



RELaTED

D2.4 – Energy flexibility and DH Control

Date: 2019/09/19

Version: 4.0

Deliverable	D2.4
Name	Energy flexibility and DH Control
Website	www.relatedproject.eu
Project Coordinator	Roberto Garay, TECNALIA, roberto.garay@tecnalia.com
Author(s)	Jakob Fester, DTI, jafe@teknologisk.dk Peter Friis Østergaard, DTI, peo@teknologisk.dk Mikel Lumbreras Mugaguren, Tecnalia, mikel.lumbreras@tecnalia.com Roberto Garay Martinez, Tecnalia, mikel.lumbreras@tecnalia.com

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 768567

This material reflects only the author's views and neither Agency nor the Commission are responsible for any use that may be made of the information contained therein



PROJECT SUMMARY

Project Acronym	RELaTED
Project Title	REnewable Low TEmpérature District
Project Coordinator	Roberto Garay, TECNALIA roberto.garay@tecnalia.com
Starting date	2017/11/01
Duration in months	48
Topic	EE-04-2016-2017 New heating and cooling solutions using low grade sources of thermal energy

Copyright notice

© 2017-2021 RELaTED Consortium Partners. All right reserved. All contents are reserved by default and may not be disclosed to third parties without the written consent of the RELaTED partners, except as mandated by the European Commission contract, for reviewing and dissemination purposes.

All trademarks and other rights on third party products mentioned in this document are acknowledged and owned by the respective holders. The information contained in this document represents the views of RELaTED members as of the date they are published. The RELaTED consortium does not guarantee that any information contained here is error free or up to date, nor makes warranties, express, implied, or statutory, by publishing this document.



DOCUMENT HISTORY

Work Package	WP 2 – System Architectures		
WP Lead	Danish Technological Institute		
Deliverable	D2.4 Energy flexibility and DH Control		
Date	2019/06/27		
Due date	2019/06/30		
Status	DELIVERED		
Date	Version	Person/Partner	Comments
2019/03/14	1.0	Jakob Fester (DTI) Peter Friis Østergaard (DTI)	List of content & input from preliminary report
2019/05/29	2.6	Jakob Fester (DTI) Peter Friis Østergaard (DTI) Mikel Lumbreras Mugaguren (Tecnalia)	Integration of results from Tecnalia
2019/06/21	3.7	Jakob Fester (DTI) Peter Friis Østergaard (DTI) Mikel Lumbreras Mugaguren (Tecnalia)	Final version
2019/07/12	3.9	Jakob Fester (DTI)	Changes according to internal review
2019/09/19	4.0	Roberto Garay Martinez (Tecnalia)	Submission



Table of content

Acronyms	8
1. Executive summary	9
2. Introduction	10
2.1. Objective	10
2.2. Methodology	10
2.3. Report content	11
3. Challenges and advanced analytics models for energy flexibility and control of ULT DH	12
4. Machine Learning (ML) as a solution tool for DH and ULT DH	13
4.1. Introduction to Machine Learning (ML)	13
4.2. ML techniques and tools	15
4.3. ML perspectives in the context of DH	18
4.4. Pioneering examples of exploitation of ML in DH	19
5. Challenges of measurement data for ML training and service use	22
5.1. Evaluation of data requirements for ML use	22
5.2. Measurement frequency	23
5.3. Data communication and resolution of stored data	24
6. A framework for advanced control procedures	25
6.1. Potential of ML for ULT DH.....	25
6.2. DH Doctor and DH Autotune concepts	26
7. Development of tools and algorithms	28
7.1. DH Doctor: Fault detection based on daily readings	30
7.2. DH Autotune: Load forecast and fault detection based on hourly values.	44
8. Conclusions and follow up for adapting	73
8.1. Summary of developed tools	73
8.2. Implementation at DH utilities	75
9. References	76
10. Annex 1: DH Doctor	78
11. Annex 2: DH Autotune	81



List of Tables

Table 1: Overview of algorithm features and data used for development of DH Doctor and DH Autotune.	28
Table 2: R^2 value, MAE and Q_{REF} values for the Q function model with decision-trees applied to test data from 2019 with hourly resolution.	52
Table 3: Average R^2 values of the bagged-tree models, having different input parameters. The R^2 values are calculated as the average values for the 38 best performing models.....	57
Table 4: R^2 values of bagged-tree model and neural network.....	57
Table 5: R^2 values for the four installations discussed in section 7.2.2	58
Table 6: Evaluation of prediction models for mass flow characterized by their combinations of input parameters.	63
Table 7: Model trained on data from all installations compared to models trained individually on each installation.	67
Table 8: Comparison of models predicting <i>cooling</i> , <i>return temperature</i> and <i>mass flow</i>	67

List of Figures

Figure 1: Illustration of two common ML techniques: a) An artificial neural network (ANN) and b) K-means clustering.....	14
Figure 2 Identification of an artificially induced fault on a single sub-station by ML-prediction of the mass flow.	20
Figure 3: Examples of 1-hour values compared to 24-hour-averages: Resolution and quality. Data from AffaldVarme Aarhus, Geding, Denmark.	23
Figure 4: Sources of information (data) in the concept of a DH smart grid.....	26
Figure 5: Illustrating example of clustering over 490 installations (data from Aarhus, Denmark) into 6 clusters.	33
Figure 6: Statistics over detected outliers in clustering using two criteria.	34
Figure 7: Temperatures and consumption: Visualization examples from Geding, Aarhus.....	35
Figure 8: Model predictions and measured values for standardized mass flow.	37



Figure 9: Visual explanation of the calculated parameters “temperature response” and “Linearity” associated with the temperature response-curve. 39

Figure 10: Collective visualization of temperature response curves..... 40

Figure 11: Output from DH Doctor showing groups of outliers resulting from clustering and calculated parameters..... 42

Figure 12: Q algorithm applied to daily average values from building 10045 for different Q_{REF} 49

Figure 13: DT1 and DT2 applied to building 10045. 50

Figure 14: Q algorithm application after DT1, building 10045. The blue dotted line indicates the value of Q_{REF} 51

Figure 15: Temporal expression of the regression models for building 10045 (apartment)..... 53

Figure 16: Actual and predicted consumption in kW. 59

Figure 17: Visualization of results from a model predicting the flow over a test period of 5 days for one building (anonymous meter number). 68

Figure 18: Statistics for all 40 analyzed installations with models trained on individual installations, using the same test period..... 69

Figure 19: Output from the Matlab® program predicting the flow during a test period with an intended fault induced on the data set. 71

Figure 20: Overview of the developed DH Doctor and DH Autotune tools: Prerequisites, output and use of output information. 74

Figure 21: Clustering of 24 installations in Geding using 4 clusters. 78

Figure 22: Clustering of 460 installations from a district in Aarhus using 15 clusters..... 79

Figure 23: Statistics output from DH Doctor. 80

Figure 24: T algorithm applied to building 10045 (residential building with Domestic Hot Water consumption), for different T_{REF} 81

Figure 25: Q algorithm application after DT1 applied to test data from 2019, building 10045. 82

Figure 26: Q algorithm application after DT1 applied to test data from 2019, building 10259. 82



Figure 27: Q algorithm application after DT1 applied to test data from 2019, building 10949..... 83

Figure 28: Q algorithm application after DT1 applied to test data from 2019, building 11718..... 83

Figure 29: Temporal expression of regression models for the buildings 10259 (apartment), 10949 (kindergarten) and 11718 (office building). 84



Acronyms

AI	Artificial Intelligence
BIST	Building Integrated Solar Thermal
DER	Distributed Energy Resources
DH	District Heating
DHC	District Heating and Cooling
DT	Decision Tree
k-NN	k-Nearest Neighbor (k-NN)
LRM	Linear Regression Model
LT	Low Temperature
ML	Machine Learning
MLPs	Multi-Layer Perceptrons
(A)NN	(Artificial) Neural Network
NZEB	Near Zero Energy Building
PEH	Plus Energy Houses
RES	Renewable Energy Sources
SVM	Support Vector Machines
ULT	Ultra Low Temperature



1. Executive summary

RELaTED task 2.4 aims at developing a framework for advanced control procedures for managing and optimizing ULT DH networks based on energy consumers equipped with smart meters and decentralized connected energy producers.

This has been done by use of Machine Learning (ML) which is a fast-developing technology that allows for handling large and complex systems with many parameters and large amounts of data, aiming to find optimal patterns and solutions as well as abnormal behavior. Machine learning is based on training of procedures by use of representative experience (training data), meaning data from LT or ULT DH networks based on energy consumers equipped with smart meters, other operational data sources as well as weather data- and forecasts.

The presented work is based on the following concepts:

- DH Doctor focusing on automatic fault detection and identification of connections/substations with optimization potential.
- DH Autotune focusing on providing recommendations for production set-points and for notification of deviations from expected parameter values.

Together with historical training data from district heating supplies in Denmark and Estonia used for development and test, this has resulted in two algorithms or tools: DH Doctor intended for real-time evaluation of DH substations (fault detection and detection of outliers) and DH Autotune intended for load forecasting (recommendations for set-points and reaction to abnormal behavior by an alarming system).

Furthermore, the tools have been divided into two groups such that algorithms based on both daily and hourly readout of smart meter data are available to widen up the applicability to most DH suppliers regarding their prerequisites of data.

Finally, it is discussed how the tools can be implemented at utilities for practical use and experience in a next step to further evaluate their potential and opportunities seen from a supplier's point of view. This will enable further refinement and tuning of the algorithms towards increased applicability and practical value.



2. Introduction

This deliverable reports on the activities carried out under RElated task 2.4 and describe new possibilities concerning measurement and control of ULT DH and other DH systems with high energy flexibility.

The activities address the need for increased energy efficiency and new management strategies in relation to ULT DH. Installing smart meters allows for increased insight into the conditions of networks and installations and by the use of machine learning to utilize these data in new algorithms and tools.

The report is part of a set of reports that define the system architecture of the RElated concept. The reports are:

- D.2.1 Low temperature concepts
- D.2.2 Interconnection schemes for consumer installations
- D.2.3 Interconnection schemes for producer installations
- **D.2.4 Energy flexibility and district heating control**
- D.2.5 Development schemes for new DH developments
- D.2.6 Transition schemes for district heating in operation

2.1. Objective

The objective of task 2.4 is “developing the framework for advanced control procedures for managing and optimizing low temperature districts with several independent energy procedures”.

2.2. Methodology

Based on a general knowledge about Machine Learning (ML) and District Heating (DH), including experiences from cooperation with various DH supplies in Denmark, the following methodology has been used:

1. Survey study concerning specific use of ML in relation to control of DH
2. Mapping out possibilities and challenges concerning smart meter data including finding specific historical data for training
3. Developing a final strategy for the DH Doctor and DH Autotune framework and for the procedures in relation to realistic demonstration possibilities



4. Developing algorithms for DH Doctor and DH Autotune
5. Agreements with DH supplies concerning smart meter data with a history of up to several years
6. Working out the DH Doctor and DH Autotune control solutions for RElated demonstration
7. Training and validation of the models
8. Description and reporting of the solutions.

2.3. Report content

The content of this report is sectioned as below:

1. Executive summary
2. Introduction including the objective and the methodology used
3. General challenges for energy flexibility and control at ULT DH
4. Machine Learning (ML) as a solution tool for DH and ULT DH - an introduction to ML including existing work in relation to DH
5. Challenges using measurement data for ML, covering data frequency, resolution and data transmission issues
6. A framework for advanced control procedures comprising DH Doctor and DH Autotune
7. Development of ML algorithms and tools (DH Doctor and DH Autotune) and training of the algorithms
8. Conclusions and discussion of follow-up possibilities for adaption of the tools at DH utilities.



3. Challenges and advanced analytics models for energy flexibility and control of ULT DH

Increasing use of distributed energy resources (DER) from renewable energy and waste, including plus energy houses (PEH), as well as the basis for the energy supply of the DH or cooling network, makes it difficult to overlook co-operation and efficient use of energy.

Using smart meters with remote communication and including still cheaper sensor solutions makes it possible to gain increased knowledge of the connections and DH network's operation. Due to large amounts of data it is, however, at the same time increasingly difficult via ordinary control and relay tools to perform overviews or simulations.

Reduction of DH network temperatures to ULT DH causes further challenges since the usual traditions of management and operation cannot in all cases be transferred. In addition, variable energy prices must be expected in the long term.

The new needs and opportunities mean that traditional operating strategies and experiences do not necessarily suffice, just as the development of very complex simulation models is resource intensive. So there is a need for methods that in themselves can handle and summarize the amounts of data and the data patterns that, for example, can provide optimal operation or which give indications when the system does not function optimally.

Advanced analytics is a broad range of analytics that are intended to provide businesses with a better and more valuable insight into their data (Big data). Automating and sophisticating by for example using pattern recognizing algorithms (e.g. machine learning) has been identified as a great potential for improving the DH system, as formulated in the Digital Roadmap for District Heating and Cooling [1].

These techniques include machine learning, data mining, predictive analytics, location analytics, big data analytics and location intelligence. Especially Machine Learning (ML) is found relevant in relation to DH and the further challenges for ULT DH, and ML will be the main focus for the following study and developments. However, also simpler regression models will be tested and compared to ML-models.



4. Machine Learning (ML) as a solution tool for DH and ULT DH

Machine learning is a fast-developing tool that allows you to handle large and complex systems with many parameters and data, with different desires for efficiency. It is a branch of artificial intelligence, and defined by Computer Scientist and machine learning pioneer Tom M. Mitchell in relation to his book Machine Learning (1997) [2] as:

“Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience.”

4.1. Introduction to Machine Learning (ML)

In recent years, Machine Learning (ML) has become one of the most used techniques when modelling relationships between different parameters. It allows, among others, researchers, data scientists and engineers to produce reliable, repeatable decisions and results and to uncover hidden insights through learning from historical relationships and trends in data that the more traditional analytic methods would not be able to grasp. The basis is pattern recognition and, e.g., imitation of neural networks in the human brain, and is strongly entangled with the concept of artificial intelligence (AI). Although ML has existed for decades, one of the major reasons behind its present success is the strong development in computational capacity of modern computers that enables very fast calculation and thereby promotes the possibility to better “train” ML-algorithms efficiently without the need of special and unique computer facilities at hand.

Numerous examples of the widespread application of ML exist today. They count exotic areas such as automatic voice- and face recognition, self-driving cars and artificial players in computer games. Daily tasks, such as reading the addresses on the letters at the post sorting office, ensuring it arrives at the right recipients, as well as the spam filter in your inbox, are based on ML.

Popularly phrased, what distinguishes ML techniques from other statistical methods, is that the algorithms are not specifically programmed; instead the purpose is to let the algorithms learn and improve from experience in order to obtain a well performing algorithm. In “Supervised learning”-techniques the algorithm uses labeled data to train models that are used for a multitude of different problems, the most common ones being classification and regression modelling. In other words, based on a set of training data it learns the general patterns and relations and uses this experience to make a prediction.

The predictions of a ML algorithm will not be fully accurate, and the precision must be evaluated on a test set of data. Along this line, ML falls in the category of “Soft computing” that differs from conventional (“Hard”) computing in that,



unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation. For example, an average of 95–99% accuracy typically achieved in the case of a so called “artificial neural network” (ANN) when compared with numerical results [3]. In this respect, ML and soft computing resembles the human mind.

Similarly, ML-algorithms cannot give a result which is outside its training area. If an algorithm is trained to distinguish between two classes of mails, regular and spam, it will not be able to distinguish between mails concerning botany or ornithology. This means that any ML-algorithm will only provide a sensible output in the field in which it has been trained.

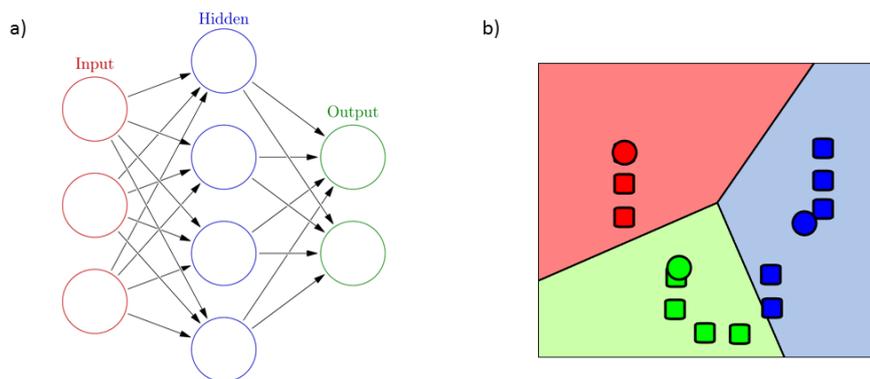


Figure 1: Illustration of two common ML techniques: a) An artificial neural network (ANN) and b) K-means clustering.

Sources: Reproduced without changes from (a) Glosser.ca and (b) Weston.pace, both licensed under the [Creative Commons Attribution-Share Alike 3.0 Unported license](https://creativecommons.org/licenses/by-sa/3.0/).



4.2. ML techniques and tools

The concept of ML covers a wide variety of specific techniques with the common denominator that they all rely on large amounts of data in order to generate a precise model.

Two examples are shown in Figure 1, illustrating the idea behind an artificial neural network (ANN) (a) in which input data is processed through hidden layers to produce an output in the end – a structure inspired by the working mechanisms of the neurons in the human brain. To the right (b) another approach through classification is shown, visualizing how a ML algorithm can recognize patterns and groups of data points to extract information from, at first sight, non- or less structured information.

Many of these techniques can be effectively used for the purpose of optimization. Neural networks and related predictive techniques are highly valuable, for instance, in the financial and economic sector to help predicting prices and work out prognoses etc. without a basis of solid and known model equations. At the other hand, classification methods can be effectively exploited to draw out anomalies, i.e. detection of abnormal behavior, in large amounts of information. As an example, this can be used to identify problems and start maintenance before a vital component fails in a windmill.

4.2.1. Specific techniques

Below, a handful of selected techniques relevant to the work performed in this report, or used by the references cited, are briefly described:

- **Linear regression models (LRMs)** are used to fit a predictive model to an observed data set of values from response and explanatory variables. LRMs include both unique variable regression with one input and output variable respectively, and multiple variable regression that includes more than one input variables. LRMs are not by definition machine learning techniques, however they might be applied together with for example decision trees (DT, see below).
- **Artificial Neural Networks (ANNs)** are a supervised information processing algorithm inspired by the way biological nervous systems, such as the brain, process information. The key element is the novel structure of the information processing system, comprising a number of so-called “Hidden layers”. It is composed of many highly interconnected processing elements (neurones) working together to solve specific problems. ANNs, like the human brain, learn by examples. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process.



- **Decision Trees (DTs)** is a type of supervised learning algorithm that can be used in both regression and classification problems. The technique is based on a number of questions that can partition a data set and split the data. Many divisions of the data over such decision nodes will form a “tree“ that may eventually (if the correct questions are made) constitute a predictive model.
- **Cluster analysis** or **clustering** is an unsupervised method that groups a set of objects in such a way that objects in the same group (cluster) are more similar to each other as judged by specified criteria than to those in other groups. It is a common technique for statistical data analysis and used in many fields. Various algorithms can be applied, for example by **k-means clustering** which is an iterative process that converges towards an final division of the input data, however with a build-in randomness due to the (random) choice of initial groups that forms the starting point for the iterative process.
- **Support-vector machines (SVMs)** are supervised learning techniques that can be used for both classification and regression analysis. It is a discriminative classifier in the way that it finds an optimal hyperplane which categorizes new data examples from training on labelled training data. The algorithm can be tuned in different ways, for example by using different “Kernels” that describe both linear and non-linear data separation functions.



4.2.2. Software tools and programming languages

Various software tools or programming languages are available for application of ML. The choice of tool depends on many factors such as preferred programming language, license- or freeware, degree of manual programming vs. ready-to-use packages as well as the options for distribution of worked out solutions and programs (e.g. export of executable files).

The most relevant ones in the present context are briefly presented below:

- **Matlab®** (see <https://se.mathworks.com/products/matlab.html>): A numerical computing environment and programming language developed by MathWorks. It allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with other programs. Matlab® is not a freeware, but is very widely used at universities and in relation to research and engineering.
- **R** (see <https://www.r-project.org/>): A language and environment for statistical computing and graphics. It provides a wide variety of statistical tools (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering etc.). One of R's strengths is the ease of use and its availability as Free Software.
- **Python** (see <https://www.python.org/>): According to its homepage Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures combined with dynamic typing and dynamic binding make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy-to-learn syntax emphasizes readability and therefore reduces the cost of program maintenance.

According to some references, Python is the most used language for application of ML techniques used commercially, whereas R is the second most used. Matlab® alone has, however, more than 3 million licensed users worldwide.

In this work Matlab® has, because of its research character, been used for the ML activities, while R is the basis for the research carried out in section 7.2.2 concerning Linear Regression Models (LRMs).



4.3. ML perspectives in the context of DH

Inspired by the successful integration of ML in many other areas, it is beginning to draw attention in the DH sector as well. Albeit only few examples of actual application of ML in the context of DH exist today, it has an obvious potential as a component of tomorrow's heating networks that can be integrated through the fast and inevitable development leading to the digitalization of the supply systems.

One of the large potentials of ML is **to predict heat loads** from the expanding data bases of customer data and operational data in combination with weather forecasts, national holidays, weekday, etc. to optimize and plan the heat production, thereby lowering heat loss and handling peak loads. In combination with economic data, prices of electricity, gas, etc., as well as integration of sustainable energy, which is not always reliant/constantly available, it will also be an efficient tool when integrating DH with the remaining energy sectors to optimize the overall use of resources, both environmentally and economically. This is of even higher interest, when more sustainable energy is integrated into the market, since electricity is not always needed, when the wind is blowing.

Another large potential is to use intelligent algorithms in **fault detection** on the basis of retrieved customer data and other data from the network. For example, such a tool will be able to identify leakages, inefficient heating systems or errors stemming from failure related to single components. It can thus apply both at the scale of individual components and the overall installation where it can be associated with the operation and settings of the sub-station. Based on real-time remote reading of smart meters distributed in the customer network, the software will typically be able to monitor data on the fly and significantly reduce the time passed from a fault appears to its detection and repair, which has until now been based on manual inspection.

Manual inspection is a time-demanding activity, and because of this, substations with the largest heat demand in the DH system are normally prioritized when performing the analysis, leading to a large share of poorly performing substations going unnoticed. Accumulating over the full DH network, this could have a significant effect on the energy efficiency. Identifying and correcting many small problems in a DH area can potentially help save energy and improve performance of the production facility.



4.4. Pioneering examples of exploitation of ML in DH

In Denmark, Sweden and several other countries, many smart meters are currently being installed or will be installed in the near future. Databases hosted by the major supply companies or at service companies and facilities comprising customer data with supply- and return temperatures, flow and energy consumption, typically with an hourly readout, already exist. Together with local weather information and forecasts, this pool of data has started to attract attention in the light of optimization. Methods have been demonstrated addressing how the information can be exploited together with ML and statistical methods for more than the original purpose of billing.

4.4.1. Heat demand prediction

The use of ML algorithms to predict heat demand has been intensively investigated in the buildings- and electricity sectors but are also strongly progressing in DH. Artificial Neural Networks (ANN) and Support Vector Machines (SVM) are two of the most widely used techniques [4][5], but also, e.g., Decision Trees (DT) are suitable to forecast energy consumption in the context of DH [6].

Recently, E. Saloux et al. [6] found that ML algorithms, including Decision Trees (DTs), can significantly improve the accuracy of predicted heat loads by incorporating the effect of additional influencing factors (e.g., time of the day, day of the week, solar radiation, etc.). Potential implementation of such models in this example is highly desirable to control integrated heat storage and solar thermal collectors to optimize the overall energy efficiency.

Integration of national and school holidays into ML models have also shown improvements in the predictions [7] and may be considered to optimize forecast models.



4.4.2. Fault detection.

S. Farouq et al. [8] performed an analysis of outliers among smart meter datasets from 800 multi-dwelling buildings by calculating and comparing a defined parameter that estimated the thermal energy demand response with respect to change in outdoor temperature. The strength of the relation between energy demand and outdoor temperature was shown to hold important information about operational efficiency of a substation. It was concluded that a robust regression method in combination with the ordinary least square's method can provide reliable estimates on the operational efficiency of DH substations.

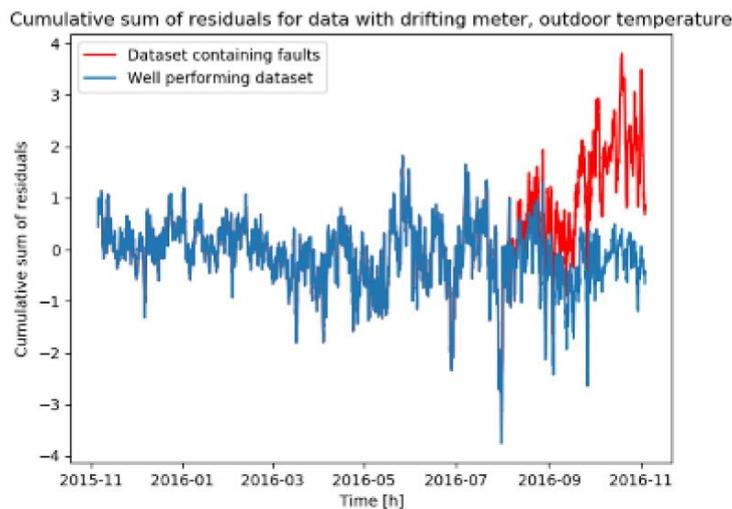


Figure 2 Identification of an artificially induced fault on a single sub-station by ML-prediction of the mass flow.

Source: Figure adapted from [9] under the [Creative Commons Attribution-NonCommercial-No Derivatives License \(CC BY NC ND\)](https://creativecommons.org/licenses/by-nc-nd/4.0/).

In a different approach, S. Månsson et al. [9] demonstrated a ML approach to fault detection using smart meter values recorded hourly during one year from a substation in Sweden. They showed that prediction of the flow and deviations from predicted values could be effectively used to spot artificially induced faults that were made up to mimic e.g. valves that are stuck, fouled heat exchanger areas in the heat exchanger and malfunctioning temperature transmitters, but also wrong settings in the control system.



4.4.3. Intelligent control

In general, the application of intelligent control in DH is a promising technology that is currently receiving strong attention, both from the side of production companies and in relation to supply. For example, digitalization was a highly present theme at the DH and cooling conference DHC2018 held in Hamburg Sept. 2018 (<https://www.dhc2018.eu/>).

Furthermore, in the EU H2020 project STORM [10-12] the aim is to develop a specific controller that enables DH suppliers and distributors to maximize the use of waste heat and renewable energy sources in DHC networks. This action incorporates the use of ML and the results have been applied at two demonstration sites in Sweden and the Netherlands.

During recent years, Danish Technological Institute (DTI) has been working with ML in several contexts. In activities focusing on DH, DTI is currently receiving support from the national DH association Dansk Fjernvarme (<https://www.danskfjernvarme.dk/english>) to promote a smaller project on utilization of smart meter data for optimization purposes by means of ML. Encouragingly, DTI was recently given the opportunity to present the project at the annual meeting in October 2018 of Dansk Fjernvarme [13], gathering participants from the approximately 400 individual Danish DH companies, and found a strong interest. Especially small and medium-sized companies not having the resources needed to perform an extensive analysis of data themselves were interested in the future outcome of these activities.



5. Challenges of measurement data for ML training and service use

A fundamental challenge using advanced analytics for optimization with smart meter data is the quality and validation of the recorded data. This includes both the actual measurement accuracy and resolution, as well as the solution for data transmittance and storage.

Depending on the circumstances, it might be necessary to discard certain pools of data for ML purposes, interpolate missing values or modify algorithms according to the data format and quality. As an example, interpolation was estimated to be required and carried out in data sets of one-minute resolution from the distributing company in Aarhus (DK) in a previous project [14] with the purpose of analyzing different strategies of optimizations for improving the operating conditions in DH network.

Furthermore, it is important to keep in mind that smart meters are constructed for billing purposes and not optimized for all kinds of data exploitation. For example, whereas the temperature difference (cooling) of an installation is relatively well measured due to selective pairing of probes for supply- and return temperature measurement, the absolute temperatures suffer from large uncertainties and compromise direct use in calculations and comparison.

5.1. Evaluation of data requirements for ML use

When recording and saving smart meter data from a DH network there are several important considerations to address. One must make a choice of time resolution as well as which parameters to record (real-time values or time averages, accumulated values etc.). Furthermore, requirements for data quality must be considered for specific analysis purposes (completeness and reading precision).

However, these decisions are rather difficult to make since actual business cases and examples of exploitation of smart meter data as implemented at supply companies are currently lacking. Pioneering studies give indications (see for example the literature examples in section 0 above), but not enough to provide a solid basis to take decisions about how much to invest in data collection infrastructure (including the actual set-up of smart meters in the network). While the purpose of billing is obvious, but does not impose many requirements on the data, it is still uncertain exactly which methods and data input formats that will generate most value.



5.2. Measurement frequency

The measurement frequency can be chosen in a wide range from seconds to daily values. Lower time resolution, i.e. beyond daily values, is not going to provide enough details for most purposes of exploitation of smart meter data.

On the other hand, collection of data at the highest time resolution is expensive in terms of data storage capabilities and computation time. Selecting a suitable measurement frequency is therefore a compromise between analytics potential and low expenses (easy-to-use and cost effective).

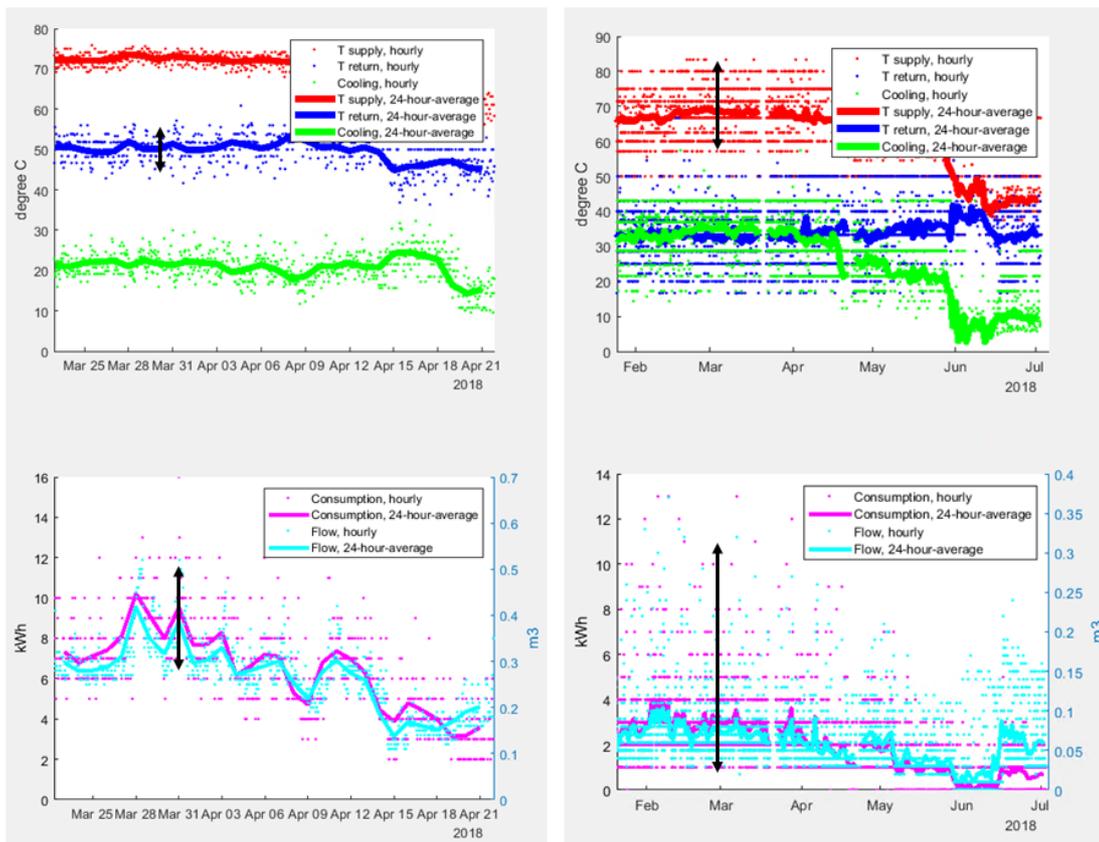


Figure 3: Examples of 1-hour values compared to 24-hour-averages: Resolution and quality. Data from AffaldVarme Aarhus, Geding, Denmark.

Arrows indicate the variations in data from: (LEFT panel) an installation with relatively high consumption and (RIGHT panel) an installation with low consumption. Note the finite resolution for especially consumption.



A measurement frequency in the range of seconds offers possibilities of very detailed analysis [15]. Single components in a home can be identified through the data patterns, e.g. the use of domestic hot water. The high resolution allows for a deep analysis of the installations, as well as it can be used to evaluate the heat loss and state of the branch pipe when used in combination with control of the system (for example opening and closing a hot water tap).

However, 24-hour values still yield enough information to overall evaluate an installation. If analysis is performed for example weekly it renders the possibility of surveilling the network for appearing faults while keeping the calculation cost and complexity of the database at a minimum. This has the advantage that a data basis is existing at most suppliers with installed smart meters and means that often the full network comprising 10,000 or more installations and households can be analyzed periodically altogether without large expenses in calculation force and time.

Hourly data is often considered an appropriate choice since it allows storage from several years of operation from an entire network and a enables relatively detailed analysis that includes the daily consumption patterns. At the same time, one approaches “real-time” surveillance of the network with hourly values.

5.3. Data communication and resolution of stored data

Data communication can be set up using a variety of options. Some suppliers use drive-by solutions with vehicles for data collection whereas it is now starting to get more common with wireless data-transfer via a number of transmission points in the net.

The resulting completeness of data is important to evaluate. For hourly values, it can reach 100 % in geographical areas with optimum transmission conditions, however on average in a middle-sized DH supply company it is usually much lower. This parameter (“completeness”, i.e. percent of received and stored data points) will to some extent determine which analysis methods to be used. In many cases, the best option can be to extract daily values as a more robust starting point, supplemented by hourly values where available.

As indicated in Figure 3, another issue is the resolution of values in the case of hourly values where, typically, the consumption is stored in whole kWh's. This might cause a problem when the consumption is low (RIGHT in Figure 3, for example taking the values 2, 3 and 4 kWh, resulting in a highly discrete dataset. The problem can be circumvented by calculation of 24-hour-averages instead, depending on the purpose of analysis and visualization.

In this report, tools working with both daily values (section 0) and hourly readout (section 0) have been developed. At the same time, the data used for test of algorithms includes stored data of different levels of quality in terms of resolution.



6. A framework for advanced control procedures

ML has a potential use for several purposes in the RELaTED project. This section describes the general possibilities of applying intelligent analysis in the light of ULT DH, as well as the developed concept behind the DH Doctor and DH Autotune tools.

6.1. Potential of ML for ULT DH

Overall, on locations where smart meter data at 1-hour resolution or better, the data can be used to model single parameters in the system, e.g. the mass flow at the level of individual substations or the heat-load of the entire system. The approach combines smart meter data with weather archives and forecasts. It has the advantage that it is merely based on actual data and measurements in the system (opposite calculations based on certain model equations). Thus, it can continuously adapt the model through a “learning” algorithm to unforeseen changes and with time improve its knowledge and precision related specifically to the actual DH network, including the case of ULT DH.

Predictions of single parameters in an Ultra-Low Temperature DH system by ML algorithms have been exploited to develop the **DH Doctor** tool. Comparing predicted values and actual measurements reveals if the system is running well, or if parts are on the verge of failing. In case of faults, the system will change its data pattern and deviate from the expected parameter output values which will be identified by the ML algorithm. This will enable a fast action or repair of faulty parts and ultimately result in less down-time and lower costs in the long perspective. On top of prediction techniques, “recognition” or “classification” methods learning the typical and “healthy” data patterns from smart meters at individual substations in the ULT DH system can contribute to the effective identification of errors.

Simulations based on a prediction model compared to the actual heat load, production and other measured parameters (temperatures, mass flows etc.), is also a basis for yet another tool: **DH Autotune**. The basic idea is an algorithm that evaluates the behavior of the system with the purpose of suggesting actions in the drift. Specifically, for ULT DH, such a tool will be of value in the process of lowering the supply temperature and rise an alarm as soon as deviating data patterns are recorded. Based on the alarm and further diagnosis from the data patterns, relevant actions include, for example, rising the temperature or supplying with additional energy from other sources. An effective way of monitoring the network while testing and exploring the limits of lower temperature is essential, and the intelligent approach exploiting ML and actual measurements is a promising candidate. Furthermore, DH Autotune aims at forecasting the heat load based on a weather forecast.



Although the natural starting point is to take stored data as input and react on this, it's important to underline that the final target and perspective along the line of digitalization and the use of intelligent software in DH is a real-time operating tool that collects data continuously and responds to the network by, ultimately, controlling and optimizing parts of the operation. Not least in the case of efforts to push the limits of conventional DH towards ULT, the higher sensitivity of the network and complexity in combination with additional heating sources, the aid from intelligent software is of great potential (see the Digital Roadmap for District Heating & Cooling [1]).

6.2. DH Doctor and DH Autotune concepts

In the overall picture of a so called smart thermal grid [16] in the 4th generation DH systems, data from smart meters play a significant role retrieving relevant operational information to the DH supplier, as sketched on Figure 4. Such future systems will be complicated by the major challenge of utilizing low-temperature heat sources and the interaction with low energy buildings.

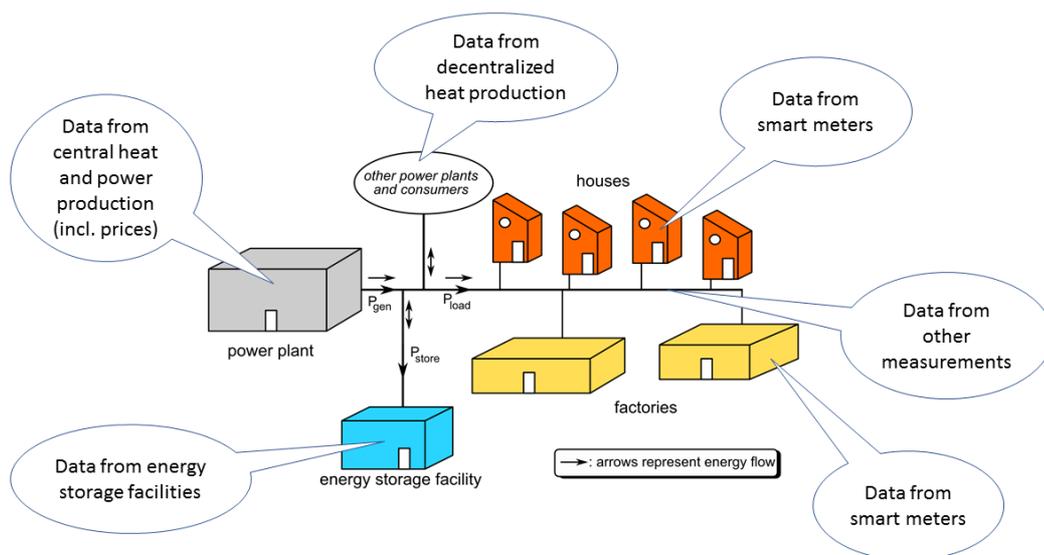


Figure 4: Sources of information (data) in the concept of a DH smart grid.

Source: Modified from Wikichesterdit: “Simplified electrical grid with grid energy storage (without distributed generation)” licensed under the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/).

Based on these sources of information as input, the overall concept behind DH Doctor and DH Autotune is aimed at



- providing network surveillance (fault detection on installations and real-time feedback from the network while adjusting operational parameters, e.g. lowering the supply temperature)
- providing data-driven heat load forecasts
- managing and optimizing heat production in low temperature districts with several independent energy producers
- optimizing heat production according to current energy prizes, local production of renewable energy and updated consumption patterns.

In this report, we describe the first steps to realize this concept. Currently, the DH Doctor and DH Autotune tools are restricting themselves to exploit only smart meter (as well as weather) data. However, data from central and decentralized heat production sources and other data inputs, e.g. energy prices for production cost optimization and savings, are planned to be integrated in future versions.

The current versions of DH Doctor and DH Autotune address the first points listed above, i.e. surveillance of the DH network aimed to identify fault occurrence (DH Doctor) as well as recommendations regarding set-points for production and temperature adjustments (DH Autotune). While the output from DH Doctor will be used as a basis for repair of malfunctioning installations, the DH Autotune provides a basis for overall energy optimization via adjustments in the power production and temperature in the distribution network.

Both tools incorporate intelligent algorithms (machine learning, ML) to enable a high degree of adaption to the actual network and its characteristics. Intelligent algorithms in this setting trains the software program to understand the current system under surveillance and to point out relevant information regarding individual parts and substations.

The output and use of DH Doctor and DH Autotune at this point consists of guiding information to the staff to enable better decisions in the daily operation. An actual automatic control of the DH system has not been addressed yet, since its applicability is still restricted due to the current level of technology which has not yet come to standard feasible possibilities regarding remote control of, e.g., pumps, units and installations.



7. Development of tools and algorithms

In this chapter, the developed tools integrated in DH Doctor and DH Autotune are described. The work has been performed by DTI using Matlab® unless otherwise stated (section 0).

Overall, DH Doctor and DH Autotune have been developed on two separate sets of smart meter data. DH Doctor was developed using data from Geding in Aarhus, Denmark, characterized as an ultra-low temperature network (ULT DH) with data of 24-hour resolution and a small number of installations (24). This data was supplied by the distributor AffaldVarme Aarhus which also provided access to its data base comprising stored smart meter data from ~56.000 traditional DH installations, of which an area with 490 installations has been used for this project. In the development of DH Autotune we used data from the distributor Fortum Tartu, Estonia, of hourly resolution, comprising 43 buildings.

This choice of training data has been made primarily according to data availability, especially in the case of ULT DH data (as supplied by AffaldVarme Aarhus for the case of Geding), which is currently hard to obtain. Overall, larger sets of data comprising more installations and longer time periods would have been preferable and improved the training of the algorithms, however it does not impose any limitations of use regarding the developed tools.

Table 1: Overview of algorithm features and data used for development of DH Doctor and DH Autotune.

	DH Doctor	DH Autotune
Supplier/district	AffaldVarme Aarhus Geding, Aarhus, Denmark	Fortum Tartu Tartu, Estonia
Number of installations	24 ULT DH 490 LT DH	43
Time resolution¹	1 day (24 hours)	1 hour
Methods implemented	Clustering (ML ²) Decision tree ensembles (ML) Statistics	Linear regression models Neural Networks (ML) Decision tree ensembles (ML)
Application to:	Data from many substations compared	Individual substations
Data input	Historical smart meter data Historical weather data	Historical smart meter data Historical weather data Real time recordings (hourly) Weather forecast
Purpose	Fault detection Outlier detection	Fault detection Load prediction

¹ Resolution applied in analysis. 24-hour values were constructed from hourly values of lower resolution.

² ML: Machine Learning technique



The division of data between DH Doctor and DH Autotune was organized such that the tools cover both input consisting of daily values and demonstration on actual ULT DH data (DH Autotune with data from Geding) and analysis exploiting hourly values and conventional DH data (DH Autotune with data from Tartu).

As summarized in Table 1, different types of algorithms were implemented in the two DH tools, including several machine learning techniques (ML). Also, their applications have different targets regarding data basis, namely analysis of a network consisting of, preferably, many installations (up to in the order of 10,000 or more) for DH Doctor, whereas DH Autotune analyzes individual substations in parallel.

DH Doctor and DH Autotune share the purpose of recognizing alarming signals regarding faults on installations in the network (fault detection), but have different algorithms implemented which complement each other in terms of methods and requirements for input data (daily/hourly time resolution).

Furthermore, DH Autotune includes a consumption prognosis algorithm that predicts the heat load of individual substations/whole network via hourly recordings and a weather forecast.

In the following sections, the tools, methods and test results will be described one by one.



7.1. DH Doctor: Fault detection based on daily readings

The aim of the DH Doctor tool is to enable a daily or weekly assessment of the performance of individual installations in a DH network. In particular, the tool is targeted situations where the number of substations exceeds what would be possible to inspect by looking manually at the available meter data.

Demonstration of the program was carried out on a small number (24) of ULT DH installations in Geding (Aarhus, Denmark), but data from a larger number of traditional DH substations in the same network have also been included to demonstrate some of the principles and methods.

It should be emphasized, that ULT DH data from Geding with merely 24 installations is not a sufficient amount of data (number of installations) for the actual purpose of DH Doctor, however it serves well as an illustrating example. Much more accurate results will be harvested when a larger number of smart meters from several hundreds or thousands of installations are analyzed together.

The construction of algorithms for DH Doctor has benefitted from previous work on analysis of smart meter data from Aarhus (AffaldVarme Aarhus, DK) in two Danish projects conducted by DTI.

7.1.1. Training and test data for the development of DH Doctor

Amount of data: DH Doctor is based on use of all available historical meter data up to date and evaluates the individual installation on this background. As such, the resulting outcome of the algorithms is not a snapshot in real time, but an overall assessment over the period with available data. However, the evaluation period can be restricted to e.g. the latest week to perform an assessment of the current operation.

Time resolution: The algorithms behind the DH Doctor tool are based on daily average values of primarily the supply- and return temperatures, mass flow and energy consumption. Advantages of daily average values here, as compared to measurements of higher time resolution, include that the size of data to handle is minimized, algorithms are executed faster and a data set covering long periods and all seasons are relatively easy to establish. Almost complete data sets without many missing dates and wrong direct meter readings can be retrieved from most installations in networks equipped with remote meter readings. With hourly values, the precision of the tool might be improved by providing more training examples as input, however it also renders the model more complex and with the issue of frequent missing values (see section 5) and time stamps not exactly matching hourly reading which calls for additional data preparation



maneuvers. Hence, daily averages are considered the best starting point for the widest possible applicability and robustness of the algorithms behind DH Doctor.

Data access: The DH Doctor tool does not require any specific data infrastructure such as type of database, but is built to take a standardized input consisting of two elements:

- Smart meter data in a specified format containing as a minimum:
 - supply- and return temperatures
 - energy consumption
 - mass flow

as well as the meter number and date for every data point. From these parameters, we calculate temperature difference (cooling), day of week, etc.

- Weather data: The local outdoor temperature (daily average).

In the present context using data from Geding (Aarhus, DK), data is stored in a Microsoft SQL database and exported in the specified format, however it can also be saved and exported from Excel or other software programs.

Data preparation: The input consisting of daily averages of meter readings is synchronized with the local outdoor temperature as well as time of week (weekday). Due to ubiquitous error readings, thresholds are applied to remove obviously wrong values, including negative temperatures, etc. Thus, the resulting data set will contain a fraction of missing values stemming from this cleaning process, as well as from originally missing meter readings. As a consequence, the “clustering” method applied in the DH Doctor tool was modified by manual programming since the available algorithm in the analysis software Matlab® (and in general) only handles complete sets of features (data input series), i.e. no missing readings.

Based on the prepared data input, the DH Doctor aims to identify and quantify anomalies or outliers among the substations analyzed. The tool applies two different machine learning methods, the first one being clustering.



7.1.2. Clustering

This algorithm compares time series of selected parameters between all installations to identify groups with similar data patterns using the unsupervised ML technique “clustering” (see section 4). The program walks through the following steps:

1. Installations that fail to resemble the main groups are often located in groups (clusters) consisting of one or few members and are identified on this background as anomalies.
2. Apart from noticing these anomalies, the distance to the average of the nearest cluster is additionally calculated for every installation. The installations with the largest distances, i.e. worst fit to main clusters, are noticed as well and regarded outliers.
3. Due to the arbitrary starting point of random initial clusters, the clustering algorithms is run n times to produce enough statistics of detected anomalies and outliers for a robust analysis (points 1 and 2 above).

The method is illustrated by example in Figure 5. A visualization has been used such that the line width of curves corresponding to the different cluster centroids (averages) is proportional to the number of installations belonging to it. Hence, it immediately shows the larger groups of similar, well performing installations (light and medium dark blue traces in this case). Notice that cluster 6 has a very small number of members that will be classified as “anomalies”. The number of 6 clusters has been chosen to better illustrate the method here, however a larger number might be chosen for actual evaluation of a whole DH network.



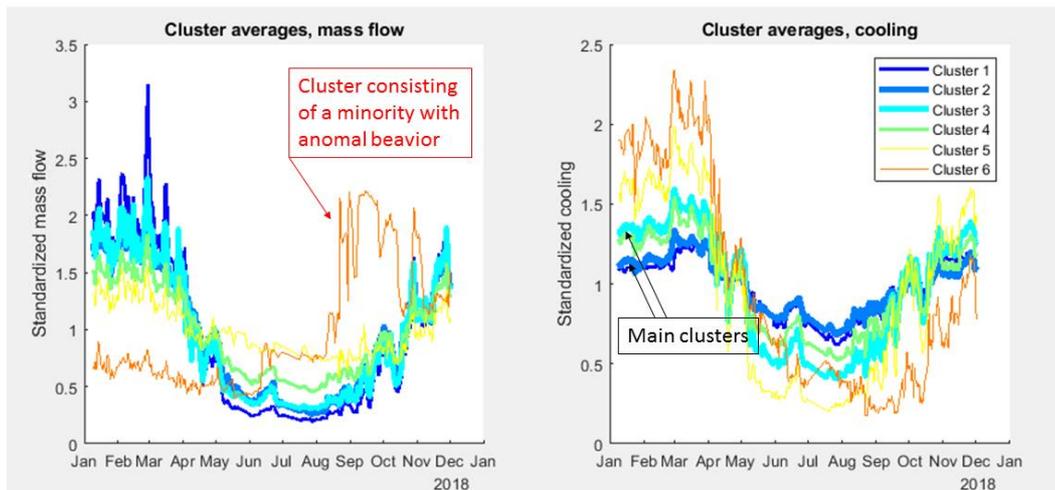


Figure 5: Illustrating example of clustering over 490 installations (data from Aarhus, Denmark) into 6 clusters.

Clustering based on standardized mass flow and cooling over approximately one year.

In Annex 1: DH Doctor (Figure 21 to Figure 23), two examples of clustering are furthermore given to illustrate the method and to evaluate it on an ULT DH network. Here, an analysis of the small ULT DH area comprising 24 installations in Geding is compared to a larger group of 460 single family houses from another district in Aarhus.

The raw and clustered data is shown in Figure 21 and Figure 22 (see the Annex 1: DH Doctor), using 4 and 15 clusters, respectively. In the raw data (upper panels) colors corresponding to the assigned clusters are used. 15 repetitions of the clustering processes yield the numbers of detected anomalies shown in the histograms in Figure 6. Noteworthy, a certain overlap by the two different anomaly criteria (upper/lower panels) can be seen, i.e. many meter numbers have large numbers of counts in both histograms for “Detected anomalies” (clusters with one or few installations) and “Worst fit to nearest cluster”. On the other hand, there are also meters with differences emphasizing that the criteria also hold different information and advantage should be taken of using them together. The algorithm later overlaps the two groups and performs a final prioritization of most important outliers depending on user input regarding the number of “most urgent outliers” that the user wishes to extract.



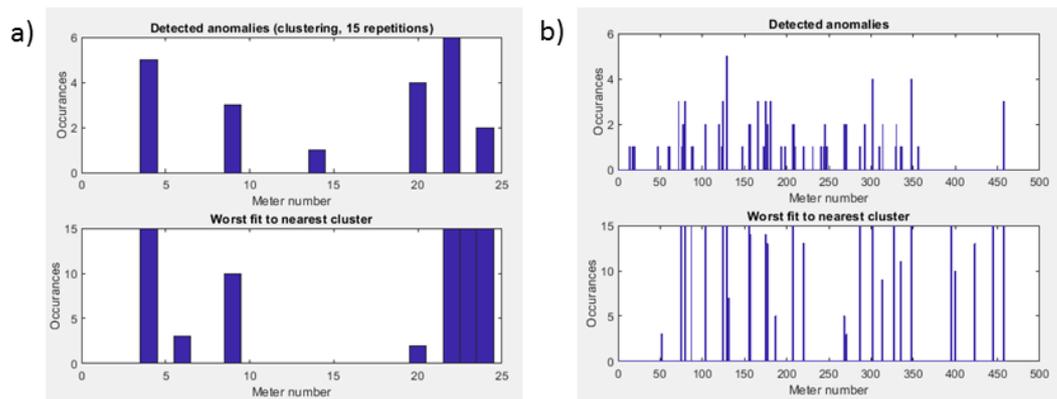


Figure 6: Statistics over detected outliers in clustering using two criteria.

a) Geding, 24 installations, b) 460 installations from another district in Aarhus.

Meter data for the detected anomalies, as well as for any specified installation, is finally displayed using the format shown in Figure 7. In a) an installation categorized as a building with “usual performance”, i.e. from one of the main clusters, is shown and compared to one detected as “anomal” in b). The reason for this classification is evident from the data pattern that indicates a fault appearing in late April 2018 with a vanishing cooling as a consequence, calling for action. The parallel visualization of the parameters in time series furthermore provides the option for diagnosis to some extent, in this case pointing to a problem with the mass flow associated with the fault.

In the examples given, the parameters mass flow and temperature difference (cooling) have been used as input, however any other combination or single parameter can potentially be used (specified in the program) and might reveal different kinds of information. For example, clustering by “supply temperature” might provide a very direct overview of heat distribution and deviations compared to expectations that will be useful to know.



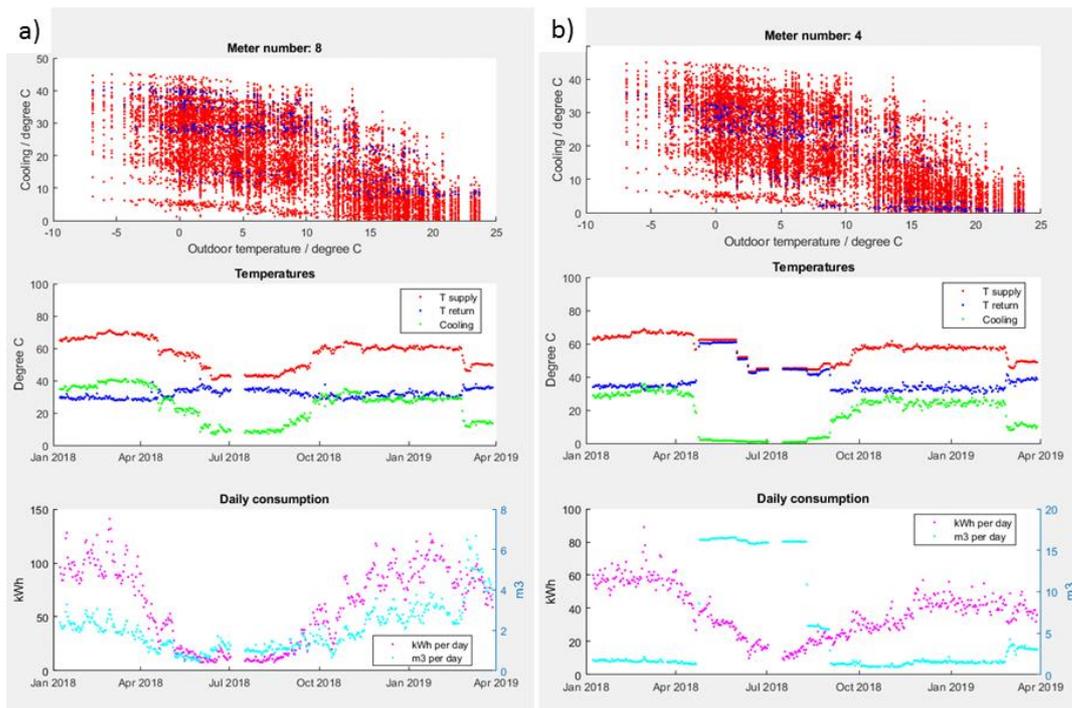


Figure 7: Temperatures and consumption: Visualization examples from Geding, Aarhus.

a) An installation assigned to a main cluster and b) an outlier cluster.



7.1.3. Predictive modelling

The second applied machine learning technique in DH Doctor is an ensemble of decision trees (DTs), a technique that is readily available and supported in Matlab®.

Using ensemble decision trees (DT), a selected parameter (e.g. flow) is modelled based on several other input parameters. The algorithm learns to predict the parameter based on experience from the total pool of data from all installations and over the full time period (without knowing the true, but unknown, physical relations). The procedure can be divided into the following steps:

1. The program utilizes a bagged decision tree algorithm accessed in Matlab® to generate a data-based model predicting the chosen output parameter.
2. With the trained model at hand, deviations between measured values and model predictions are used to identify installations with anomalous behavior.

In Figure 8, examples are given for a model predicting the mass flow taking an input consisting of *supply temperature, energy consumption, outdoor temperature* and *day of week*. An installation displaying a normal behavior is shown in the top-left panel (Meter No. 8) where agreement can be observed between predicted and measured values. The three remaining examples show installations with a measured flow that yields significantly larger values than expected. In one case (Meter No. 24) the observed fault appears to be rectified during November 2018, however for Meters No. 5 and 20, the high flow rate does not disappear at any point. This might suggest that further inspection of these installations is needed.



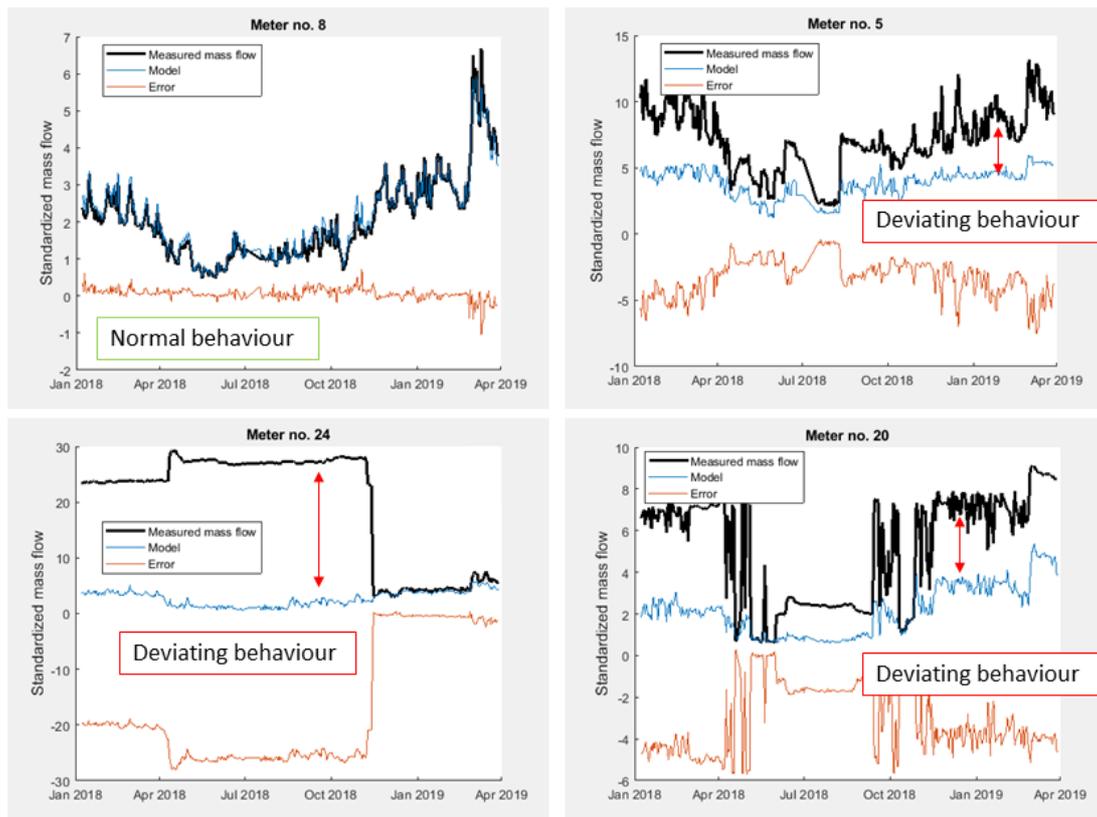


Figure 8: Model predictions and measured values for standardized mass flow.

Data from four particular installations in Geding. Notice the critical deviations marked by red arrows indicating too large values of flow compared to the expected values (model predictions).

Quantification of the error between model and measured value is additionally an option to evaluate if discrepancies are at a critical level. The Mean Absolute Error (MAE see definition in Section 7.2.1) indicates the average size of the residuals and can, hence, be used to prioritize inspections. Another option is to use an average “overestimate” that sums the errors and reports on over- and undershooting in general. An overestimate might provide a better interpretation of the type of fault, e.g. whether the flow, cooling or other output parameters is too large or small compared to the average well-performing installation.



7.1.4. Calculated parameters

In addition to the implemented ML methods (clustering and decision trees), DH Doctor supplements the analysis by calculating descriptive parameters for every installation, most importantly:

1. The temperature response: The relationship between outdoor temperature and temperature difference (cooling) for every installation (calculated for outdoor temperatures below 10 °C). In general, a hypothesized linear relation with better cooling at lower temperature is expected [17].
2. The linearity of the temperature response curve: The average value of the absolute distance from every point to the linear regression of the temperature response curve. A large deviation from the hypothesized linearity often indicates an (unresolved) error and a threshold is defined above which occurrences are regarded as outliers/anomalies.

DH Doctor displays statistics for all analyzed installations. In this panel, histograms of *consumption*, *cooling*, *slope* and *linearity* of the temperature response curves are given (see Figure 23 in Annex 1: DH Doctor).

We conclude that a large number of installations (comparing 24 and 490 in Figure 23) is required to form a satisfactory statistical output. Outliers regarding *average cooling* and *linearity* of the temperature response curve, etc. can be extracted most robustly when having smooth distributions. Examples of such outliers are indicated for the dataset of 460 installations by red arrows and vertical lines that represent thresholds specified by the user (Figure 23 in Annex 1: DH Doctor).



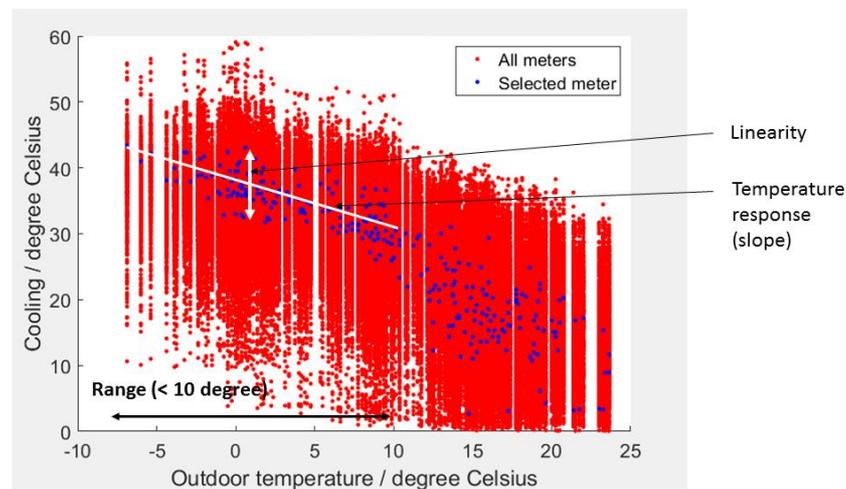


Figure 9: Visual explanation of the calculated parameters “temperature response” and “Linearity” associated with the temperature response-curve.

One specific group of outliers detected using this statistical approach is a number of installations displaying an “opposite” temperature response, i.e. a positive slope, whereas the expected behavior is a negative slope corresponding to larger cooling with lower outdoor temperature. An example can be seen in Figure 9 a) (installation 2). An opposite temperature response is typically seen for a small percentage of the installations (e.g. none examples in Geding) and can potentially result from certain faults. Hence, this group of outliers represents a target for further inspection.

Another example in Figure 10 is highlighted due to its deviation from a straight line. This anomaly will result in a value characterizing the linearity outside the normal range and points to a potential problem occurring at the installation. In the zoom, Figure 10 a) a certain group of points is seen to deviate from the linear behavior due to unexpectedly low cooling which calls for a further diagnosis.



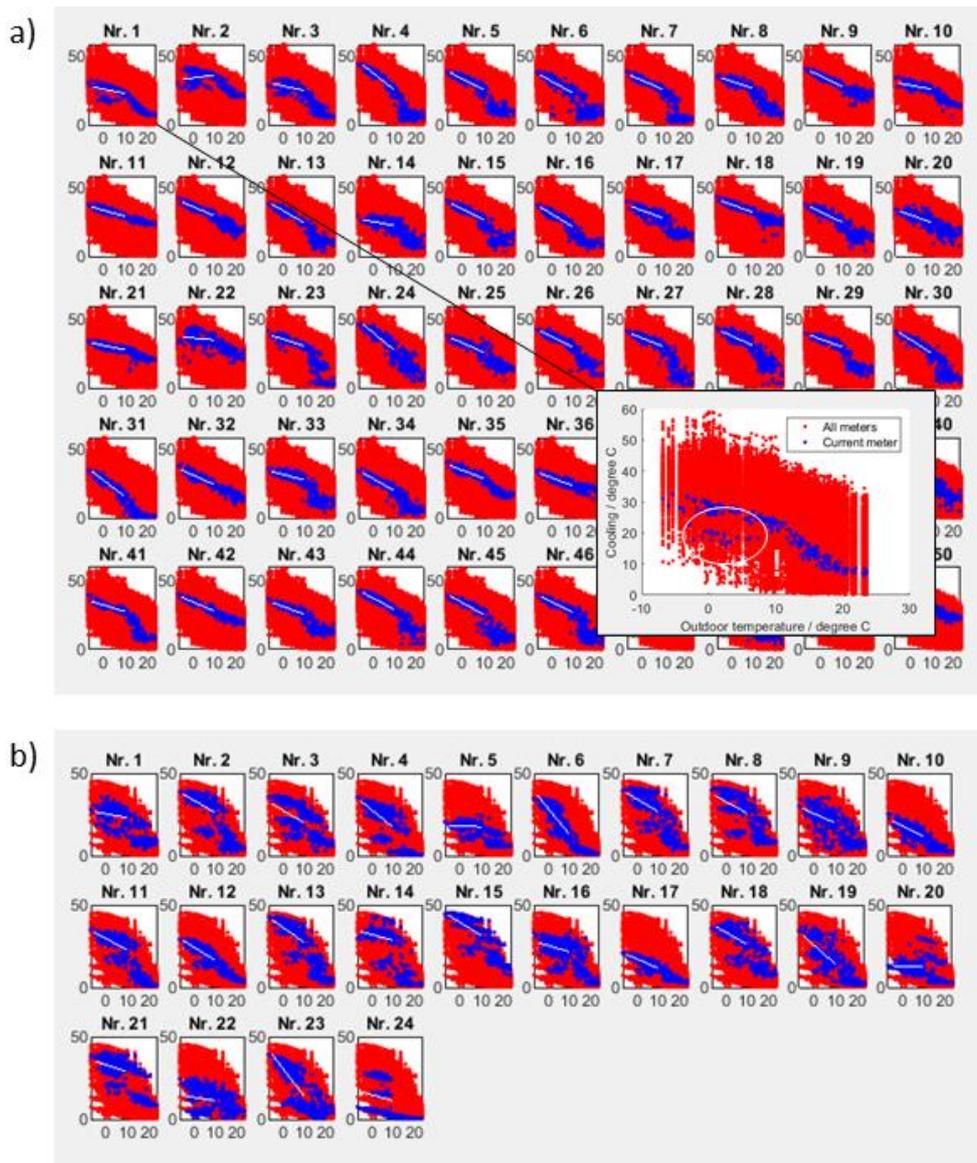


Figure 10: Collective visualization of temperature response curves

a) traditional DH (50 random single family houses in Aarhus) and b) ULT DH (Geding).

A remarkable difference in the data patterns between the installations in Figure 10 a) and b) is the linearity. The general trend of the temperature response curves follows a straight line significantly better in the data from traditional DH compared to the many different patterns in data originating from Geding. Hence, including the 24 installations from Geding in a total analysis together with traditional installations will most likely result in assignment of most installations in Geding as outliers among the much more homogeneous data patterns that are usually encountered. Inspecting the meter data for each of the individual installations



from Geding, as exemplified in Figure 7, indeed, points to faults and anomalous behavior for a very large percentage of the buildings, suggesting that ULT-implementation in this sub-network still has a significant optimization potential.



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 768567

This material reflects only the author's views and neither Agency nor the Commission are responsible for any use that may be made of the information contained therein

7.1.5. User input

The DH Doctor has been developed such that it provides a handful of analysis opportunities for the supplier. The individual features can be regarded as a palette of options that might come useful, depending on the demands, wishes and aims put forward by the user.

At the same time, the output from the program will be dependent on user input and the tuning of various parameters. Therefore, the true value of each individual part is to be uncovered only when implemented and used in practice, as planned in the continuation of the RELaTED project (see section 8).

Since the methods do not anticipate or require any prior knowledge related to geographical information or type of network, it applies, in principle, to every individual DH network. The algorithms will learn from the recorded meter data and adapt clusters and models to the specific case. Therefore, it can be trained on conventional as well as ULT DH networks without modifications to the program, as already demonstrated in sections 0 and 0.

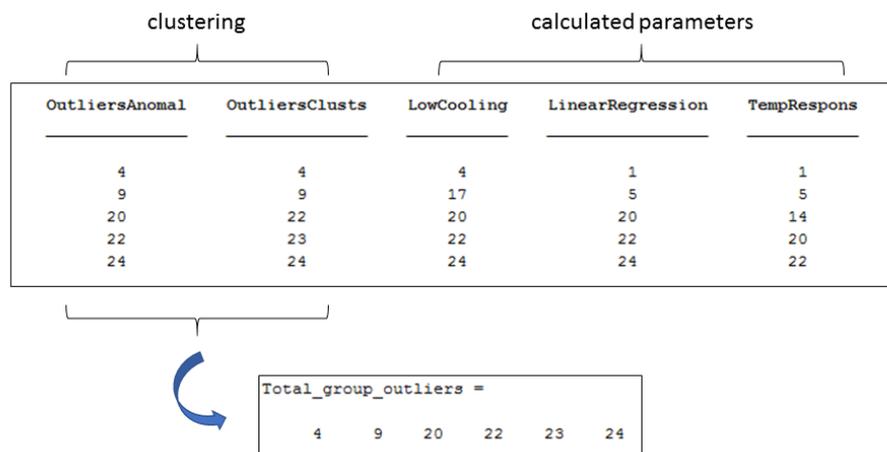


Figure 11: Output from DH Doctor showing groups of outliers resulting from clustering and calculated parameters.



A schematic of output from DH Doctor is shown in Figure 11. Five groups indicate a number of “worst” outliers in these respective categories (results from predictive models in section 0 excluded). The installations in each category can be further visualized for diagnosis, but also, depending on experience with the tools, certain types of groups might be prioritized or emphasized by the algorithms by tuning various parameters (user input), most importantly:

- *Number of clusters* (typically in the range 4 to 20), as well as the upper limit defining how many installations that can belong to an anomal cluster (typically less than 3). These numbers will depend on the total number of substations included in the analysis.
To help the user, an option was added that calculates a suitable number of clusters, K , in the specific case based on several runs of the algorithm while varying K . However, this can be a time-consuming calculation and is not recommended for daily use.
- *Number of clustering repetitions* to produce a repeatable outcome. Typically 15 or, preferably, more repetitions should be executed depending on the resulting calculation time.
- *Definition of outlier criteria*: Fraction of total number of installations to be included in the group of outliers, i.e. how many of the worst performing installations are considered critical.
- *Definition of thresholds* for critically low cooling and the linearity of the temperature response curve.

In addition to these definitions, it is possible to specify which parameter time series the clustering should take into account. The default setting is “*flow and cooling*”, but it might be useful to compare single parameters one at a time (*consumption, supply temperature, etc.*).

For predictive modeling, the input and output parameters can similarly be selected manually, however the program allows only one output parameter. The default setting is *flow* as the output parameter modeled on the basis of *energy consumption, supply temperature, outdoor temperature and day of week*.



7.2. DH Autotune: Load forecast and fault detection based on hourly values

DH Autotune is designed to give a **prediction of the future consumption** in the DH network, but also to predict single parameters continuously to enable **alarming signals** if significant deviations from models occur related to, e.g. critically low supply temperatures.

The future consumption in the DH network predicted by DH Autotune will allow the utility to produce the required energy at the correct time. This ensures enough heat for the consumer, while lowering the costs of the utility, as the utility only produces what is required. Two methods for load forecast have been investigated (sections 0 and 0, respectively).

Finally, the last section (7.2.4) addresses an alarming system based on predictive ML by decision tree (DT) ensembles.

7.2.1. Training and test data for the development of DH Autotune

DH Autotune was developed using the measured power and other parameters from 43 sub-units in Tartu, Estonia, and weather data from the University of Tartu. The data from the 43 meters has been recorded at an hourly rate since January 1st, 2019. However, method 1 for load forecast in section 0 also incorporates values from 2017 and 2018.

A limitation of the hourly data, originating from only 5 months in 2019, means that the derived load forecast models (and possibly also the alarming system presented in Section 7.2.4) as currently trained, will face difficulties when predicting consumption during later seasons of the year, where the data used for the model does not have many similarities to the weather conditions and power requirements already recorded. However, the model can be retrained at periodical intervals, e.g. once per week, reducing the risk of making predictions which are diverting more than expected. Over time, as more data than from a full year will be stored in the database, this problem is expected to be overcome.

While the model is trained on data from installations in Tartu, the same procedure can be used at other locations. In particular, it is also anticipated that although the data used for training does not originate from an ULT DH network, the approach will be fully functioning at such sites.

Time resolution: To obtain a model useful for predicting the heat profile during the day, 1-hour resolution might represent the lowest resolution that can be used for a valuable daily prognosis. Ideally, time resolution should be higher, although this would start challenging the computational power available in the project.



The algorithms behind DH Autotune is exclusively based on 1-hour resolution, opposite DH Doctor that takes an input of daily values. Since the consumption in the network is heavily dependent on the time of the day, a time resolution with longer time between each measurement would not generate much value for the utility, since the utility needs to know, when the power is required during the day.

A higher time resolution, e.g. one measurement every 5 minutes, would be interesting to investigate. However, it would also imply certain issues to use such data, as the amount of data generated by reading the meters every 5 minutes will strain the computational powers. Furthermore, day-to-day fluctuations of domestic hot water can generate random noise to the model which might result in a model with limited use. It can also be questioned whether the utilities will actually benefit from a model of higher resolution, since the significant inertia in the system makes it difficult to make abrupt changes in the power production.

Data access: In its current form, the DH Autotune imports meter data in the form of an Excel-sheet, and weather data using a http-call to the University of Tartu's webserver.

Since the meter data is only used during the training of the model, there is no need to obtain real-time data. If the model needs to be re-trained once every week, the new data points can be downloaded and added to the existing data.

For use with data from other locations, it is a requirement that the power at the sub-station is available. Whether this can be read directly from the meter or found by calculations, i.e. derived from inlet temperature, return temperature and water flow is not of high importance.

The weather parameters loaded and used in the following sections are:

- Outdoor dry bulb temperature [°C]: T_{OUT}
- Humidity [% rh]
- Wind direction [°]
- Wind speed [m/s]: W_s
- Global solar radiation in horizontal plane [W/m^2]: G_T

In general, such data can be retrieved from either a local weather station or from the national weather services.

For live predictions of the consumption, a precise weather forecast is required. This has yet to be implemented into the algorithm, but should not cause a large problem, as such services are readily available from multiple online providers.



Modal evaluation: To be able to compare the accuracy of the different developed models behind DH Autotune, the R^2 -value (coefficient of determination) is calculated, defined as

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}},$$

$$SS_{res} = \sum_i (y_i - f_i)^2$$

$$SS_{tot} = \sum_i (y_i - \bar{y})^2$$

where \bar{y} is the mean value of the measured values and y_i and f_i the measured / predicted values for each data point, i . The coefficient takes by definition values smaller than 1. Higher values (approaching 1) indicate better quality of the fit or model predictions, and a value equal to 1 corresponds to predicted values being exactly equal to measured values.

Furthermore, comparison and quantification of agreement between measured values and predictions have been characterized by the Mean Absolute Error:

$$MAE = \frac{\sum_i |f_i - y_i|}{n}$$

where f_i is the predicted value and y_i the measured value for each point, i , considering n points. Small MAE values indicate accurate predictions.



7.2.2. Consumption prognosis, method 1: Linear regression models³

To model the heat load of individual installations, Linear Regression Models (denoted T and Q functions) reported in this section were tested for their ability to predict loads based on weather information. Furthermore, the models have been formulated to also extract information from recorded smart meter data regarding separation of space heating (SH) and domestic hot water (DHW).

For the visualization of the model performance in this section, four buildings were selected representing different types of consumers (load patterns and uses). The presented buildings are: 10045 (residential building + DHW (Domestic Hot Water)); 10259 (residential + no DHW); 10949 (kindergarten + DHW) & 11718 (office building + DHW). The buildings will be used as a reference for comparison with other methods for load prediction in section 0.

The T and Q functions focus on the separation of the heating demand (Q) into space heating (SH) and domestic hot water (DHW) – one part of the heating demand (DHW) being completely independent of the climatic variables whereas the other one (SH) has a high dependency. Hence, the T function algorithm is designed to separate the data by a threshold in the outdoor dry bulb temperature (T), considering that this is the main variable that limits the use of the space heating. Similarly, the Q function algorithm separates the data by a threshold in the heat load (Q).

Data: Data from the seasons 2017 & 2018 were used as training data. Since data from the energy metering in 2019 heating season displays no lack of information, this allows an appropriate test set for the algorithms proposed. The recorded energy meter values represent the sum of loads for space-heating (SH) and domestic hot water (DHW), since the energy meter is located before the heat exchangers in the secondary side of the DH network.

The climatic or external (to the buildings) variables introduced in the LRMs models are:

- Outdoor dry bulb temperature, in °C: T_{out}
- Global solar radiation in horizontal plane [W/m^2]: G_T
- Wind speed, in m/s: W_s

³ Work performed by Tecnalía. The algorithms were developed in the program R.



T algorithm: The equation defining the system performance by the proposed T algorithm is the following (for each of the buildings):

$$Q_{alg}(T_{REF}, T_{OUT}, G_T, W_S) = \begin{cases} C_1 + C_2 \times T_{OUT} + C_3 \times G_T + C_4 \times W_S, & T < T_{REF} \\ C_0, & T \geq T_{REF} \end{cases}$$

This algorithm may be applied for any time-step frequency. If the hourly data in which data is received is transformed to daily data, intra-daily transitory effects are avoided, and it is possible to disaggregate loads for SH and DHW, considering that the DHW does not vary during the heating season. The algorithm is applied to a loop of T_{REF} , so that the one with minimal error can be found.

The results for one of the buildings for daily average values and different values of T_{REF} is given in Figure 24 in Annex 2: DH Autotune. Looking at the results, it can be concluded that part of the DHW consumption is also made below the reference temperature and that the methodology needs to be revised.

Q algorithm: A different algorithm, the Q algorithm, was then proposed, in which data is divided by another reference value, Q_{REF} , which allows the separation of the two load contributions. The model equation now takes the form:

$$Q_{alg}(Q_{REF}, T_{OUT}, G_T, W_S) = \begin{cases} C1 + C2 \times T_{OUT} + C3 \times G_T + C4 \times W_S, & Q > Q_{REF} \\ B, & Q \leq Q_{REF} \end{cases}$$



As in the case of the T algorithm, Q_{REF}^4 is obtained by executing a loop over a span for values, until the minimal error is obtained. To illustrate the process, results for one building are presented in Figure 12. Here, the horizontal blue dotted line indicates the current value of Q_{REF} .

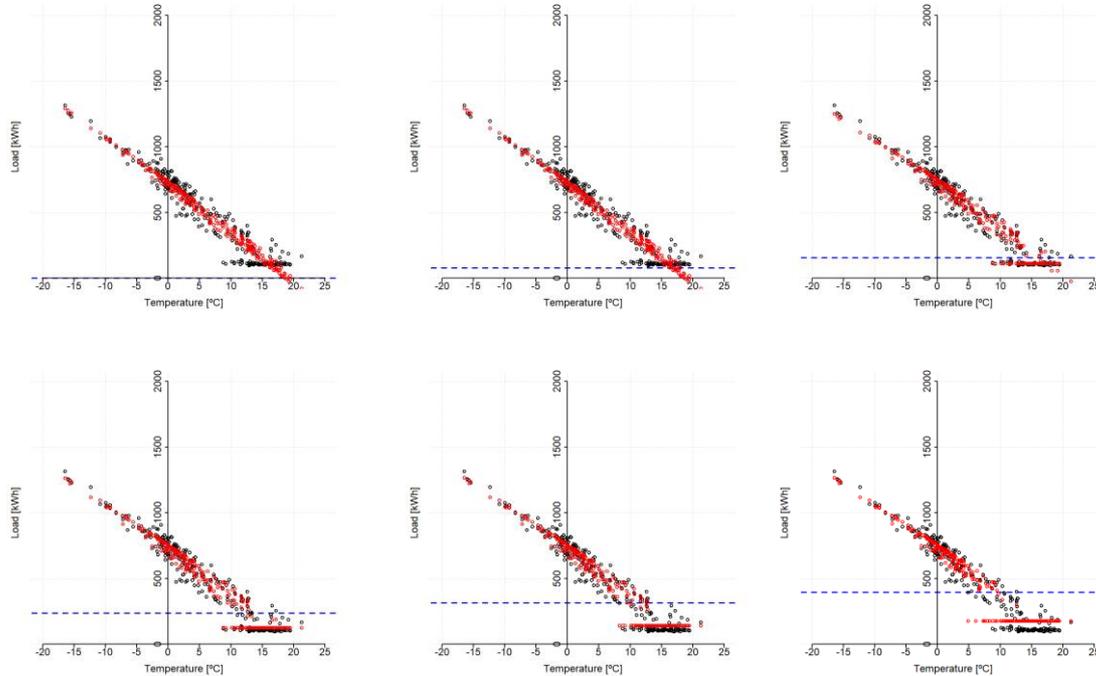


Figure 12: Q algorithm applied to daily average values from building 10045 for different Q_{REF} .

The blue dotted line indicates the value of Q_{REF} .

⁴ Note that Q_{REF} does not match the DHW consumption. It is only a mathematical variable corresponding to the building.



Decision-Tree models: For analysis on hourly data, much more training data is available, however the loads are more complicated to predict, and the simple Q algorithm does not apply.

In hourly frequency data it is possible to distinguish patterns of use along the day. As an example, see Figure 13, setback for night hours in the SH demand can be distinguished. For this reason, application of decision trees (DT) to the previous algorithm is necessary and implemented by in two ways (models):

- DT1: Clustering of the whole data by the day of the week
- DT2: Clustering by the hour of the day

In this way, the equation of the Q algorithm is applied to each of the data groups divided by the two decision trees applied, getting the coefficients in the previous equation for each of the groups. Figure 13 illustrates the division of raw data (left) into separation by weekday (middle) and hour (right).

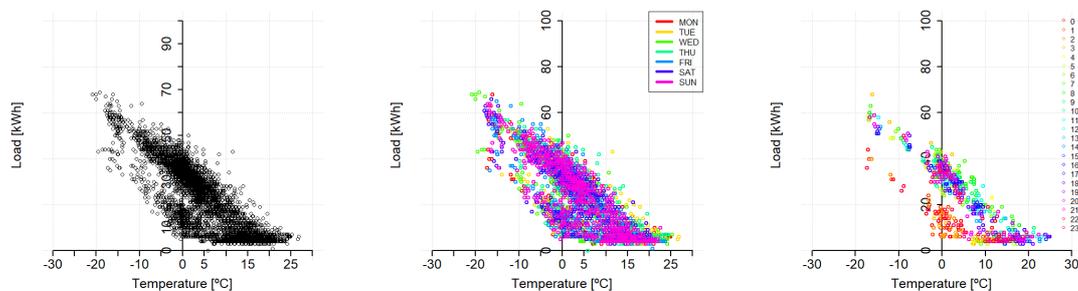


Figure 13: DT1 and DT2 applied to building 10045.

LEFT: Raw data, MIDDLE: Data separated by weekday, RIGHT: Data separated by hour.



Using the model DT1, the algorithm describes the heating demand satisfactorily. Results for characterization of building 10045 after applying the Q algorithm with the decision-trees DT1 are presented in Figure 14 with predicted load values (red) laid on top of the measured readings.

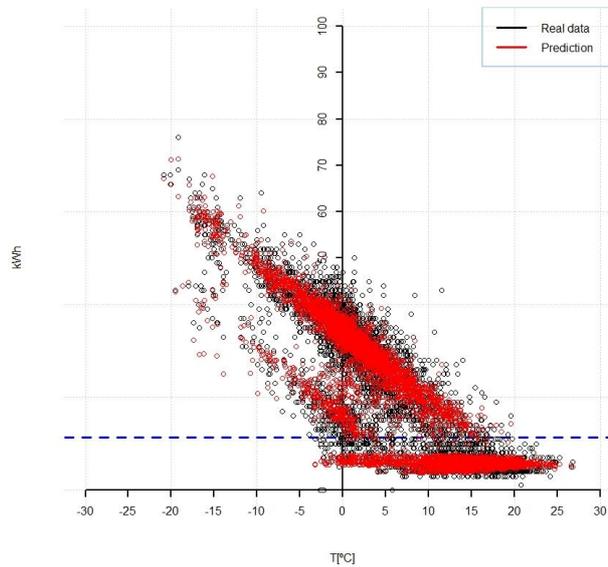


Figure 14: Q algorithm application after DT1, building 10045. The blue dotted line indicates the value of Q_{REF} .

Test on data from 2019: The developed model was applied to test data from the heating season in 2019 (until April, i.e. to four months of available data). Due to the low available amount of data, only results for hourly frequency will be presented for the four selected buildings.

For prediction of the heat load in 2019 the coefficients from training data stemming from 2017 and 2018 are used, anticipating that the use patterns for the respective buildings have not changed. At the other hand, for characterization of the buildings in 2019 (via the value of Q_{REF}), the same model is applied, but trained on data from 2019 to calculate new coefficients including Q_{REF} .

Heating load predictions for 2019 for the four buildings are plotted and shown in Annex 2, Figure 25 - Figure 28, together with the raw data.



Table 2 shows the R^2 values for the regression model applied to the test data, and it is seen to result in different success rates. The major reason is that the algorithm is very dependent on the quality and the quantity of data available for the specific building. The method aimed at disaggregation of SH and DHW loads requires data ranging over more than one season (as currently available for 2019). As a consequence of lacking input data, the error for the regression increases (e.g. Building 10259).

Table 2: R^2 value, MAE and Q_{REF} values for the Q function model with decision-trees applied to test data from 2019 with hourly resolution.

	Building 10045	Building 10259	Building 10949	Building 11718
R^2 value	0.8067	0.6990	0.8905	0.8191
MAE (kWh)	3.124	3.000	5.826	7.408
Q_{REF} (kWh)	12.42	8.64	8.4	21.45



Finally, results from the regressions are displayed versus time in Figure 15. The predictions (red) are observed to fit the real load (black) up to a high degree for all four selected buildings (the remaining three test buildings are shown in Figure 29 in Annex 2: DH Autotune).

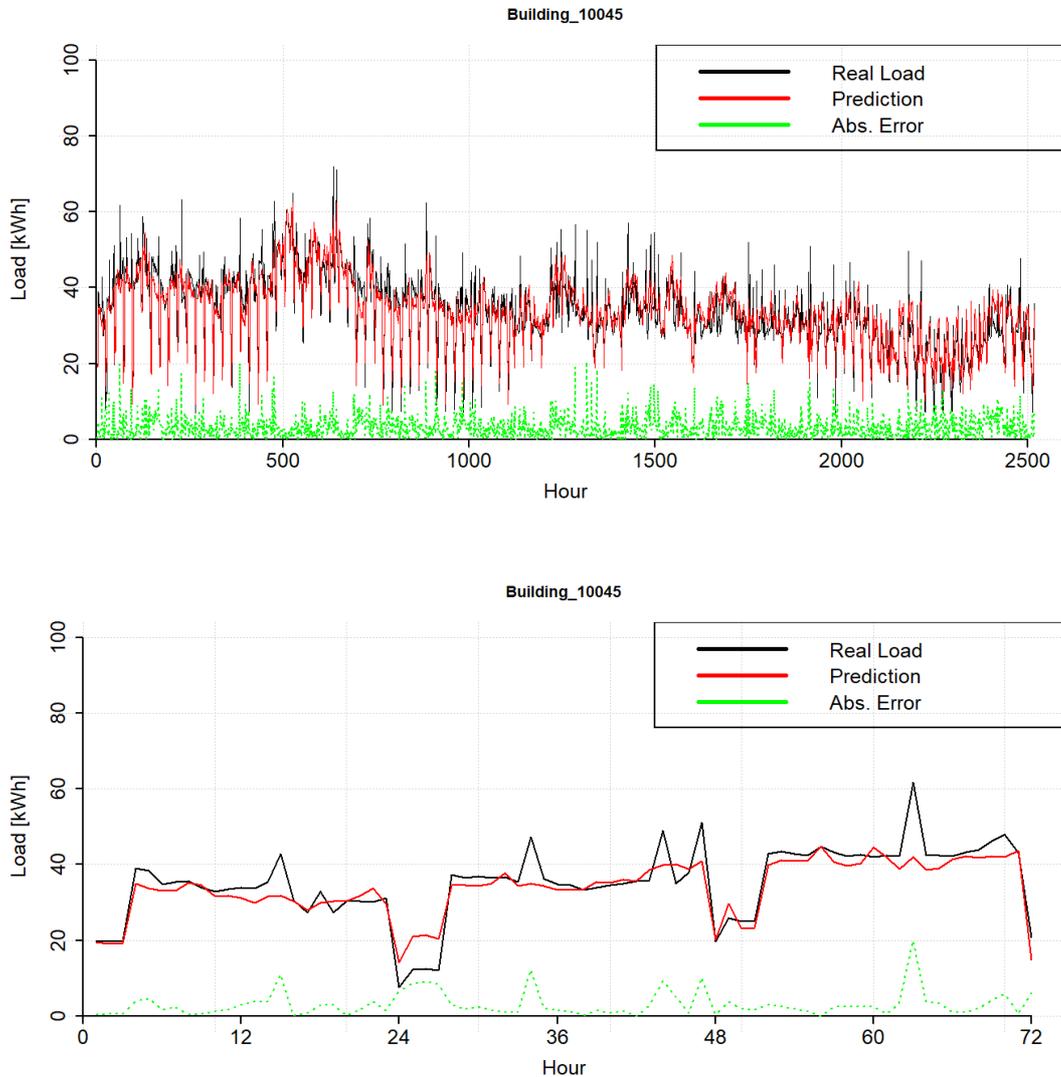


Figure 15: Temporal expression of the regression models for building 10045 (apartment).

In conclusion, the investigated Linear Regression Models (LRM) are able to predict the heat load at a satisfactory level when combining the approach with decision-tree nodes dividing the data into weekday and hour of day. However, using the reference threshold T_{REF} (outdoor temperature) in the models (T function) was not a viable solution. Hence, a reference value based on the load, Q_{REF} , was introduced (Q function) that produced satisfactory results for heat load prediction.



Since the Q function needs the power as an input, the method is of limited use regarding consumption prognosis, but can still be applied to historical data to characterize buildings regarding their consumption and division between space heating (SH) and domestic hot water use (DHW).

Additionally, although the Q function predicts the heat load well, the ML methods presented in the next section (0), turns out to be more accurate and will therefore be suggested for final implementation.



7.2.3. Consumption prognosis, method 2: Decision-tree ensembles and neural network

Method 2 for load prediction uses meter data from Tartu joined with local weather data. This provides a series of potential parameters for use in a ML model predicting the heat load.

When many parameters are available, some can be expected to correlate well with the heat load, while others might only introduce noise to the system which can have a negative effect. Hence, parameters in this method were added one by one in the development procedure to evaluate their impact on the models by comparing the R^2 values.

The parameters considered to be of interest for the heat load prediction in this section are:

- Outside temperature
- Outside humidity
- Wind speed and direction
- Time of day
- Day of year
- Weekend/not weekend
- Day of week

As a metric for comparing the models, the R^2 value was primarily used. The models use 90 % of the available data for training while the remaining 10 % were used as a test set. Since there is a risk of overfitting a model, the R^2 value calculated from the test set evaluates a model. To ensure consistency in the test data entire days were used for testing. Effectively, 16 random days were used for testing, while the remaining 135 days were used for training.

Since the training of a neural network takes significantly longer than training of a bagged-tree model, the parameters used in the models were chosen by optimization using the bagged-tree models. It is possible that another combination of parameters would do better in a neural network, however the parameters found from the bagged-tree model can be assumed to form a good starting point.

For neural networks, much work was devoted to selection of the best input parameters, however much efforts are also laid into tuning hyper-parameters such as the size of the network (number of hidden layers), learning rate during training, dampening of weights, etc.



Data preparation: From the weather data, the wind direction is converted from degrees to sinus and a cosine value. This removes the discrete step going from 360 ° to 1 °, and, instead, two continuous values are obtained. Similarly, the time of the day, and the day of the year is transformed from a linear value to a sinus and a cosine value. In this way, jumps in the value at around midnight and around new year are avoided.

The meter data is synchronized with the weather data from the previous three hours in order to incorporate any hysteresis in the building mass.

Algorithms: The basis of the analysis is a bagged-tree model available in Matlab®.

The performance of the model was taken as the average R^2 value of the 38 best performing meters. Some of the meters were consistently having low R^2 values regardless of the models being used, indicating that these buildings were used in a different way than the remaining.

Training the bagged-tree model with just one parameter at a time showed, as expected, that outside temperature was the single parameter explaining heat load best.

Adding the day of the year, followed by time of day to the model increased the precision of the model, and adding information regarding weekends improved the models further.



Table 3 shows the average R^2 values for the test data from the bagged-tree models. Interestingly, information regarding humidity and wind (direction and speed) did not improve the performance of the models. It can be speculated whether this is due to these parameters changing significantly over the area of the city of Tartu, or if the weather station at the University of Tartu is not accurate enough.

Table 3: Average R^2 values of the bagged-tree models, having different input parameters. The R^2 values are calculated as the average values for the 38 best performing models

Parameters	Temperature	+Day of year	+Time of day	+Weekend	+Humidity	+Wind
R^2 value	0.713	0.814	0.874	0.881	0.880	0.876

After training the bagged-trees, a neural network was trained for each of the installations. As mentioned, these neural networks took the same input parameters as the bagged-trees. Results from the neural network appear from Table 4.

To simplify the training of the neural network, a network was defined containing five fully connected layers, each containing 100 nodes. ReLU activation functions were used during the back propagation of the network. When training the network, a learning rate of 0.0005 was settled, and the networks were trained for 3000 iterations.

With the network structure and the hyper parameter settings used in this network, it is seen that the bagged-tree resulted in higher R^2 values than the neural network. While there is a possibility of improving the performance of the neural network by optimizing the network structure, learning rate, L2 regularization parameter, dropout rate, activation functions, along with changing the network to be a Long-Short Term Memory-network (the network is taking the last predicted values into account) the R^2 values for the bagged-tree models are in general remarkably high, making it altogether difficult to find a better overall algorithm.

Table 4: R^2 values of bagged-tree model and neural network.

The two models have the same input parameters. The bagged-tree performs, on average, better than the neural network.

Algorithm	Bagged-tree	Neural network
R^2 value	0.881	0.853



Performance for individual installations: In order to compare the models above with the linear regression models in section 0, we evaluated the R^2 values of the same installations.

The R^2 values corresponding to the four selected installations, also analyzed in section 7.2.1 by linear regression models, have been collected in Table 5. As indicated by green, the neural network and bagged decision tree models perform best, although the Linear Regression Model has the highest R^2 value for building 10949 (the kindergarten).

Table 5: R^2 values for the four installations discussed in section 7.2.2

The bagged decision tree and neural network are compared with the linear regression model (section 0). The best R^2 values are indicated with green. MAE values for the bagged-trees for the four installations are also shown and compared to the linear regression model. Note that the MAE values for the bagged-tree are consistently better compared to the regression model.

Installation	10045	10259	10949	11718
R^2 -value, Bagged-tree	0.9097	0.8809	0.8838	0.9116
R^2 -value, Neural Net	0.8455	0.8674	0.8177	0.8864
R^2 -value, Linear Regression Model	0.8067	0.6990	0.8905	0.8191
MAE-value, Bagged-tree (kWh)	2.677	2.222	5.770	6.373
MAE-value, Linear Regression Model (kWh)	3.124	3.000	5.826	7.408

Prediction of total consumption: When a model has been trained for each of the 43 substations, the expected consumption in the system is found by adding up the expected consumption from the individual installations.

Executing the modelling on the individual installations, rather than on the system in full, has the clear advantage that it is always possible to find enough data to train a model for a single sub-station. If the entire network is to be trained as a single unit, data from all meters must be available for any specific time. If it can be expected that 99 % of all meter readings are recorded, there is only a $0.99^{43} = 64$ % chance that data from all meters are recorded when the network consists of 43 meters. In a larger network with thousands of meters, the chance that data from all meters make it to the database at any one hour is practically 0.

While the models of the individual sub-stations, see Figure 16, left, might have rather high residuals when comparing the modeled value with the meter value, the combination of values from the models of all the sub-stations, Figure 16, right, tends to make the large residuals disappear because of the central limit theory.



As such, it can also be expected that the calculated total consumption will perform better, the more sub-units that are included in the network, since the fluctuations arriving from domestic hot water consumption is smeared out over more models.

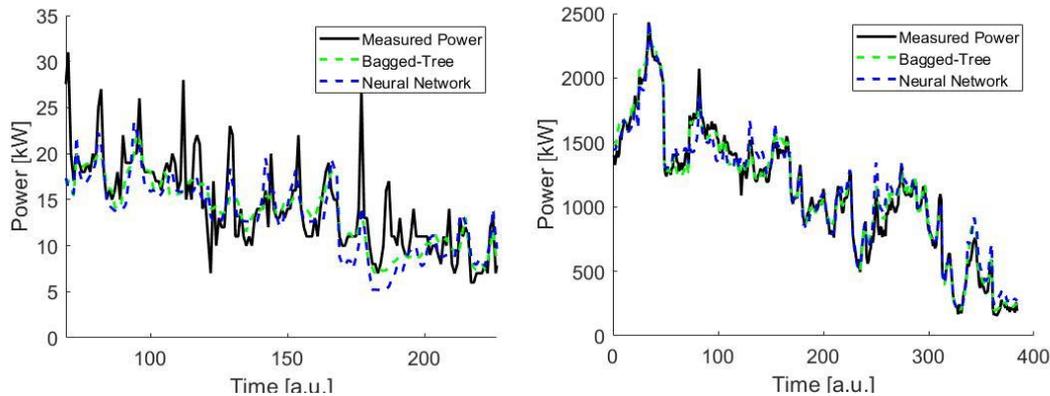


Figure 16: Actual and predicted consumption in kW.

Left: Predictions for a single installation. Right: Prediction of the total load of all test buildings.

Calculating the R^2 value of the total power consumption yields a value of 0.9807 for the bagged-tree models and 0.9608 for the neural network. These numbers are even including data and predictions for the meters which were excluded previously, due to exceptionally bad performance agreement with their respective models. The MAE value of the total power consumption can be found to 52 kW out of a total consumption of on average 1.12 MW. This corresponds to an error of less than 5 %. Whether this will have any practical implications, depends on how well the energy is stored in the district heating system.

The high R^2 values also show that when many installations are in service, fluctuations on a few of these do not perturbate the entire network to a large extend. Modelling the power load, based on machine learning algorithms, thus appears to provide a reliable forecast.



Notes on implementation of heat load prediction: In summary, the results on application of method 2 (this section) for load forecast (decision trees and a neural network) show that the power consumption in a network consisting of multiple users can be found to a high precision.

In this work, ensembles of decision trees can be concluded to perform better than a neural network and linear regression models and is therefore the suggested implemented method. With this algorithm, R^2 values characterizing the quality of the prognosis was observed close to 1 (0.98 for the total load of 43 buildings), indicating that a reliable prediction is viable, but of course dependent on the weather forecast. In addition, the results show that accurate prediction for individual buildings is also feasible, yielding typical R^2 values of 0.9.

7.2.4. Alarming system sensitive to critical limits regarding supply temperature

The purpose of modeling and predicting selected parameter values such as flow, return temperature, etc. in DH Autotune is to enable an alarming system, designed to react when an installation fails to perform well.

In particular, this kind of alarming system is intended to be used in the gradual process of lowering supply temperature towards critical values and this way gain a better insight and documentation while searching for the limits of the system.

The models are based on the same hourly resolved smart meter data from 43 installations in Tartu used for consumption prognosis in sections 0 and 0. A variety of measured parameters have been modeled through their relation to other simultaneously recorded parameters used as input variables, as well as their dependence on complementary information such as weather, weekday and time of day. The developed models describe the relationships for a well performing installation. On this ground, deviations from the model are regarded as a sign of abnormal behavior and could for example occur in connection with lowering the supply temperature beyond critical limits.

Data and model algorithm: The models use real-time recordings (alternatively hourly average might be used in the same algorithms, but these were not available for all parameters in the present study).

The data used comprise 43 installations in Tartu over a time period from 1/1 2019 to 30/4 2019 (four months). The data was divided into training and test periods as specified below.

An ensemble regression tree algorithm (ensemble of decision trees) was used by accessing Matlab®'s build-in functions. It was found that feature normalization did not improve the results and, hence, was not built into the program. Generation of polynomial features was temporarily implemented but did also not improve the



algorithms significantly and was therefore not included to simplify the coding and minimize sources of error.

Model features: As a starting point, the parameter flow (l/h) was modeled with different combinations of the available input parameters. One important restriction is that only two of the parameters *flow*, *power* and *cooling* (difference between supply and return temperature) must be selected since the third one can in principle be calculated from any two of them. Consumer types were “apartment building”, “private house”, “school”, “shopping center”, “office” and “kindergarten”.

A comparison between models with different input parameter combinations is summarized in



Table 6. The models were evaluated on the basis of the R^2 value as well as the Mean Absolute Error (MAE) divided by the mean value of the output, which can be interpreted as the average deviation in percent between model and measurement.

The models were trained on all data from a total of 40 installations, where a complete data set was at hand (including information on “consumer type”). R^2 and MAE values were calculated for a test set consisting of the same 16 days excluded from the training data. The R^2 and MAE values displayed in the table are averages of individual calculations for every single installation.

It can be concluded from the table that addition of the categorical variable “consumer type” significantly enhances the accuracy of the model predictions (maximized R^2 value). This points to a different behavior depending on the type and use of the building which encouraged a further training on individual houses (next section).

The optimum parameter combination was found to be the model predicting the flow on the basis of the inputs: *Supply temperature*, *Outdoor temperature*, *Power*, *Weekday* (1-7), *Hour of day* (1-24) and *Consumer type*.



Table 6: Evaluation of prediction models for mass flow characterized by their combinations of input parameters.

The best performing model is indicated with bold, judged by the maximum R² value and minimum Mean Absolut Error (MAE).

T supply	T outdoor (hourly)	Power	Weekday	Hour of day	Consumer type	R ² value	MAE (rel. to mean)
Degree C	Degree C	kW	Categorical	Categorical	Categorical	-	%
x	x	x				0.927	11.73
x	x	x	x			0.928	11.41
x	x	x		x		0.929	11.24
x	x	x	x	x		0.931	11.29
x	x	x	x		x	0.971	6.69
x	x	x		x	x	0.971	6.66
x	x	x	x	x	x	0.971	6.95

Training on individual installations: The best performing combination of input parameters for the prediction of flow according to



Table 6 was used to train a model for each installation in order to assess if the accuracy of predictions could be improved. The result is presented in



Table 7.

As expected, due to the positive effect of introducing “consumer type” as parameter, the accuracy of the model predictions on the test set was improved, reflected by the R^2 value increasing from 0.819 to 0.924. Note, that these values cannot be compared directly to



Table 6 since a different test set was used. Similarly, the improvement was reflected in a decrease of the average MAE value from 7.92 % to 4.57 %, i.e. an average size of error of roughly 5 % of the measured flow values.



Table 7: Model trained on data from all installations compared to models trained individually on each installation.

The R^2 and MAE values are calculated from the same test set for each installation.

Model training:	R^2 value (average for all installations)	MAE (rel. to mean) (average for all installations)
Trained on all data	0.819	7.92 %
Trained on each installation	0.924	4.57 %

Table 8: Comparison of models predicting *cooling*, *return temperature* and *mass flow*.

The R^2 values are averages over the individual installations (same dates applied as test set for each installation).

Model (output parameter)	Input parameters	R^2 -value
Flow	power, T outdoor, T supply, weekday, hour of day, consumer type	0.924
Cooling	flow, T supply, T outdoor, weekday, hour of day, consumer type	0.815
Return temperature	flow, T supply, T outdoor, weekday, consumer type	0.755

Prediction of other parameters: In addition to prediction of “flow”, other parameters were modeled using the same method and algorithm. The models were trained on individual installations due to the improvement observed compared to training on the full set of data from all buildings.

A model was set up to predict the cooling, defined as the difference between supply and return temperatures (real-time values). With the input parameters stated in

Table 8, the model predicts the cooling with an R^2 value of 0.815, i.e. the model is somewhat less accurate compared to prediction of the flow.

Similarly, the return temperature was predicted with an R^2 value of 0.755.

It should be noted, that since these values are averages, many individual models for the respective installations perform significantly better. However, a few installations also display variations in the data patterns that result in lower R^2 value for reasons that have not been diagnosed by the models but might be of interest from the supplier’s point of view.



Visualization of modeling results: The developed Matlab® script for model training was furthermore used to visualize the results from calculations.

In the selected presentation of data, the training set is indicated by green whereas the test set is marked red. Since the aim of the program is to assess the current performance of an installation, the test period was in the example shown in Figure 17 defined as a short period a few days back in time such that an arising problem can be spotted. In this case, no problems are seen, indicated by the agreement between predicted and measured values (left and middle) as well as the straight line of the test points in the plot to the right.

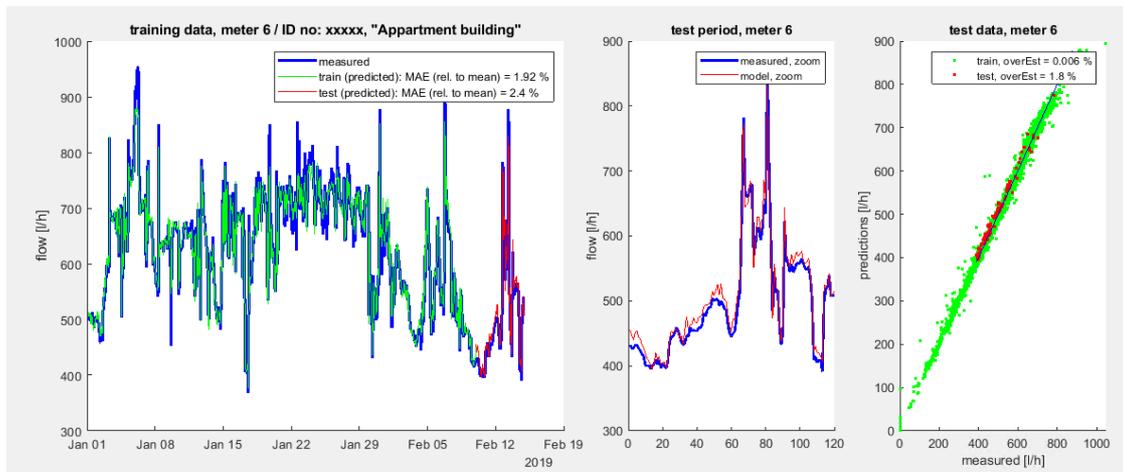


Figure 17: Visualization of results from a model predicting the flow over a test period of 5 days for one building (anonymous meter number).

Predictions for the test period (red) are compared to the actual measured values (blue). Training data is marked with green.



A few key numbers and information, including the “consumer type”, have been added in the visual output from the program:

- **MAE** in percent of the average output parameter value (rel. to mean), describing the average size of the error between measured and predicted values. The MAE (rel. to mean) varies somewhat depending on the choice of test period. In the statistics given in Figure 18, the average MAE was 3.95 %, however a few meters have values reaching 10 % and 15 %.
- **Overestimate** relative to mean (overEst) defined as the average value of the residuals, stated in percent of the mean value of the output parameter. This value is a measure of how much the model overestimates (or underestimates) e.g. the flow or cooling. This information is aimed at better interpretation by the user. The overestimate for a well working installation typically lies within 5 % (Figure 18).

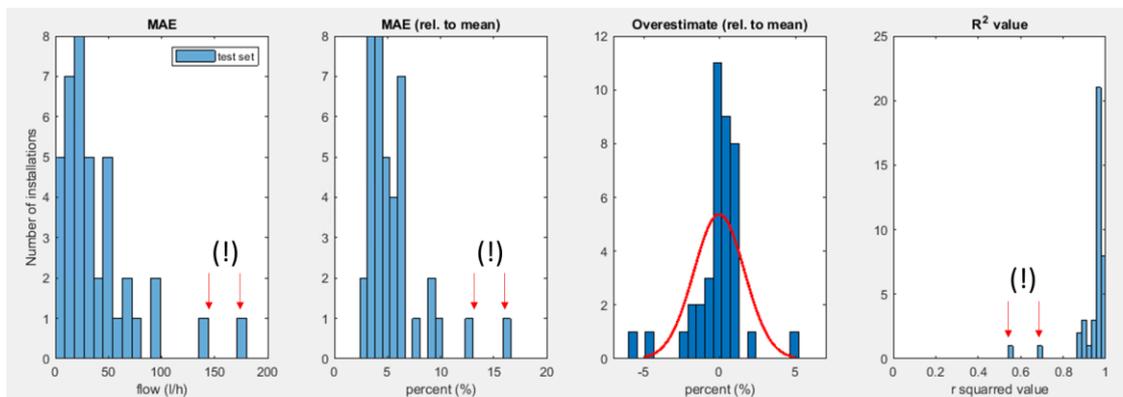


Figure 18: Statistics for all 40 analyzed installations with models trained on individual installations, using the same test period.

Two outliers are indicated by arrows with deviating performance due to a few datapoints with abnormal behavior in the test set.



Detection of induced faults: The sensitivity to occurring faults at installations was tested by inducing faults in the raw data set. Over a test period of five days, the parameter “flow” for a specific installation was increased stepwise hour by hour up to 150 % relative to the original values at the end of the period.

The consequence in terms of deviation from model predictions was evaluated using all three different model setups for prediction of *flow*, *cooling* and *return temperature*, respectively. The results are displayed in Figure 19, where mismatch between measured values and predictions are now easily spotted, both in the time series (left) and plot of measured vs. predicted values that should form a straight line for a well performing installation (right).



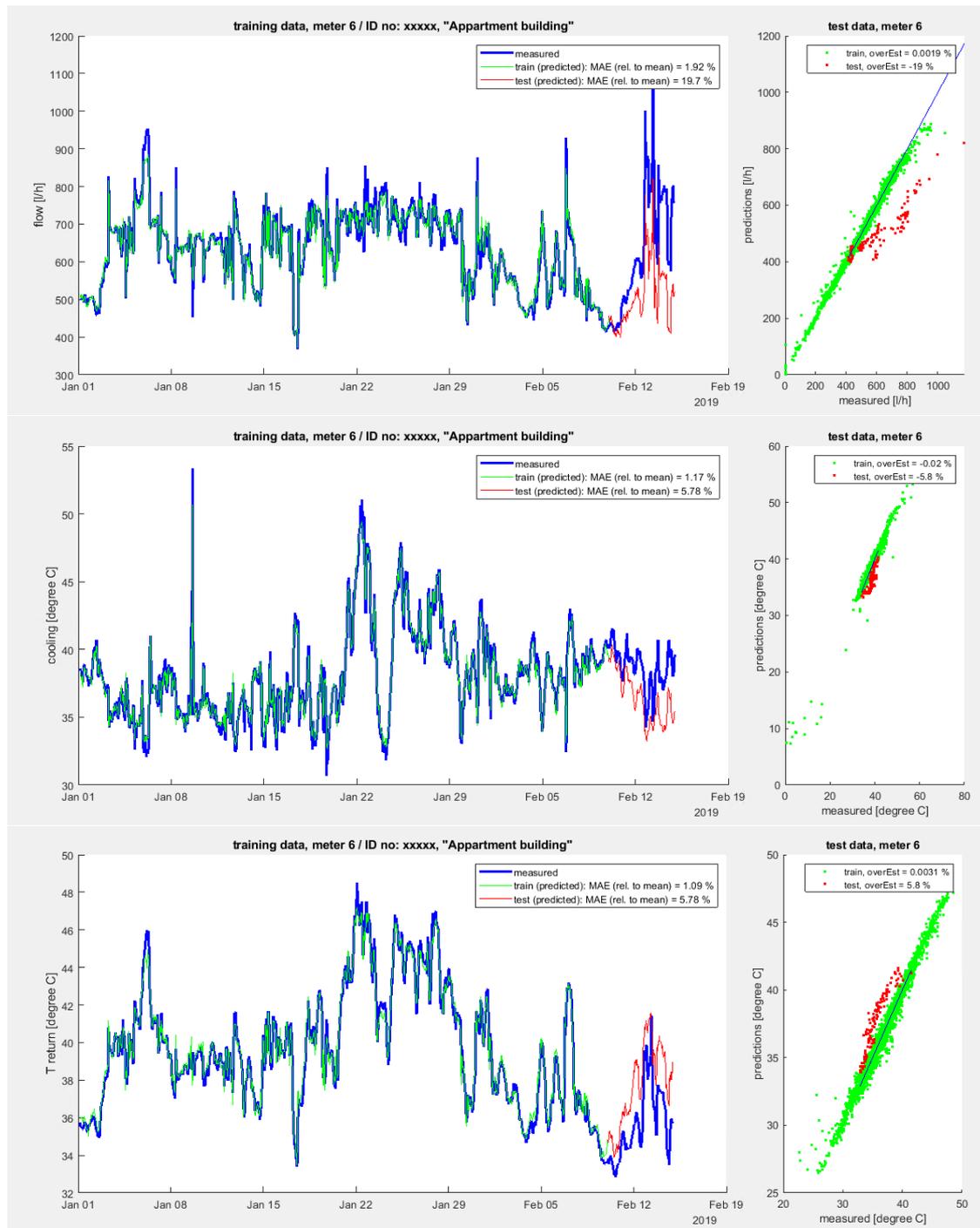


Figure 19: Output from the Matlab® program predicting the flow during a test period with an intended fault induced on the data set.

The test period (five days) and induced fault were the same for all three models.



The MAE value for flow prediction increases to 19.7 % relative to the mean value which is well above the detection threshold when compared to the situation without an induced fault that yielded 2.4 % in Figure 17. The overestimate yields -19 % reflecting that the measured flow is significantly higher than expected.

In the case of choosing *cooling* or *return temperature* as output the parameter, the same induced fault can be easily detected, as seen in the two other panel snapshots displayed in Figure 19. In these plots, the fault is slightly less obvious, a difference that might be related with the circumstance that the fault now enters through the input data (flow being input rather than output in these models). However, for various other sources of error than on the flow, it is anticipated that the most sensitive model will vary between the three proposed combinations of input and output parameters presented in Figure 19. For this reason, it is suggested to evaluate installations with all models such that the most reliable and robust surveillance is obtained.

In summary, the alarming system presented here as part of DH Autotune complements the predictive models in DH Doctor which were based on the same method – now offering surveillance from hourly readout instead of daily values. In this way, provided that hourly values are available and that the data are of sufficient quality (completeness and resolution), this tool offers evaluation of the network on an even higher time frequency, e.g. once per day.



8. Conclusions and follow up for adapting

In conclusion, tools have been developed targeted analysis of smart meter data from ULT DH networks to provide guidance and control in the daily operation by the distributor company. The tools incorporate intelligent algorithms (ML), as well as statistical analysis based on research on the state-of-the-art and fast progress of these analysis methods over the recent years.

Application on test data from Aarhus (AffaldVarme Aarhus, Denmark) comprising 24 ULT DH installations in Geding and 490 LT installations in Aarhus) and 43 installations in Tartu (Estonia) demonstrate the working principles of the DH Doctor and DH Autotune. Examples are given of the capability of the **algorithms to detect faults** on individual installations in the distribution system, as well as to provide a **data-driven consumption prognosis**.

8.1. Summary of developed tools

The work carried out has resulted in two tools with the following main features and input requirements:

DH Doctor (input data: Daily values):

- Clustering of installations: **Fault detection** indicating malfunctioning installations that require repair. Mainly targeted for use on historical data over a longer period (preferable minimum a full year) of 24-hour-resolution. An entire network with several thousand smart meters installed can be regularly analyzed together with a calculation time in most cases below 1-2 hours.
- Statistical analysis on calculated parameters: Grouping and **detection of outliers** in various categories to point at possible faults or optimization potentials. Based on historical data over, optimally, minimum a full year, but can also be applied with less available data.
- Predictions of flow: **Fault detection** linked to deviations from the typical relation to the model input-parameters. The algorithm is trained on historical data from an entire network. The evaluation period can be chosen to, e.g., “last week”, to approach a continuous surveillance.

DH Autotune (input data: Hourly values):

- Load prediction via different machine learning methods: **Prediction of the heat load** a few hours or days ahead based on the weather forecast. Preferably, a minimum of a full year of training data should be available. The load forecast can run continuously and provide guidance in the daily operation by the supplier.



- Alarming system: Enabling **fast reaction to abnormal behavior** regarding measured parameters (demonstrated for *flow*, *return temperature* and *cooling*) intended for evaluation of system performance and faults while lowering the supply temperature during the transformation from traditional DH to ULT DH networks. This tool includes visualization of hourly data and model data in time series as well as quantification by key numbers, such as MAE values, comparing recent measurements and expected values (model predictions).

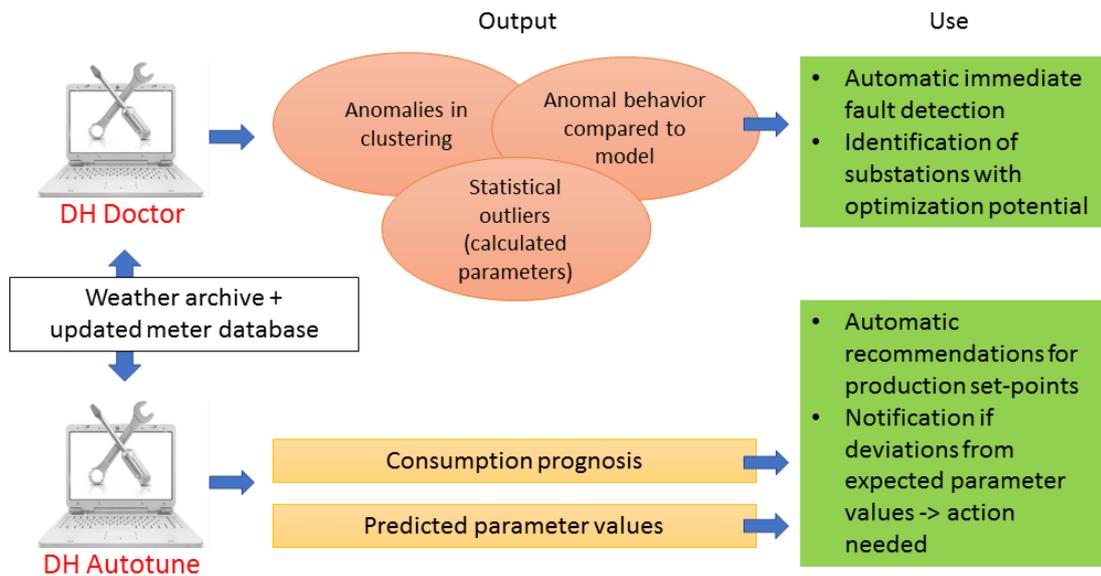


Figure 20: Overview of the developed DH Doctor and DH Autotune tools: Prerequisites, output and use of output information.

The relation between the listed elements and parts of the programs in the current version of DH Doctor and DH Autotune have been overviewed in Figure 20.

From an operational point of view, these tools will be considered for final development and adjustment in collaboration with relevant demonstration sites.



8.2. Implementation at DH utilities

The algorithms developed and described in this report will be deployed at demonstration sites in the course of WP5 of the RElated project.

At the current stage (end of WP2), the algorithms are executed by DTI via data transfer from the supply companies in Tartu, Estonia and Aarhus, Denmark. This procedure involves an initial download every time the surveillance system must be updated with the latest training data. Hence, it will be desirable to implement a new solution through the IT infrastructure at the location of the company.

This is a non-trivial task regarding the implementation, and it will be addressed in WP5 in the context of adapting control systems at demonstration sites.

Potential solutions are

- **remote access** from the utility to the calculation machine hosted by DTI with a scheme for automatic data transfer, i.e. an online service in principle.
- **export of the algorithms** developed by DTI and Tecnia as executable files or, eventually, adaption of the tools from their original code in Matlab® to other free available analysis platforms such as Python or R.

Based on further considerations in collaboration with DH suppliers, e.g. in Tartu, a viable solution regarding data access and hosting of the algorithms will be worked out and applied.



9. References

- [1] Digital Roadmap for District Heating & Cooling, DHC+ Technology Platform, Euroheat and Power, 2017-05-25
- [2] Machine Learning, Tom M. Mitchell, McGraw Hill, 1997
- [3] Ekici, Betül Bektas, and U. Teoman Aksoy. "Prediction of building energy consumption by using artificial neural networks." *Advances in Engineering Software* 40.5 (2009): 356-362.
- [4] Daut, Mohammad Azhar Mat, et al. "Building electrical energy consumption forecasting analysis using conventional and artificial intelligence methods: A review." *Renewable and Sustainable Energy Reviews* 70 (2017): 1108-1118.
- [5] Ahmad, A. S., et al. "A review on applications of ANN and SVM for building electrical energy consumption forecasting." *Renewable and Sustainable Energy Reviews* 33.1 (2014): 102-109.
- [6] Saloux, Etienne, and José A. Candanedo. "Forecasting District Heating Demand using Machine Learning Algorithms." *Energy Procedia* 149 (2018): 59-68.
- [7] Dahl, Magnus, et al. "Improving short-term heat load forecasts with calendar and holiday data." *Energies* 11.7 (2018): 1678.
- [8] Farouq, Shiraz, Stefan Byttner, and Henrik Gadd. "Towards understanding district heating substation behavior using robust first difference regression." *Energy Procedia* 149 (2018): 236-245.
- [9] Månsson, Sara, et al. "A machine learning approach to fault detection in district heating substations." *Energy Procedia* 149 (2018): 226-235.
- [10] STORM District Energy Controller, <https://storm-dhc.eu/> (fact sheet, brochure and presentation)
- [11] Operational Demand Forecasting in District Heating Systems Using Ensembles of Online Machine Learning Algorithms" *Energy Procedia*, Volume 116, June 2017: 208-216
- [12] New management system for DHC networks, Hot-Cool, Journal No. 2, 2019: 18-21
- [13] Fester, Jakob - Danish Technological Institute "Driftsoptimering med målerdata og sensorer - Hvordan kan eksisterende målerdata bedst anvendes til at optimere driften?", Dansk Fjernvarme, Landsmøde, (2018-10-26), <https://www.danskfjernvarme.dk/arrangementer/moedematerialer/2018-okt-dansk-fjernvarmes-landsmoede>



- [14] "Driftsoptimering af smart grid-fjernvarmesystemer for lavtemperatur", EUDP-2013, Journal No. 64012-0238, submitted by Danish Technological Institute 28-10-2015.
- [15] READY project, deliverable 3.3.2. <http://www.smartcity-ready.eu/d-3-3-2-documentation-of-model-development-for-optimised-dh-network-operation-based-on-smart-metering-data/>, accessed 06-06-2019
- [16] Lund, Henrik, et al. "4th Generation District Heating (4GDH): Integrating smart thermal grids into future sustainable energy systems." Energy 68 (2014): 1-11
- [17] Gadd, Henrik, and Sven Werner. "Achieving low return temperatures from district heating substations." Applied energy 136 (2014): 59-67.



10. Annex 1: DH Doctor

In this Annex, supplementary data regarding the development of DH Doctor is presented.

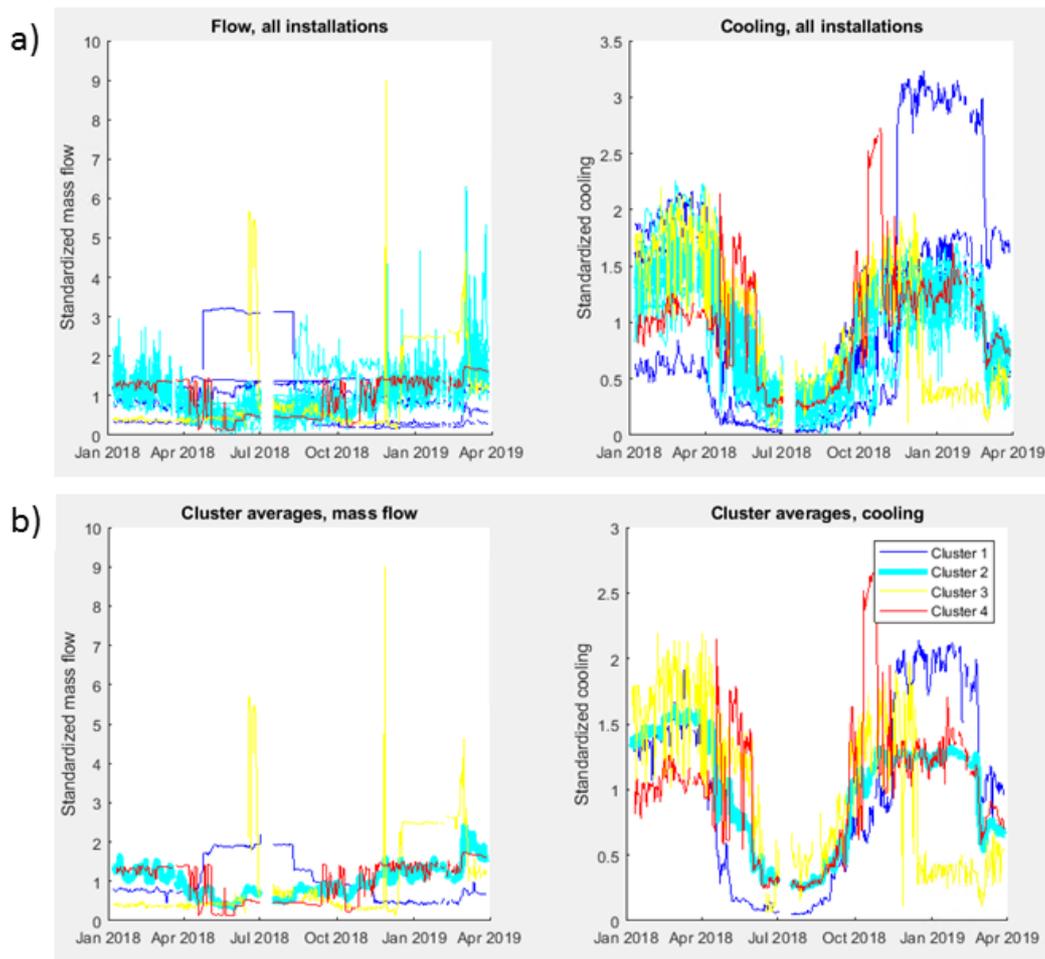


Figure 21: Clustering of 24 installations in Geding using 4 clusters.

a) All installations, b) cluster averages (centroids). Anomalies are detected in the clusters 1, 3 and 4.



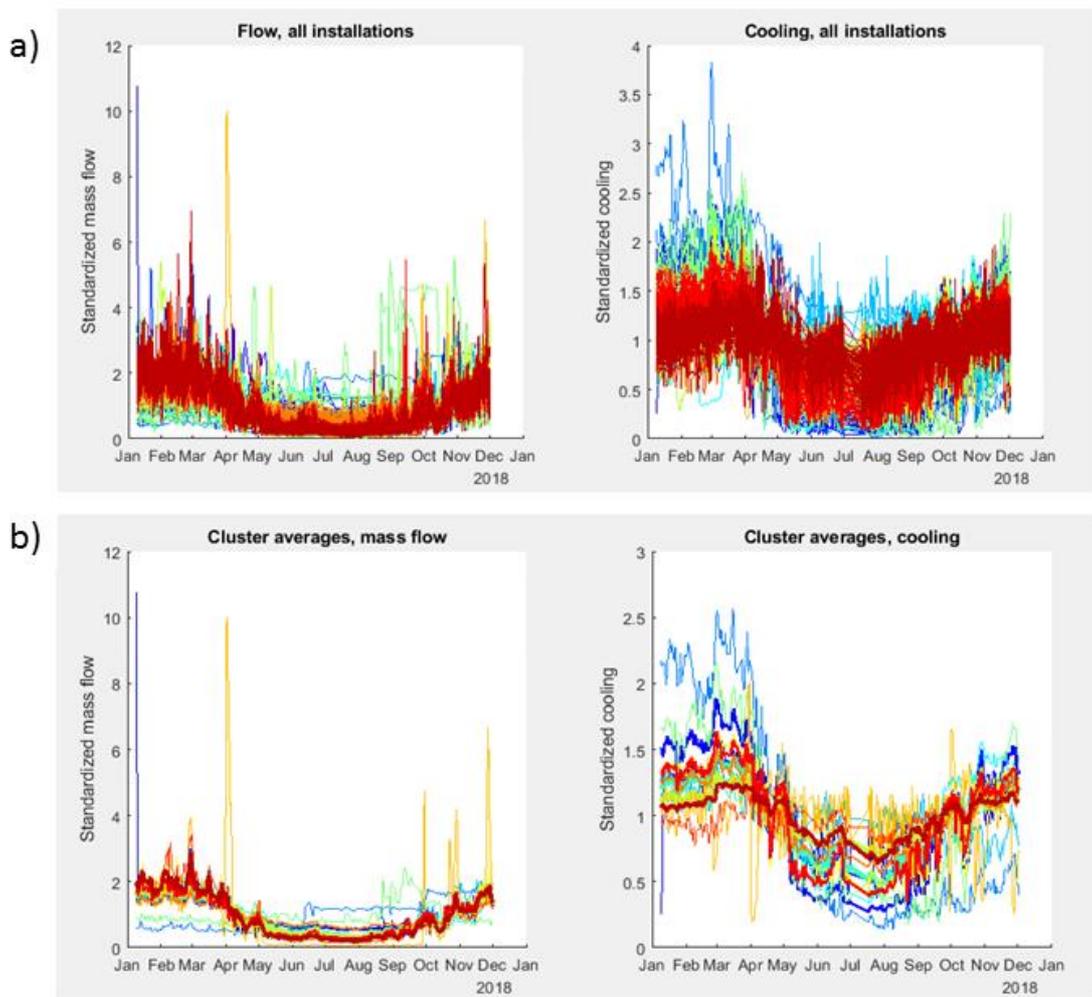


Figure 22: Clustering of 460 installations from a district in Aarhus using 15 clusters.

a) All installations, b) cluster averages (centroids).



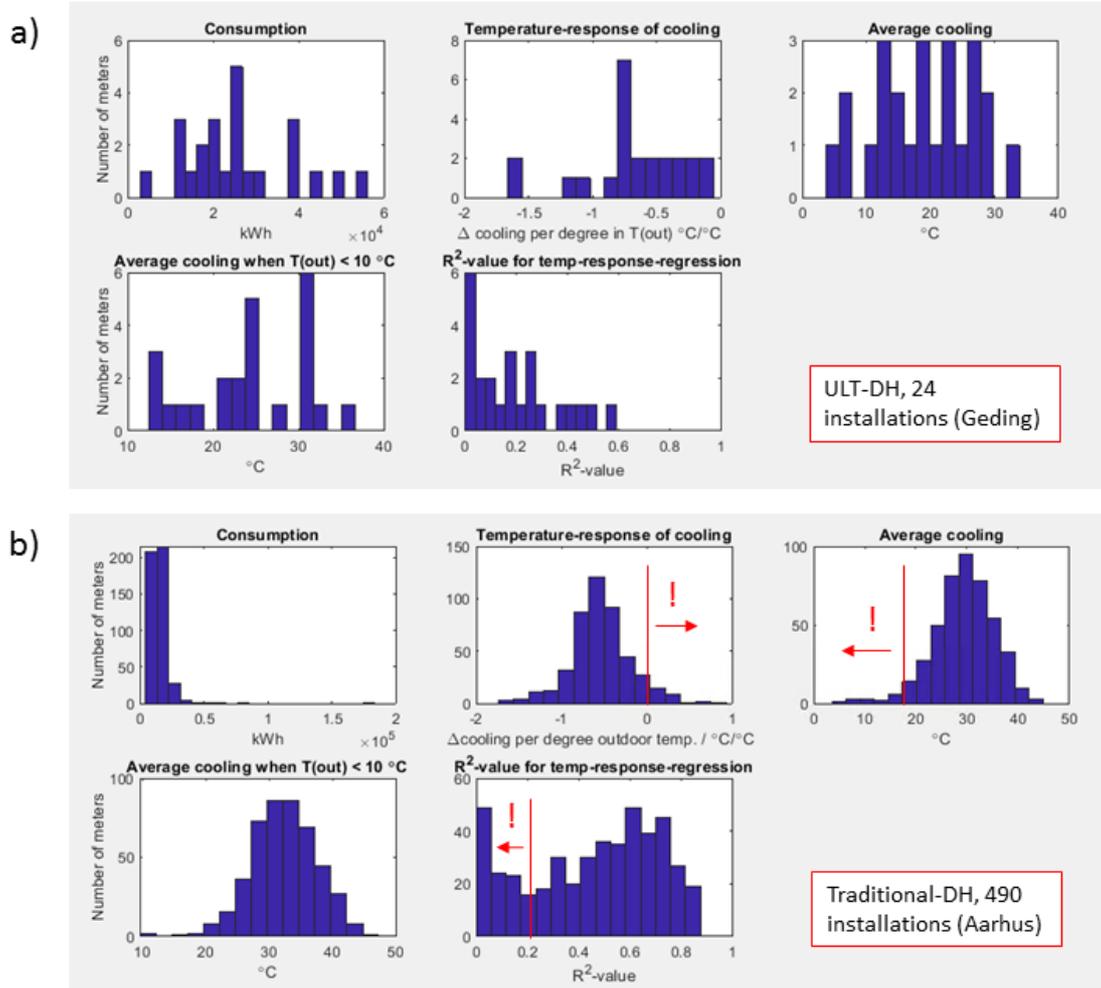


Figure 23: Statistics output from DH Doctor.

a) 24 ULT installations (Geding) and b) 460 installations from a traditional network (Aarhus).



11. Annex 2: DH Autotune

Annex 2 presents supplementary data to the sections in the main text describing the development and functions of DH Autotune.

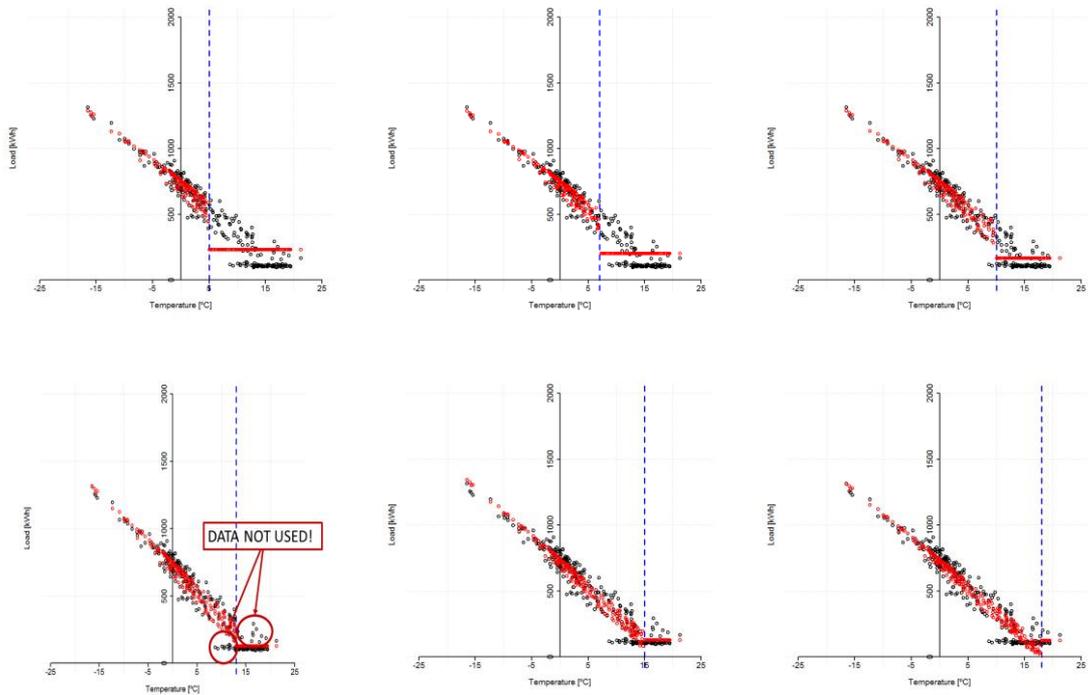


Figure 24: T algorithm applied to building 10045 (residential building with Domestic Hot Water consumption), for different T_{REF} .

The blue dotted line indicates the value of T_{REF} .



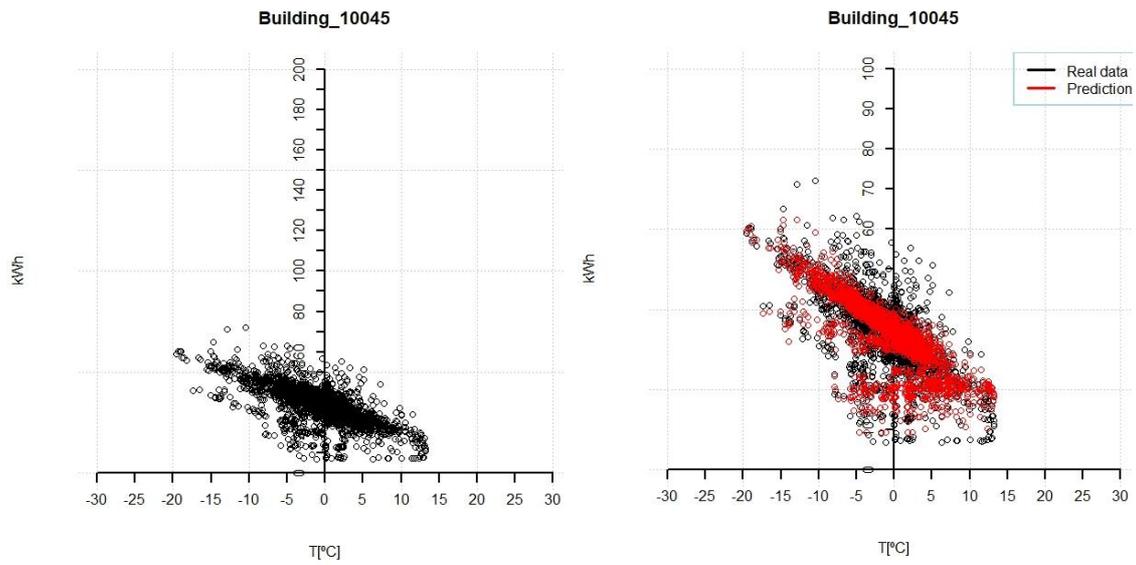


Figure 25: Q algorithm application after DT1 applied to test data from 2019, building 10045.

LEFT: Raw data. RIGHT: Predictions (red) on top of raw data (black).

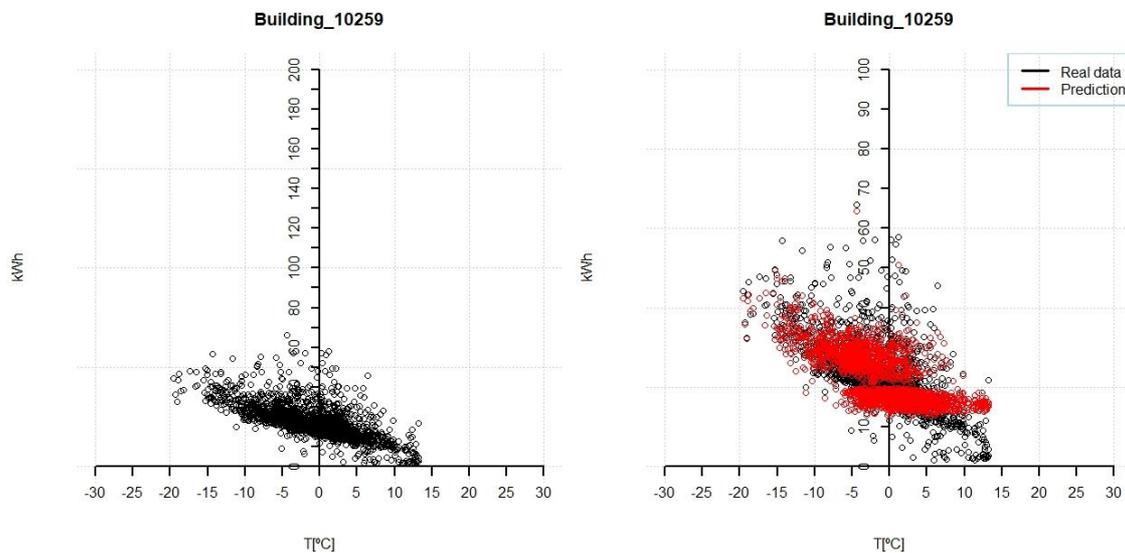


Figure 26: Q algorithm application after DT1 applied to test data from 2019, building 10259.



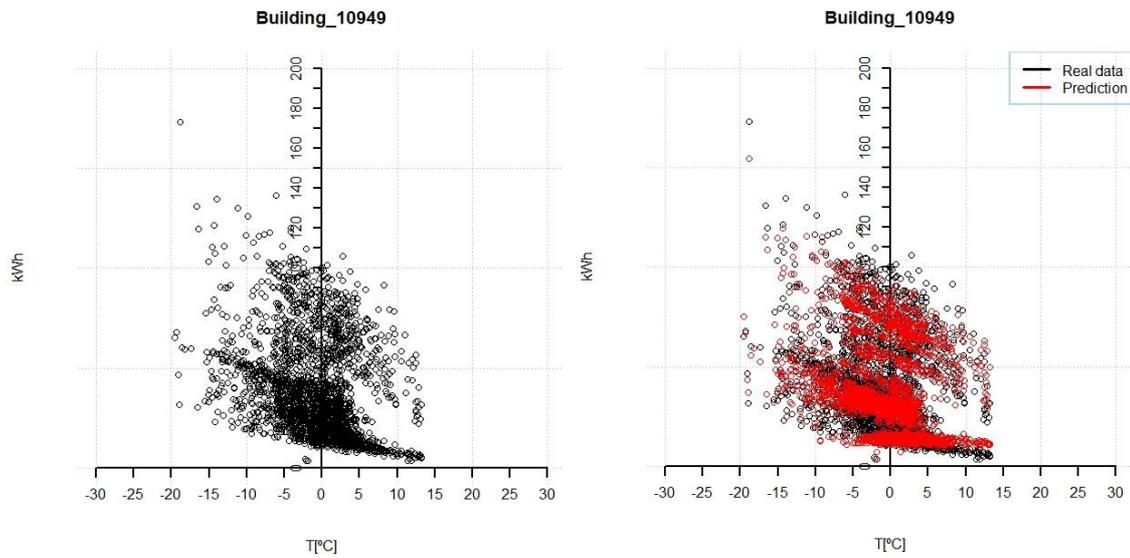


Figure 27: Q algorithm application after DT1 applied to test data from 2019, building 10949.

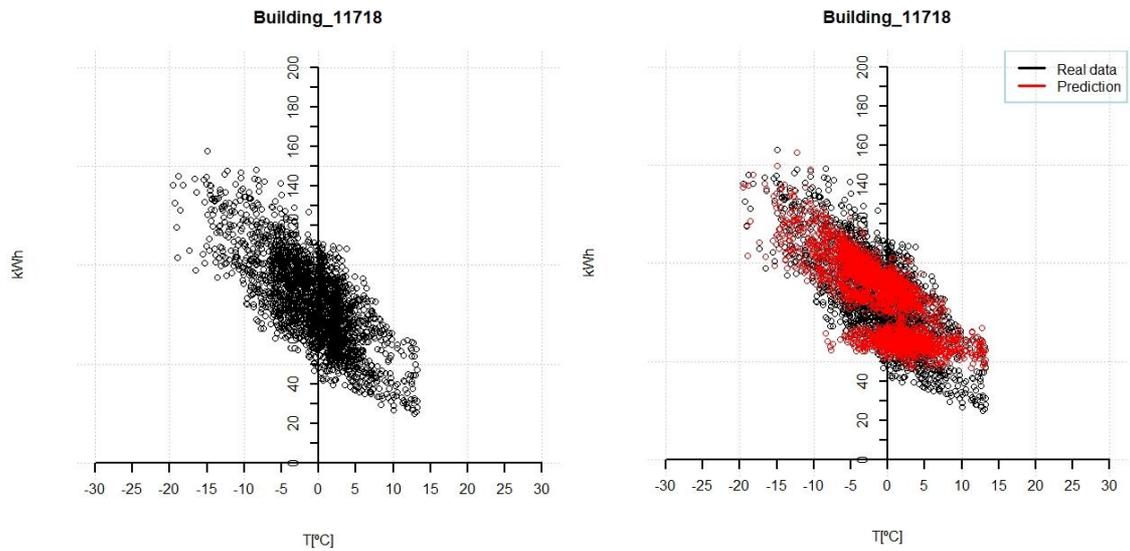


Figure 28: Q algorithm application after DT1 applied to test data from 2019, building 11718.



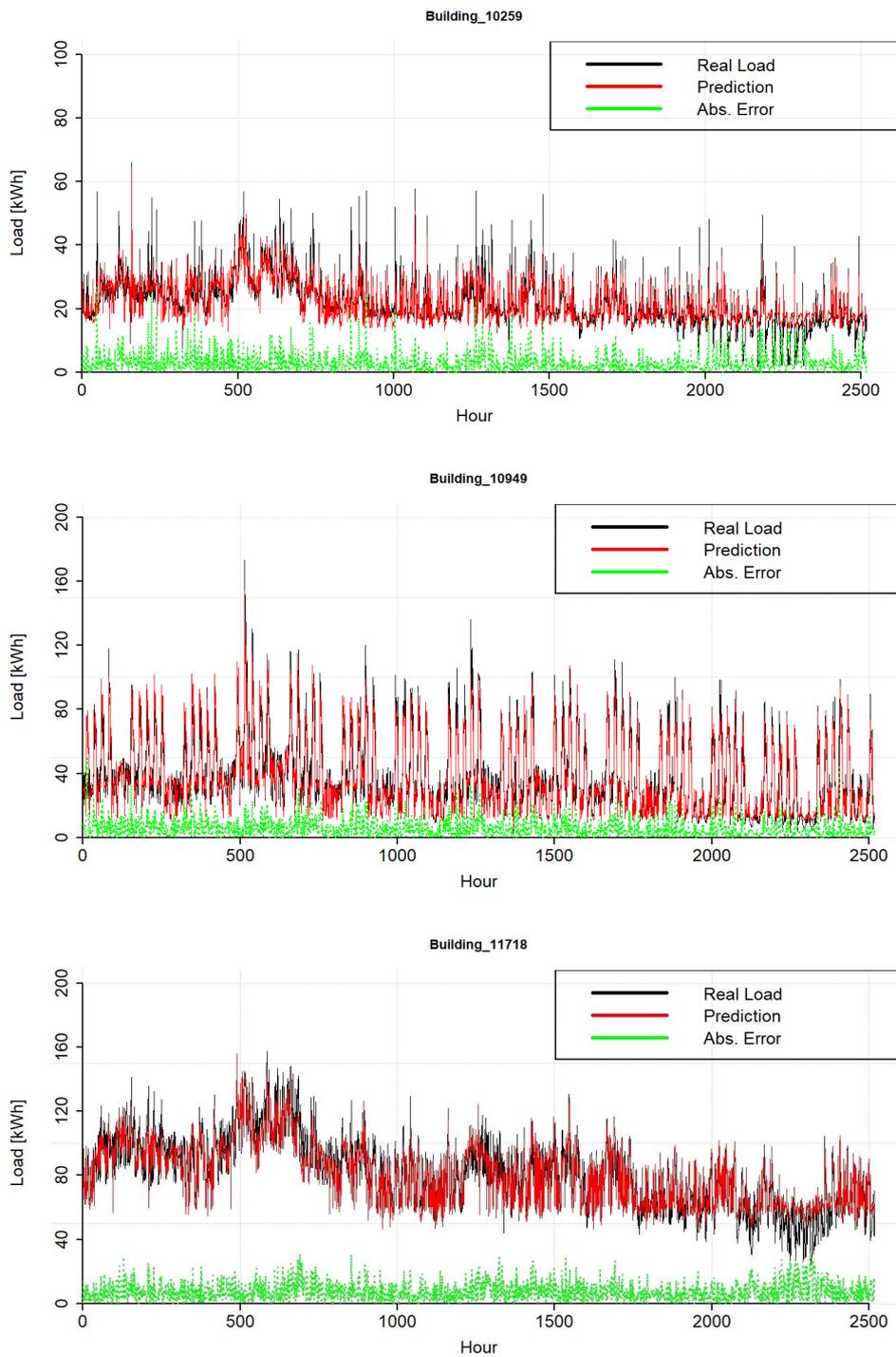


Figure 29: Temporal expression of regression models for the buildings 10259 (apartment), 10949 (kindergarten) and 11718 (office building).

