Energy and Buildings*, 165:301–320, 4 2018.

[48] Clayton Miller and Forrest Meggers. Mining electrical meter data to predict principal building use, performance class, and operations strategy for hundreds of non-residential buildings. *Energy and Buildings*, 156:360–373, 2017.

[49] Theophano Mitsa. *Temporal data mining*. CRC Press, 2010.

[50] Frederick R Macaulay et al. The smoothing of time series. *NBER Books*, 1931.

[51] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[52] Twitter. Breakout detection via robust e-statistics. `https://github.com/twitter/BreakoutDetection`, 2014.

[53] Nicholas A James, Arun Kejariwal, and David S Matteson. Leveraging cloud data to mitigate user experience from 'breaking bad'. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 3499–3508. IEEE, 2016.

[54] Wiebke Toussaint. Evaluation of clustering techniques for generating household energy consumption patterns in a developing country. Master's thesis, Faculty of Science, 2019.

[55] Alexander Martin Tureczek and Per Sieverts Nielsen. Structured literature review of electricity consumption classification using smart meter data. *Energies*, 10(5):584, 2017.

[56] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[57] Bishnu Nepal, Motoi Yamaha, Hiroya Sahashi, and Aya Yokoe. Analysis of building electricity use pattern using k-means clustering algorithm by determination of better initial centroids and number of clusters. *Energies*, 12, 2019.

[58] Gianfranco Chicco. Overview and performance assessment of the clustering methods for electrical load pattern grouping. *Energy*, 42(1):68–80, 2012.

[59] Alexander Tureczek, Per Sieverts Nielsen, and Henrik Madsen. Electricity consumption clustering using smart meter data. *Energies*, 11(4):859, 2018.

[60] João Pedro Gouveia and Júlia Seixas. Unraveling electricity consumption profiles in households through clusters: Combining smart meters and door-to-door surveys. *Energy and Buildings*, 116:666–676, 2016.

[61] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[62] Kemal Erdem. t-sne clearly explained. `https://erdem.pl/2020/04/t-sne-clearly-explained`, 2020. Accessed: 2021-05-30.

[63] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[64] Yoshua Bengio. *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[65] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[66] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[67] Christopher Olah. Understanding lstm networks.

[68] Salvador García, Julián Luengo, and Francisco Herrera. *Data preprocessing in data mining*, volume 72. Springer, 2015.

[69] 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, 2020.

[70] Baran Yildiz, Jose I Bilbao, and Alistair B Sproul. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. *Renewable and Sustainable Energy Reviews*, 73:1104–1122, 2017.

[71] Soheil Fathi, Ravi Srinivasan, Andriel Fenner, and Sahand Fathi. Machine learning applications in urban building energy performance forecasting: A systematic review. *Renewable and Sustainable Energy Reviews*, 133:110287, 2020.

[72] Ramon Granell, Colin J Axon, and David CH Wallom. Impacts of raw data temporal resolution using selected clustering methods on residential electricity load profiles. *IEEE Transactions on Power Systems*, 30(6):3217–3224, 2014.

Appendix A

Literature review search result

Figure A.1 and Figure A.2 show the search results based on using the key words as defined search term in Table 3.1.

Research terms for literature review on effect of temporal resolution							
No.	Group	Problem	Intervention	Intervention	Intervention	Hits	New Hits
		Fixed	Focus 1	Focus 2	Outcome		
1	A	temporal	building	energy consumption	forecasting	15	15
2	B	temporal	building	energy consumption	accuracy	19	13
3	C	temporal	building	energy demand	forecasting	3	1
4	D	temporal	building	energy demand	accuracy	9	6
5	E	temporal	household	energy consumption	forecasting	4	2
6	F	temporal	household	energy consumption	accuracy	3	3
7	G	temporal	household	energy demand	forecasting	3	1
8	H	temporal	household	energy demand	accuracy	4	0
9	I	temporal	residential	energy consumption	forecasting	6	0
10	J	temporal	residential	energy consumption	accuracy	6	0
11	K	temporal	residential	energy demand	forecasting	4	1
12	L	temporal	residential	energy demand	accuracy	6	1
13	M	temporal resolution	building	energy consumption	forecasting	2	0
14	N	temporal resolution	building	energy consumption	accuracy	2	0
15	O	temporal resolution	building	energy demand	forecasting	0	0
16	P	temporal resolution	building	energy demand	accuracy	0	0
17	Q	temporal resolution	household	energy consumption	forecasting	1	0
18	R	temporal resolution	household	energy consumption	accuracy	0	0
19	S	temporal resolution	household	energy demand	forecasting	0	0
20	T	temporal resolution	household	energy demand	accuracy	0	0
21	U	temporal resolution	residential	energy consumption	forecasting	0	0
22	V	temporal resolution	residential	energy consumption	accuracy	0	0
23	W	temporal resolution	residential	energy demand	forecasting	1	0
24	X	temporal resolution	residential	energy demand	accuracy	1	0
25	Y	temporal granularity	building	energy consumption	forecasting	0	0
26	Z	temporal granularity	building	energy consumption	accuracy	1	0
27	AA	temporal granularity	building	energy demand	forecasting	0	0
28	AB	temporal granularity	building	energy demand	accuracy	1	0
29	AC	temporal granularity	household	energy consumption	forecasting	0	0

Figure A.1: Literature review search results based on the defined search terms-1

30	AD	temporal granularity	household	energy consumption	accuracy	0	0
31	AE	temporal granularity	household	energy demand	forecasting	0	0
32	AF	temporal granularity	household	energy demand	accuracy	1	0
33	AG	temporal granularity	residential	energy consumption	forecasting	0	0
34	AH	temporal granularity	residential	energy consumption	accuracy	0	0
35	AI	temporal granularity	residential	energy demand	forecasting	0	0
36	AJ	temporal granularity	residential	energy demand	accuracy	1	0
37	AK	resampling	building	energy consumption	forecasting	0	0
38	AL	resampling	building	energy consumption	accuracy	1	1
39	AM	resampling	building	energy demand	forecasting	0	0
40	AN	resampling	building	energy demand	accuracy	0	0
41	AO	resampling	household	energy consumption	forecasting	0	0
42	AP	resampling	household	energy consumption	accuracy	0	0
43	AQ	resampling	household	energy demand	forecasting	0	0
44	AR	resampling	household	energy demand	accuracy	0	0
45	AS	resampling	residential	energy consumption	forecasting	0	0
46	AT	resampling	residential	energy consumption	accuracy	0	0
47	AU	resampling	residential	energy demand	forecasting	0	0
48	AV	resampling	residential	energy demand	accuracy	0	0
49	AW	sampling frequency	building	energy consumption	forecasting	0	0
50	AX	sampling frequency	building	energy consumption	accuracy	1	1
51	AY	sampling frequency	building	energy demand	forecasting	0	0
52	AZ	sampling frequency	building	energy demand	accuracy	0	0
53	BA	sampling frequency	household	energy consumption	forecasting	0	0
54	BB	sampling frequency	household	energy consumption	accuracy	0	0
55	BC	sampling frequency	household	energy demand	forecasting	0	0
56	BD	sampling frequency	household	energy demand	accuracy	0	0
57	BE	sampling frequency	residential	energy consumption	forecasting	0	0
58	BF	sampling frequency	residential	energy consumption	accuracy	0	0
59	BG	sampling frequency	residential	energy demand	forecasting	0	0
60	BH	sampling frequency	residential	energy demand	accuracy	0	0

Figure A.2: Literature review search results based on the defined search terms-2

Appendix B

Correlation heat map

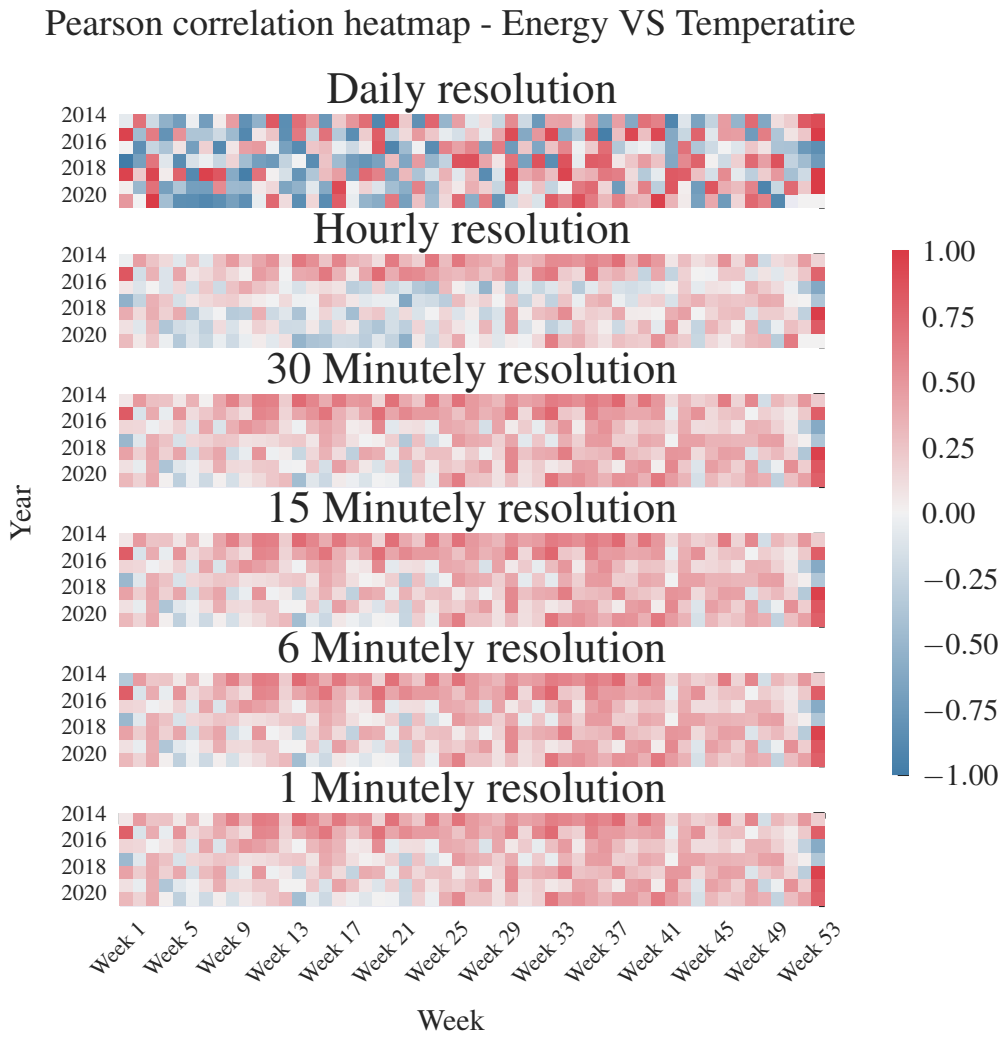

Figure B.1: Pearson correlation heat map energy v.s. temperature

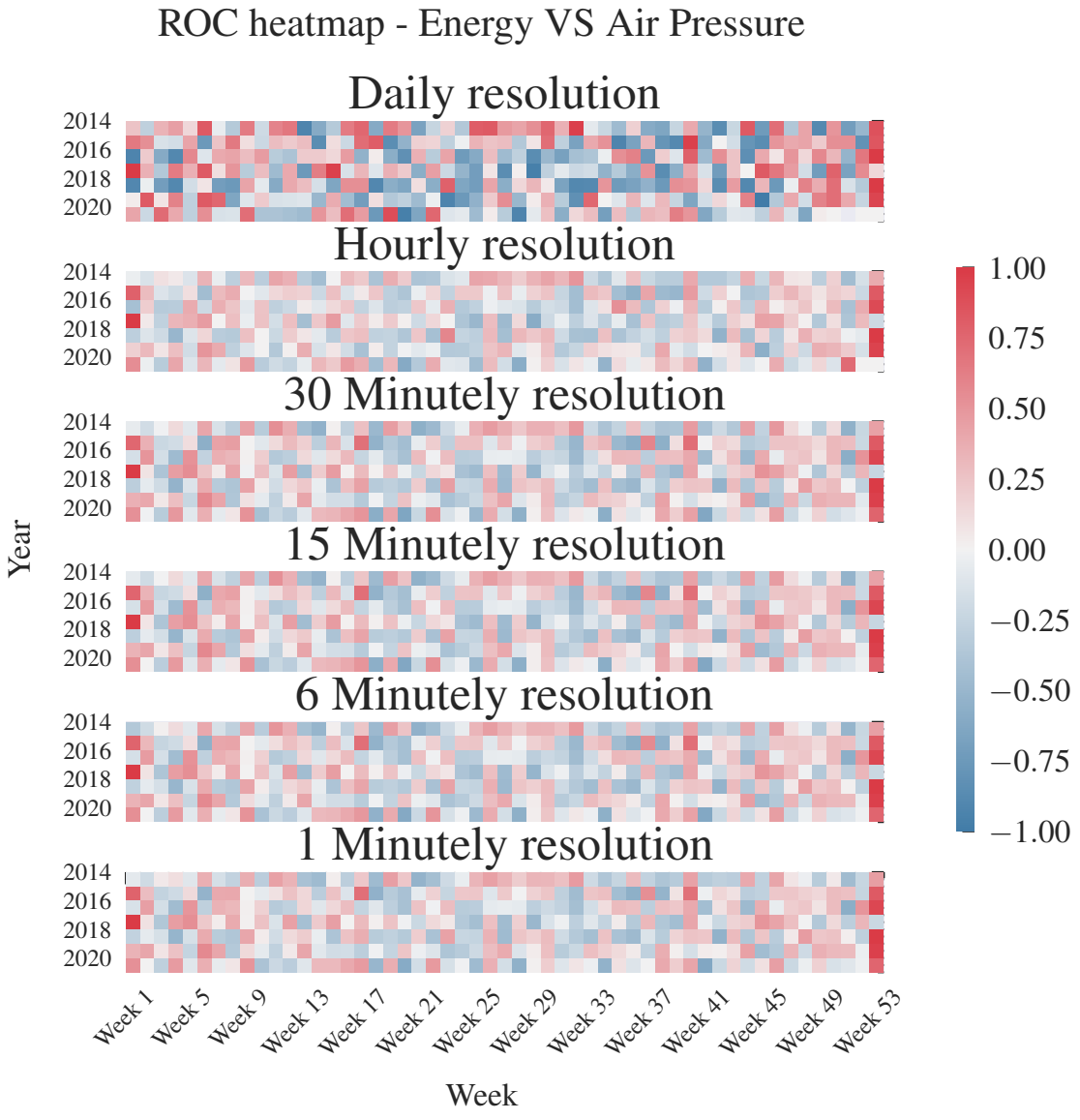

Figure B.2: Spearman rank order correlation heat map energy v.s. pressure

Appendix C

Pattern identification

Figure C.1: Decomposition of 1-hourly data with 24 hours frequency

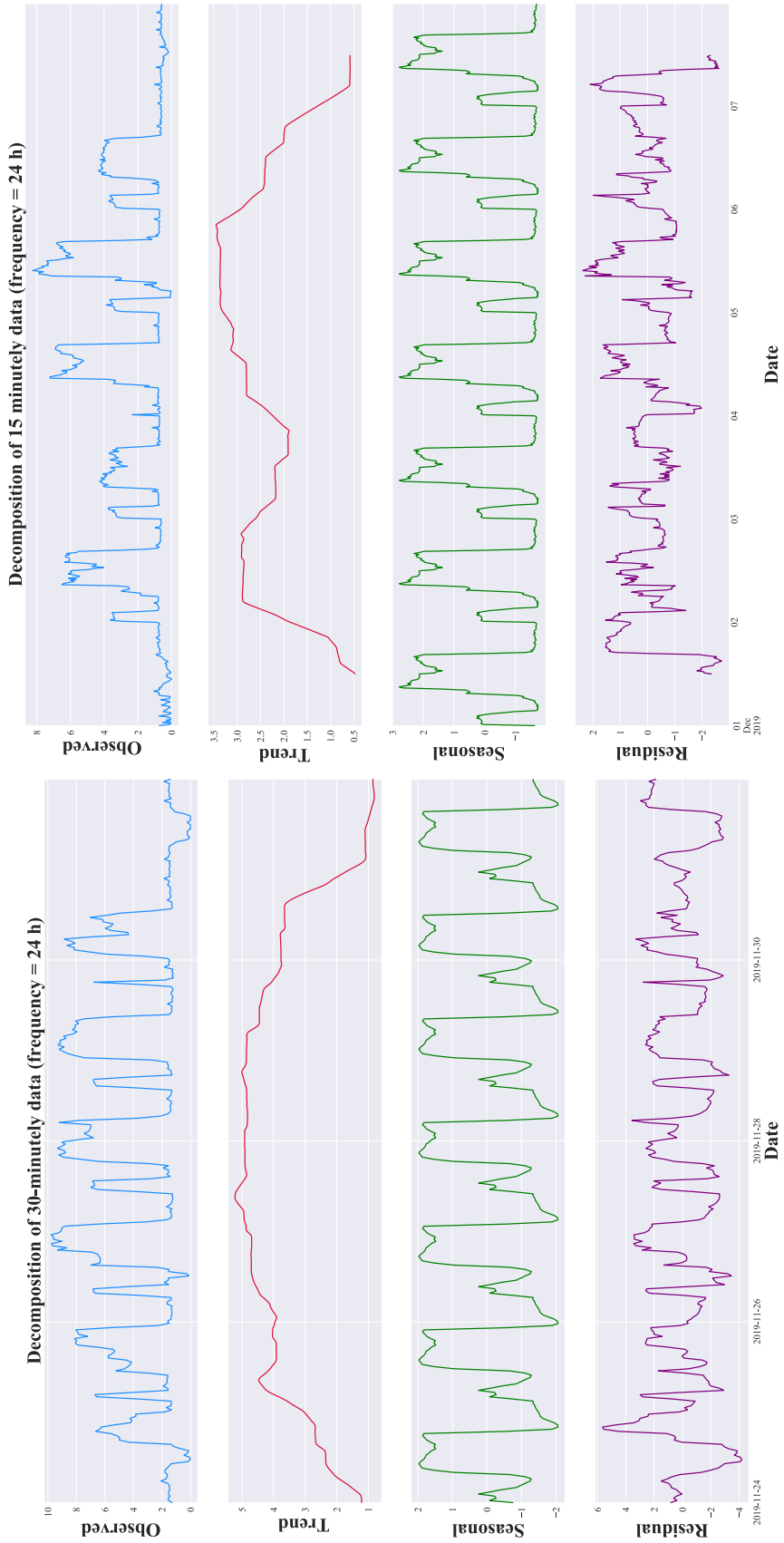

Figure C.2: Decomposition of 30-minutely data and 15-minutely data with 24 hours frequency

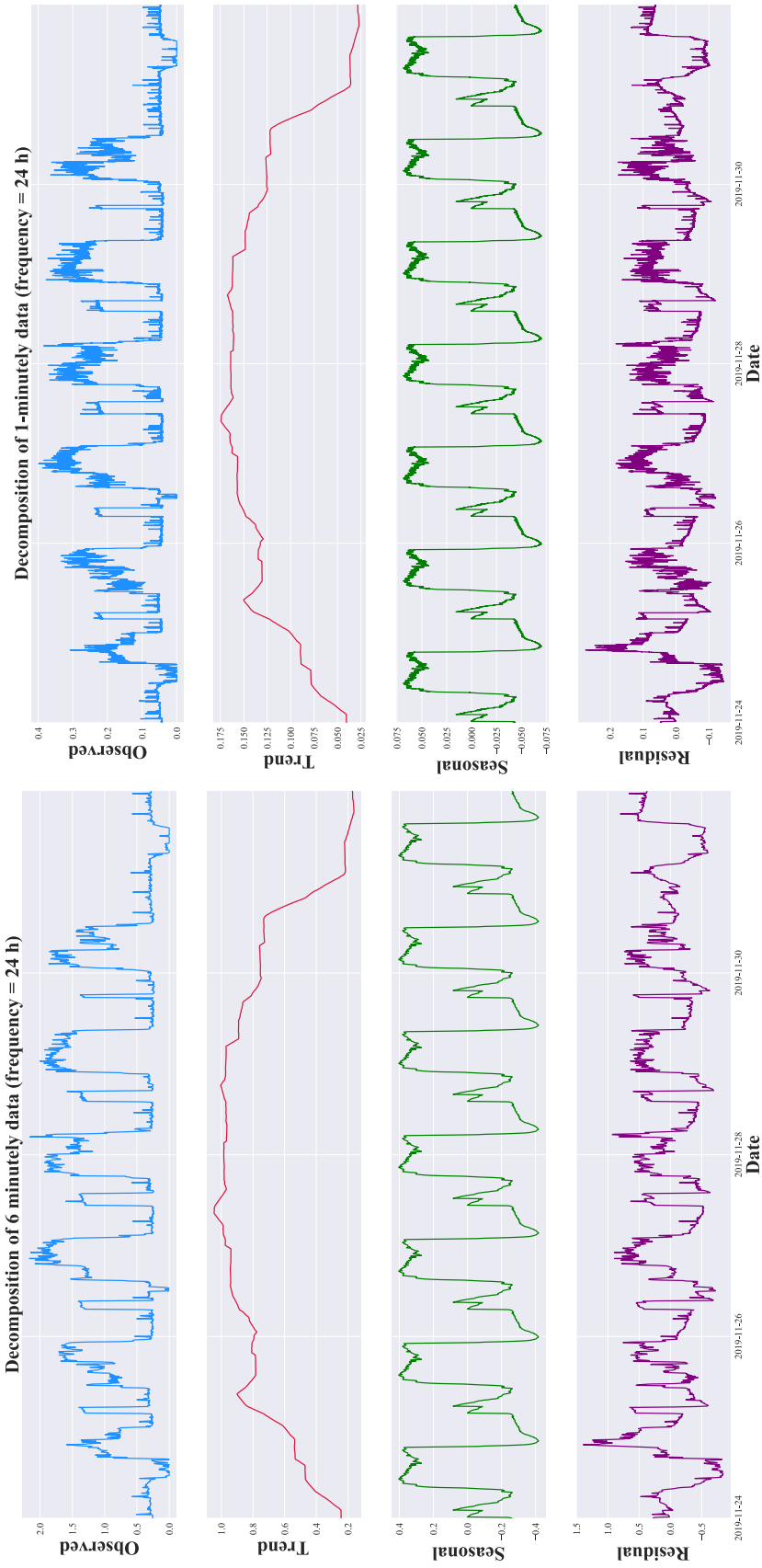

Figure C.3: Decomposition of 6-minutely and 1-minutely data with 24 hours frequency

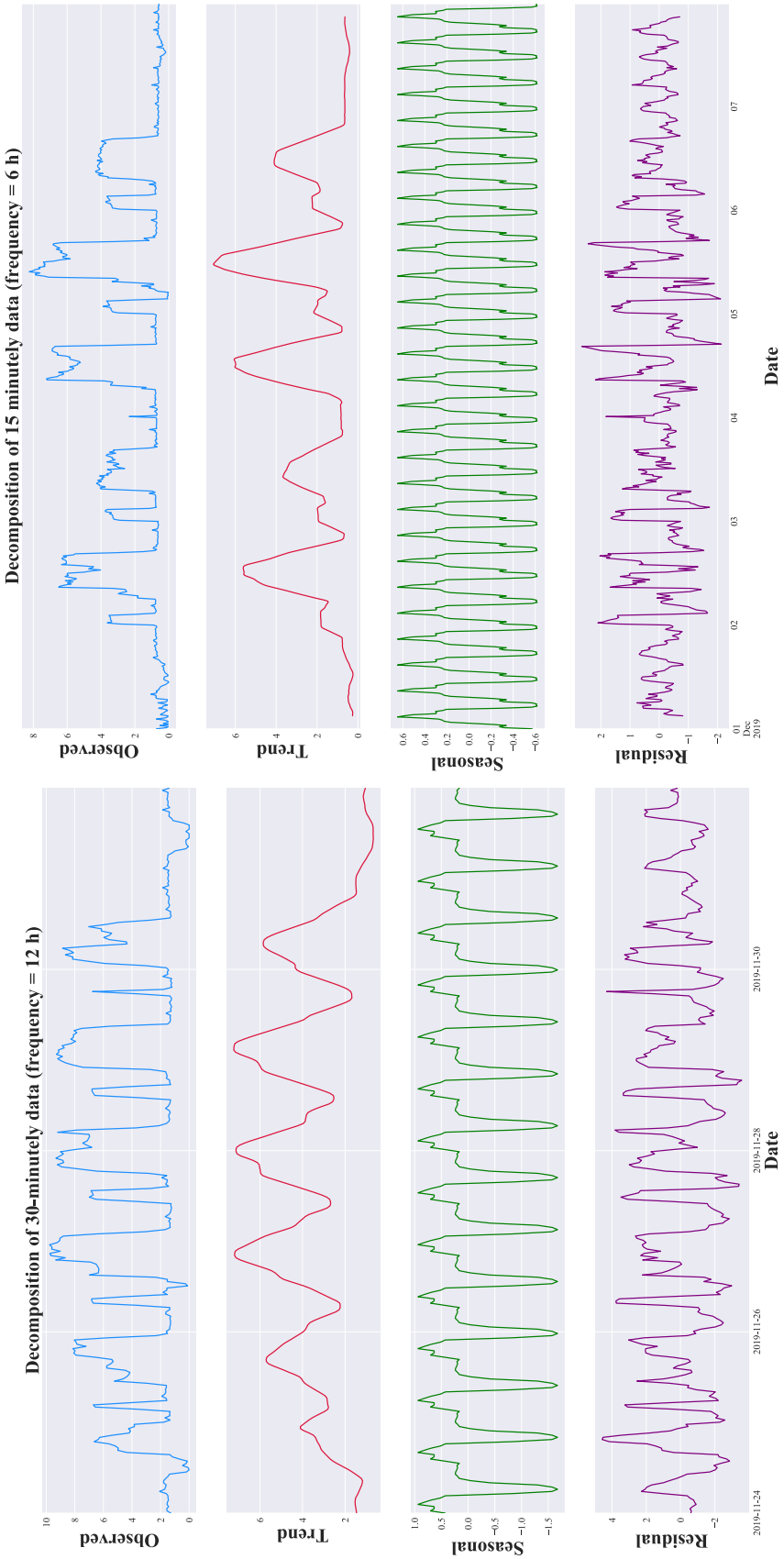

Figure C.4: Decomposition of 30-minutely data and 15-minutely data with various frequencies

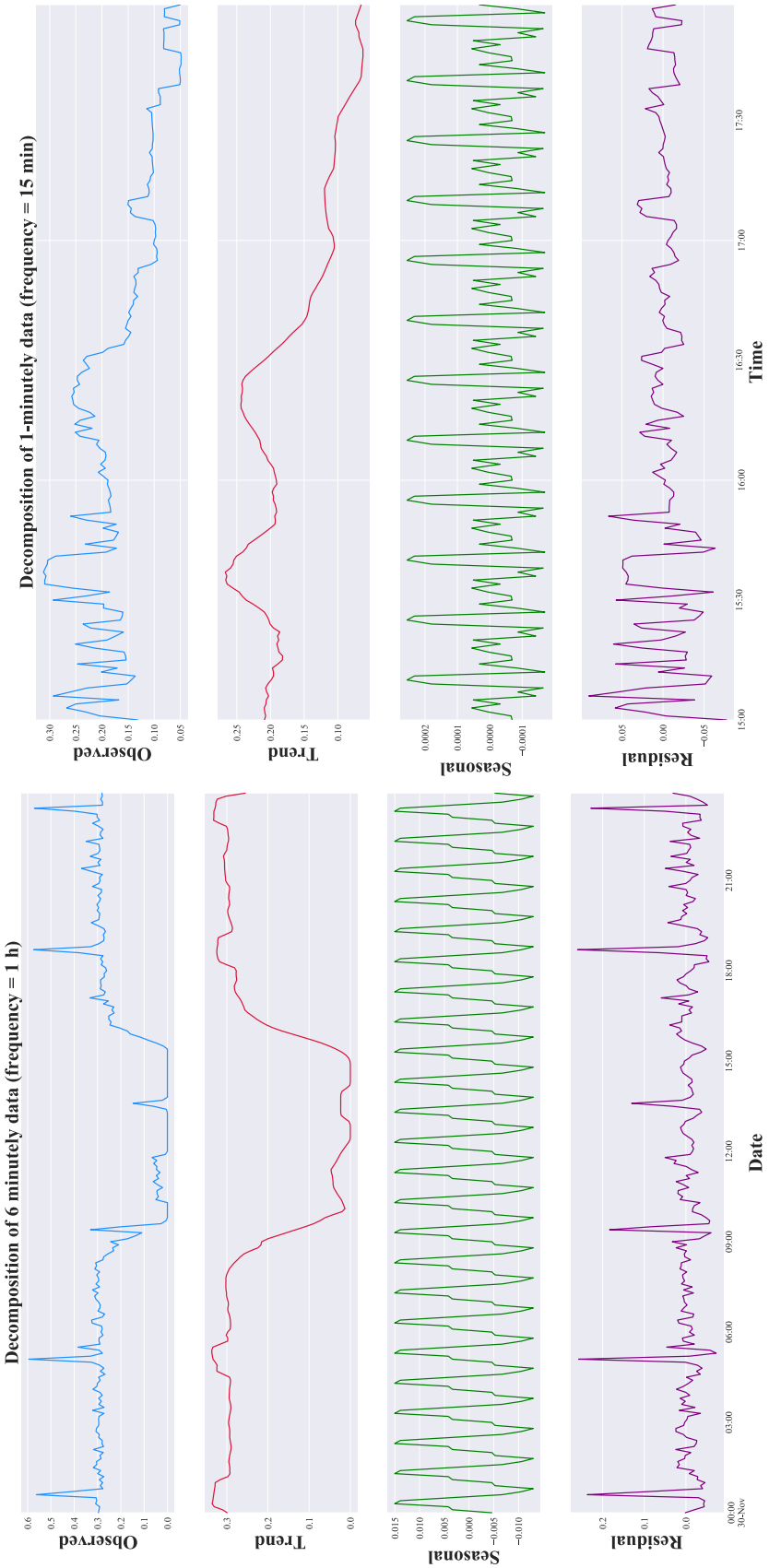

Figure C.5: Decomposition of 6-minutely and 1-minutely data with various frequencies