Neural Network Forecast Model Development of Energy Consumption in Houses

EcoGenie

Bauke Bergsma

The Hague University

THE HAGUE
UNIVERSITY OF
APPLIED SCIENCES

Neural Network Forecast Model Development of Energy Consumption in Houses

EcoGenie

by

Bauke Bergsma

in partial fulfillment of the requirements for the degree of

Master of Science
in Applied Physics

at The Hague University of Applied Sciences,

Student number:	14099853
Project duration:	February 5, 2018 – July 6, 2018
External Supervisor:	Dr.-Ing. P. Breithaupt, Shell Global Solutions
Supervisor:	Prof. dr. R. Vellekoop
Administrator, Coordinator:	Prof. dr. A. Lock
Study Career Counsellor:	Prof. dr. N. van der Houwen

THE HAGUE
UNIVERSITY OF
APPLIED SCIENCES

Preface

My internship at Shell Global Solutions has given me the opportunity to learn many skills and its application thereof. I am extremely thankful for the guidance my external supervisor, Dr. Peter Breithaupt, has given me. He has given me focus in a sea of knowledge, where otherwise I would get lost learning (interesting and) irrelevant areas, loosely connected to the objectives. Our discussions always inspired me, and it has been delightful to gain insight in the complexity of neural networks, machine & deep learning. I will take his advice and give back to the (open-source) community, by sharing my gained knowledge within the R-programming-language. I would also like to thank him for giving me crucial feedback on this internship report. I thank Vaidehi Parab for the insight into earlier research at the EcoGenie project.

I would like to thank my supervisor, René Vellekoop, for his support, advice and insight. I also would like to express my gratitude of the opportunity of doing this internship to my coordinator Arjan Lock and study career counselor Nico van der Houwen.

Finally, I would like to express my dearest gratitude towards my parents Marja & Gerard Bergsma, who have always supported me with love and inspiration, and who have shown a great deal of curiosity in this project and report.

Bauke Bergsma
Delft, July 2018

Summary

The objectives of the internship were to learn and apply ”big data” analysis techniques, and to explore and model relevant energy data from the EcoGenie house, spanning several years. The primary objectives were learning, applying and analyzing measurement data, with the secondary objective being modelling (with limited) measurement (input) data with training a neural network to derive an energy forecast model. The Master Thesis by Parab, V. (2016) has proven a valuable reference to describe The EcoGenie Project and the Thermal Network Model analyzed and applied in 2016. Literature on physics and the basics of neural networks and deep learning have been studied. The R programming language has been practiced and code refined for many specific and general purposes.

The first 2 months had a focus on learning the R-language (programming) and its Shiny package (used to set up servers and convert R code into HTML), while getting to know the content of the acquired data of the measuring equipment. The Anaconda Navigator, R-Studio, and Jupyter Notebook have been used for coding and making notes of written code. On DataCamp, an account has been used to learn the programming language ”R”, and some extra courses on other subjects. Python has been considered, but learning it has been postponed until after the internship.

The essence of (Big) Data Analytics has been studied, and research has been done on the functionality of neural networks and their possible applications. The Linux operating system has been studied, and several versions (Mint and Debian environments) have been practiced with.

The EcoGenie house’s heating and energy equipment comprise air-source heat pumps, micro CHP’s, boilers,hot water storage, Photo Voltaic cells, solar thermal and batteries. These have been studied for developing integrated energy balances. The raw data set consists of a great deal of missing values and different names for the same measuring equipment due to various changes overt he years. Data Cleaning techniques have been studied. The raw data used were written as comma-delimited files (csv files), and importing techniques have been refined for general and specific purposes.

The second 2 months mainly consisted of first identifying and formulating the objective and problems. Importing and preliminary cleaning techniques were the first parts being applied of the studied programming language. The files were examined with a list of technical names and their dimensions analyzed before manipulating the data in usable forms. The actual total cleaning time for the entire data set of 1950 csv files with each 1440 observations for every 350 variables took about 4 months, overlapping other phases of learning the R programming language, and studying neural networks and deep (and machine-) learning algorithms.

Data Exploration has been done throughout the entire internship, and there has been a focus on just 5 variables (6 in the last month). In order to correlate ambient conditions with energy demand patterns, weather and climate data from the Royal Netherlands Meteorological Institute (KNMI) have been explored and applied during the last month, for training the machine learning algorithm. This internship report has been written during the last few weeks.

An objective has been to develop a Reliable Energy Forecast Model with a granularity of 15-minute averages and there has been a focus on training a neural network, with the objective to derive an energy forecast model based on limited input data. The (in)accuracy of the measuring equipment has been taken into account, and a model has been developed to visualize the accuracy of the measurements. Thermodynamic effects have been visualized to create a clear view of the individual benefits of each solution.

The Machine Learning Model (using R) is an interesting tool in providing predictions which may prove to provide a reliable energy forecast. It is however, very difficult to understand, and the development of the neural network model used during this internship shows the many hurdles this process causes.

This internship has been very insightful in understanding the provided EcoGenie data and in many areas it has also provided insight in subjects for future research. Due to time constraints, this internship has been limited in learning the R language, understanding, cleaning, transforming, and visualizing the data, whilst preparing this data for preliminary machine learning models.

Contents

1

Introduction

The objectives of the internship were to learn and apply ”big data” analysis techniques, and to explore and model relevant energy data from a multi-years, detailed single household measurement campaign, the EcoGenie house. The primary objectives were learning, applying and analyzing measurement data, and the secondary objective was to model (with limited) measurement (input) data with training a neural network to derive an energy forecast model. The Master Thesis by Parab, V. (2016) has proven a valuable reference to describe The EcoGenie Project and the Thermal Network Model analyzed and applied in 2016.

1.1. Different Stages of the Internship

The internship consisted of 22 weeks, starting on February 5th and ending on July 6th. The main objective was to first learn skills (2 months), then apply them (2 months) to analyze the measurement data, and then if there was enough time left (1 month) for the secondary objective; train a neural network to derive a reliable 72-hour energy forecast model with a granularity (accuracy) of 15-minute averages (averages of data over intervals of 15 minutes).

1.1.1. Learning Phase

The first 2 months had a focus on learning the R-language (programming software) and its Shiny package (used to set up servers and convert R code into HTML), while getting to know the content of the data acquired by measuring equipment. There has also been a focus on using the Anaconda Navigator, which R-studio has been used for coding in the R-language and Yupiter Notebook for making notes on code. An account on DataCamp has been used to follow a skill track with many relevant courses in mostly R. Initially Python (programming language) had been considered to be learned, however a focus on only R had been chosen due to time constraints and practicality.

Tools Used

Slack is a messenger application that is used by many programmers, and it has also proven helpful for communication in the EcoGenie project. GitHub is a place where code can be shared, and collectively worked on with special version controls, however this has been gradually understood over the entire internship. Many open source courses, videos on YouTube and searches on Google have proven very helpful. Stack-Overflow is one of the many sites with a lot of examples of simple solutions to many problems.

Further Areas of Study

The essence of (Big) Data Analytics has been studied, and research has been done on functionality of neural networks and their possible applications. There has also been some practice on working with the Linux operating system (Mint and Debian environments). LaTeX, JabRef and ShareLaTeX have been used to write this internship report (JabRef produces bibTex code for the bibliography).

Raw Data

The EcoGenie house has measuring equipment such as air-source heat pumps, micro CH-P's, boilers, hot water storage, Photo Voltaic cells, solar thermal and batteries. These are all important for understanding and developing integrated energy balances. The raw data set consists of a great deal of missing values and different names for the same measuring equipment due to various changes over the years. The raw data used were written as comma-delimited files (csv files).

1.1.2. Analyzing Phase

After the first month of the internship, having learned some programming skill in R, the analyzing phase started. This process continued until the end of the internship and eventually also during the (energy forecast) modelling phase. This process was an iterative feedback loop, consistently changing techniques, writing new code and debugging old code. Many encounters of specific errors, for example due to combined packages of different libraries inside the R programming language, prompted large structural changes in code scripts, modelling designs and also the structure of this internship report.

1.1.3. Modelling Phase

During the last 2 months of the internship, there has been a focus on training a neural network for the energy forecast model and writing this internship report. Initially there were plans to train a neural network with Google Tensor-flow, however it proved to also be possible with the R-language. This had several advantages, due to its simplicity, time constraints and online documentation. A recommendation is to use Python for training a neural network energy forecast model in the future.

1.2. The EcoGenie Project

The EcoGenie Project is a research program to test deep decarbonization strategies for residential energy supply and demand under North Western European climate conditions, carried out by Shell Global Solutions since 2011. In 2016, energy consumed for heating was 4-5 times more than electricity consumption in residential housing. The primary objective of the EcoGenie project is investigating CO_2 influences and cost performances of modern heating appliances (Parab, V., 2016).

Investigating technologies which will allow increased usage of renewable energy is the secondary objective. There is a particular focus on renewable energy gained from PV (photo-voltaic solar cells) and wind. The tertiary objective is studying technologies and business models which integrate houses into a smart grid. This is done by providing useful information and allowing load-shifting through connected energy devices (Parab, V., 2016).

The Shell EcoGenie house is a $200\mathrm{m}^2$ terraced house in a residential area of The Hague, Netherlands and was build in the 1930's. Apart from double glazing of the windows, the house remains unchanged. It is being used as a laboratory and has over 200 sensors measuring data at intervals of 1 minute, sampled by a Siemens SCADA system. Indoor air, radiator in- and out flow temperatures are measured in each room. To close the energy balance of individual heating appliances, energy flux meters are used. To test hybrid heating strategies, there have been several combinations of appliances deployed (Parab, V., 2016).

1.2.1. Hybrid Heating Systems

Hybrid Heating Systems that have been deployed are: Condensing Boiler, Micro-CHP, Ground Source Heat Pump (GSHP), Air Source Heat Pump (ASHP), Hot Water Storage, Energy Storage through Phase Change (wax), Solar Thermal Water Heater, Solar PV Cells, and Electricity Storage through Batteries (Parab, V., 2016).

It has been concluded that (Parab, V., 2016) an ASHP-Condensing Boiler hybrid system leads to a 66% reduction in natural gas demand, which also increases the lifespan of the boiler and contributes to a significant reduction of carbon emissions.

1.2.2. Decarbonization Solutions

Demand-side energy management combined with physical decarbonization solutions are, and will continue to be researched. Aggregated predictions from many houses could be used to effectively regulate and manage energy generation and storage (Parab, V., 2016). Increased effectiveness is essential due to rising renewable energy penetration, being highly intermittent, into the grid.

Thermal Networks

Using thermal networks, a dynamic thermal model had been developed for the Shell EcoGenie house. In order to determine heat transfer properties, a data-driven grey-modelling procedure had been deployed for the EcoGenie house. The heat transfer mechanisms were represented by class linear time-invariant state-space equations (Parab, V., 2016). To estimate the coefficients representing the heat transfer properties, statistical solution algorithms based on the maximum likelihood method had been applied.

Data had been prepared by converting raw data from EcoGenie to hourly averaged solar radiation flux, in-door/out air temperature and heating energy flux (Parab, V., 2016). Instead of point-wise data, hourly average data had been used, since point-wise data fails to capture the pulse-type nature of the heating energy flux.

Thermal Network Model Accuracy

The Thermal Network Model used by Vaidehi Parab (Parab, V., 2016) was validated and tested with a data set of 48 hours, starting February 23rd and ending February 24th, 2015. The objective of training a neural network as a reliable energy forecast model, had as a target to provide a 72 hour forecast with a granularity (accuracy) of 15 minute-averages. This target has not been met during this internship, and a section in this report discusses the issues and suggestions for future study. A new goal has been set for future research, where the accuracy is of 1 hour, due to implications of the KNMI (Koninklijk Nederlands Meteorologisch Instituut, or Royal Dutch Meteorological Institute) data set having an accuracy of 1 hour (affecting machine learning reliability).

Energy Forecast Models using Neural Networks

The *Thermal Network Energy Forecast Model* (Parab, V., 2016) concluded that further study was required on transient heat transfer effects and relevant mechanisms to deliver accurate energy demand forecasts within a 3 percent error margin. Training (with machine learning techniques) a Neural Network on the EcoGenie data, in combination with data from the KNMI, was the next step in developing a reliable energy forecast model.

1.3. Decisions During the Project

There has been a focus on learning and using the programming language R to read, clean and analyze the EcoGenie measurement data. Before any information could be gained from the acquired data, the data needed to be cleaned for analysis. Data preparation is an iterative process, where there are feedback loops for improvements and corrections.

Postponing Python

The use of Python has been postponed until after the internship program via the HHS (The Hague University), since it became apparent that it is possible to use machine learning in R also. Having developed a structure of R code to read, clean and analyze the measurement data, implementing it in R code to model a neural network (machine learning) was very time consuming. The development of alterations of R code, to be used in the Shiny (server) package within R, was also postponed due to the inherent complexity and necessity to have a working version of the machine learning R code.

Importing csv Files

The measurement data that has been used for analysis, was imported from csv (comma-delimited) files. Every fully intact csv file contained 1440 minutes of observations (measurements). Some did not contain the full amount or were erroneous. These had to be cleaned or removed during the importing process. From November 21st 2012 to September 30th 2016, there have been 9 phases of additions to the number of variables (measurement equipment data as columns). Therefor there are 10 different sizes (number of columns versus the ideal 1440 observations) of csv files. The number of variables range from 292 to 379.

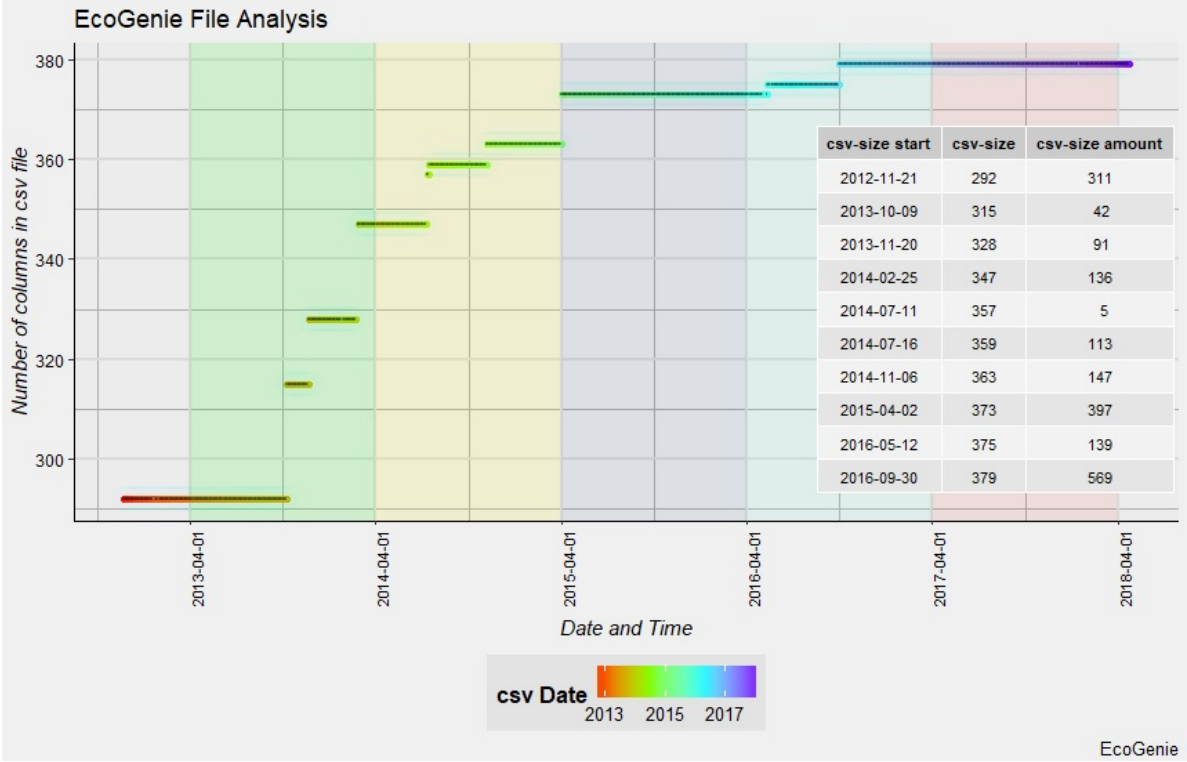
Joining Variables

To successfully join variables from csv files with a different number of variables, the specific variables in question had to be carefully examined, before being combined into the complete data set being used for analysis. The first variable selected, which had some overlapping with differently sized csv files, was the heat going to the radiator. Since its data was from different measurement equipment,

measuring separately, the data points could be added up. Taking into account the non-existent (NA, a.k.a not-available) values, missing data points were replaced by approximated values.

Data Import Size

In figure 1.1, the size of csv files is represented on the y-axis, time is represented by date and time on the x-axis, and the number of csv files with a certain size is displayed inside the table inside the figure. From the first to last csv file, the colors range from red to purple, as displayed in the legend. Note that an energy year starts on April 1st and ends on March 31st, represented by the green, yellow, blue, aquamarine and red rectangles in the background.



csv-size start	csv-size	csv-size amount
2012-11-21	292	311
2013-10-09	315	42
2013-11-20	328	91
2014-02-25	347	136
2014-07-11	357	5
2014-07-16	359	113
2014-11-06	363	147
2015-04-02	373	397
2016-05-12	375	139
2016-09-30	379	569

Figure 1.1: *Analysis of the EcoGenie csv files. On the x-axis is the time and date, while the y-axis is represented by the number of columns (variables, measurement types), the table inside the figure displays the amount of each type (size) of csv file, with its starting date. The time of the csv files range from red to purple, as seen in the legend. Each csv file ideally contains 1440 observations, which is the amount of minutes in one day.*

Import Errors

The way the data was imported changed in various ways during the iterative feedback processes. In order to ignore empty csv files and erroneous ones, a line of code had completely removed them in the process of importing. This caused a problem where the original list of files was of a different length (amount of files), than the new one (after importing), which caused errors concerning naming the imported files inside the list and prompting the necessity to check every list inside the list of lists (nested lists) for empty lists, which had been created in that process. It was at first unclear that this was an unnecessary process.

Revisiting the code for importing the csv files, it became apparent that replacing the erroneous and/or empty csv files with NULL (non-existing) instead of NA values, caused the specific error function to remove the file from the list completely (changing its size). When this was changed to only produce NA values for empty or erroneous file readings, not only was it possible to give the files read into the list names, but it was also easier to remove the NA values and it was not necessary to check inside nested lists (because the nested lists of NULL values weren't even created in this way). This made reading the imported files faster and more transparent.

Selecting Variables before Joining into a Data Frame

After importing the data, certain variables are chosen before the cleaning process begins. This is because R-Studio, the program in which the R code is run, has difficulties combining the data from 1950 csv files with over 300 variables that each have 1440 observations. Because the primary focus has been on mainly 5 variables, those variables were selected before joining the data from the csv files into one data frame. Since each csv file should have 1440 observations, 1950 csv files will result in about 3 million observations for every variable. A non-leap year has 525600 minutes (there is an observation each minute) and a leap year has 527040 minutes, which shows the current data set spans about 6 years. As seen in figure 1.1 there are 5 full energy years available.

Visualize Data for Exploration of Integrity

Before the data set is cleaned, it is called raw data. To understand the data set, plots are constantly made throughout all processes, prompting quick changes in the code to analyze and clean it. Raw plots are especially important when starting to explore the data, since extreme values almost always influence the plots dramatically. A very extreme y-value (like an erroneous value), could make the entire graph seem like a flat horizontal line on the x-axis. Filtering these values early after importing is very important. They are replaced by NA values, which are then quickly approximated with interpolation.

2

Theoretical Analysis

To properly analyze data, the data sets should be cleaned or already clean (Dasu, 2003). Any decision made during the cleaning process or thereafter, is one that should generally be motivated by some theoretical explanation. To convert data into a format which is easy to manipulate, it is possible to use mathematical techniques and or formulas. It is however, very important to consider physical boundaries of the system at all times. Even one mistake can potentially nullify any use the presented information holds.

There are many ways to clean a data set, and ultimately the choices should be made considering the desired result(s). When preparing a data set with time stamps, it is potentially necessary to join the data set onto a specified time frame. This may cause the creation of missing values (represented by NA values) which are unusable in calculations. It is therefore essential to replace these missing values. A reliable way to replace missing values is using linear spline interpolation.

2.1. Energy Consumption

To heat a house and use electrical appliances, a house needs to receive energy. Energy can be acquired in different forms and the most common ones at the moment (in The Netherlands) are electricity and natural gas. There is a shift towards using more electricity and less natural gas (Dieperink *et al.*, 2004), which is an effort to reduce global carbon emissions. Renewable Energy production providing electricity further decreases the carbon footprint (Boyle, G., 2004). Due to its inconsistent output, an energy forecast model may provide assistance in predicting where energy will be needed, to further decrease energy waste and decrease the carbon footprint.

2.1.1. Heat Sources

Heat Sources in houses require 4-5 times the amount of energy than electricity consumption (Parab, V., 2016). It is therefor important to be able to predict heat requirements in order to provide energy which is intermittent in nature. There are several different heat sources applicable in houses. These are for example natural gas heating systems, heat pumps, heat from (micro) co-generation, solar collectors and passive solar heating (Hermans, 2011). There has mainly been a focus on the natural gas heating system and heat pumps.

2.2. Physics

The data analyzed consists of measurements. These measurements should be constrained by the laws of physics, although errors may arise and alter actual measurements. It is essential to filter these extreme values, and to do so it is necessary to have sufficient knowledge on the physics related to the measurement data.

It is relevant to model the physical system before selecting, cleaning and analyzing variables in the data set. The first law of Thermodynamics states that there is a conservation of energy (Hirosawa and Wirth, 2009), meaning that if the temperature (and amount of energy) stays constant inside the EcoGenie house, the energy going into the house must be equal to the energy leaving the house.

The EcoGenie house receives natural gas and electric power. The **Gas Flow** $\dot{m}_{gas-flow}$ is measured in $\left(\frac{\mathrm{m}^3}{\text{minute}}\right)$ and the **Electric Power** $P_{electric}$ in (kW) with intervals of 1 minute. In order to combine and analyze data, it is necessary to evaluate the dimensions of the variables. Gas Flow and Electric Power can both be converted to calculate their *Energy* over an amount of *Time*.

2.2.1. Electric Power and Electric Energy

Electric Power is a unit *Electric Energy* per unit of *Time*. Since the measured **Electric Power** $P_{electric}$ or *Electricity Consumption* values are measured in (kW) with intervals of 1 minute, adding 60 consecutive minutes of values results in equation 2.1.

$$E_{electric-energy,1-hour} = \sum_{i=1}^{60} P_{electric-power,i} t_i \tag{2.1}$$

In equation 2.1, the Electric Energy $E_{electric-energy,1-hour}$ is in kWh.
The Electric Power $P_{electric-power,i}$ is in (kW), and the Time Stamp t_i is in intervals of minutes.

Joule to kWh Conversion

Electric Energy (Wolfson, R., 2014a, Volume 2) $E_{electric}$ and *Gas Flow Energy* $E_{gas-flow}$ can both be expressed in (MJ or in (kWh), however they have to first be converted, since these expressions are not equal per unit of measurement. Since Watt (W) can be expressed as $\left(\frac{\mathrm{J}}{\mathrm{s}}\right)$, Electric Energy can also be expressed as $\left(\frac{\mathrm{kJh}}{\mathrm{s}}\right)$. Since there are 3600 seconds in 1 hour, and k is equal to 1000, the equation 2.2 can be expressed as:

$$1 \bullet \mathrm{kWh} = \left(\mathrm{k}\frac{\mathrm{J}}{\mathrm{s}}\mathrm{h}\right) = \left(\frac{1000}{1} \bullet \frac{3600\mathrm{s}}{\mathrm{s}} \bullet \mathrm{J}\right) = 3.6 \bullet \mathrm{MJ} \tag{2.2}$$

This conversion is used when calculating the Gas Flow Energy.

2.2.2. Heat Capacity

The Heat Q in (J) transported to an object with its resulting Temperature Difference $\mathrm{d}T$ are equal to the *Heat Capacity* of an object, as seen in equation 2.3 (Wolfson, R., 2014b, Volume 1).

$$Q = C\mathrm{d}T \tag{2.3}$$

Where C is the Heat Capacity in $\left(\frac{\mathrm{J}}{\mathrm{K}}\right)$ and the Temperature Difference $\mathrm{d}T$ in (°C).

2.2.3. Specific Heat

The *Specific Heat* of a material is the heat capacity that depends on its type of (inner) structure and its mass. The Specific Heat c in $\left(\frac{\mathrm{J}}{\mathrm{mK}}\right)$ characterizes the Heat Capacity C of a material per unit of mass (Wolfson, R., 2014b, Volume 1), as seen in equation 2.4.

$$Q = mc\mathrm{d}T \tag{2.4}$$

Where the Heat Q is in (J), the Mass m is in (kg) and the Temperature Difference $\mathrm{d}T$ is in (°C).

2.2.4. Gas Flow Energy

To calculate the Gas Flow Energy, the measured Gas Flow in cubic meters needs to be multiplied with its calorific value and converted from Joule to kWh with equation 2.2. The calorific value used is the Lower Calorific Value (LCV), meaning the lower Energy value is being assumed, since a higher calorific value would result in more Energy (per unit of m^3.

The net calorific value of Groningen natural gas is 31.669$\left(\frac{\mathrm{MJ}}{\mathrm{m}^3}\right)$ at a Temperature T of 273.15K (°C), with a Pressure P of 101.325kPa its Density ρ is 0.781$\frac{\mathrm{kg}}{\mathrm{m}^3}$ (Geersen, T. M., 1988). The real Temperature is higher than 0°C, but this value has been assumed for simplicity. The calorific value

chosen is 31.6$\left(\frac{\text{MJ}}{\text{m}^3}\right)$, due to uncertainties produced by assumptions. In equation 2.5 the full calculation can be seen.

$$E_{gas-flow} = CalorificValue \bullet \dot{m}_{gas-flow} = 31.6\left(\frac{\text{MJ}}{\text{m}^3}\right) \bullet \dot{m}_{gas-flow} = (31.6 \bullet 3.6) \bullet kWh \tag{2.5}$$

Where the calculated Gas Flow $E_{gas-flow}$ is in kWh and the $CalorificValue$ is in $\left(\frac{\text{MJ}}{\text{m}^3}\right)$ at a Temperature T of 273.15K (°C), with a Pressure P of 101.325kPa its Density ρ is 0.781$\frac{\text{kg}}{\text{m}^3}$ (Geersen, T. M., 1988). The Gas Flow $\dot{m}_{gas-flow}$ is in $\left(\frac{\text{m}^3}{\text{minute}}\right)$.

2.2.5. Mechanisms of Heat Transfer

Different mechanisms of heat loss are examined in order to minimize its effects. There are 3 different mechanisms for heat transfer (Wolfson, R., 2014b; Boles, M. A. and Cengel, Y. A., 2009; Kreith and Black, 1980) when modelling heat flow in houses. These mechanisms are conduction, convection and radiation.

Conduction is energy transference between particles (more energetic particles to less energetic ones), due to interactions (Wolfson, R., 2014b; Boles, M. A. and Cengel, Y. A., 2009). *Convection* is energy transference between a solid and an adjacent fluid (in motion), which is a combination of conduction and the motion of the adjacent fluid (Wolfson, R., 2014b; Boles, M. A. and Cengel, Y. A., 2009). *Radiation* is energy transference due to electromagnetic wave (photons) emission (Wolfson, R., 2014b; Boles, M. A. and Cengel, Y. A., 2009).

Conduction

Conduction is direct (physical) contact heat transference and occurs as molecules collide with other molecules. Molecules in a hotter region transfer energy to those in an adjacent cooler region (Wolfson, R., 2014b, Volume 1). This process is characterized by the *Thermal Conductivity* k in $\frac{\text{W}}{\text{mK}}$.

Heat Flow H is measured in Watts. 1 Watt is equal to 1 Joule per second $\left(\frac{\text{J}}{\text{s}}\right)$. The heat flow is proportional to the temperature difference, the volume it moves through, and the thermal conductivity k. This relation is displayed in equation 2.6 (Wolfson, R., 2014b, Volume 1).

$$H = -kA\frac{\text{d}T}{\text{d}x} \tag{2.6}$$

Where x is the distance in meters inside the volume with the direction of the heat flow, A the area of the surface perpendicular to x in m^2, and $\text{d}T$ the temperature difference in °C. The negative sign means that the heat flow is in the opposite direction of the increasing temperature along x in the volume.

Conduction is the most important form of heat loss in houses.

Convection

Convection is heat transfer due to the motion of fluids. Movement is influenced by changing densities in the fluids, due to their physical properties of density relating to temperature. Calculating convective heat loss is complicated due to associated fluid motion (Wolfson, R., 2014b, Volume 1). It is approximately proportional to the temperature difference, such as with conduction.

The air heated by a radiator will warm the room (or other parts of the house) due to convection. This air flow is however (usually) limited to inside the house only, since insulation (especially in The Netherlands and similar cold regions) will prevent convection from having any significant effect of total house heat loss.

Radiation

Radiation is the transfer of energy by electromagnetic waves (photons). It has a small effect on heat flow in the house, but this can be considered insignificant. The energy received by radiation is almost always from the sun. This has the potential to capture energy with solar panels, which can be used as electrical power, which in turn can be used to generate heat.

An object that absorbs perfectly is called a *Black-body*, which is black at room temperature. Poor emitters can also be shiny objects, because they reflect radiation (Wolfson, R., 2014b, Volume 1).

A heated body (such as a house or a person) always radiate some energy. Having people inside a house will cause it to heat up slightly due to the people radiating heat. This effect is also insignificant (unless there are a lot of people for a long period of time.

2.3. Thermal-Energy Balance

A house is usually kept at a comfortable temperature when people are at home, and sometimes also when they are away. During the winter (The Netherlands has relatively cold winters), more heat is needed to keep the house at the same comfortable temperature than during other seasons. Gas and Electricity can be used to heat a house. There is a growing trend worldwide, and especially in the Netherlands (Dieperink *et al.*, 2004), to use less gas and switch to using electricity to generate heat.

There is a need to keep the *Thermal-Energy Balance* (Wolfson, R., 2014b, Volume 1) constant inside the house, when requiring a constant (comfortable) temperature. Because of the greater temperature difference between the cold winter (outside) and the desired temperature (inside), that difference needs to be supplied to the house in order to keep its interior temperature constant.

2.3.1. Internal Energy

According to the first law of Thermodynamics, the total energy of an isolated system stays constant. The change in the internal energy $\mathrm{d}U$ of a closed system is therefor always *be equal* to the heat Q *entering* the system minus the work W done *by* the system externally (Kimmenaede, A. J. M., 2010), as seen in equation 2.7.

$$\mathrm{d}U = Q - W \tag{2.7}$$

Where $\mathrm{d}U$ is the internal energy in Joule (J), Q heat in (J), and W work in (J). If a house has to stay the same temperature, then $\mathrm{d}U$ has to equal 0, meaning the heat Q leaving the house must equal the work W needed to be done *on* the system (the system being the house).

2.3.2. Heat Equilibrium

A *Heat Equilibrium* is when the Heat is assumed to stay constant inside the house (system), the Heat Loss H_{loss} in (kWh) is equal to the Heat Provided $H_{provided}$ in (kWh), as seen in equation 2.8.

$$H_{loss} = H_{provided} \tag{2.8}$$

This means that isolating a house, reducing its Heat Loss H_{loss}, results in reducing the Heat Provided $H_{provided}$, meaning less heat is needed to keep the house at the same temperature.

2.3.3. Environmental Effects

Environmental Effects are effects due to the surrounding environment of a house. Factors such as wind speed and wind direction play a role in cooling a house. When there is no wind, only conduction plays a significant role in heat loss. When there is wind however, a combination of conduction and convection cool the house. Conduction stays the most significant part of heat loss, however more wind will increase convective heat loss.

Rain and Cloud Cover also have relevant effects, due to the absence of sunlight (radiation) and rain having a low temperature due to its origin from higher (and cooler) altitudes.

Energy-Year

In The Netherlands there are 4 seasons, of which the winter is the most important one when examining an *Energy-Year* (starting April 1st and ending March 31st). An Energy-year (by definition) has been chosen to have the winter in the middle, so it is easier to see the energy use in a single winter. A normal year starting on January 1st has 2 halves of different winters, meaning the data is incomparable.

2.3.4. Physical Boundaries

Physical Boundaries are an essential part in data analysis, since it sets certain limits and filters non-sensical data. There are restrictions, such as the maximum amount of (a type of) energy able to be supplied to a house.

There are boundaries due to the laws of Physics (and common sense), such as measurements of energy consumption that are higher than the amount of energy of several millions of galaxies nearby, or Ambient Temperatures higher than 4000°C. Whereas those values are not physically impossible, they would indeed imply either the destruction of our planet (and solar system), or the immediate inflammation of the neighborhood. Since these events obviously did not happen, it is safe to assume such values are erroneous.

Some boundaries do shift due to climate change. Whereas an upper limit of 40°C has been chosen, this value may need to be increased due to rising global temperatures.

Electricity Consumption Limits

In The Netherlands, most houses use in between 2500 kWh and 5000 kWh (van Wezel, B., 2015) in 1 year. Assuming a house uses 5000 kWh in 1 (non-leap) year, it uses $\left(\frac{5000}{365 \bullet 24 \bullet 60} = 0.0095\right)$ kWh per minute. This is an average Electric Energy Consumption $E_{electric-energy}$ of 0.0095 kWh per minute (= 0.57 kWh per hour), or a constant average Electric Power Consumption $P_{electric-power}$ of 0.57 kW (= 570 W).

Assuming a high value such as the usage of 20000 kWh in 1 (non-leap) year, the average Electric Energy Consumption $E_{electric-energy}$ is 2.3 kWh per hour.

Gas Flow Limits

Gas Flow is limited by physical constraints of the system providing the (natural) gas.

Ambient Temperature Limits

Ambient Temperature Limits in The Netherlands are rising, with more average summer days being higher than 30°C (Bessembinder, J., 2009). For the data analysis of the EcoGenie Ambient Temperature, an upper limit of 40°C has been chosen, however the Ambient Temperature of the KNMI (Koninklijk Nederlands Meteorologisch Instituut) Data Set is more reliable.

House Temperature Limits

House Temperature Limits can be higher than the Ambient Temperature Limits, since insulation decreases the rate of outward heat flow, designed to keep houses warm (with wasting less energy) during the winter.

Heat to Radiator Limits

The heat going to the radiator is heat coming from an Air Source Heat Pump (ASHP) and is limited by the dimensions of the appliance.

2.4. Uncertainties

Measurements are never precise and any result should be accompanied with an explanatory uncertainty. It is important to realize what uncertainties exist in the measurement techniques and unknown factors. It is useful to make a distinction between incorrect systematic and coincidental measurements (van den Eijnde, P., 2016).

2.4.1. Inaccuracy

Measurement Equipment can differ in accuracy and may cause *Reproducible Errors*. This can be expressed as the Systematic Error, or Bias e_s. *Non-Reproducible Errors* are Coincidental Errors e_i caused by Imprecision, such as different forms of applying the same measuring instrument.

The *True Value* y_t is the sum of the *Measurement Value* y_m, the *Systematic Error* e_s, and the *Imprecision* e_i, as seen in equation 2.9 (van den Eijnde, P., 2016). It is important to remember that each measuring equipment and environment, influence measurements.

$$y_t = y_m + e_s + e_i \tag{2.9}$$

2.4.2. Imprecision

Coincidental Errors are not reproducible and therefor require a statistical approach. With an Expectancy Value $E(e_i) = 0$ and a Standard Deviation σ, a Normal Distribution Function can be used as seen in equation 2.10 (van den Eijnde, P., 2016).

$$\sigma \approx s = \sqrt{\left(\frac{\Sigma(y_m - \overline{y_m})^2}{n-1}\right)} \tag{2.10}$$

$\overline{y_m}$ is the *Average Value of the Measurements* and n is the *Number of Measurements*. If the standard deviation of the coincidental error has a reasonable amount of measurements (n > 15), then 95% of the measurements will be in between $y_m - 2s$ and $y_m + 2s$ as seen in equation 2.11 and 2.12.

$$e_i = \pm 2s \tag{2.11}$$

$$y_m = \overline{y_m} \pm 2s \tag{2.12}$$

If the Systematic Error can be neglected, the Real Measurement Value becomes as seen in equation 2.13 (van den Eijnde, P., 2016).

$$y_t = \overline{y_m} \pm \frac{2s}{\sqrt{n}} \tag{2.13}$$

If assumed that the Coincidental Error stays the same and the Systematic Error is zero, then y_m could be seen as equal to $\overline{y_m}$ with respect to time(van den Eijnde, P., 2016).

2.4.3. Systematic Error

A Systematic Error is reproducible, and when known, can be used to correct measurement data by adjusting this Bias. This error is more difficult to estimate than the Incidental Error, which becomes clear when doing a Number of Measurements. An approximation for the Systematic Error is displayed in equation 2.14 (van den Eijnde, P., 2016).

$$e_s \approx y_t - \overline{y_m} \tag{2.14}$$

This approximation becomes more precise when the $\overline{y_m}$ is defined by a larger Number of Measurements, because then the Coincidental Error decreases. To estimate e_s, one must first know y_t, which is impossible. The Independent Measuring Method is an alternative which is relatively more precise in acquiring the Systematic Error (van den Eijnde, P., 2016).

2.4.4. Rounding Measurements

It is incorrect to copy measurements on displays, or for example always rounding to 2 decimals, without having a specific reason for it. This would suggest an accuracy which is based on nothing. The standard deviation s may be used with a set of rules for rounding (van den Eijnde, P., 2016).

Rule 1:

Define the largest decimal unit a, which is greater than $\frac{s}{2}$ e.g. equation 2.15 (van den Eijnde, P., 2016).

$$s = 0.03 \rightarrow a = 0.01;\ s = 0.01 \rightarrow a = 0.001 \tag{2.15}$$

Rule 2a:

Round to the nearest multiple of the largest decimal unit a. An example is given in equation 2.16 (van den Eijnde, P., 2016).

$$s = 0.03,\ y_m 8.314 \rightarrow a = 0.01 \text{ and } y_m = 8.31 \tag{2.16}$$

Rule 2b:
When the nearest multiple of the largest decimal is a 5, round towards the next higher decimal position with an *even* number. An example is given in equation 2.17 and equation 2.18 (van den Eijnde, P., 2016).

$$s = 0.03,\ y_m 8.315 \rightarrow a = 0.01 \text{ and } y_m = 8.32 \tag{2.17}$$

$$s = 0.03,\ y_m 8.345 \rightarrow a = 0.01 \text{ and } y_m = 8.34 \tag{2.18}$$

Rule 2c:
If more than 1 decimal is removed, round in only 1 step. An example is given in equation 2.19 (van den Eijnde, P., 2016).

$$s = 0.03,\ y_m 8.345 \rightarrow a = 0.01 \text{ and } y_m = 8.3 \tag{2.19}$$

2.4.5. Absolute and Relative Error
An *Absolute Error* is a fixed value such as the ones earlier described in the examples given in this section. A *Relative Error* uses a percentage sign and can be a better indicator of an error, because it is less ambiguous (van den Eijnde, P., 2016).

2.5. Replacing Missing Values
Replacing missing values can be done in various ways. A simple method is to take the average value of a data set, and replace all missing values with this average value. This is however a poorly designed way to clean a data set, since it introduces very large uncertainties in the data analysis. It is also an incorrect way of imagining what those missing values should have been, while also being dependant on only the average of the known values, which may also vary considerably.

A mathematical model that can be used to replace missing values, is linear spline interpolation (De Boor *et al.*, 1978).

2.5.1. Linear Spline Interpolation
Spline interpolation is a mathematical method of numerical analysis using a hybrid function of polynomials (Schumaker, 2015). There are many interpolation techniques that can be used (Press *et al.*, 1992). Approximations of equidistant data can by done by analytic splines of any order (Schoenberg, 1988).

Linear spline interpolation (straight lines, instead of higher-order polynomials) has been used for the approximations of missing data.

2.6. Neural Network
The chosen method for calculating the energy forecast is a neural network. A standard neural network (NN) contains many simple, connected processors called neurons (Schmidhuber, J., 2014). Input neurons are activated through the environment perceived by sensors (Schmidhuber, J., 2014), or measurements, while other neurons are activated through connections from previously activated neuron, and are weighted for influential relevance during back-and-forth calculations.

Machine Learning is a process in which Credit Assignment finds the weights which cause the NN desired behavior (Schmidhuber, J., 2014). Neural Network structure, and neuron connectivity affect computational stages. Each stage transforms the aggregate activation of the neural network (Schmidhuber, J., 2014). Deep Learning is the process in which accurately assigning credit across multiple stages is applied.

The neuralnet package in R has been used for all neural net calculations during this internship for fitting a neural network (Alice, M., 2015).

2.6.1. Numerical Computations, Probability and Information Theory

For Numerical Computations, Probability and Information Theory functions are used in Neural Networks (Goodfellow *et al.*, 2016). A mathematical framework for representing uncertain statements is probability theory, quantifying uncertainty and deriving new uncertain statements (Goodfellow *et al.*, 2016). Probability laws should explain how artificial intelligence (AI) systems should reason while probability and statistics can analyze the behavior theoretically. Information theory enables quantification of the amount of uncertainty in a probability distribution (Goodfellow *et al.*, 2016).

Random Variables and Probability Distributions

Random Variables are variables that can randomly take different values (Goodfellow *et al.*, 2016). They may be discrete or continuous. Discrete random variables have a finite or countably infinite number of states, while continuous random variables are associated with real values (Goodfellow *et al.*, 2016).

Probability Distributions describe the likelihood a random variable or set of random variables take on each of its possible states (Goodfellow *et al.*, 2016). Common Probability Distributions are the Bernoulli Distribution, Multinoulli Distribution, Gaussian Distribution, Exponential and Laplace Distributions, The Dirac Distribution and Empirical Distribution and Mixtures of Distributions (Goodfellow *et al.*, 2016).

Logistic Sigmoid

A function that is used a lot with probability distributions in deep learning models is the logistic sigmoid (Goodfellow *et al.*, 2016), which is the $\sigma(x)$ in equation 2.20 and has a graphical representation in figure 2.1. x is a random (dimensionless) variable, which can be any number from -Infinity to Infinity. The $\sigma(x)$ is dimensionless and ranges from 0 to 1, high values of x result in a $\sigma(x)$ of 1 and high negative values of x result in a $\sigma(x)$ of 0.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.20}$$

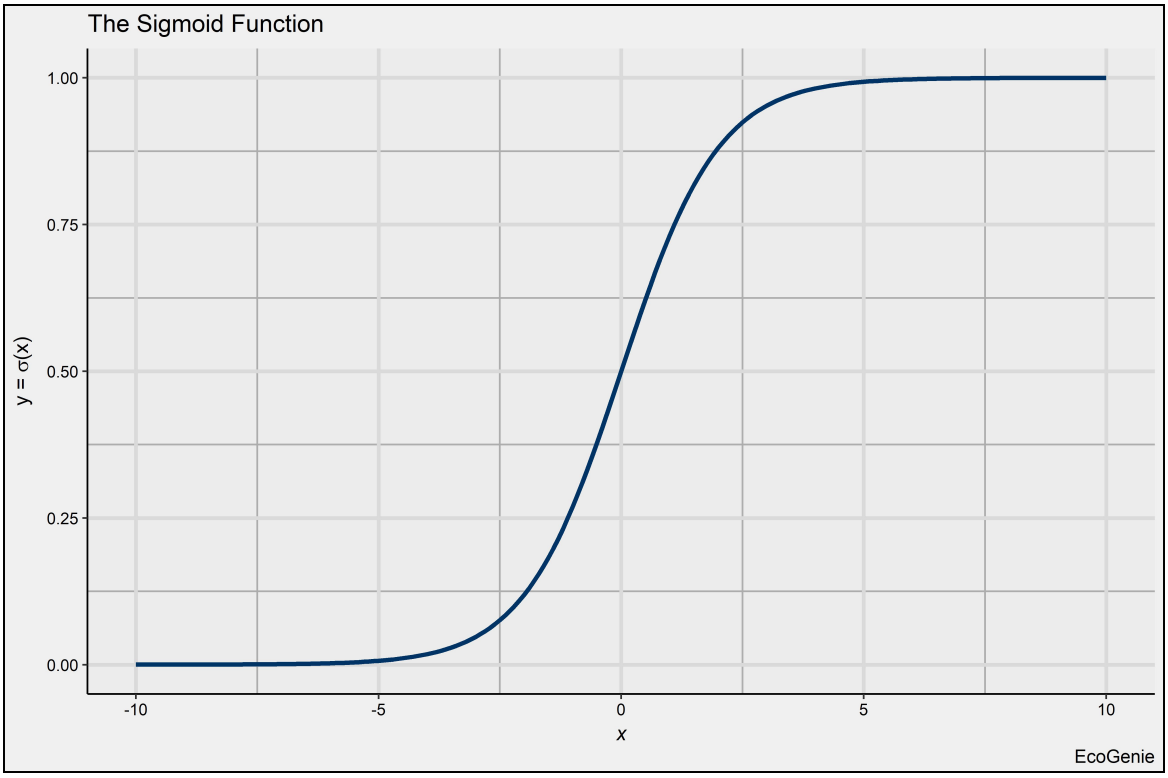


Figure 2.1: *The Logistic Sigmoid with x as a random (dimensionless) variable, which can be any number from -Infinity to Infinity, and the $\sigma(x)$ dimensionless ranging from 0 to 1, with high values of x resulting in a $\sigma(x)$ of 1 and high negative values of x resulting in a $\sigma(x)$ of 0 (Goodfellow et al., 2016). This figure has been rendered in R-Studio using R.*

Since the range of the logistic sigmoid is between 0 and 1, it is commonly used to produce the ϕ parameter of Bernoulli Distributions (Goodfellow *et al.*, 2016). With very positive or negative values, the functions saturates and becomes flat, meaning it is insensitive to small input changes.

Softplus Function

The Softplus Function, as seen in equation 2.21 and in figure 2.2, with $\zeta(x)$ having a range of 0 to Infinity, it is useful for producing the β or σ parameter of a normal distribution (Goodfellow *et al.*, 2016). It is commonly used in combination with sigmoids. The behavior of $\zeta(x)$ is practically linear with high positive values of x. Its behavior is flat with high negative values of x. When x equals zero, $\zeta(x)$ equals log(2), which is slightly positive.

$$\zeta(x) = \log(1 + e^x) \tag{2.21}$$

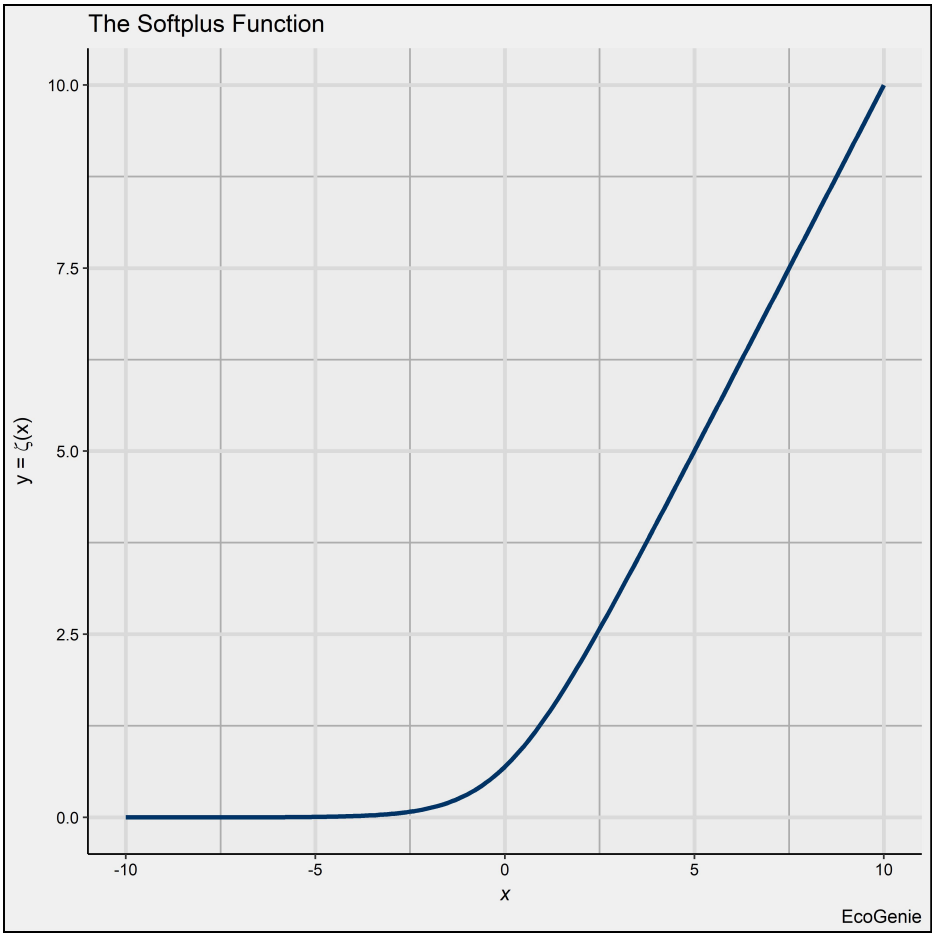


Figure 2.2: *The Softplus Function, with $\zeta(x)$ having a range of 0 to Infinity, it is useful for producing the β or σ parameter of a normal distribution (Goodfellow* et al.*, 2016). This figure has been rendered in RStudio using R.*

Information Theory

The amount of information present in a signal can be quantified. Information Theory assists in designing optimal codes and calculating the expected length of messages samples from specific probability distribution using a variety of encoding schemes (Goodfellow *et al.*, 2016). In machine learning, information theory is applied to continuous variables, where message length interpretations do not always apply. Events that are likely to have low information content, while guaranteed events have no information. Less likely events have higher information content (Goodfellow *et al.*, 2016). Independent events have additive information.

Numerical Computation

Machine Learning Algorithms that solve mathematical problems by methods that update estimates of the solution via an iterative process usually require a high amount of Numerical Computations (Goodfellow *et al.*, 2016). These algorithms differ from analytically deriving a formula to provide a symbolic expression for the correct solution (Goodfellow *et al.*, 2016). Optimization operations, which finds values maximizing and minimizing functions, and solving systems of linear equations are commonly included.

Overflow and Underflow

Infinitely many real numbers are represented by a finite number of bit patterns on digital computers. This causes approximation errors, which is usually a rounding error. Underflow is a form of rounding error which occurs when numbers near zero become zero. Divisions by zero can cause problems, as software environments may have different solutions for it (Goodfellow *et al.*, 2016). Overflow occurs when high positive and negative numbers become Infinite or -Infinite (Goodfellow *et al.*, 2016). A solution to avoiding this problem is the *Softmax Function* (Goodfellow *et al.*, 2016).

Gradient-Based Optimization

A Gradient-Based Optimization scheme is *Gradient Descent*. Gradient Descent is used by finding derivatives to minimize or maximize a function, or to find saddle points. There is a distinction between local and global minimums and maximums. With multiple inputs, partial derivatives are used (Goodfellow *et al.*, 2016). The Directional Derivative is used in the Method of Steepest Descent, or Gradient Descent. It affects the Learning Rate, which is a positive scalar determining the size of the step (Goodfellow *et al.*, 2016).

2.6.2. Calculus Chain Rule in the Context of Networks

The entire (neural) network is a function. A form to describe the calculations from one layer in a neural network, to the next, is displayed in equation 2.22 (3Blue1Brown, 2017). The First Activation Layer, which is the Input Layer a_0^1 (a dimensionless value ranging from 0 to 1), is equal to the Sigmoid Function $\sigma(x)$ (which could also be another special function, but in this case is the sigmoid function, also known as the logistic curve) with a matrix (the chosen notation) inside, which consists of a matrix representing the Weighted Sum $\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,n} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \cdots & w_{k,n} \end{bmatrix}$ used in the vector-matrix product with the vector of the Activation Layer $\begin{bmatrix} a_0^0 \\ a_1^0 \\ \vdots \\ a_n^0 \end{bmatrix}$ plus a vector of the Bias $\begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}$ (3Blue1Brown, 2017). The superscripts are the layer numbers and the subscripts are the neuron numbers inside a layer. The Bias means how high the Weighted Sum needs to be, before being Activated. The Weights, Activation, Biases, and the Sigmoid are all dimensionless values ranging from 0 to 1.

$$a_0^1 = \sigma\left(\begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,n} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{k,0} & w_{k,1} & \cdots & w_{k,n} \end{bmatrix}\begin{bmatrix} a_0^0 \\ a_1^0 \\ \vdots \\ a_n^0 \end{bmatrix} + \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix}\right) \tag{2.22}$$

A *Cost Function* is a measure of how a neural network functioned with respect to its training data and predicted output. It can depend on variables such as weights and biases. A goal can be to understand how sensitive the cost function is to these variables (3Blue1Brown, 2017). Specific adjustments to these variables could efficiently decrease the Cost Function, which can be desirable.

The Neuron Activation a is dimensionless and ranges from 0 to 1. The superscript L in a^L is not an exponent, but indicates the layer in which the activation neuron exists. The Desired Output y is dimensionless and ranges from 0 to 1. The Cost of One Training Example C_0, which is dimensionless, can be described as equation 2.23 (3Blue1Brown, 2017). The Activation a^L is also dimensionless and varies from 0 to 1. All variables (and variations thereof) in the Cost Function are also dimensionless.

$$C_0 = (a^L - y)^2 \tag{2.23}$$

The Weighted Sum z^L, is the Weight w^L times the Previous Activation (neuron) a^{L-1} plus a Bias b^L, as seen in equation 2.24 (3Blue1Brown, 2017).

$$z^L = w^L a^{L-1} + b^L \tag{2.24}$$

The weighted sum is used inside a function such as the sigmoid, to acquire the Activation a^L as seen in equation 2.25 (3Blue1Brown, 2017).

$$a^L = \sigma(z^L) \tag{2.25}$$

Chain Rule

The first goal is to understand how sensitive the cost function is with respect to the weight. This is the derivative of the Cost Function C_0 with respect to the Weight w^L, written as $\frac{\partial C_0}{\partial w^L}$ (3Blue1Brown, 2017). As seen in equation 2.29, there are 3 separate ratios that are examined in order to acquire $\frac{\partial C_0}{\partial w^L}$; the Ratio between the Cost Function and the Activation $\frac{\partial C_0}{\partial a^L}$, the Ratio between the Weighted Sum and the Weight $\frac{\partial z^L}{\partial w^L}$, and the Ratio between the Activation and the Weighted Sum $\frac{\partial a^L}{\partial z^L}$ (3Blue1Brown, 2017). The derivative of the sigmoid function is $\sigma'(z^L)$.

The Ratio between the Cost Function and the Activation $\frac{\partial C_0}{\partial a^L}$ is equal to 2 times the Activation a^L minus the Output y, displayed in equation 2.26.

$$\frac{\partial C_0}{\partial a^L} = 2(a^L - y) \tag{2.26}$$

The Ratio between the Weighted Sum and the Weight $\frac{\partial z^L}{\partial w^L}$ is equal to the previous Activation a^{L-1}, displayed in equation 2.27.

$$\frac{\partial z^L}{\partial w^L} = a^{L-1} \tag{2.27}$$

The Ratio between the Activation and the Weighted Sum $\frac{\partial a^L}{\partial z^L}$ is equal to derivative of the Sigmoid over the Summed Weight $\sigma'(z^L)$, displayed in equation 2.28.

$$\frac{\partial a^L}{\partial z^L} = \sigma'(z^L) \tag{2.28}$$

Equation 2.29 displays the 3 ratios which multiplied equal the Ratio between the specific Cost of one training example and the layered Weight $\frac{\partial C_0}{\partial w^L}$ (3Blue1Brown, 2017).

$$\frac{\partial C_0}{\partial w^L} = \frac{\partial C_0}{\partial a^L}\frac{\partial z^L}{\partial w^L}\frac{\partial a^L}{\partial z^L} = 2(a^L - y)a^{L-1}\sigma'(z^L) \tag{2.29}$$

The full derivative of the cost function is the average of all training examples, as seen in equation 2.30.

$$\frac{\partial C}{\partial w^L} = \frac{1}{n}\sum_{k=0}^{n-1}\frac{\partial C_k}{\partial w^L} \tag{2.30}$$

The full derivative of the cost function (3Blue1Brown, 2017) is one component of the Cost Gradient Vector ∇C, as seen in equation 2.31.

$$\nabla C = \begin{bmatrix} \frac{\partial C}{\partial w^1} \\ \\ \frac{\partial C}{\partial b^1} \\ \\ \vdots \\ \\ \frac{\partial C}{\partial w^L} \\ \\ \frac{\partial C}{\partial b^L} \end{bmatrix} \tag{2.31}$$

The sensitivity of the Cost Function towards the Bias is similar to equation 2.29. Equation 2.32 displays the sensitivity of the Cost Function towards the Bias (3Blue1Brown, 2017).

$$\frac{\partial C_0}{\partial b^L} = \frac{\partial C_0}{\partial a^L} \frac{\partial z^L}{\partial b^L} \frac{\partial a^L}{\partial z^L} = 2(a^L - y)\sigma'(z^L) \tag{2.32}$$

The sensitivity of the Cost Function with respect to the Activation of the previous Layer is $\frac{\partial C_0}{\partial a^{L-1}}$, and is shown in equation 2.33 (3Blue1Brown, 2017). This is where the idea of back-propagation arises. Note that $\frac{\partial C_0}{\partial a^{L-1}} = \frac{\partial C_0}{\partial b^L} w^L$.

$$\frac{\partial C_0}{\partial a^{L-1}} = \frac{\partial C_0}{\partial a^L} \frac{\partial z^L}{\partial a^{L-1}} \frac{\partial a^L}{\partial z^L} = 2(a^L - y) w^L \sigma'(z^L) \tag{2.33}$$

The subscript indicates which neuron it is in that layer. k indicates the layer $L-1$ and j indicates the layer L (3Blue1Brown, 2017). Now the squares of the differences between the Activation a_j^L and the Desired Output y_j are added to gain the Cost C_0 as seen in equation 2.34.

$$C_0 = \sum_{j=0}^{n_{L-1}} (a_j^L - y_j)^2 \tag{2.34}$$

The Weight $w_{j,k}^L$ connects the Previous Activation Neuron a_k^{L-1} with the Activation Neuron a_j^L. The Relative Weighted Sum z_j^L is displayed in equation 2.35 and its general version in equation 2.36, which can be used inside a special function, such as the sigmoid, to calculate the activation (3Blue1Brown, 2017).

$$z_j^L = w_{j,0}^L \; a_0^{L-1} + w_{j,1}^L \; a_1^{L-1} + w_{j,2}^L \; a_2^{L-1} + b_j^L \tag{2.35}$$

$$z_j^L = \cdots + w_{j,k}^L \; a_k^{L-1} + \cdots \tag{2.36}$$

Equations 2.37, 2.38, and 2.39 are chain-rule expressions which give the derivatives of each component of the gradient that minimizes the cost of the network by repeating downward steps (3Blue1Brown, 2017).

$$\frac{\partial C}{\partial a_j^l} = \sum_{j=0}^{n_{l+1}-1} w_{j,k}^{l+1} \sigma'(z_j^{l+1}) \frac{\partial C}{\partial a_j^{l+1}} \tag{2.37}$$

or

$$\frac{\partial C}{\partial a_j^l} = 2(a_j^L - y_j) \tag{2.38}$$

and

$$\frac{\partial C}{\partial w_{j,k}^{l}} = a_k^{l-1}\sigma'(z_j^l)\frac{\partial C}{\partial a_j^l} \tag{2.39}$$

2.7. Influential Relevance

Variables may differ in relevance and must be examined before selection. Variables which are irrelevant in an energy forecast model decrease its reliability and also slow down the calculation process. Multiple co-linearity between explanatory variables can be examined in R (Ghosh, B., 2017) for influential relevance assessment and further variable selection for machine learning.

2.7.1. Selecting Variables for Machine Learning

It is important to select variables for machine learning, because random input variables will give ambiguous results.

EcoGenie Data Set

The House Energy Balance must be taken into account when choosing variables. Choosing variables which do not significantly affect the variable(s) necessary to predict, may result in useless predictions and can also slow down calculation processes.

KNMI Data Set

The imported KNMI Data Set consisted of several variables. These were:

- HH: Time (hour)
- DD: Average Wind Direction of the last 10 minutes of the previous hour (°)
- FH: Hour Average Wind Speed ($0.1\frac{\mathrm{m}}{\mathrm{s}}$)
- FF: Average Wind Speed of the last 10 minutes of the previous hour ($0.1\frac{\mathrm{m}}{\mathrm{s}}$)
- FX: Greatest Gust of Wind of the previous hour ($0.1\frac{\mathrm{m}}{\mathrm{s}}$)
- T: Temperature on 1.5m height (0.1°C)
- $T10N$: Minimum Temperature on 10cm height of the previous 6 hours (0.1°C)
- TD: Dew Temperature on 1.5 height (0.1°C)
- SQ: Duration of Sunshine per hourly period, calculated from overall radiation (in 0.1 hour, with -1 for < 0.05 hour)
- Q: Overall Radiation per hourly period ($0.1\frac{\mathrm{J}}{\mathrm{cm}^2}$)
- DR: Duration of Rain per hourly period (in 0.1 hour)
- RH: Hour Sum of Rain (in 0.1mm, with -1 for < 0.05mm)
- VV: Horizontal Sight during Observation (0 = less than 100m, 1=100-200m, 2=200-300m,..., 49=4900-5000m, 50=5-6km, 56=6-7km, 57=7-8km, ..., 79=29-30km, 80=30-35km, 81=35-40km,..., 89 = more than 70km)
- N: Cloud Cover (degree of coverage of the upper air in eighths), during the observation (9 = upper air is invisible)
- U: Relative Humidity (in percent) on 1.50m height during observation

The variables were analyzed for NA values, and those without were selected for machine learning to keep the size of the data (time) frame intact. Some values were converted to coincide with the dimensions of the variables within the entire data frame, to prevent unintended offsets.

3

Data Preparation

Data can be acquired in a number of ways. It can be measured by equipment or entered by hand as numerical values or character strings. The data used was measured by equipment and saved into daily csv files, with each minute being an observation or measurement.

There are 1440 minutes in a day, so this accounts for 1440 observations or measurements for each variable (unique measurement type) available. As seen in figure 1.1, the number of variables has changed several times. The amount of variables ranges from 292 to 379, with a total 1950 (non-empty) csv files and 1440 observations each, this accounts for a total of 997 million observations, which is a little less than a billion observations.

Importing a billion time-stamps may cause computational problems, so there has been a focus on primarily 5 variables for analysis during most of the internship. Selecting 5 variables results in selecting an ideal $1,4040,000$ observations, however not all 1962 csv files are complete.

In a non-leap year, there are 525600 minutes and in a leap year, there are 527040 minutes. 2013, 2014, 2015, 2017 and 2018 are non-leap years, whereas 2016 is a leap year. An energy year starts April 1st and ends March 31st. The leap day of 2016 is in the energy-year of 2015-2016. When 4 non-leap energy-years and 1 leap energy-year are selected, there is a total of $2,629,440$ minutes, or observations per variable.

3.1. Identify and Formulate Problem

The measuring equipment in the EcoGenie house produce data and a large amount of that data is stored in comma-delimited excel (csv) files. Before any analysis of the data can be done, the data types must be examined. An excel file with all the names, technical names and special data types is provided. From this list it is possible to find measuring equipment's technical names, in order to be able to select them in the comma-delimited files.

The EcoGenie Project has had a few changes in the names and technical names of the measuring equipment over the years. The list of available equipment grew, and some old names stayed the same, however one important one did change a few times over the years. The name for the time stamp of the measurements changed a few times, which is problematic for the data analysis. To be able to properly analyze the data, the names should be the same as to be able to join the different data sets into one big one.

Analyzing data integrity is a vital part of the preparation, or cleaning, of the entire set. An unprepared data set is useless for acquiring the desired information.

3.2. Data Selection

There are 1950 csv files with almost each containing (the ideal) 1440 observations (or minutes per day) per variable are imported using a specially written R function that at first makes a list of all the 1950 non-empty data frames (each csv file is read as a data frame into a list of data frames). This list of 1950 (elements) of data frames has a size 7.7GB. It holds around 997 million data points, which is almost a billion. Before this imported list of data frames can be joined into 1 data frame, a selection of some variables has to be made (considering current computational limits).

While analyzing data, it is best to focus on 1 variable first, before introducing new variables into the data frame for a more complex analysis. Since there are hundreds of variables, it is unwise to join it all into 1 data frame without having complete knowledge of every variable joined in question. Many variables may be dependant on other variables. Joining data which is unnecessary for a certain evaluation or analysis, will unnecessarily increase computation time, while possibly introducing erroneous behavior and decrease overall information transparency.

3.2.1. Data Import

A function written to rename technical names to unambiguous names and a function written to select several variables (5 variables during most of the internship), both manipulate and select the names of the data frames in the list of data frames. The list of data frames is then joined into 1 data frame.

3.2.2. Exploring Imported Data

The 5 variables chosen during most of the internship were the *Electricity* (Consumption), *Gas Flow*, *House Temperature*, *Ambient Temperature*, and *Heat to Radiator*. While exploring these variables with calculations and graphical representations, it became clear that the Heat to Radiator variable was incomplete. The variable had been discontinued and a new variable was required to be joined onto this one. New equipment and measurements had been logged into a variable renamed to *Heat to Radiator 2*. The old Heat to Radiator was renamed to *Heat to Radiator 1*. To successfully join these data sets, the individual variables had to be analyzed first.

Overlapping (in time) observations had to be separated and summed for each time-stamp. The now three separated data sets could be joined together into one, with its 2 variables being the Time-Stamp and Heat to Radiator. While exploring the Heat to Radiator data, the power P in (kW) is summed over each hour to gain the energy E in (kWh). In 3.1, Heat to Radiator (part) 1 is displayed in red, whereas Heat to Radiator (part) 2 is displayed in blue. There are 172 minutes overlapping Heat to Radiator part 1 and 2, which were separately summed as a temporary part 3, before having all parts combine into 1 data set.

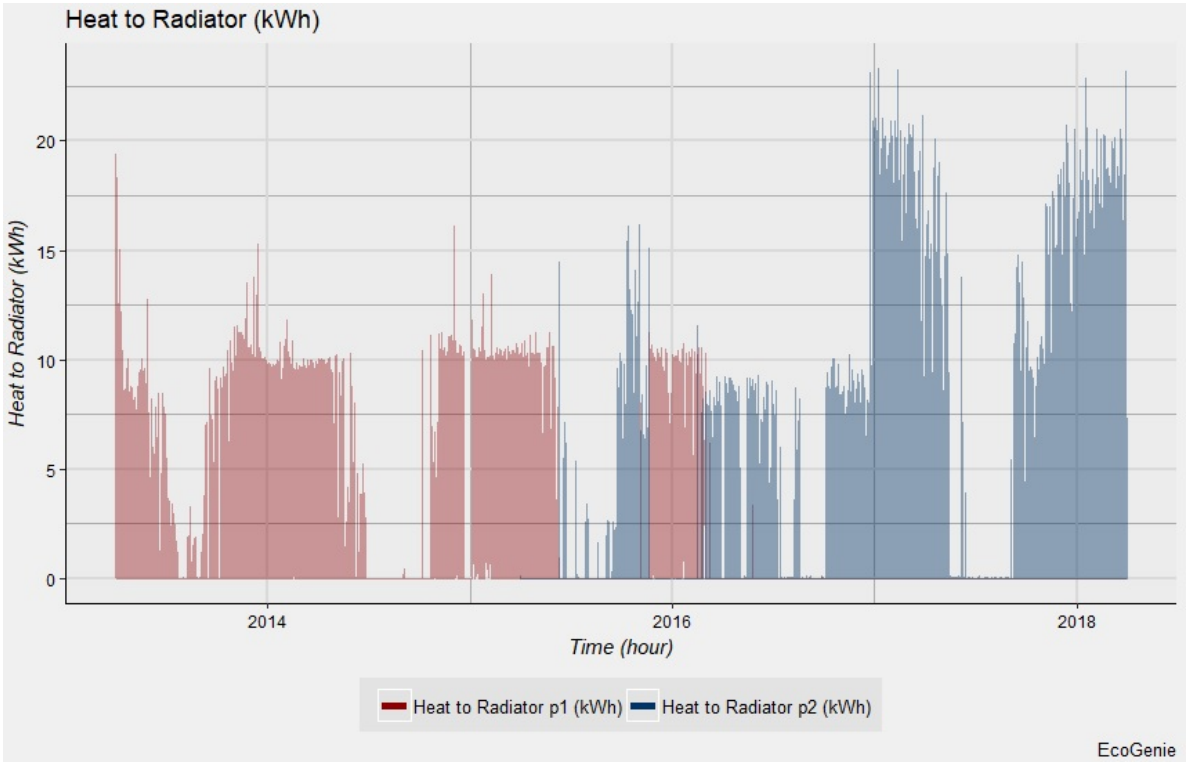


Figure 3.1: *Heat to Radiator (part) 1 is red and Heat to Radiator (part) 2 is blue. There is less than 3 hours overlap with respect to the 2 parts. The overlap had to be separately summed, before being joined with part 1 and 2 into the entire data set for the variable Heat to Radiator.*

3.3. Data Cleaning

Cleaning data is an iterative process. The first few steps of data cleaning are immediately after importing the data. The entire process of understanding, cleaning and then producing graphs of these selected data, took about 3 months.

3.3.1. Time-Stamp

The *Time-Stamp*, *iteration-stamp*, *time stamp* or *observation*, which lowest value is 1 minute in time *t* is arguably the most important variable in the data set. The first part of the cleaning process, is to convert the time stamp into a format that can be easily manipulated with R code.

The time stamp (Time-Stamp) needs to be converted to a generic date-time style, required by the software being used. This is necessary because the software may otherwise not understand the time stamps. Once the time stamp has been converted to the required generic style, it is possible to join the data set onto a generated complete time scale. This gives the possibility to identify missing data, because missing time stamps will generate NA (not available) values due to the joining of the data set to the generated time scale.

If a generic time frame is not built, the graphical function will ignore these time stamps, and simply join them together as if it were a complete data frame. It is essential to create a generic time frame, for the specified analysis (in this case there are 5 energy-years, starting in April 1st 2013 and ending in March 31st 2018). This is because summaries consisting of summed values, averages and standard deviations, will be incorrect, since missing time stamps are ignored if the data has not been joined onto a full time frame (to produce the NA values, which are later approximated to complete the data set).

3.3.2. Renaming Variables

The second part of cleaning the data is renaming variables from their technical names into easily readable names. This happens immediately after formatting the time stamp, which is immediately after importing the data. It is also before the list of data frames is joined into one. There are exceptions, where a combination of measuring equipment produce different values for a common complete variable. The renaming of these variables into one joined variable happens after several other steps of cleaning the data. The Heat to Radiator variable as shown in figure 3.1 is an example of 2 variables being joined into 1 during the iterative cleaning process.

3.3.3. Selecting Variables

When analyzing data and producing graphs, there is usually only a focus on 1, 2 or on rare occasions, several variables. This is to be able to quickly find extreme values and erroneous occurrences.

3.3.4. Data Integrity

In unprepared data, it is almost certain that there will be a great amount of missing (NA, or not-available) values in the data set. This may be due to varying reasons, like an error in the electrical circuit which provides the data from the measurement equipment. Unprepared data might also appear to not have any NA values, while after joining the data set with a full time frame it may show to have a considerable amount of NA values, which could potentially influence certain calculations.

It is possible to apply desirable calculations at undesired locations within the code structure. Figure 3.2 is an example of plotting incorrectly applied calculations. One may assume such a plot holds some kind of valuable information, however after careful analysis it became apparent that an essential part of data cleaning was misplaced.

Due to the gas flow being measured as a current summed total flow into the house (since a certain starting date of meter implementation), a calculation had to be made to acquire the actual flow per minute. This conversion was placed before filtering extreme values or assessing missing data. This caused the calculations to include extreme values and even erroneous values, such as sudden zero values in between the total summed flow, resulting in the curiously shaped figure 3.2.

In figure 3.2, the x-axis is time in minutes and the y-axis is the Gas Flow in cubic meters. This Gas Flow is physically impossible in this system. For comparison: 5000 cubic meters of gas is about the size of 5 tennis courts filled with African elephants. Since this is time in minutes, it would mean the house is regularly receiving and sending these extreme amounts of gas each minute. If data is not critically examined and dimensional analysis is not performed, erroneous implementations of code

could be overlooked, resulting in misinformation and useless data.

The problem that had occurred in this misplaced calculation had been masked for over a month, because it was filtered out during cleaning processes which should have partially been done before the process of calculating any rate of change per minute. Having returned to the start of the implemented coding structure to analyze and debug issues, this issue was solved by creating an extra cleaning step before the specific calculation.

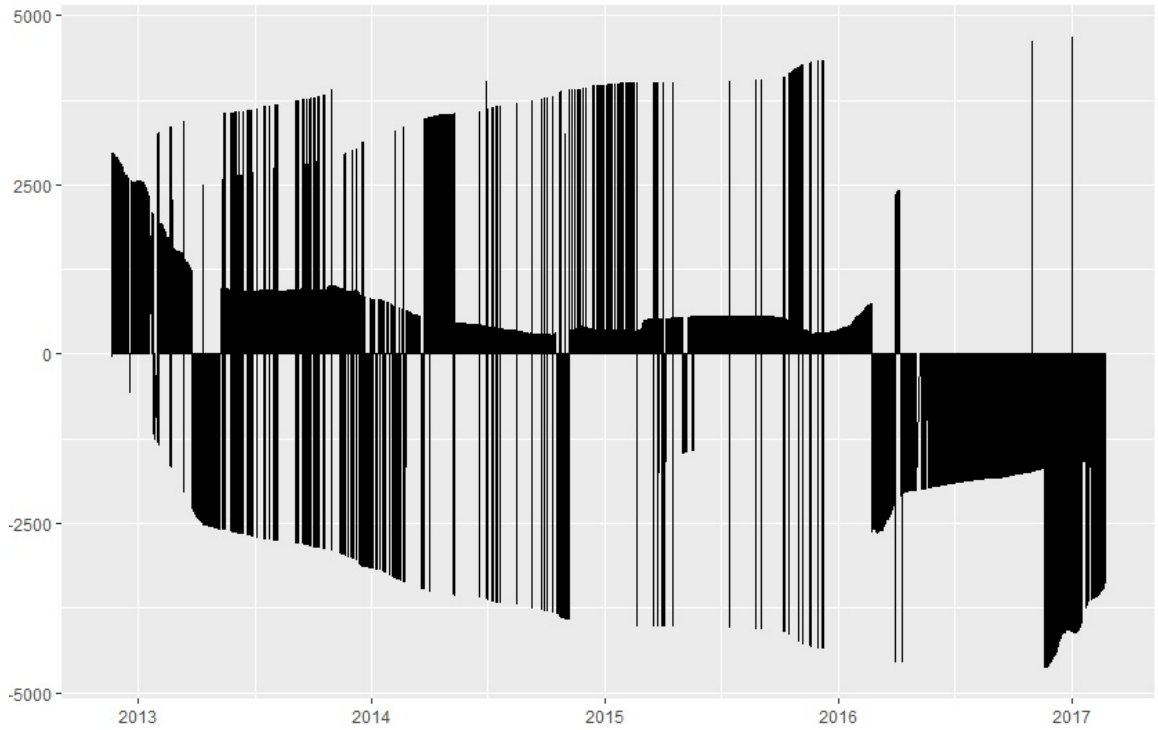


Figure 3.2: *An example of incorrectly applied desirable calculations resulting in misinformation. The Gas Flow is displayed here on the y-axis (supposedly in cubic meters), whereas the x-axis represents time in minutes. Note the extreme positive and negative values, both being physically impossible for a normal house system to receive. Negative values indicate giving back gas, which would mean very large amounts of gas are being sent back and forth. This graph (or plot) is an excellent example on how it may seem an informative figure has been produced, however it only shows a calculation performed on systematic errors produced by other incorrectly placed code.*

After solving this problem, the amount of ”missing” data was reduced. This missing data became missing because of incorrectly placed calculations, causing new extreme values which were filtered out before producing certain graphs.

3.3.5. Extreme Values

Extreme values are values that are possibly produced by erroneous behavior of measuring equipment. There are also real ”extreme” values in data, but one must always consider physical constraints of a system. In that sense, certain boundaries have been chosen to filter out any values that are outside of these limits.

In figure 3.3 the x-axis is time in minute and the y-axis the Heat to Radiator in kWh. Note the lack of boundaries causing all the data to flatten towards zero, because of an extreme value. This extreme value translates to the amount of energy produced by a million galaxies (Tsao and Waide, 2010) (assuming each galaxy has the amount of stars in the milky way, with each star similar to our sun), for a billion years.

It can be assumed that this amount of energy will most likely not ever be produced by humans, let alone be supplied to the EcoGenie house. If this amount of energy was supplied to the EcoGenie house in one minutes, the EcoGenie house would transform into a black hole slowly consuming possibly thousands of galaxies nearby.

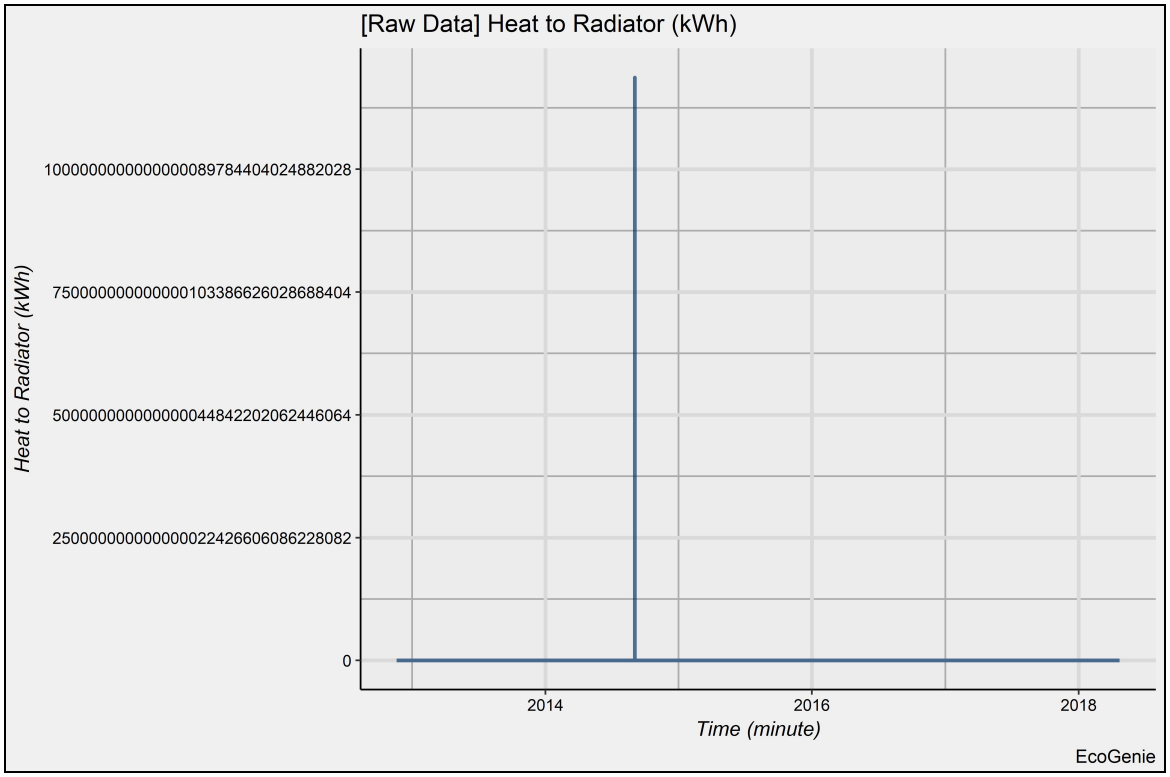


Figure 3.3: *Heat to Radiator (only first variable, or technical name selected, which is part 1 of the total Heat to Radiator later used), is on the y-axis in kWh, whereas time is in minutes on the x-axis. Note the extreme values displayed on the y-axis, which reach values in the* 10^{33} *kWh, which is the amount of energy produced by a million galaxies (Tsao and Waide, 2010) (assuming each galaxy has the amount of stars in the milky way, with each star similar to our sun), for a billion years. It can be assumed that this amount of energy cannot by supplied per minute, to or by the EcoGenie house, or even will be supplied by humans ever.*

3.3.6. Missing Data

When joining data onto a specified time frame with units such as minutes, hours, days, weeks, months or years, only existing data will occupy their time stamps, and any time stamps without data will be ”missing data” (usually NA values) in these data variables (other than the time stamp variable itself).

Data can be accidentally filtered by incorrectly placed code, producing more missing data. As seen in figure 3.2, misplaced code can produce new extreme values which are then filtered by the boundaries set in place to avoid problems such as in figure 3.3.

This over-filtering process had taken place during the first half of the internship, producing a graph like figure 3.4. On the y-axis in this figure, there is the Electricity Consumption (in blue points on the graph) on the left in kWh and the Ambient Temperature (in red points on the graph) in °C on the right. The x-axis is the time in minutes. The yellow vertical lines indicate time sections with missing data. Note that the precision of the electricity measuring equipment is limited to two decimals, and therefor is equally spaced vertically on the graph.

The horizontal red line in figure 3.4 represents the replacement value used for missing ambient temperature data. The horizontal blue line represents the replacement value used for missing electricity data. These values were based on an overall average value, and were a placeholder before more accurate replacements were created.

The use of replacing missing values with a single averaged value has been changed to using linear spline interpolation.

3.4. Data Exploration

Data Exploration is also an iterative process. During this process, it may become apparent that new variables should be selected during importing the data or that the time-stamp is incomplete, duplicated or has an off-set of a number of seconds (whereas the data is per minute, causing problems with

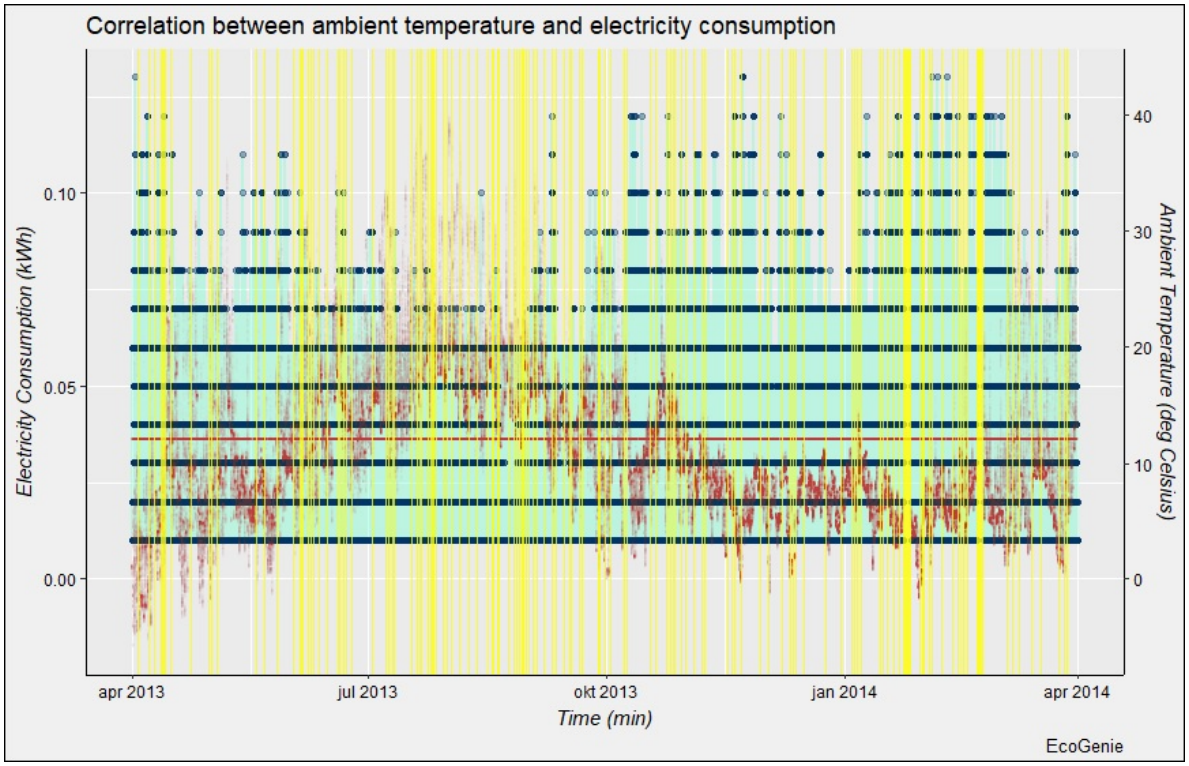


Figure 3.4: *Correlation between the Ambient Temperature and Electricity Consumption, with yellow vertical lines indicating time sections with missing data, either by actual missing data or unintentionally caused missing data. On the y-axis there is the Electricity Consumption in kWh on the left and the Ambient Temperature in °C on the right. Electricity on the graph is represented by blue dots, whereas the temperature is represented by red dots. The light greenish hue are lines connecting the blue dots representing the Electricity Consumption. Time is in minutes on the x-axis. Note that the precision of the electricity measuring equipment is limited to two decimals, and thus is equally spaced vertically. There is a horizontal red line representing a replacement value for missing ambient temperature data and a horizontal blue line representing a replacement value for missing electricity data. These values were based on an overall average value.*

joining the data). During the data exploration, missing data could be found or accidentally created due to some incorrect function of code, or incorrect implementation of code. Data exploration happens in many stages, or phases, and is also a process prompting recurrences of previous stages.

This feedback loop of program development, to analyze the available data, is a different process than the linear structure of explanation in this report. Whereas a report has a well-defined linear structure, explaining as if everything happened linearly, the feedback loop of developing code is more like a fractal. There can be many surprises, and much development happens with trail and error.

3.4.1. Estimating Errors

To estimate the error of the measurements, a combination of equations can be used for several variables. The 5 variables that have been studied intensively are *Electricity*, *Gas Flow*, *House Temperature*, *Ambient Temperature* and *Heat to Radiator*.

Since the measurements are on variables which change in time, using equation 2.10 can result in very high Coincidental Errors. Values such as Gas Flow, House Temperature, Ambient Temperature, and Heat to Radiator may vary during the seasons, with winter-summer differences being the greatest.

Coincidental Error Evaluation

Coincidental (Imprecision) Errors result in standard deviations (taken from April 1st 2013 to March 31st 2018), such as $\sigma_{ambient-temperature} = 7.04$°C for the Ambient Temperature. House Temperatures do not differ as much during the year and its standard deviation is $\sigma_{house-temperature} = 2.36$°C, which is not as extreme however it is still very large in comparison to its Measurement Accuracy. The standard deviation of the Heat to Radiator is $\sigma_{heat-to-radiator} = 0.07$kWh, and the Gas Flow is $\sigma_{gas-flow} = 0.06$kWh, which are both also high in comparison to the Measurement Accuracy, since these Observations are defined for intervals of 1 Minute.

Electricity Consumption changes during night and day, however does not change much throughout the year (and its seasons). The standard deviation of the Electricity Consumption $\sigma_{electricity} = 0.02$kWh and has been estimated to be lower than its Measurement Accuracy.

Measurement Accuracy Evaluation

Not all details were known about the specifications of the measuring instruments. The Accuracy of the Electricity Consumption has been defined as $a_{electricity} = 0.01$kW·min per minute, which can be converted to $a_{electricity} = 0.6$kWh per minute. Gas Flow Accuracy has been estimated to be $a_{gas-flow} = 0.0001$kWh, and the Heat to Radiator Accuracy (having had altering measuring equipment incidents) has been estimated to be $a_{heat-to-radiator} = 0.00001$kWh. The actual accuracy of these measurements are unknown, and this could be a subject for future study.

General Accuracy Estimation

In appendix A, table A.1 displays the amount of observations in a given time frame. These values had been used in calculation to scale the error, however the amount of unknowns were too great. In order to use a conservative error margin, all tables show their values with only 2 significant digits and their errors with 1 significant digit chosen to be an Estimated Relative Error of 5%.

3.4.2. Electricity Consumption

A slow and steady increase in electricity consumption has been observed. During the winter there is more electricity being used, which reflects the darkness of the longer nights and the use of house lighting. A baseline of electricity consumption is observed, which represents the usage of appliances such as washing machines and coffee machines. In the winter of 2015-2016, The EcoGenie Project had turned off the electricity, gas flow and heat to radiator, to see how long it takes for a house to lose all the heat (e.g. to estimate the heat capacity of a house). The table for the electricity consumption summary (table A.3) is in appendix A.

In figure 3.5 the *Electricity Consumption* is in kWh on the y-axis. $Time$ is in hour on the x-axis. Five energy-years are displayed, starting April 1st 2013 and ending March 31st 2018. Dark blue lines are data per hour, dark red points are the mean day averages, and light green lines are the mean week averages. Very sharp notches in winter times represent winter holidays where there has been no electricity consumption during that time in the EcoGenie house. During the summer (holidays), there are also periods of very low electricity usage.

In figure B.1, the *Electricity Consumption* is in kWh on the left y-axis, and the *Average Electricity per Month* is in kWh on the right y-axis. $Time$ is in hour on the x-axis. Five energy-years are displayed in 5 colors (displaying the average value of that year), starting April 1st 2013 and ending March 31st 2018. The Energy-Year 2013-2014 is green, the Energy-Year 2014-2015 is yellow, the Energy-Year 2015-2016 is dark blue, the Energy-Year 2016-2017 is aquamarine, and the Energy-Year 2017-2018 is red. The *Average Electricity per Month* is a dark red line in the graph.

3.4.3. Gas Flow

The Gas Flow was highest during the Energy-Year 2013-2014, which was before many changes in the EcoGenie house, such as improving insulation. The Gas Flow during the Energy-Years 2014-2015 and 2015-2016 was half the Gas Flow of the first Energy-Year (2013-2014), which has been the most important improvement in reducing the cost of energy usage and preventing to waste energy. The Energy-Years 2016-2017 and 2017-2018 see an increase in Gas Flow, which correlates with an increase in Average House Temperature, as seen in figure B.4. The table for the gas flow summary (table A.4) is in appendix A.

In figure 3.6 the *Gas Flow* is in kWh on the y-axis. $Time$ is in hour on the x-axis. Five energy-years are displayed, starting April 1st 2013 and ending March 31st 2018. Dark blue lines are data per hour, dark red points are the mean day averages, and light green lines are the mean week averages.

Note that in figure 3.6 the hour-data consists of many spiky data points, proving the importance of averaging these values into meaningful time frames (such as days and weeks), to visualize the data in order to make sense of it.

In figure B.2, the *Gas Flow* is in kWh on the left y-axis, and the *Average Gas Flow per Month* is in kWh on the right y-axis. $Time$ is in hour on the x-axis. Five energy-years are displayed in 5 colors (displaying the average value of that year), starting April 1st 2013 and ending March 31st 2018. The

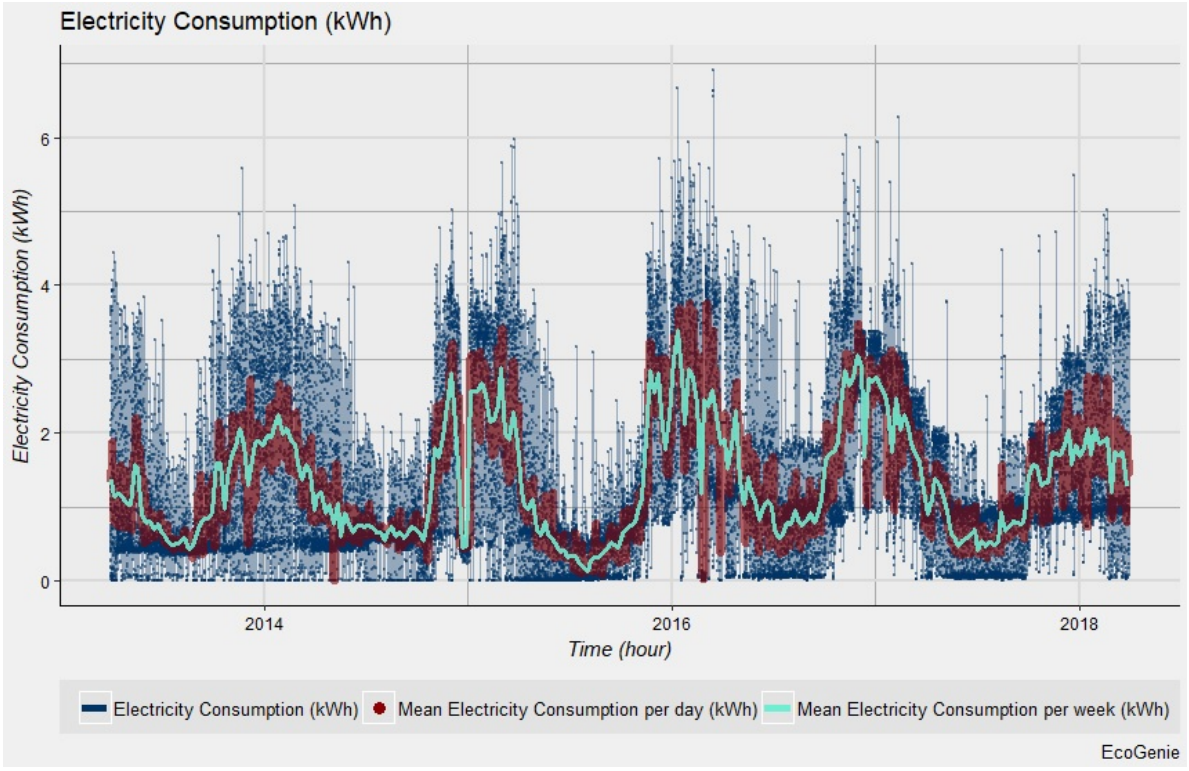


Figure 3.5: *Electricity Consumption is in kWh on the y-axis.* $Time$ *is in hour on the x-axis. Five energy-years are displayed, starting April 1st 2013 and ending March 31st 2018. Dark blue lines are data per hour, dark red points are the mean day averages, and light green lines are the mean week averages. Sharp notches during winter times indicate the absence of people and electricity usage, due to winter holidays. A decrease in electricity consumption can also be seen during the summer holidays.*

Energy-Year 2013-2014 is green, the Energy-Year 2014-2015 is yellow, the Energy-Year 2015-2016 is dark blue, the Energy-Year 2016-2017 is aquamarine, and the Energy-Year 2017-2018 is red. The *Average Gas Flow per Month* is a dark red line in the graph.

3.4.4. House Temperature

A range of 0°C to 40°C has been chosen for the house temperature, however in figure B.3 it appears that the house temperature may have been below zero around march. There are however similar spikes in the hour data, being much lower than the day-average values. This could be an area for future study if one wishes to have more precise knowledge on the (EcoGenie) house temperature.

In figure B.3 the *House Temperature* is in °C on the y-axis. $Time$ is in hour on the x-axis. Five energy-years are displayed, starting April 1st 2013 and ending March 31st 2018. Dark blue lines are data per hour, dark red points are the mean day averages, and light green lines are the mean week averages.

In figure B.4, the *Average House Temperature* is in °C on the y-axis. $Time$ is in hour on the x-axis. Five energy-years are displayed in 5 colors (displaying the average value of that year), starting April 1st 2013 and ending March 31st 2018. The Energy-Year 2013-2014 is green, the Energy-Year 2014-2015 is yellow, the Energy-Year 2015-2016 is dark blue, the Energy-Year 2016-2017 is aquamarine, and the Energy-Year 2017-2018 is red. The *Average House Temperature per Month* is a dark red line in the graph.

3.4.5. Ambient Temperature

A range of -10°C to 40°C has been chosen for the ambient temperature. The figures of the ambient temperatures have been placed in appendix B. In figure B.5 the *Ambient Temperature* is in °C on the y-axis. $Time$ is in hour on the x-axis. Five energy-years are displayed, starting April 1st 2013 and ending March 31st 2018. Dark blue lines are data per hour, dark red points are the mean day averages, and light green lines are the mean week averages. Note how change in ambient temperatures is cyclic over

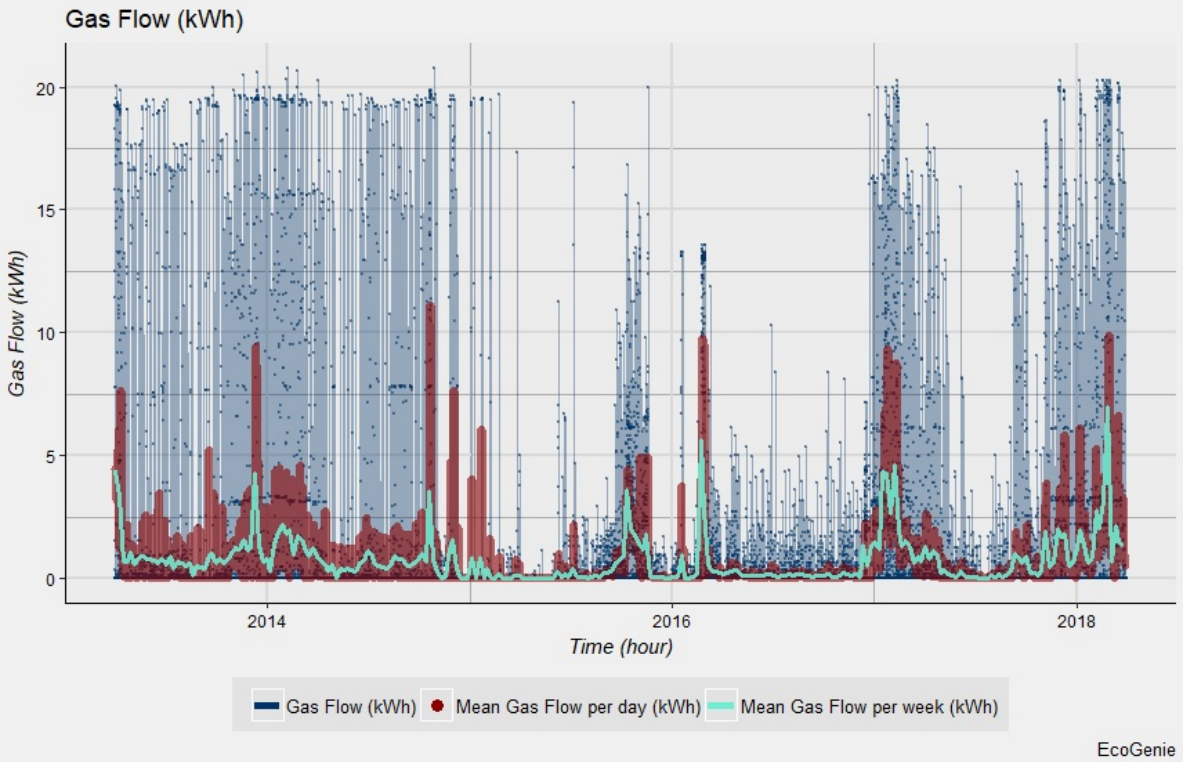


Figure 3.6: *Gas Flow is in kWh on the y-axis. $Time$ is in hour on the x-axis. Five energy-years are displayed, starting April 1st 2013 and ending March 31st 2018. Dark blue lines are data per hour, dark red points are the mean day averages, and light green lines are the mean week averages.*

time.

In figure B.6, the *Average Ambient Temperature* is in °C on the y-axis. $Time$ is in hour on the x-axis. Five energy-years are displayed in 5 colors (displaying the average value of that year), starting April 1st 2013 and ending March 31st 2018. The Energy-Year 2013-2014 is green, the Energy-Year 2014-2015 is yellow, the Energy-Year 2015-2016 is dark blue, the Energy-Year 2016-2017 is aquamarine, and the Energy-Year 2017-2018 is red. The *Average Ambient Temperature per Month* is a dark red line in the graph.

3.4.6. Heat to Radiator

The Heat to Radiator data frame has been joined from 2 different (heat to radiator) variables out of the EcoGenie import data. *Heat to Radiator* is in kWh on the y-axis. $Time$ is in hour on the x-axis. Five energy-years are displayed, starting April 1st 2013 and ending March 31st 2018. Dark blue lines are data per hour, dark red points are the mean day averages, and light green lines are the mean week averages. Note how in the last 2 Energy-Years, there are spikes of hour-data up to 20kWh, while the Heat to Radiator only provides up to 10kWh. This is because of maintenance-related aspects of the measurements and proves the usefulness of taking day- and week-averages.

In figure 3.8, the *Heat to Radiator* is in kWh on the left y-axis, and the *Average Heat to Radiator per Month* is in kWh on the right y-axis. $Time$ is in hour on the x-axis. Five energy-years are displayed in 5 colors (displaying the average value of that year), starting April 1st 2013 and ending March 31st 2018. The Energy-Year 2013-2014 is green, the Energy-Year 2014-2015 is yellow, the Energy-Year 2015-2016 is dark blue, the Energy-Year 2016-2017 is aquamarine, and the Energy-Year 2017-2018 is red. The *Average Heat to Radiator per Month* is a dark red line in the graph.

A steady decrease in Heat to Radiator usage can be seen over the 5-year measuring period, as seen in figure 3.8.

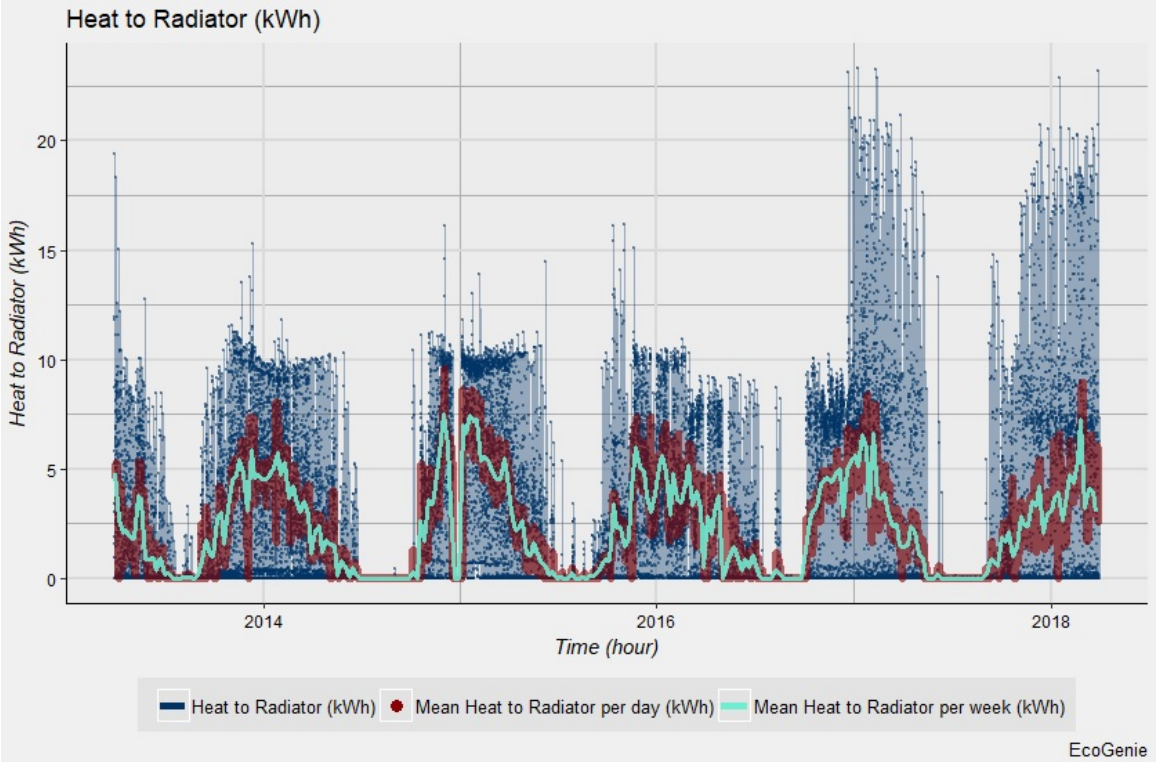


Figure 3.7: *Heat to Radiator is in kWh on the y-axis. Time is in hour on the x-axis. Five energy-years are displayed, starting April 1st 2013 and ending March 31st 2018. Dark blue lines are data per hour, dark red points are the mean day averages, and light green lines are the mean week averages.*

3.5. The Winter of 2015-2016

In the winter of 2015-2016, the electricity, gas flow and radiators have been turned off, to see how fast the EcoGenie house loses its heat (e.g. to find its heat-capacity). Figures B.7, B.8, B.9, B.10, B.11 display these effects.

In figure B.7 the *Electricity Consumption* is in kWh on the y-axis, and the *Time* (hour) is on the x-axis. Note the amount of time it takes for the electricity consumption to become zero.

In figure B.8 the *Gas Flow* is in kWh on the y-axis, and the *Time* (hour) is on the x-axis. Note how the gas flow becomes zero rapidly.

In figure B.9 the *House Temperature* is in °C on the y-axis, and the *Time* (hour) is on the x-axis. Note the time it takes for the house temperature to become zero, and how much time it takes for the house to heat up again.

In figure B.10 the *Ambient Temperature* is in °C on the y-axis, and the *Time* (hour) is on the x-axis. Note the correlation the ambient temperature has with the house temperature, seen in figure B.9.

In figure B.11 the *Heat to Radiator* is in kWh on the y-axis, and the *Time* (hour) is on the x-axis. Note the curve going towards zero once turned off, and the fairly linear line moving from zero once turned on.

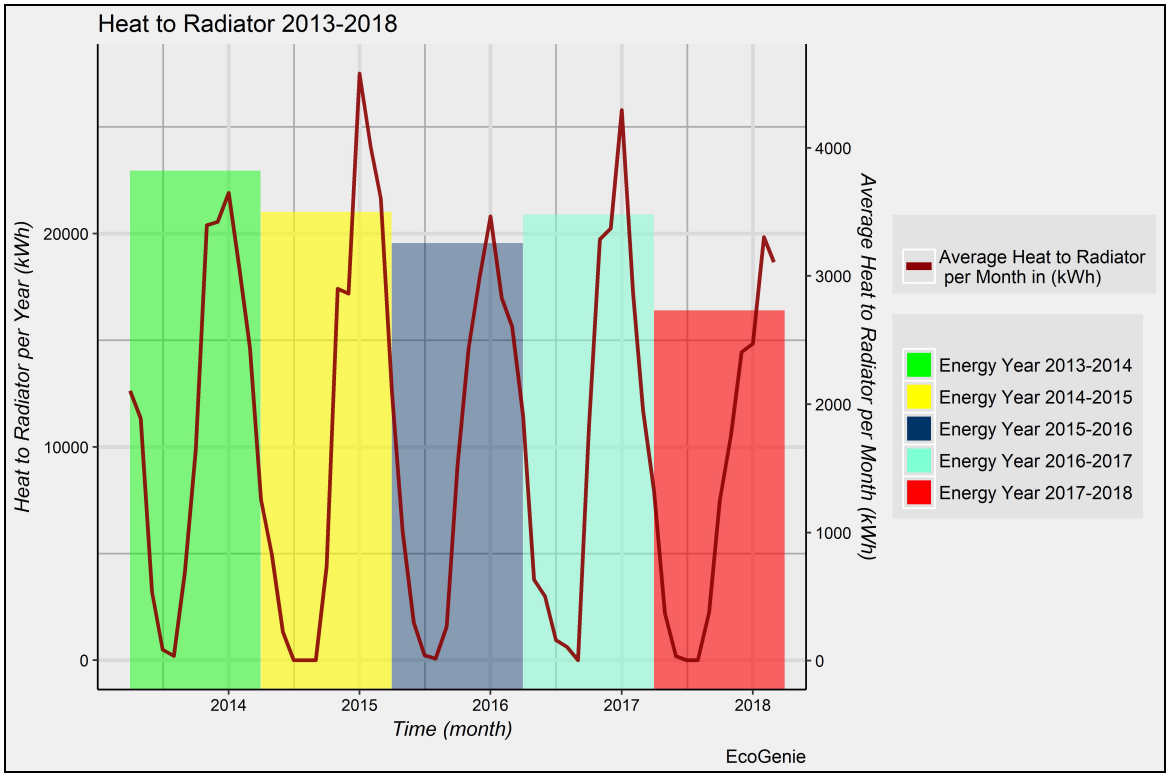


Figure 3.8: *The Heat to Radiator is in kWh on the left y-axis, and the Average Heat to Radiator per Month is in kWh on the right y-axis.* $Time$ *is in hour on the x-axis. Five energy-years are displayed in 5 colors (displaying the average value of that year), starting April 1st 2013 and ending March 31st 2018. The Energy-Year 2013-2014 is green, the Energy-Year 2014-2015 is yellow, the Energy-Year 2015-2016 is dark blue, the Energy-Year 2016-2017 is aquamarine, and the Energy-Year 2017-2018 is red. The Average Heat to Radiator per Month is a dark red line in the graph.*

4

Energy Forecast Model

An objective has been to develop a *Reliable Energy Forecast Model* with a granularity of 15-minute averages and there has been a focus on training a neural network, with the objective to derive an energy forecast model based on limited input data. The (in)accuracy of the measuring equipment has been taken into account, and a model has been developed to visualize the accuracy of the measurements. Thermodynamic effects are visualized to create a clear view of the individual benefits of each solution.

4.1. Model Development

The *Development of the Model* is an iterative process, meaning that each model building stage is an improvement on the previous one. The energy forecast model has not been finalized during this internship, but there have been numerous developments. There are suggestions and recommendations on future development of the model, which are also discussed in this report.

An example of an applicable feedback loop for model development is figure 4.1 (P. David, 2013). This feedback loop first starts with identifying and formulating the problem. It then continues with data preparation and data exploration, followed by transforming and selecting data. The model is built (applied) after selecting data. Validation of the model test, provides insight in model integrity. The model is deployed (applied) after validation and is closely monitored and evaluated.

This feedback process is purposefully vague, as details may vary in the applied context of the model. Every step (iteration) may reset to its starting point, or have similar feedback processes within its iteration. Validation, Deployment and Evaluation are also all a part of Building the model. Transforming and Selecting data also may be done before the new Data Preparation. As such, The example in figure 4.1 (P. David, 2013) may be interpreted in many forms.

4.1.1. Identifying and Formulating the Problem

Identifying and Formulating the problem is a useful component in understanding objectives and is usually the first step of each loop. It is basically the start of the modelling process, since it can also be seen as *Formulating the Objective*, such as describing the model itself.

Problems may go unnoticed for a period of time, which indicates the importance of actually identifying the problem first. Certain code may seem useful and prove useful for an amount of time, however during development it may become a problem once combined with new code structures, prompting the necessity of identifying this issue, formulating it, and address it.

4.1.2. Preparing the Data

Preparing the Data is a very substantial process, which in this internship has taken about 80% of the time. The largest part of Data Preparation is Data Cleaning, which has been discussed in chapter 3. Unprepared data is useless, since no effort has gone into understanding or analyzing it. Without preparing data, it is just one big unknown, with its only real value being random nonsensical data. Even poorly prepared data, while still being prepared, may contain just a few values which will break the entire analysis, such as energy measurement values which are greater than the energy of several galaxies. Another very important analysis is the *Dimensional Analysis*.

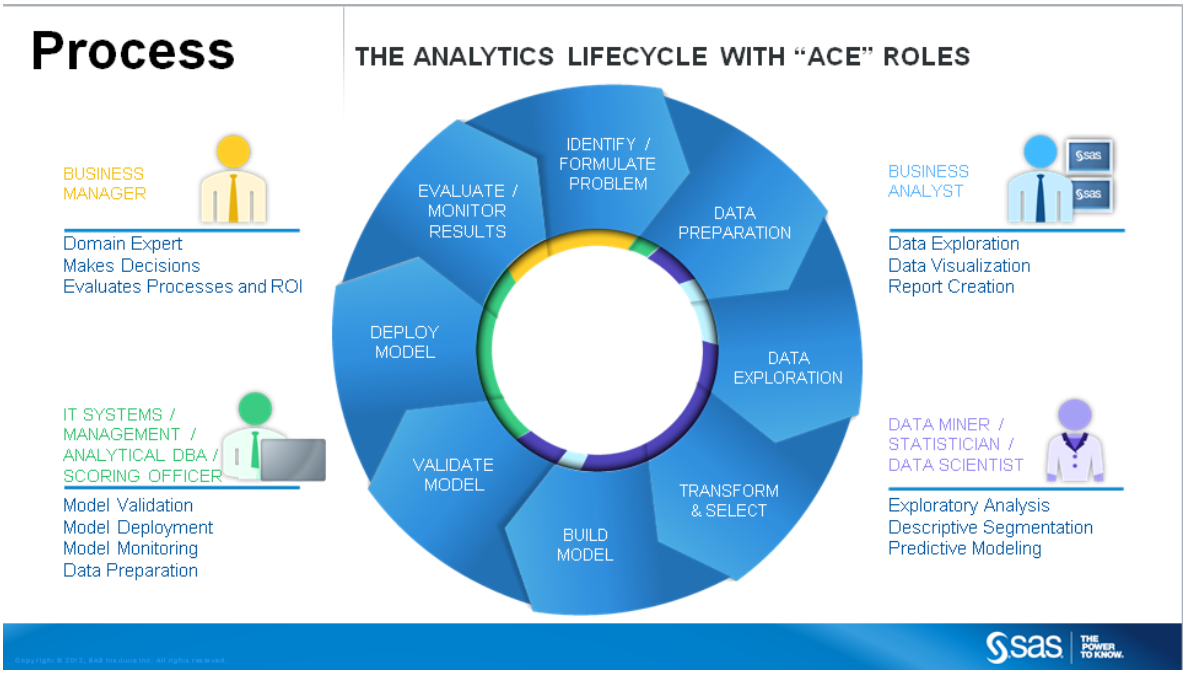


Figure 4.1: *An example of an applicable feedback loop for the development of a model. It starts with identifying and formulating the problem and ends when the model is completed, having taken any number of loops (P. David, 2013).*

If units are not correctly converted, the data is useless. Any mistake, however tiny it may seem, is disastrous. An example is the conversion of metric to imperial units. NASA once lost a $125 million Mars orbiter because a Lockheed Martin engineering team used imperial units, while the agency's team used the metric system (Lloyd, R., 1999).

4.1.3. Transforming and Selecting Data

Transforming and Selecting data is a part of the Data Preparation and precedes the actual buildup of the model. Data Selection is done in various steps during the pre-modelling process, such as during data importing and the machine learning algorithms. Transforms such as using different time frames and iteration steps, sums and averages, standard deviations and similar summaries, are useful in providing clearer data visuals and understanding the data, while providing insight into the next steps in the "analytic life-cycle".

4.1.4. Building the Model

Building the Model is essentially the process of the entire feedback loop, however is also the part just before the implementation of the machine learning algorithms in the neural network. Figure 4.2 is an example of a neural network. It consists of 4 layers, the left one being the input layer, the right one the output layer, and the 2 middle ones the hidden (black box) layers. A neural network (as explained in chapter 2, section 2.6) usually consists of an input, output, and hidden layers. Several variables have been chosen here as input variables; the Wind Direction (n, n-1, n-2), the Heat to Radiator (n-1, n-2), and the Temperature (n, n-1, n-2). The output variable is the Heat to Radiator (n), being the desired energy prediction.

4.1.5. Validating the Model

Validating the Model provides a decision-making-process which in turn is evaluated. A large neural network, such as figure B.12, may provide more precise predictions, however also slow down calculation processes. This figure has many variables with several previous iterations. This model, while being previously validated, is now outdated and did not prove to provide useful predictions. It did however visualize during this process which changes were necessary, and are subject to future study and development.

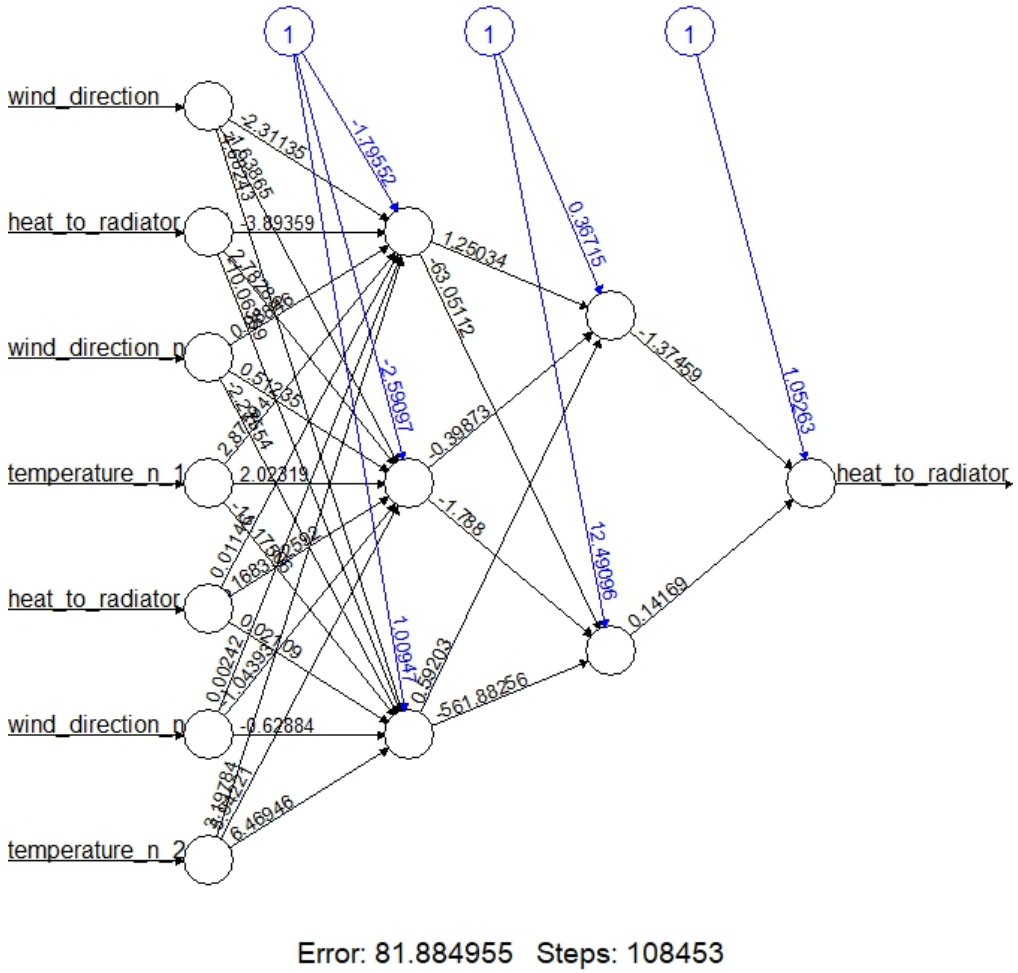


Figure 4.2: *This neural network consists of 4 layers, the left one being the input layer, the right one the output layer, and the 2 middle ones the hidden (black box) layers. A neural network (as explained in chapter 2, section 2.6) usually consists of an input, output, and hidden layers. Several variables have been chosen here as input variables; the Wind Direction (n, n-1, n-2), the Heat to Radiator (n-1, n-2), and the Temperature (n, n-1, n-2). The output variable is the Heat to Radiator (n), being the desired energy prediction.*

4.1.6. Deploying the Model

Deploying the Model initiates a testing phase, or is the actual finalized product. An example of a deployed model, as seen in figure B.13, displays the use of many inputs. This neural network contains 3 hidden layers and its many inputs were provided to improve the previously deployed model. This test however proved that an extra hidden layer, without truly understanding its effects, caused the algorithm to converge to nonsensical outputs as seen in figure 4.3.

Figure 4.3 consists of 2 (sub-) figures; one with the Real Values vs those Predicted by the Neural Network and one with the Real Values vs those Predicted by the Linear Model. The x-axis of both sub-figures are the (Heat to Radiator) Real Values in kWh, while the y-axis is the Heat to Radiator Predicted by the Neural Network (kWh) in the left sub-figure and the y-axis is the Heat to Radiator Predicted by the Linear Model (kWh) in the right sub-figure. Note that the left figure is flat, due to an extreme value produced by the unnecessary extra hidden layer. The linear model shows a vertical distribution near the Heat to Radiator Real Value of 0kWh, which indicates an effect caused by the linear approximation on certain values. Extra (control) outputs, such as a zero-value output, could increase effectiveness of the models.

4.1.7. Evaluating and Monitoring Results of the Model

Evaluating and Monitoring Results of the Model assist possible and necessary insights into the model and its provided (prepared) data. Choices in visualization techniques are relevant for providing information to those involved in the project. Figure 4.4 combines both predictions of the Neural Network & the Linear Model into one figure, to provide a clear comparison of the two methods. The x-axis is the Heat to Radiator Real Value (kWh) and the y-axis is the prediction in kWh of both the Neural Network and the Linear Model, with the Linear Model being blue and the Neural Network being red. Note the vertical, horizontal, and other forms of clustering (of the real values vs the predicted ones); indications of necessary changes to be made to the model. Ideally the real vs predicted values should cluster around the intersecting line(s), with the intersecting line ideally being exactly linear in the form $y = x$.

Only several of figures have been included in this internship report (and the appendices), however countless figures and calculations have been made, each being evaluated, before returning to the process of identifying and formulating the problem(s).

Figure B.14 has both predictions of the Neural Network & the Linear Model in one figure, providing a clear comparison of the two methods. The x-axis is the Heat to Radiator Real Value (kWh) and the y-axis is the prediction in kWh of both the Neural Network and the Linear Model, with the Linear Model being blue and the Neural Network being red. Note large empty space in the left, left-bottom, and bottom area, caused by only 1 extreme value being -20kWh. This figure has been designed to have the x-axis and the y-axis to have the same length, to clearly visualize the offset the intersection lines have compared to the ideal $y = x$ form.

4.2. Machine Learning Model using R

The Machine Learning Model (using R) is an interesting tool in providing predictions which may prove to provide a reliable energy forecast. It is however, very difficult to understand, and the development of the neural network model used during this internship shows the many hurdles this process causes. The hidden layers sometimes seem a mystery, however (sometimes logical, depending on the application,) mathematical equations are the cause of the inputs providing an output, or multiple outputs. There is an ever-growing amount of literature able to provide more insight into the subject of deep-learning, or able to confuse someone into a complete brain-freeze. In R, different algorithms, different packages, different methods of implementing them, all provide solutions, problems, insight, and confusion, depending on the iteration of the modelling process and on (seemingly) random events.

Several articles, YouTube videos or books, such as Schmidhuber, J. (2014); Alice, M. (2015); Goodfellow *et al.* (2016); 3Blue1Brown (2017); Ghosh, B. (2017); Berel, D. (2018), all assist into the introduction of machine learning models, neural networks, deep learning, and so on; however one must always take care and not just assume all is quickly understood. Truly understanding Machine Learning Models, probably takes years of study, with all the bumpy mistakes necessary to give insight into the curious aspects of the approach of artificially simulating a neural network, loosely based on the (human) brain.

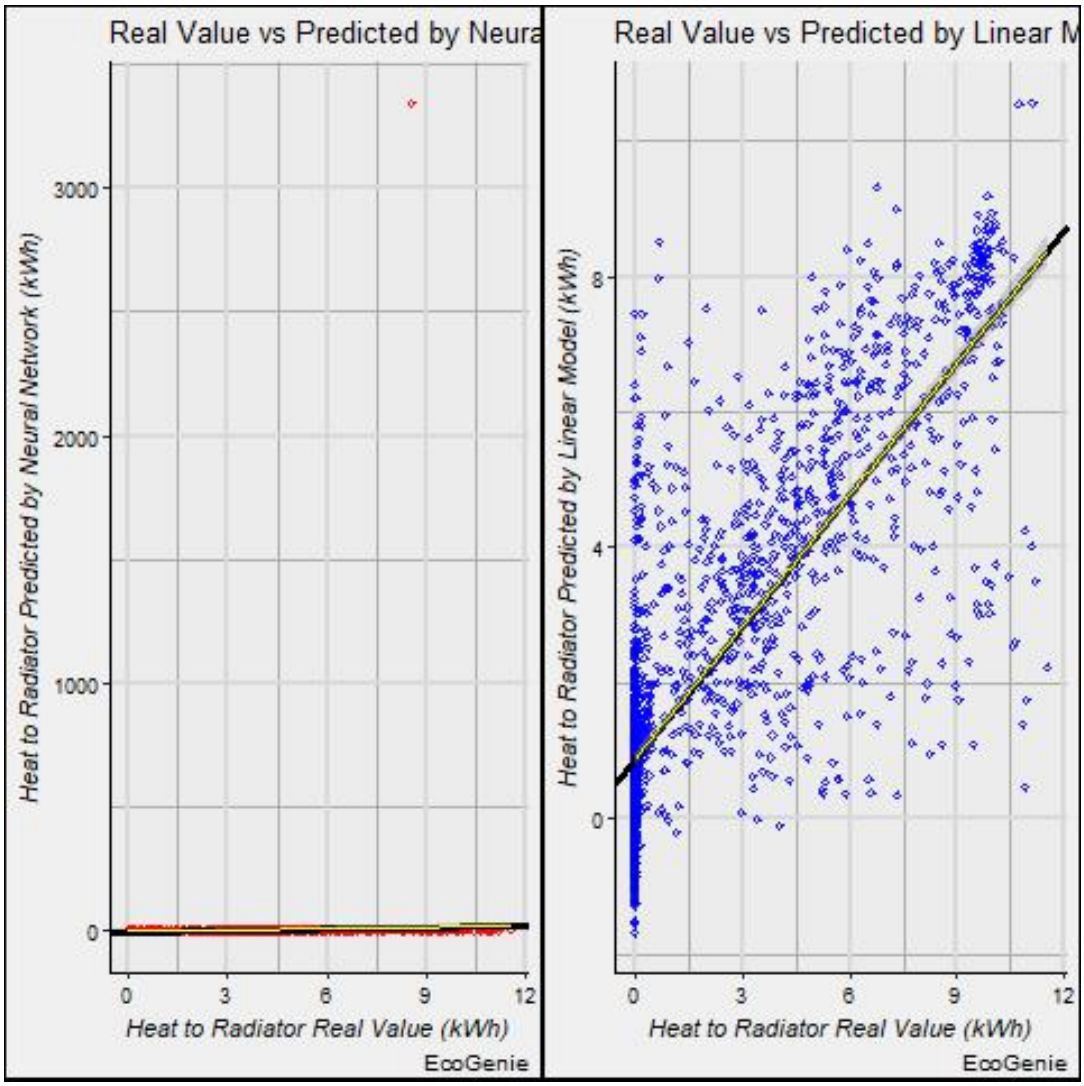


Figure 4.3: *2 (sub-) figures; one with the Real Values vs those Predicted by the Neural Network and one with the Real Values vs those Predicted by the Linear Model. The x-axis of both sub-figures are the (Heat to Radiator) Real Values in kWh, while the y-axis is the Heat to Radiator Predicted by the Neural Network (kWh) in the left sub-figure and the y-axis is the Heat to Radiator Predicted by the Linear Model (kWh) in the right sub-figure. Note that the left figure is flat, due to an extreme value produced by the unnecessary extra hidden layer. The linear model shows a vertical distribution near the Heat to Radiator Real Value of 0kWh, which indicates an effect caused by the linear approximation on certain values. Extra (control) outputs, such as a zero-value output, could increase effectiveness of the models.*

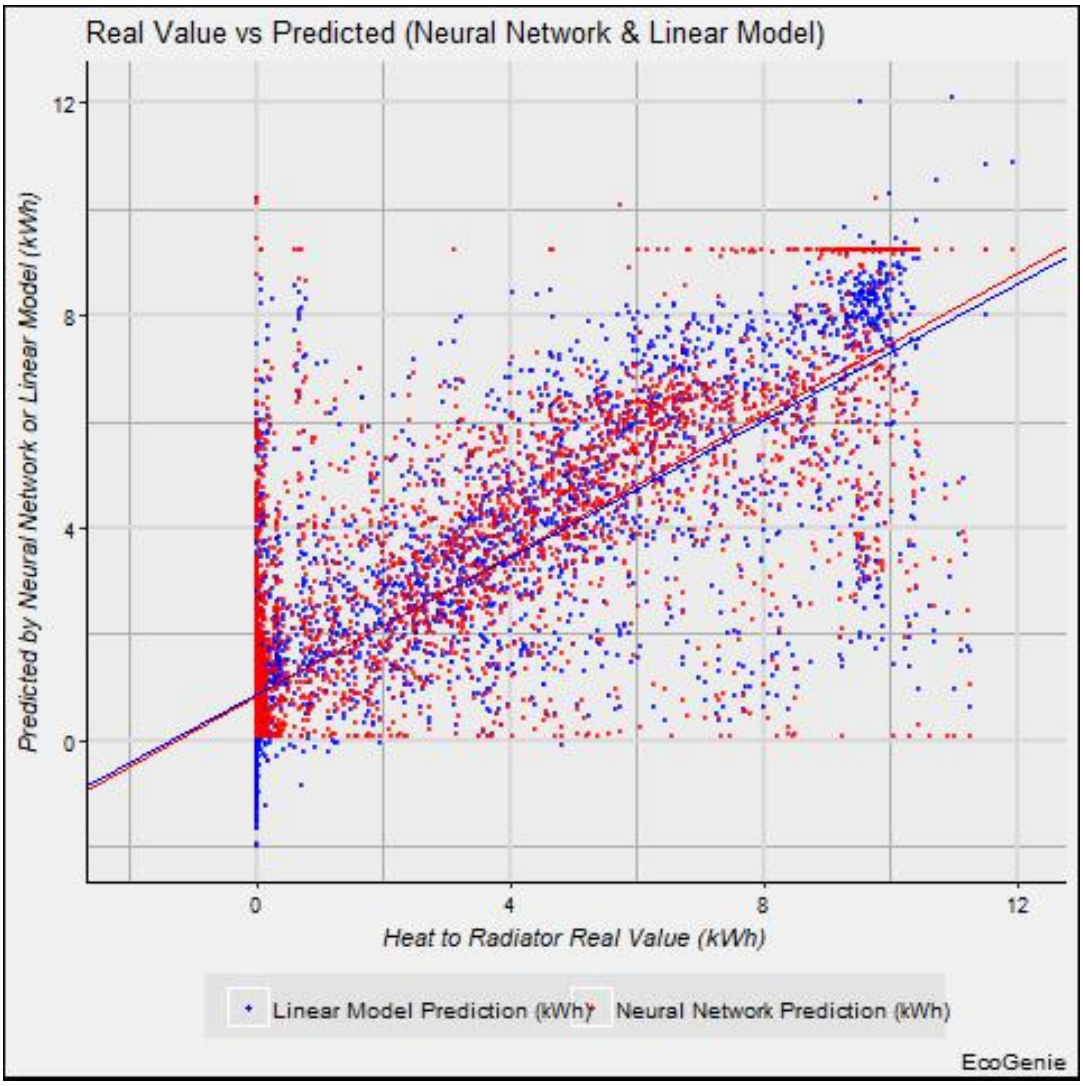


Figure 4.4: *Both predictions of the Neural Network & the Linear Model into one figure, providing a clear comparison of the two methods. The x-axis is the Heat to Radiator Real Value (kWh) and the y-axis is the prediction in kWh of both the Neural Network and the Linear Model, with the Linear Model being blue and the Neural Network being red. Note the vertical, horizontal, and other forms of clustering (of the real values vs the predicted ones); indications of necessary changes to be made to the model. Ideally the real vs predicted values should cluster around the intersecting line(s), with the intersecting line ideally being exactly linear in the form* $y = x$.

5

Recommendations

This internship has been very insightful for understanding the provided EcoGenie data, however in many areas it has also provided insight in subjects for future study. Due to time constraints, this internship has been limited in learning the R language, understanding, cleaning, transforming, and visualizing the data, whilst preparing this data for preliminary machine learning models. The development of a neural network to predict a (reliable) 72-hour energy forecast is ambitious, and it was not completed during the last month. It was a secondary objective and cleaning the data beforehand is a very time-consuming process, however there has been quite some progress in developing the first steps.

This internship report is also meant to assist future studies into the development of an energy-prediction model, for the EcoGenie project. This section provides some ideas and recommendations for future study and development.

Further Neural Network Development using R

There are many machine learning packages in R, of which some are even considered ”meta” packages (Berel, D., 2018), which are highly integrated packages covering a wide range of functions. Many aspects of these packages are hard to understand and taking a package while plugging inputs for immediate results may cause more confusion than insight, since the actual calculation processes within the ”black box” are hidden.

Library Integration

Packages, libraries, neural network types, different machine learning algorithms, deep learning techniques, mathematical equations, hidden layers, coding styles (different people may use different ways of defining functions and variables), may increase in complexity over time, causing possible future problems in integrating them. It is advised to have a focus on just one or a few things at a time. A working model is developed by slowly changing the model in an iterative process, relying on identifying (new) problems and repeating this indefinitely.

Shiny Server using R

In the first month of the internship, there has been some focus on understanding the Shiny Server package, which translate R code into HTML code. The coding structure for the Shiny Server package is similar to ”normal” code in (base) R, however it may require a significant amount of understanding in its inner structure, since some ”communicating” variables may not work as one may be used to in base R coding.

Machine Learning Model using Python

To further automate the Neural Network model, a recommendation is to study Machine Learning in Python. There is a significant amount of documentation, there are very functional packages in Python, and it is faster than R. R is easy to start out with, and Python has been a choice for future study for quite some time, however due to time constraints the focus has been mainly R during this internship (considering producing results and an internship report). DataCamp can be useful in studying the Python coding language.

Visualizing/Mapping the Energy-Balance
A detailed visualization and mapping of the entire energy-balance of the EcoGenie house can be studied. This can take into account (all) the measuring equipment and visualize it in a (possible interactive) way, which can potentially provide other fields of study an advantage, such as insightful problem formulation/solving.

Deep Learning and Selecting Variables
Deep Learning and the process of Selecting Variables can be further studied, in order to improve Neural Network prediction results and computations.

Further Error Analysis
Each variable can be further studied to understand in detail its errors and subsequently its effects on being an input in a neural network.

Machine Learning Package Integrity and Validity
A machine learning algorithm can be studied in detail, with developing methods to test and analyze these algorithms. A ”random” package in either R or Python can seem to be useful (to the person who provided, or created it), but in another model it might be useless. Also there is a possibility the network makes no sense, and is actually completely useless. Developing a method to test machine learning algorithms, can help prove its functionality and therefor assisting in the reliability of its predictions.

Deploying a Reliable Multi-House Energy-Forecast Model in The Netherlands
A developed neural network machine learning algorithm can be applied to multiple houses in The Netherlands in order to validate its effectiveness in reliably predicting energy consumption.

Cost-Benefit Analysis
A detailed cost-benefit analysis can be studied in several areas, such as the benefits of a neural network, or machine learning algorithm, and their implementation versus the cost of doing so (on a large scale).

6

Conclusion

The objectives of the internship were to learn and apply ”big data” analysis techniques, and to explore and model relevant energy data from the EcoGenie house, spanning several years. The primary objectives were learning, applying and analyzing measurement data, with the secondary objective being modelling (with limited) measurement (input) data with training a neural network to derive an energy forecast model. The Master Thesis by Parab, V. (2016) has proven a valuable reference to describe The EcoGenie Project and the Thermal Network Model analyzed and applied in 2016. Literature on physics and the basics of neural networks and deep learning have been studied. The R programming language has been practiced and code refined for many specific and general purposes.

6.1. Learning

The first 2 months had a focus on learning the R-language (programming) and its Shiny package (used to set up servers and convert R code into HTML), while getting to know the content of the acquired data of the measuring equipment. The Anaconda Navigator, R-Studio, and Yupiter Notebook have been used for coding and making notes of written code. On DataCamp, an account has been used to learn the programming language ”R”, and some extra courses on other subjects. Python has been considered, but learning it has been postponed until after the internship.

Several tools have been used in order to communicate and to learn open-source code solutions from websites with a forum on code discussions and questions. Slack is a messenger application used by many programmers, GitHub is a website where people share code with collective online version controls, YouTube and Google have been used to search for a variety of solutions and explanations, Stack-Overflow is website with many code discussions and questions/answers, and LaTex, ShareLaTeX, JabRef and bibTex have been used to write this internship report.

The essence of (Big) Data Analytics has been studied, and research has been done on the functionality of neural networks and their possible applications. The Linux operating system has been studied, and several versions (Mint and Debian environments) have been practiced with.

The EcoGenie house has measuring equipment such as air-source heat pumps, micro CH-P’s, boilers,hot water storage, Photo Voltaic cells, solar thermal and batteries. These have been studied for developing integrated energy balances. The raw data set consists of a great deal of missing values and different names for the same measuring equipment due to various changes overt he years. Data Cleaning techniques have been studied. The raw data used were written as comma-delimited files (csv files), and importing techniques have been refined for general and specific purposes.

6.2. Applying Learned Subjects

The second 2 months mainly consisted of first identifying and formulating the objective and problems. Importing and preliminary cleaning techniques were the first parts being applied of the studied programming language. The files were examined with a list of technical names and their dimensions analyzed before manipulating the data in usable forms. The actual total cleaning time for the entire data set of 1950 csv files with each 1440 observations for every 350 variables took about 4 months,

overlapping other phases of learning the R programming language, and studying neural networks and deep (and machine-) learning algorithms.

Data Exploration has been done throughout the entire internship, and there has been mainly a focus on just 5 variables (6 in the last month). Almost all variables of the KNMI data set have been explored and implemented during the last month, for training the machine learning algorithm. This internship report has been written during the last few weeks.

Figures 3.5, 3.6, B.3, B.5, 3.7, B.1, B.2, B.4, B.6, and 3.8, display the results of extensive data cleaning and its visualization.

6.3. Energy Forecast Model

An objective has been to develop a Reliable Energy Forecast Model with a granularity of 15-minute averages and there has been a focus on training a neural network, with the objective to derive an energy forecast model based on limited input data. The (in)accuracy of the measuring equipment has been taken into account, and a model has been developed to visualize the accuracy of the measurements. Thermodynamic effects have been visualized to create a clear view of the individual benefits of each solution.

The Development of the Model is an iterative process, meaning that each model building stage is an improvement on the previous one. The energy forecast model has not been finalized during this internship, but there have been numerous developments. There are suggestions and recommendations on future development of the model, which are also discussed in this report.

Preparing the Data is a very substantial process, which in this internship has taken about 80% of the time. The largest part of Data Preparation is Data Cleaning. Problems sometimes went unnoticed for a period of time, which indicated the importance of actually identifying the problem first. Certain code seemed useful and proved useful for a while, however during development it may become a problem once combined with new code structures, prompting the necessity of identifying this issue, formulating it, and address it. Without preparing data, it is just one big unknown, with its only real value being random nonsensical data.

The Machine Learning Model (using R) is an interesting tool in providing predictions which may prove to provide a reliable energy forecast. It is however, very difficult to understand, and the development of the neural network model used during this internship shows the many hurdles this process causes.

This internship has been very insightful in understanding the provided EcoGenie data and in many areas it has also provided insight in subjects for future study. Due to time constraints, this internship has been limited in learning the R language, understanding, cleaning, transforming, and visualizing the data, whilst preparing this data for preliminary machine learning models. The development of a neural network to predict a (reliable) 72-hour energy forecast is somewhat of compelling ambition, however doing so within 1 month did not seem realistic. It was a secondary objective, since cleaning the data is a very time-consuming process, and there has been quite some progress in developing the first steps.

A

Tables

Appendix-A is dedicated to host all the tables which did not have the necessary requirement to be displayed within the accompanying text, as to avoid unnecessary clutter in the internship report structure. The tables provide additional information for either improving the context of the internship report, or for future reference.

Table A.1: Amount of Observations n in a certain time-frame of minute-data.

Amount of Observations	n
Hour	60
Day	1440
(7) Days	10080
(28) Days	40320
(29) Days	41760
(30) Days	43200
(31) Days	44640
(Non-Leap) Year	525600
(Leap) Year	527040
(April 1st 2013) to (March 31st 2018)	2629440

Table A.2: Standard Deviation of 5 different variables, with "Data Breaks" being the slicing of time-frames for calculations of sums and averages, before being used for standard deviation calculations.

Standard Deviation (σ)	Data Break by: Minute	Data Break by: Hour	Data Break by: Day	Data Break by: Week	Data Break by: Month	Data Break by: Year
Electricity Sum	$2.1 \cdot 10^{-2}$	1.1	$1.9 \cdot 10^{1}$	$1.3 \cdot 10^{2}$	$5.1 \cdot 10^{2}$	$1.5 \cdot 10^{3}$
Gas Flow Sum	$5.8 \cdot 10^{-2}$	2.9	$3.2 \cdot 10^{1}$	$1.7 \cdot 10^{2}$	$5.3 \cdot 10^{2}$	$2.3 \cdot 10^{3}$
Average House Temperature	2.4	2.3	2.1	1.9	1.5	$4.4 \cdot 10^{-1}$
Average Ambient Temperature	7.0	7.0	5.9	5.6	5.3	$9.3 \cdot 10^{-1}$
Heat to Radiator Sum	$7.0 \cdot 10^{-2}$	3.5	$5.3 \cdot 10^{1}$	$3.5 \cdot 10^{2}$	$1.4 \cdot 10^{3}$	$2.4 \cdot 10^{3}$

Table A.3: Electricity Consumption (kWh) per hour, day, week, month, and year.

Electricity Consumption	kWh ($\pm 5\%$) per Hour	kWh ($\pm 5\%$) per Day	kWh ($\pm 5\%$) per Week	kWh ($\pm 5\%$) per Month	kWh ($\pm 5\%$) per Year
Minimum	$0.0 \cdot 10^{-1}$	$0.0 \cdot 10^{-1}$	$2.4 \cdot 10^{1}$	$2.0 \cdot 10^{2}$	$1.1 \cdot 10^{4}$
1st Quartile	$5.3 \cdot 10^{-1}$	$1.6 \cdot 10^{1}$	$1.2 \cdot 10^{2}$	$5.3 \cdot 10^{2}$	$1.1 \cdot 10^{4}$
Median	$9.8 \cdot 10^{-1}$	$2.8 \cdot 10^{1}$	$2.0 \cdot 10^{2}$	$9.4 \cdot 10^{2}$	$1.1 \cdot 10^{4}$
Mean	1.4	$3.2 \cdot 10^{1}$	$2.3 \cdot 10^{2}$	$9.9 \cdot 10^{2}$	$1.2 \cdot 10^{4}$
3rd Quartile	2.0	$4.6 \cdot 10^{1}$	$3.2 \cdot 10^{2}$	$1.3 \cdot 10^{3}$	$1.2 \cdot 10^{4}$
Maximum	6.9	$9.0 \cdot 10^{1}$	$5.7 \cdot 10^{2}$	$2.1 \cdot 10^{3}$	$1.5 \cdot 10^{4}$

Table A.4: Gas Flow (kWh) per hour, day, week, month, and year.

Gas Flow	kWh ($\pm 5\%$) per Hour	kWh ($\pm 5\%$) per Day	kWh ($\pm 5\%$) per Week	kWh ($\pm 5\%$) per Month	kWh ($\pm 5\%$) per Year
Minimum	$0.0 \cdot 10^{-3}$	$0.0 \cdot 10^{-3}$	$0.0 \cdot 10^{-2}$	1.7	$4.6 \cdot 10^{3}$
1st Quartile	$0.0 \cdot 10^{-3}$	$1.8 \cdot 10^{-1}$	$1.7 \cdot 10^{1}$	$1.1 \cdot 10^{2}$	$4.6 \cdot 10^{3}$
Median	$0.0 \cdot 10^{-3}$	3.7	$7.8 \cdot 10^{1}$	$4.4 \cdot 10^{2}$	$5.9 \cdot 10^{3}$
Mean	$7.5 \cdot 10^{-1}$	$1.8 \cdot 10^{1}$	$1.3 \cdot 10^{2}$	$5.5 \cdot 10^{2}$	$6.6 \cdot 10^{3}$
3rd Quartile	$0.0 \cdot 10^{-3}$	$2.4 \cdot 10^{1}$	$1.6 \cdot 10^{2}$	$8.2 \cdot 10^{2}$	$7.9 \cdot 10^{3}$
Maximum	$2.1 \cdot 10^{1}$	$2.7 \cdot 10^{2}$	$1.2 \cdot 10^{3}$	$2.0 \cdot 10^{3}$	$9.8 \cdot 10^{3}$

Table A.5: Heat to Radiator (kWh) per hour, day, week, month, and year.

Heat to Radiator	kWh (±5%) per Hour	kWh (±5%) per Day	kWh (±5%) per Week	kWh (±5%) per Month	kWh (±5%) per Year
Minimum	$0.0 \cdot 10^{-4}$	$0.0 \cdot 10^{-4}$	$0.0 \cdot 10^{-1}$	0.0	$1.6 \cdot 10^{4}$
1st Quartile	$0.0 \cdot 10^{-4}$	$9.6 \cdot 10^{-2}$	$3.3 \cdot 10^{1}$	$2.9 \cdot 10^{2}$	$2.0 \cdot 10^{4}$
Median	$2.1 \cdot 10^{-2}$	$4.3 \cdot 10^{1}$	$3.4 \cdot 10^{2}$	$1.7 \cdot 10^{3}$	$2.1 \cdot 10^{4}$
Mean	2.3	$5.5 \cdot 10^{1}$	$3.9 \cdot 10^{2}$	$1.7 \cdot 10^{3}$	$2.0 \cdot 10^{4}$
3rd Quartile	4.2	$9.8 \cdot 10^{1}$	$6.6 \cdot 10^{2}$	$2.9 \cdot 10^{3}$	$2.1 \cdot 10^{4}$
Maximum	$2.3 \cdot 10^{1}$	$2.3 \cdot 10^{2}$	$1.3 \cdot 10^{3}$	$4.6 \cdot 10^{3}$	$2.3 \cdot 10^{4}$

B

Figures

Appendix-B is dedicated to host all the figures which did not have the necessary requirement to be displayed within the accompanying text, as to avoid unnecessary clutter in the internship report structure. The figures provide additional information for either improving the context of the internship report, or for future reference.

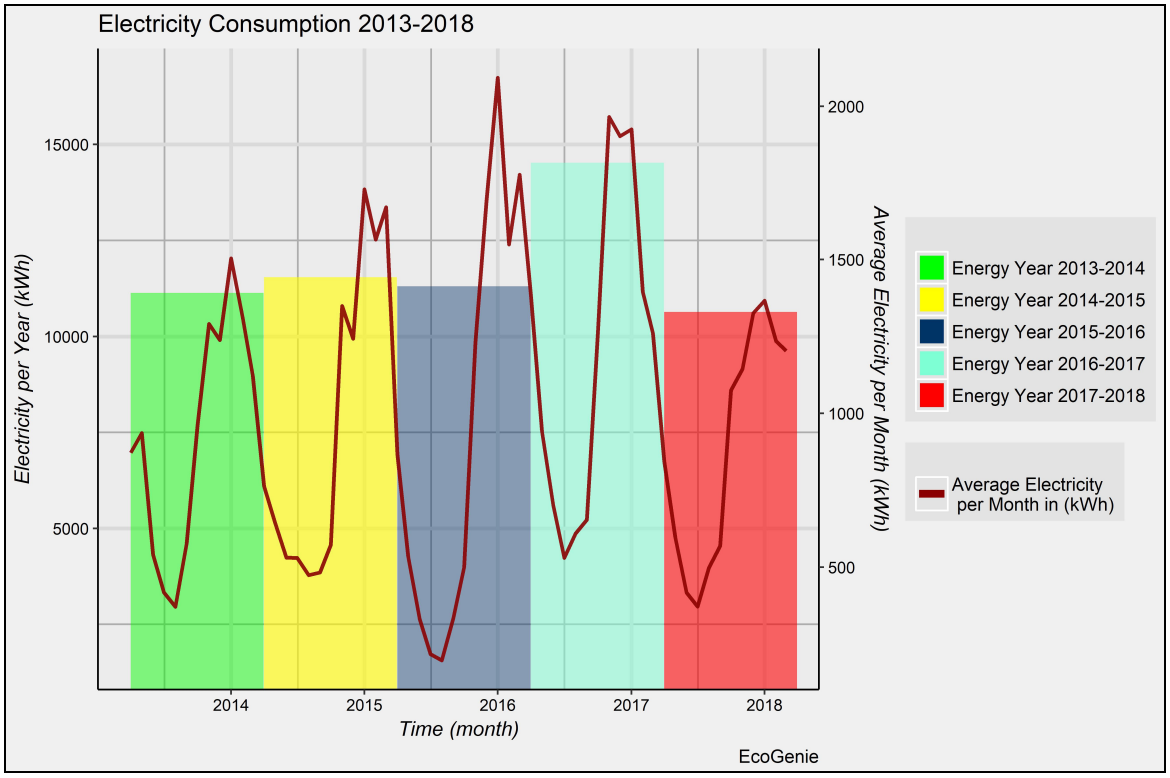


Figure B.1: *The Electricity Consumption is in kWh on the left y-axis, and the Average Electricity per Month is in kWh on the right y-axis. $Time$ is in hour on the x-axis. Five energy-years are displayed in 5 colors (displaying the average value of that year), starting April 1st 2013 and ending March 31st 2018. The Energy-Year 2013-2014 is green, the Energy-Year 2014-2015 is yellow, the Energy-Year 2015-2016 is dark blue, the Energy-Year 2016-2017 is aquamarine, and the Energy-Year 2017-2018 is red. The Average Electricity per Month is a dark red line in the graph.*

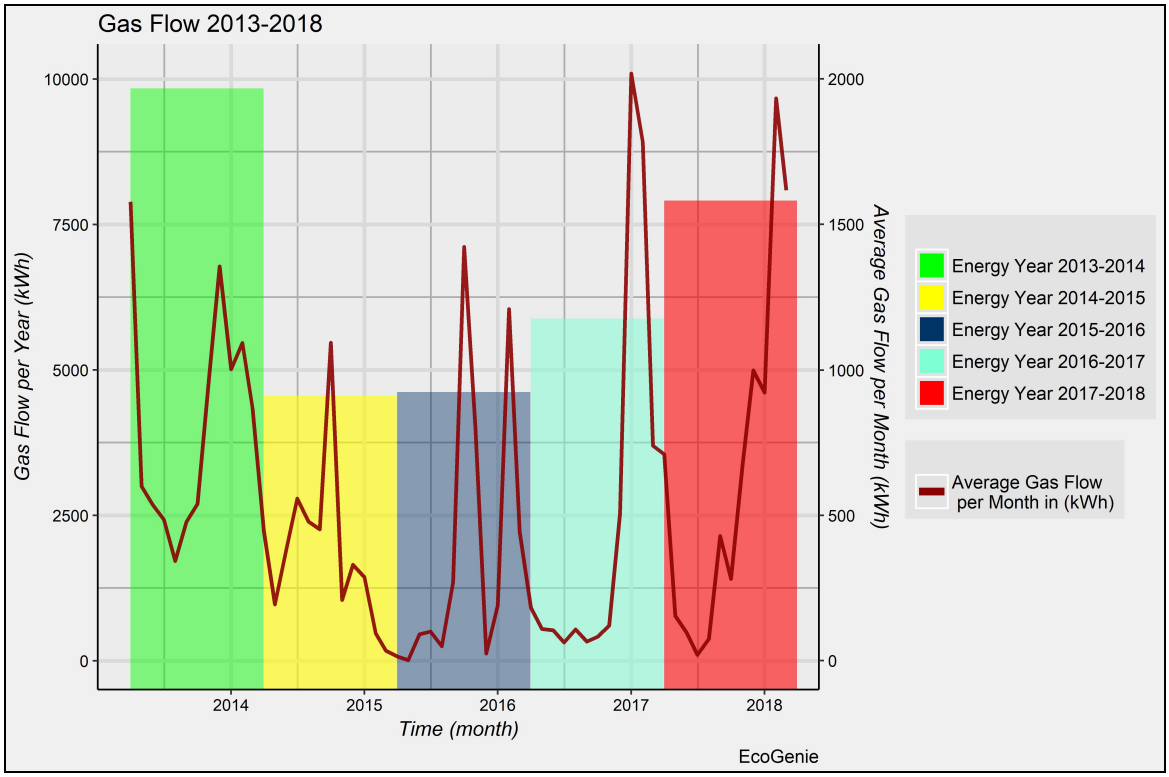


Figure B.2: *The Gas Flow is in kWh on the left y-axis, and the Average Gas Flow per Month is in kWh on the right y-axis.* $Time$ *is in hour on the x-axis. Five energy-years are displayed in 5 colors (displaying the average value of that year), starting April 1st 2013 and ending March 31st 2018. The Energy-Year 2013-2014 is green, the Energy-Year 2014-2015 is yellow, the Energy-Year 2015-2016 is dark blue, the Energy-Year 2016-2017 is aquamarine, and the Energy-Year 2017-2018 is red. The Average Gas Flow per Month is a dark red line in the graph.*

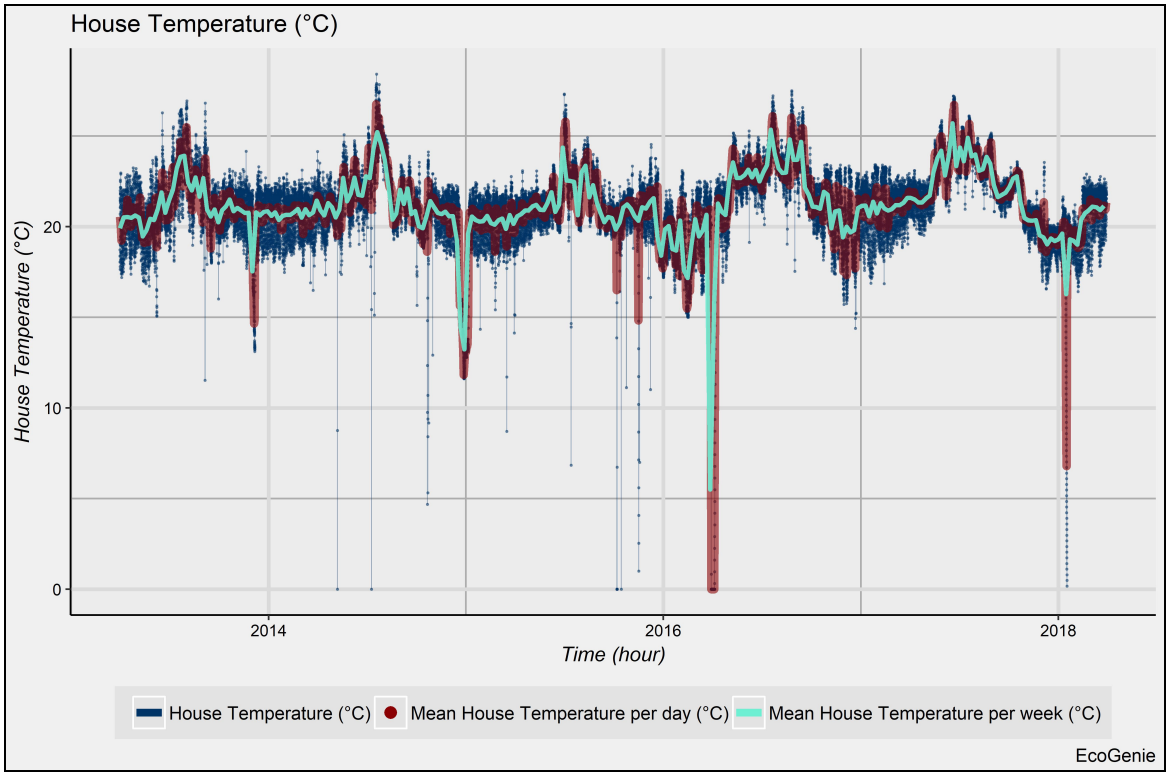


Figure B.3: *House Temperature is in °C on the y-axis.* $Time$ *is in hour on the x-axis. Five energy-years are displayed, starting April 1st 2013 and ending March 31st 2018. Dark blue lines are data per hour, dark red points are the mean day averages, and light green lines are the mean week averages.*

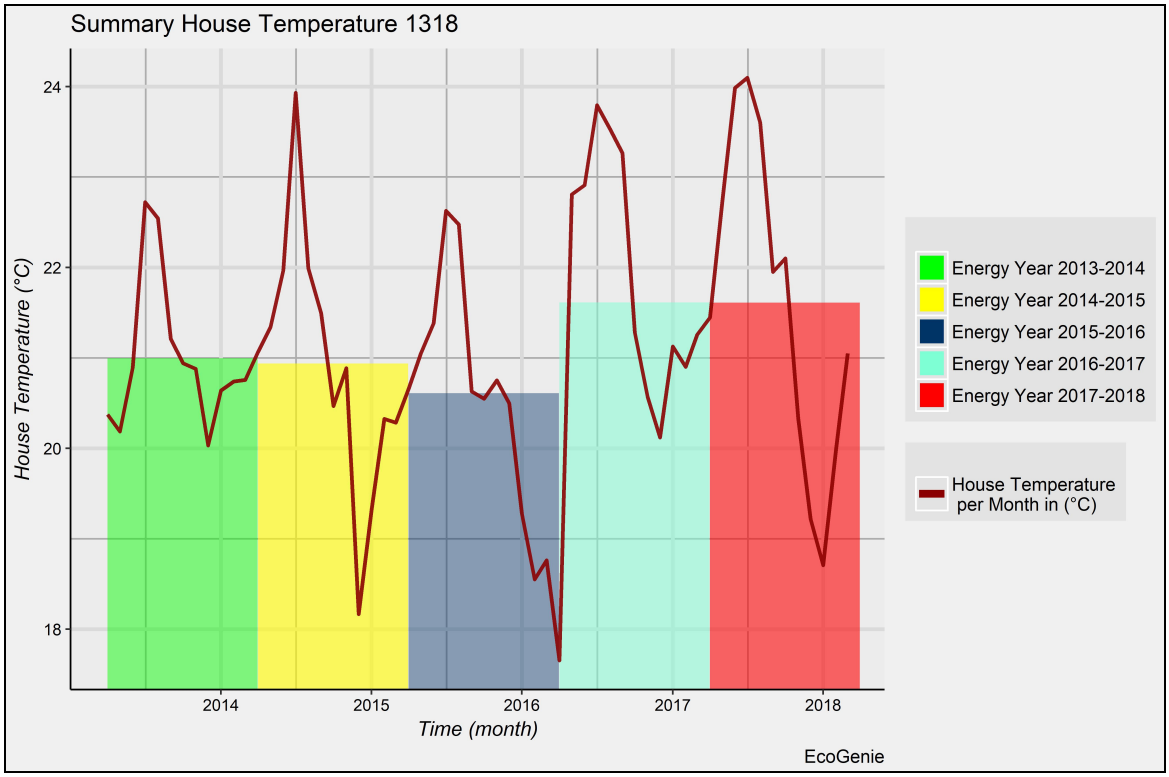


Figure B.4: *The Average House Temperature is in °C on the y-axis.* $Time$ *is in hour on the x-axis. Five energy-years are displayed in 5 colors (displaying the average value of that year), starting April 1st 2013 and ending March 31st 2018. The Energy-Year 2013-2014 is green, the Energy-Year 2014-2015 is yellow, the Energy-Year 2015-2016 is dark blue, the Energy-Year 2016-2017 is aquamarine, and the Energy-Year 2017-2018 is red. The Average House Temperature per Month is a dark red line in the graph.*

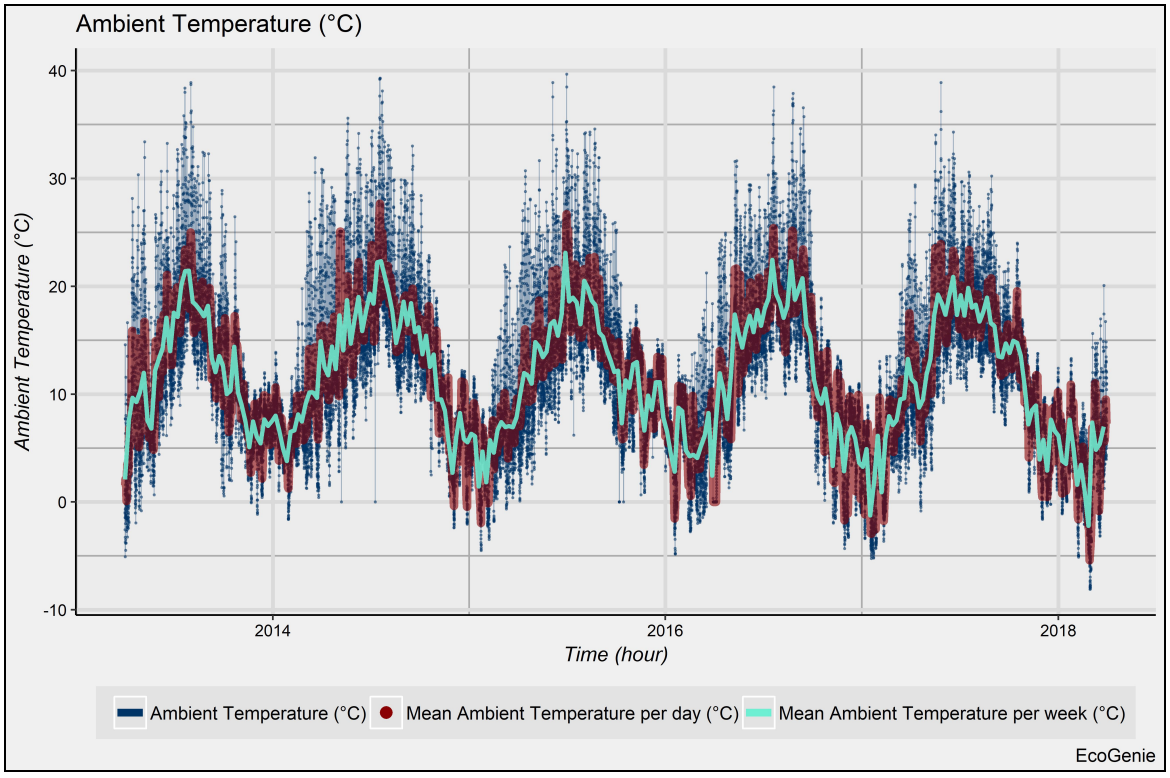


Figure B.5: *Ambient Temperature is in °C on the y-axis.* $Time$ *is in hour on the x-axis. Five energy-years are displayed, starting April 1st 2013 and ending March 31st 2018. Dark blue lines are data per hour, dark red points are the mean day averages, and light green lines are the mean week averages.*

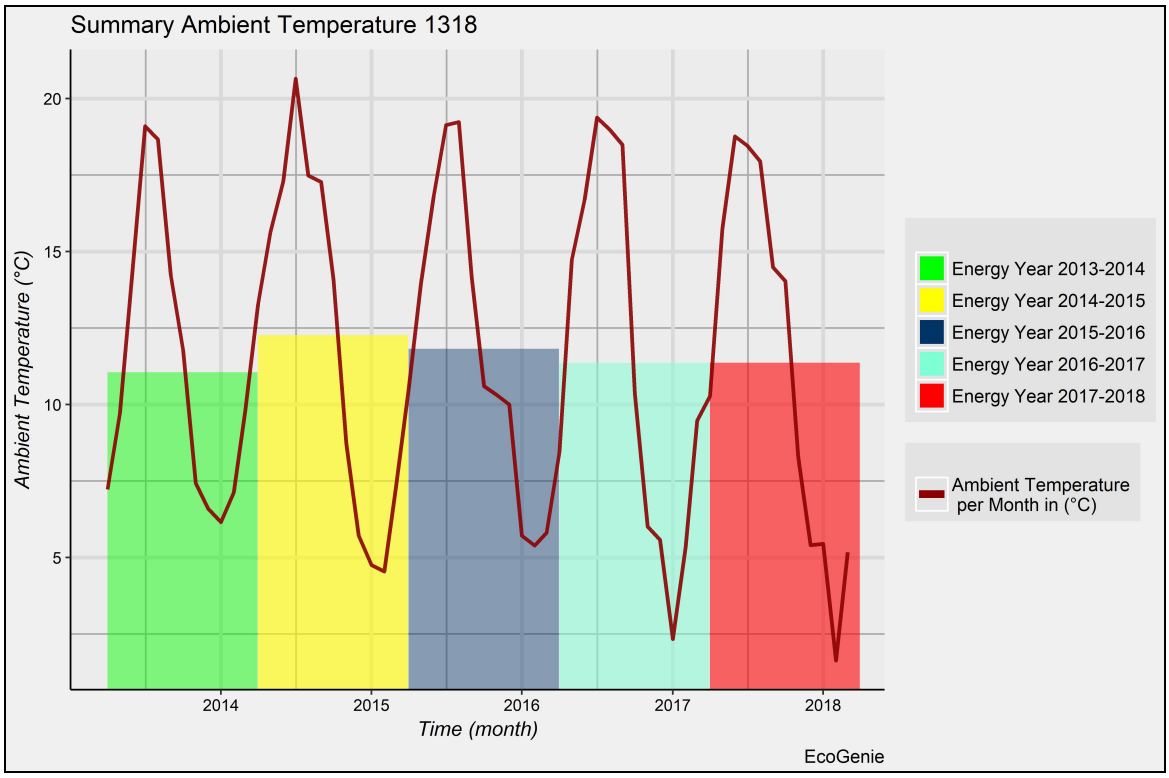


Figure B.6: *The Average Ambient Temperature is in °C on the y-axis. `Time` is in hour on the x-axis. Five energy-years are displayed in 5 colors (displaying the average value of that year), starting April 1st 2013 and ending March 31st 2018. The Energy-Year 2013-2014 is green, the Energy-Year 2014-2015 is yellow, the Energy-Year 2015-2016 is dark blue, the Energy-Year 2016-2017 is aquamarine, and the Energy-Year 2017-2018 is red. The Average Ambient Temperature per Month is a dark red line in the graph.*

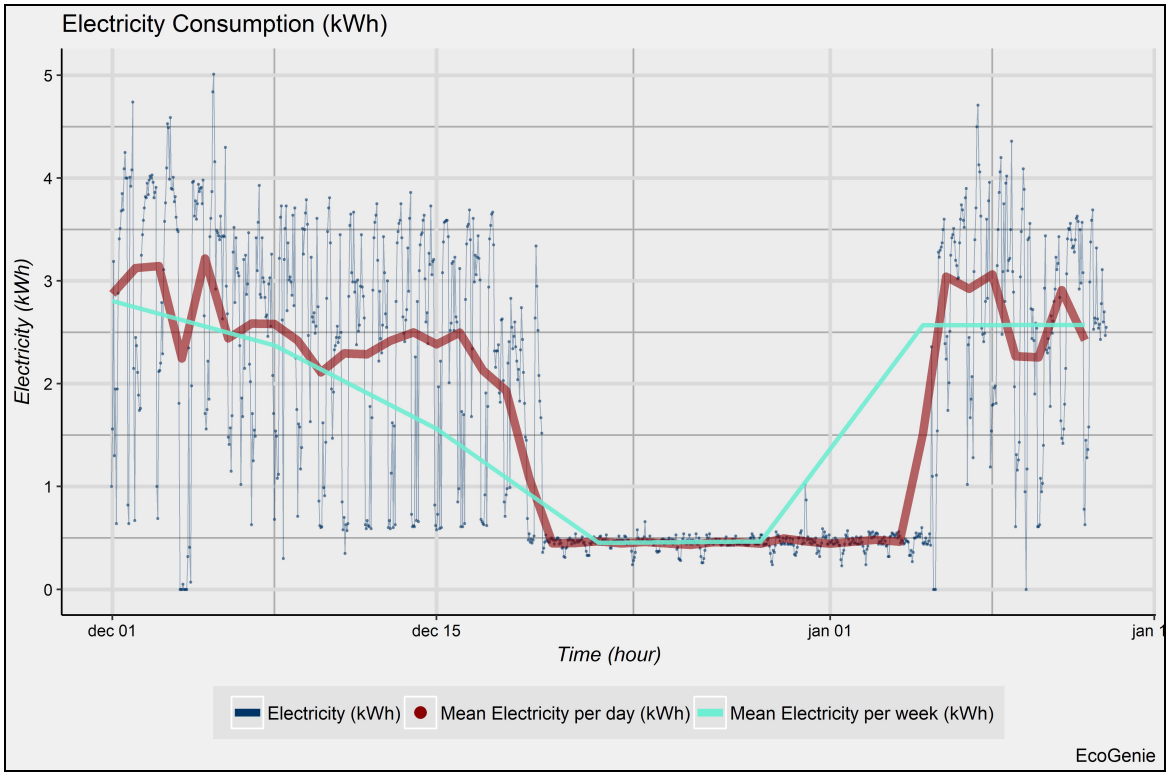


Figure B.7: *The Electricity Consumption is in kWh on the y-axis, and the Time (hour) is on the x-axis. Note the amount of time it takes for the electricity consumption to become zero.*

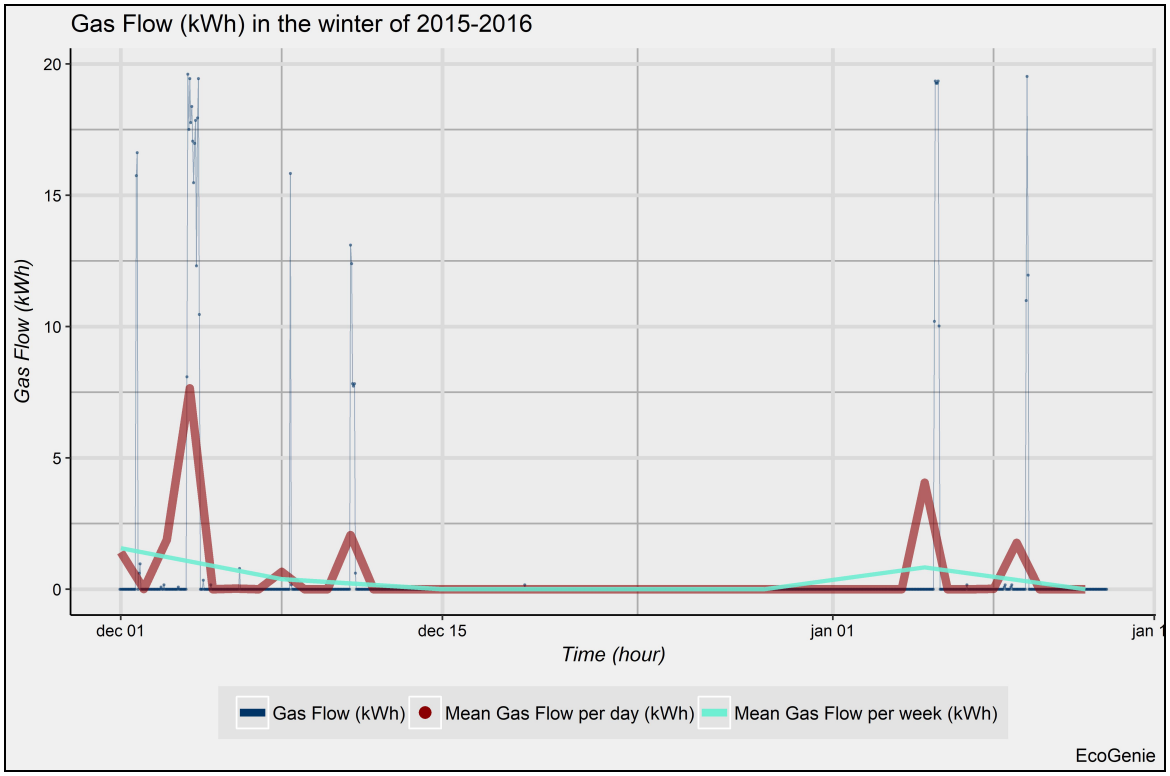


Figure B.8: *The Gas Flow is in kWh on the y-axis, and the Time (hour) is on the x-axis. Note how the gas flow becomes zero rapidly.*

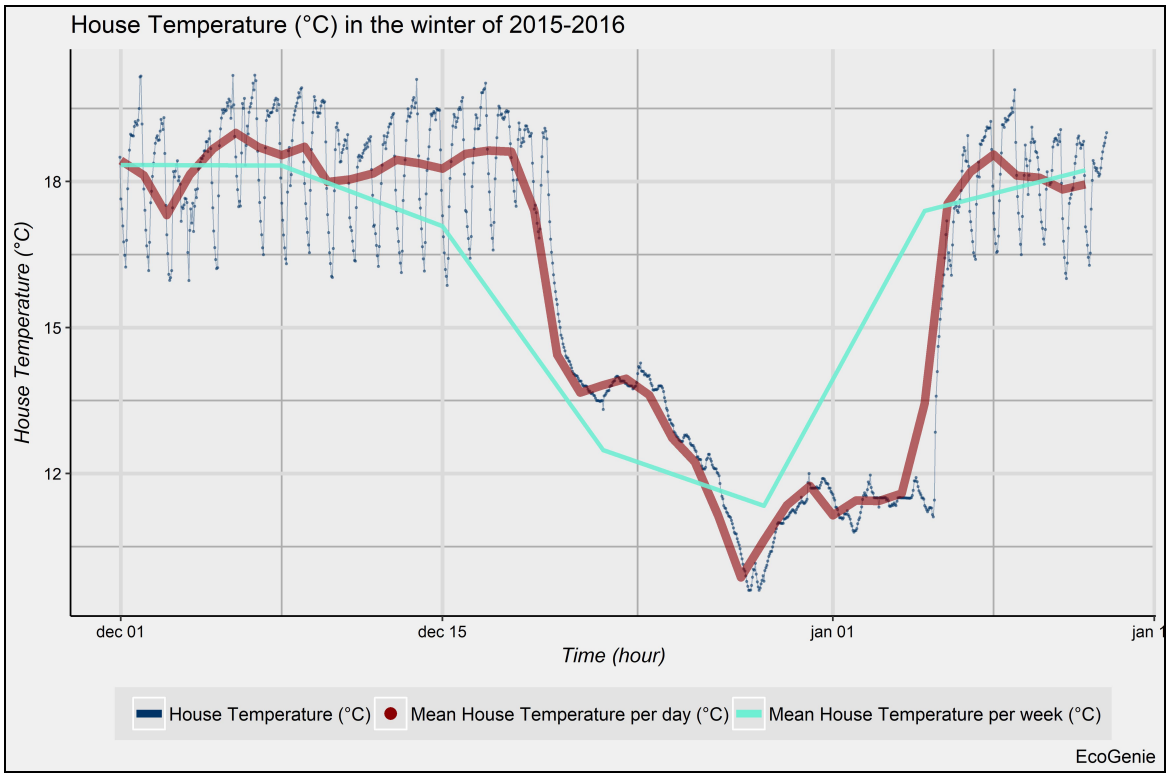


Figure B.9: *The House Temperature is in °C on the y-axis, and the Time (hour) is on the x-axis. Note the time it takes for the house temperature to become zero, and how much time it takes for the house to heat up again.*

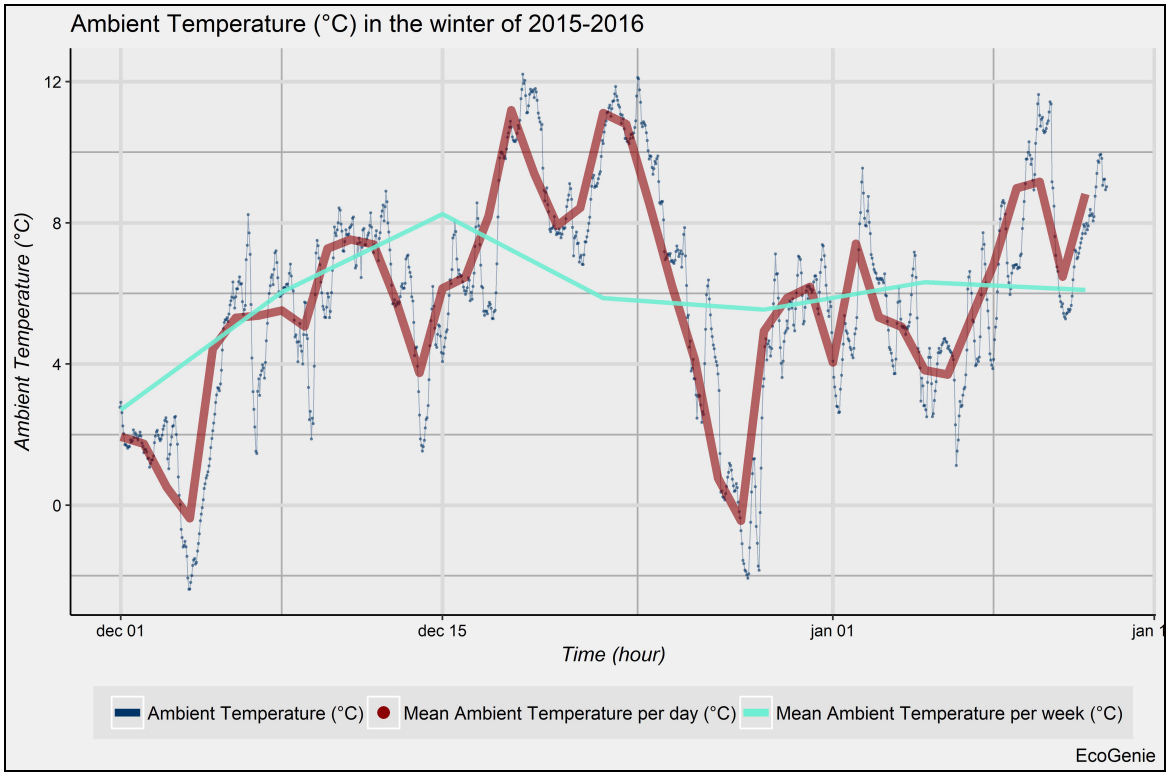


Figure B.10: *The Ambient Temperature is in °C on the y-axis, and the Time (hour) is on the x-axis. Note the correlation the ambient temperature has with the house temperature, seen in figure B.9.*

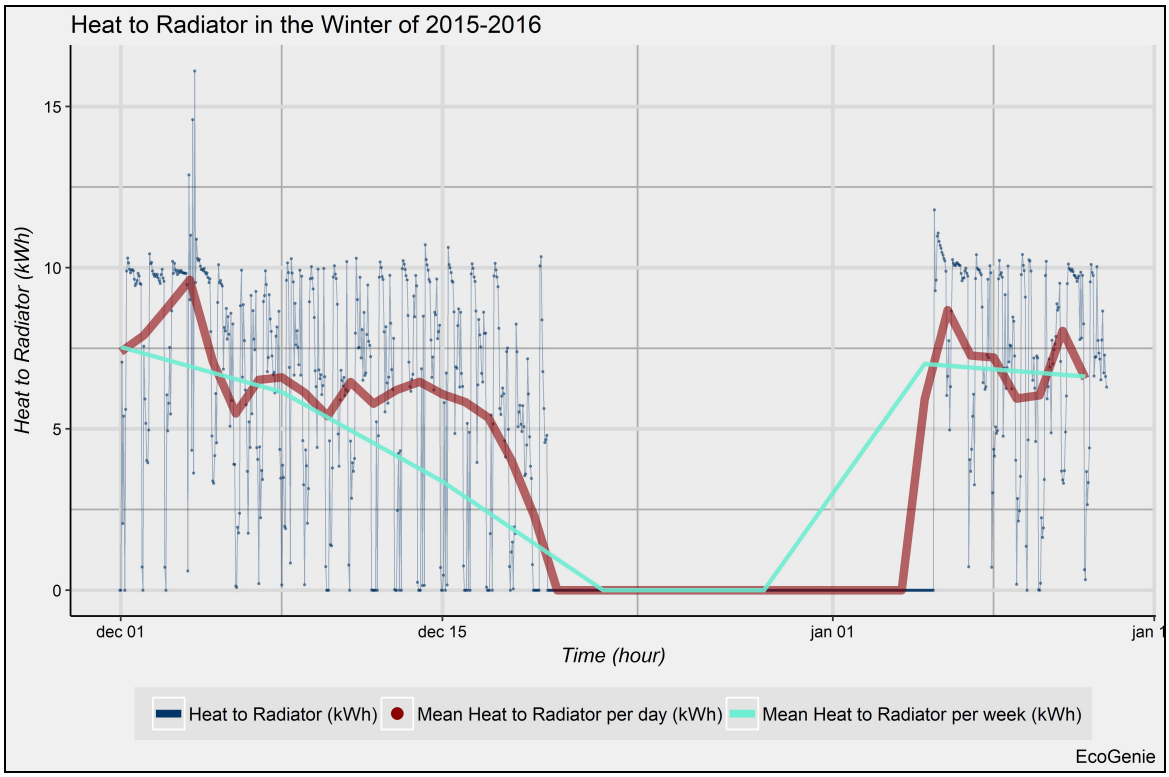


Figure B.11: *The Heat to Radiator is in kWh on the y-axis, and the Time (hour) is on the x-axis. Note the curve going towards zero once turned off, and the fairly linear line moving from zero once turned on.*

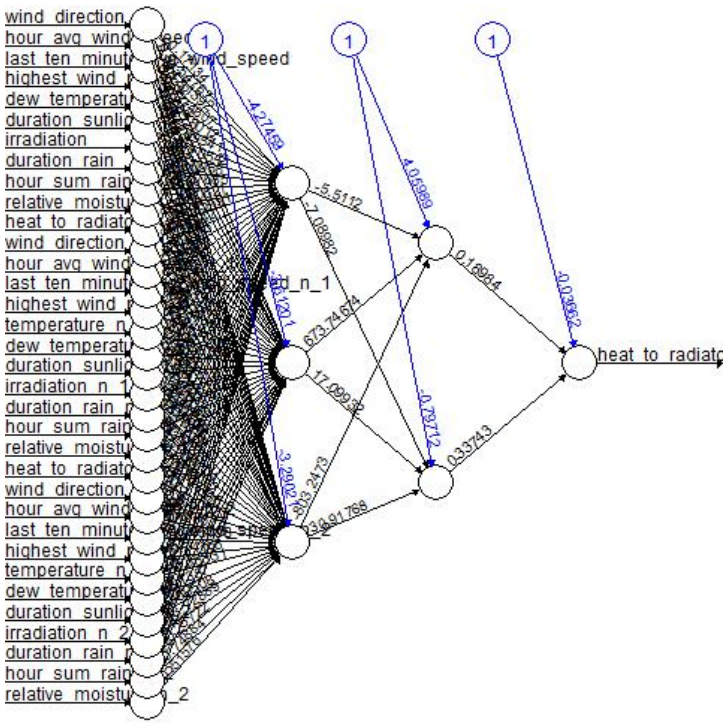


Figure B.12: *This neural network has many variables with several previous iterations. This model, while being previously validated, is now outdated and did not prove to provide useful predictions. It did however visualize during this process which changes were necessary, which are subject to future study and development.*

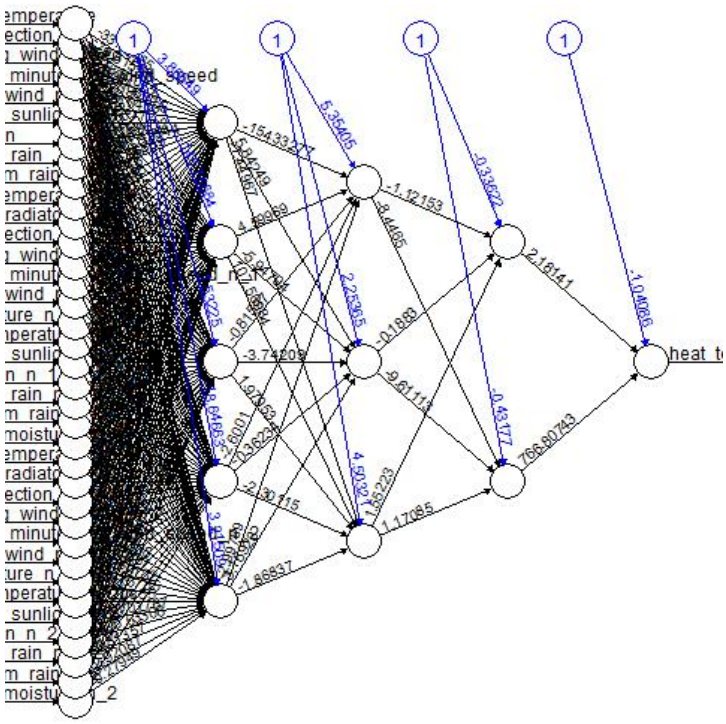


Figure B.13: *This neural network contains 3 hidden layers and its many inputs were provided to improve the previously deployed model. This test however proved that an extra hidden layer, without truly understanding its effects, caused the algorithm to converge to nonsensical outputs as seen in 4.3.*

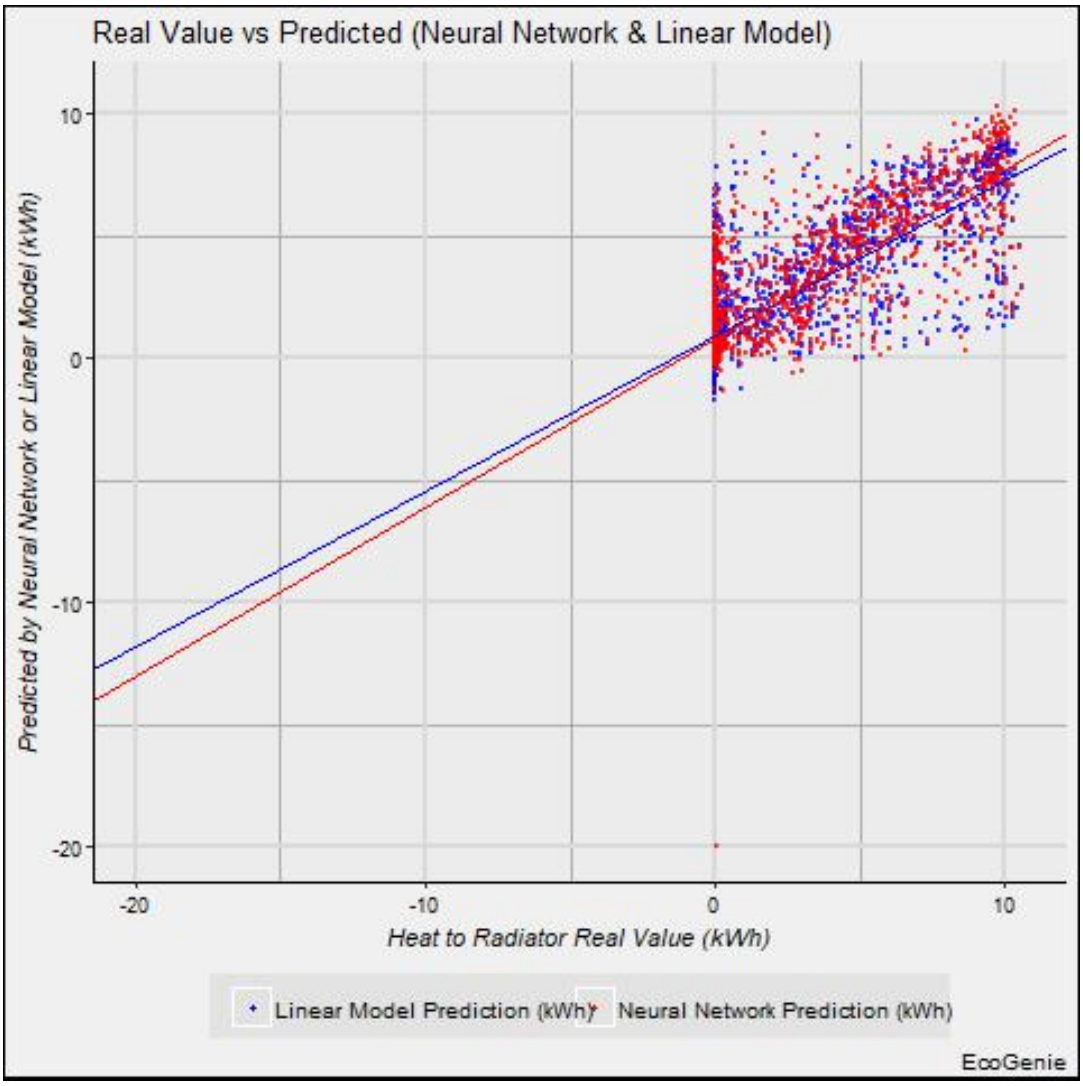


Figure B.14: *Both predictions of the Neural Network & the Linear Model in one figure, providing a clear comparison of the two methods. The x-axis is the Heat to Radiator Real Value (kWh) and the y-axis is the prediction in kWh of both the Neural Network and the Linear Model, with the Linear Model being blue and the Neural Network being red. Note large empty space in the left, left-bottom, and bottom area, caused by only 1 extreme value being -20kWh. This figure has been designed to have the x-axis and the y-axis to have the same length, to clearly visualize the offset the intersection lines have compared to the ideal* $y = x$ *form.*

Bibliography

Parab, V., *Thermal Modelling of Existing Residential Buildings in North-Western Europe*, Master's thesis, TU Delft (2016).

T. D. Dasu, Johnson, *Exploratory Data Mining and Data Cleaning*, edited by Balding, D. J. and Bloomfield, P. and Cressie, N. A. C. and Fisher, N. I. and Johnstone, I. M. and Kadane, J. B. and Ryan, L. M. and Scott, D. W. and Smith, A. F. M. and Teugels, J. L. (Wiley-Interscience, AT&T Labs, Research Division Florham Park, NJ, 2003).

C. Dieperink, I. Brand, and W. Vermeulen, *Diffusion of energy-saving innovations in industry and the built environment: Dutch studies as inputs for a more integrated analytical framework,* Energy Policy **32**, 773 (2004).

Boyle, G., *Renewable Energy*, edited by Boyle, G. (Oxford University Press, 2004).

J. Hermans, *Energy Survival Guide* (Leiden University Press, 2011).

F. Hirosawa and J. Wirth, *Generalised energy conservation law for wave equations with variable propagation speed,* Journal of Mathematical Analysis and Applications **358**, 56 (2009).

Wolfson, R., *Essential University Physics: Volume 2* (Pearson Education Limited, 2014).

Wolfson, R., *Essential University Physics: Volume 1*, 2nd ed., edited by Pearson New International Edition (Pearson Education Limited, 2014).

Geersen, T. M., *Physical Properties of Natural Gases*, edited by Geersen, T. M. (N.V. Nederlandse Gasunie, 1988).

Boles, M. A. and Cengel, Y. A., *Thermodynamics: An Engineering Approach*, edited by McGraw-Hill (McGraw-Hill, 2009).

F. Kreith and W. Z. Black, *Basic heat transfer* (Harper & Row New York, 1980).

Kimmenaede, A. J. M., *Warmteleer voor Technici*, tenth ed., edited by Kimmenaede, A. J. M. (Noordhoff Uitgevers, 2010).

van Wezel, B., *Elektriciteit in Nederland*, edited by van der Brie, R. (Centraal Bureau voor de Statistiek, Henri Faasdreef 312, 2492 JP Den Haag, 2015).

Bessembinder, J., *Klimaatschetsboek Nederland: het huidige en toekomstige klimaat,* KNMI, De Bilt (2009).

van den Eijnde, P., *Meettechnieken en Meetsystemen*, derde druk ed., edited by T. Meulenhoff (ThiemeMeulenhoff, 2016).

C. De Boor, C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor, *A practical guide to splines*, Vol. 27 (Springer-Verlag New York, 1978).

L. L. Schumaker, *Spline Functions: Computational Methods*, Vol. 142 (SIAM, 2015).

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in Fortran 77: the art of scientific computing*, Vol. 2 (Cambridge university press Cambridge, 1992).

I. J. Schoenberg, *Contributions to the problem of approximation of equidistant data by analytic functions,* in *IJ Schoenberg Selected Papers* (Springer, 1988) pp. 3–57.

Schmidhuber, J., *Deep Learning in Neural Networks: An Overview*, Tech. Rep. (Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, 2014).

Alice, M., *Fitting a neural network in R; neuralnet package,* https://www.r-bloggers.com/fitting-a-neural-network-in-r-neuralnet-package/ (2015), In this post we are going to fit a simple neural network using the neuralnet package and fit a linear model as a comparison.

I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning* (MIT Press, 2016) http://www.deeplearningbook.org.

3Blue1Brown, *But what *is* a Neural Network? | Chapter 1, deep learning,* https://www.youtube.com/watch?v=tIeHLnjs5U8 (2017), For those who want to learn more, I highly recommend the book by Michael Nielsen introducing neural networks and deep learning: https://goo.gl/Zmczdy.

Ghosh, B., *Multicollinearity in R,* https://www.r-bloggers.com/multicollinearity-in-r/ (2017), One of the assumptions of Classical Linear Regression Model is that there is no exact collinearity between the explanatory variables.

J. Y. Tsao and P. Waide, *The world's appetite for light: Empirical data and trends spanning three centuries and six continents,* Leukos **6**, 259 (2010).

P. David, *The key to solving problems is to use ppt,* https://blogs.sas.com/content/sascom/2013/01/31/the-key-to-solving-problems-is-to-use-ppt/ (2013), Figure: The analytics lifecycle with roles from an analytic center of excellence (ACE).

Lloyd, R., *Metric mishap caused loss of NASA orbiter,* http://edition.cnn.com/TECH/space/9909/30/mars.metric.02/ (1999).

Berel, D., *'Meta' machine learning packages in R,* https://towardsdatascience.com/meta-machine-learning-packages-in-r-c3e869b53ed6 (2018), towards Data Science.