An Online Machine Learning Algorithm for Heat Load Forecasting in District Heating Systems

Spyridon Provatas

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden
This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 10 weeks of full-time studies.

Contact Information:
Author:
Spyridon Provatas
E-mail: sppr13@student.bth.se

External advisor:
Christian Johansson, PhD
Chief Technology Officer
NODA Intelligent Systems AB

University advisor:
Niklas Lavesson, PhD
Associate Professor of Computer Science
Dept. Computer Science & Engineering

Faculty of Computing
Blekinge Institute of Technology
SE-371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57
Abstract

Context. Heat load forecasting is an important part of district heating optimization. In particular, energy companies aim at minimizing peak boiler usage, optimizing combined heat and power generation and planning base production. To achieve resource efficiency, the energy companies need to estimate how much energy is required to satisfy the market demand.

Objectives. We suggest an online machine learning algorithm for heat load forecasting. Online algorithms are increasingly used due to their computational efficiency and their ability to handle changes of the predictive target variable over time. We extend the implementation of online bagging to make it compatible to regression problems and we use the Fast Incremental Model Trees with Drift Detection (FIMT-DD) as the base model. Finally, we implement and incorporate to the algorithm a mechanism that handles missing values, measurement errors and outliers.

Methods. To conduct our experiments, we use two machine learning software applications, namely Waikato Environment for Knowledge Analysis (WEKA) and Massive Online Analysis (MOA). The predictive ability of the suggested algorithm is evaluated on operational data from a part of the Karlshamn District Heating network. We investigate two approaches for aggregating the data from the nodes of the network. The algorithm is evaluated on 100 runs using the repeated measures experimental design. A paired T-test is run to test the hypothesis that the choice of approach does not have a significant effect on the predictive error of the algorithm.

Results. The presented algorithm forecasts the heat load with a mean absolute percentage error of 4.77%. This means that there is a sufficiently accurate estimation of the actual values of the heat load, which can enable heat suppliers to plan and manage more effectively the heat production.

Conclusions. Experimental results show that the presented algorithm can be a viable alternative to state-of-the-art algorithms that are used for heat load forecasting. In addition to its predictive ability, it is memory-efficient and can process data in real time. Robust heat load forecasting is an important part of increased system efficiency within district heating, and the presented algorithm provides a concrete foundation for operational usage of online machine learning algorithms within the domain.

Keywords: District Heating, Heat Load Forecasting, Machine Learning, Incremental/Online Learning
I would like to express my profound gratitude to my advisor Dr. Niklas Lavesson for motivating and challenging me throughout this work. I also want to thank him for helping and guiding me to complete this thesis, thus enabling me to strive for higher goals.

Moreover, I sincerely thank my external advisor Dr. Christian Johansson for introducing me to the district heating domain as well as for his valuable insights and feedback.

Last but not least, I am deeply grateful to my parents, Christos and Roula, for their relentless support throughout my life.
# Contents

Abstract

Acknowledgements

1 Introduction

2 Background
   2.1 District heating
   2.2 Machine learning

3 Related Work

4 Aim and Objectives

5 Research Questions

6 Method
   6.1 Algorithm
   6.2 Data collection
   6.3 Software Platform
   6.4 Experimental design
   6.5 Validity Threats

7 Results

8 Analysis

9 Conclusions and Future Work
   9.1 Practical Impact
   9.2 Future Work

References


Chapter 1

Introduction

Over the last few decades, district heating (DH) has been increasingly used for space heating, domestic hot water and industrial processes (Frederiksen & Werner, 2013). The reason lies in the fact that DH is an energy-efficient and environmentally sound way to supply heating (Lund, Möller, Mathiesen, & Dyre-lund, 2010; Mahapatra & Gustavsson, 2009). Combined Heat and Power (CHP) plants can be used to generate heat in DH systems (Johansson, Wernstedt, & Davidsson, 2012a). In CHP plants, electricity and heat are produced simultaneously. This results in efficiency that ranges between 80-90%, since CHP plants require less resources to produce the same amount of energy as separate heat and power systems (Johansson et al., 2012a; Johansson, Wernstedt, & Davidsson, 2012b). The efficiency expresses the ratio of the produced power and thermal output to the total fuel input. Regarding environmental benefits, CHP plants can use environmentally friendly resources such as biomass and renewable energy.

An important issue that exists in DH systems is the long delivery time of heat to the customers. When a production unit is activated, it might take hours to carry out all the processes that are required to generate heat. In addition, customers are usually geographically dispersed and many kilometres away from the production units. As a result, it can take several hours to produce and distribute the heat to the customers. This can lead to a situation where the heat is no longer needed at the moment of delivery, e.g. due to a raise in outdoor temperature. Consequently, heat suppliers aim at producing the amount of heat that is required to satisfy the customers’ demand at any given time, including heat losses in the distribution network. To achieve this, a reliable prediction of the heat load would be beneficial.

Heat load forecasting enables effective planning and management (Kvåranström, Liljedahl, & Dotzauer, 2006; Feinberg & Genethliou, 2005). By estimating the heat demand, heat suppliers can avoid producing superfluous heat. At the same time, heat suppliers can schedule which production units to activate. Normally, heat suppliers want to initially activate production units with lower operational costs (Frederiksen & Werner, 2013; Johansson, Wernstedt, & Davidsson, 2009). Moreover, heat load forecasting enhances the effectiveness of techniques such as demand side management and load control (Johansson et al., 2009; Johansson,
Chapter 1. Introduction

Wernstedt, & Davidsson, 2010; Wernstedt, Davidsson, & Johansson, 2007). These techniques are used to coordinate the heat consumption at the customers’ side. That way, heat suppliers can avoid the use of peak load boilers, which mostly use expensive and environmentally unfriendly fossil fuels. In addition, demand side management enables load shifting. Load shifting can be used in relation to CHP plants in order to synchronize peak demands in electricity and district heating. This technique is profitable for heat suppliers, since they can match their production with high spot-prices on the power market (Johansson et al., 2012a, 2012b).

The heat load is used for space heating and heating tap water. The demand for space heating mostly depends on the outdoor temperature. The demand for hot tap water is influenced by the social behaviour of the consumers, e.g. there is a difference in the need for hot tap water between weekdays and weekends (Frederiksen & Werner, 2013). The overall heat load exhibits a nonlinear, stochastic and non-stationary behaviour (Bacher, Madsen, Nielsen, & Péters, 2013), which limits the predictive capabilities of current approaches. To achieve an accurate prediction of the heat load, there is a need for models that can be incrementally updated in order to capture changes in the heat load over time. At the same time, modern systems for supervision collect massive amounts of data during the operation of heating grids.

For the aforementioned reasons, there is a need for adaptive, robust, memory-efficient, high-speed models that will be able to process real-time data. Traditional statistical and machine learning approaches cannot meet these requirements. A solution that has emerged over the last few years is online learning (Fon tenla-Romero, Guijarro-Berdiñas, Martínez-Rego, Pérez-Sánchez, & Peteiro-Barral, 2013; Shalev-Shwartz, 2007). Online learning algorithms possess the ability to adapt to changes in the distribution of the predictive target variable over time and are characterized by considerable computational efficiency. Online learning has been successfully used in several domains such as robotics (Sofman et al., 2006), autonomic wireless networks (Shiang & van der Schaar, 2010), handwritten character recognition (Almaksour, Mouchère, & Anquetil, 2008), web page classification (Singh, Sandhawalia, Monet, Poirier, & Coursimault, 2012) and semiconductor manufacturing (Guo, Hao, & Liu, 2012).
2.1 District heating

District heating is a system that supplies heat to residential, commercial and industrial buildings (Frederiksen & Werner, 2013; Kvarnström et al., 2006). Heat is generated in CHP plants or heat-only boilers and is distributed in the form of pressurized hot water—or steam in some DH systems—through a network of insulated pipelines. The temperature of the water normally varies between 60-120°C. At the consumer’s side, heat exchangers are used to transfer the heat from the DH network to the internal heating system of the building. Then, the water returns to the production units to be reheated. Figure 1 illustrates a DH system.

![Figure 1 A District Heating System](image)

The heat load in a DH system is defined as the amount of energy that is required for space-heating and heating tap water at any given moment in time. Space heating is highly dependent on outdoor temperature (Frederiksen & Werner, 2013). When the outdoor temperature is lower than the desired indoor
temperature, heat is transmitted from the inside of the building to the external environment through walls and windows. To cancel out this heat loss and maintain a desired indoor climate, it is necessary to supply heat to the building. The amount of heat that has to be supplied is proportional to the difference between the indoor and outdoor temperature. Space heating also includes heating the cold air that is used in ventilation systems. On the other hand, the demand for hot tap water is affected by the social behaviour of the consumers. For example, the heat demand is lower during the night since most people are asleep.

The input for heat load forecasting includes factors that affect both space heating and heating tap water. The most important factors that are usually used are outdoor temperature, time of the day and day of the week.

2.2 Machine learning

A branch of artificial intelligence that is used for forecasting is machine learning (ML). ML aims at building models that learn from data (Smola & Vishwanathan, 2008). The most common task in ML is supervised learning (Smola & Vishwanathan, 2008; Maimon & Rokach, 2005). In supervised learning, we build a model using training data that consist of N data records \((x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\) called instances. Each instance is described by an input vector \(x_i\) which consists of a set of attributes \(A = A_1, A_2, \ldots, A_m\) - and a label \(y_i\) of the target attribute that denotes the desired output. The goal is to train a model that can predict the label of new instances. Supervised learning tasks mainly include classification and regression. In classification problems, the label \(y_i\) takes discrete values, whereas in regression problems it takes continuous values. Table 1 shows the first five instances of three attributes from the Electric Bill dataset, which represents a regression problem (McLaren & McLaren, 2003). The outdoor temperature and the Heating Degree Days are used to predict the monthly household electric bill charges.

<table>
<thead>
<tr>
<th>Temperature (°F)</th>
<th>Heating Degree Days</th>
<th>Amount of bill (dollars)</th>
</tr>
</thead>
<tbody>
<tr>
<td>29.1</td>
<td>1229</td>
<td>162.10</td>
</tr>
<tr>
<td>31.5</td>
<td>999</td>
<td>256.90</td>
</tr>
<tr>
<td>41.9</td>
<td>734</td>
<td>151.15</td>
</tr>
<tr>
<td>53.4</td>
<td>373</td>
<td>118.76</td>
</tr>
<tr>
<td>63.7</td>
<td>162</td>
<td>100.71</td>
</tr>
</tbody>
</table>

Table 1 First five instances of the Electric Bill dataset

One of the most popular classification and regression models is the decision tree. Decision trees are easy for humans to interpret and can achieve high levels of predictive accuracy. On the other hand, decision trees tend to overfit, meaning that the tree that is generated might be very complex and therefore not generalize
Chapter 2. Background

the data well (Rokach, 2008). The effect of overfitting is reduced with mechanisms such as pruning, which decrease the complexity and the size of the tree. Another disadvantage of decision trees is their instability, since small variations on the training set can lead to a completely different tree. This problem is mitigated by the use of ensemble methods.

Decision trees consist of internal nodes, branches and terminal nodes (Quinlan, 1986; Maimon & Rokach, 2005). Each internal node corresponds to a particular attribute that splits the instance space into two or more subspaces, according to the possible values that the attribute can take. Each possible value represents a branch. The terminal nodes are the endpoints of the tree and represent the final output of the algorithm. In order to predict the label of an instance, the algorithm navigates the instance from the root node to a terminal node, according to the instance's attribute values along the path. Figure 3 shows a decision tree that can be constructed from the training set in Figure 2. Decision trees have been used in a plethora of real-world applications such as power systems (Rovnyak, Kretsinger, Thorp, & Brown, 1994), medicine (Kokol, Pohorec, Štiglic, & Podgorelec, 2012), object recognition (Wilking & Röfer, 2005), smart homes (Stankowski & Trnkoczy, 2006). The fact that decision trees have not been used in the DH domain, gives the opportunity for this work to fill the research gap.

![Figure 2 Structure of a decision tree for the Electric Bill Dataset](image)

Decision trees are widely used in ensemble learning. The main idea behind ensemble learning is to combine the output of multiple predictive models using voting mechanisms (Dietterich, 2000). Generating multiple predictive models provides diversity that often leads to more reliable and accurate predictions. Two of the most popular techniques for constructing an ensemble are bagging and
Chapter 2. Background

boosting (Oza, 2005). Bagging uses a sampling technique called Bootstrap Aggregating to generate multiple training datasets. Each dataset trains one model in the ensemble. The output of bagging is either the majority vote (in classification problems) or the average (regression problems) of all the models in the ensemble. Below we can see the pseudocode for bagging:

**Algorithm Bagging**

1: Given training data \((x_1, y_1), \ldots, (x_N, y_N)\)
2: for \(m = 1, \ldots, M\)
3: form training dataset \(S_m\) by selecting \(N\) random instances from the original training set \(S\) with replacement
4: compute \(h_m\), which is the output of the \(m^{th}\) classifier
5: end for
6: Final output of the ensemble: \(H = \text{MajorityVote}(h_1, \ldots, h_m)\) or \(H = \text{Average}(h_1, \ldots, h_m)\)

With boosting, the models are created sequentially, and at each step the weights of misclassified training instances are increased. Each model is assigned with a weight, according to the model’s predictive error. The pseudocode for boosting is as follows:

**Algorithm Boosting**

1: Given training data \((x_1, y_1), \ldots, (x_N, y_N)\)
2: Initialize a weight \(w_N\) for each instance and a weight \(k_m\) for each classifier
3: for \(m = 1, \ldots, M\)
4: Train the \(m^{th}\) classifier on all instances
5: Set a weight \(k_m\) for the \(m^{th}\) model according to its predictive error
6: Update the weight \(w_N\) of all instances
7: end for
8: Final output of the ensemble: \(H = \sum_{m=1}^{M}(k_my_m)\) where \(y_m\) is the output of the \(m^{th}\) model

Traditional ML algorithms are trained only once on a particular set of data, called the training set. The main drawback of this approach is that the algorithms cannot adjust to potential changes in the distribution of the target attribute. This behaviour is called concept drift (Elwell & Polikar, 2011) and it is very common in real-world settings. Incremental learning is an effective way to handle concept drift. Incremental algorithms possess the following capabilities (Giraud-Carrier, 2000; Jain, Lange, & Zilles, 2006):

- Update their model whenever new information becomes available
Chapter 2. Background

- Maintain previously acquired useful knowledge
- Learn new concepts that might be introduced
- Do not require access to the data that were used to train the model

Incremental learning can be done offline or online. In online learning, each example is processed only once to update the model, and then it is discarded without the need to store it in the memory (Smola & Vishwanathan, 2008). Offline learning involves the storage of the instances in the memory, so that they can be used in case the model has to be retrained.
Heat load forecasting in DH systems has received a lot of attention in the last 15 years. There are several proposed approaches, which mainly include statistical and ML models.

Most statistical models comprise a temperature-dependent component and a social component, which is related to the social behaviour of the consumers. Dotzauer (2002) proposed a simple model, in which the temperature-dependent component is represented as a piecewise linear function, and the social component is equal to a constant value for each day of the week. Nielsen and Madsen (2006) used a grey-box approach that combines physical knowledge with statistical modelling. The physical knowledge provides a general structure of the model, considering heat transfer through walls, heat transfer through windows, ventilation as well as the social behaviour of the consumers. A statistical modelling process is then used to calculate the actual coefficients of the model. Chramcov (2010) used a polynomial function for the temperature-dependent component, and a Box-Jenkins methodology for the social component. The Box-Jenkins methodology applies an autoregressive moving average (ARMA) model. Grosswindhager, Voigt and Kozek (2011) published a study in which the non-stationary behaviour of the heat load is captured with a Seasonal Autoregressive Integrated Moving Average (SARIMA) that is embedded in a state space framework. The forecasting values are calculated using classical Kalman Recursion. The influence of outdoor temperature is described as a piecewise linear function. ML algorithms are capable of dealing with the nonlinear and non-stationary behaviour of the heat load. The most widely used ML approach to forecast the heat load in DH systems is Neural Networks (Grzenda & Macukow, 2006; Kato, Sakawa, Ishimaru, Ushiro, & Shibano, 2008; Park, Kim, Kim, Jo, & Yeo, 2010).

To our knowledge, neither online ML algorithms nor decision tree-based ML algorithms have been applied to heat load forecasting. We address this identified research gap by investigating an ML method that mixes both techniques. This method efficiently processes streaming data and it generates and calibrates comprehensible models that can perform heat load forecasting in DH systems.
The aim of this thesis is to propose a robust model that can accurately predict the heat load in DH systems. To accomplish this aim, we follow a sequence of measurable objectives. First of all, the operational data from the DH system contain measurement errors, missing values and outliers. We implement a mechanism for data preprocessing to impute missing values and handle outliers. The next step entails the selection of a decision-tree based algorithm to be used in the experiments. We make the algorithm compatible to regression problems and perform parameter tuning. Then, we design and conduct experiments using the operational data. The final step is to perform statistical analysis to test the significance of the experimental results.
Chapter 5

Research Questions

1. Are incremental online decision trees more accurate than other incremental online models in predicting the heat load in DH systems?

   In this RQ, we investigate whether incremental online decision trees can predict the heat load more accurately than other incremental online state-of-the-art algorithms. Moreover, the results will indicate the extent to which the predicted values can approximate the actual values, in terms of Kilowatts. The acceptable error may vary among different DH systems.

2. Which approach for aggregating the heat load in a DH system results in lower prediction error?

   The customer heat load in a DH network is calculated by aggregating the heat load of all the nodes in the network. This study investigates the impact of two different approaches for heat load aggregation. Approach 1 is to create one model for each node and aggregate the predictions of all the nodes. Approach 2 is to aggregate first the heat load of all the nodes and then create one model to predict the aggregated heat load.

3. What is the impact of using different ensemble methods on the prediction error of a decision tree?

   The purpose of this RQ is to compare the effect of bagging and boosting on the prediction error of a decision tree.

4. What is the impact of using an incremental learning technique on the prediction error of a decision tree?

   In this RQ, we investigate the effect of increasing sequentially the weight of new instances on the prediction error of a decision tree.
Chapter 6

Method

6.1 Algorithm

We apply online bagging to create an ensemble of decision trees (Dietterich, 2000). For the purpose of this study, we extend the implementation of online bagging to make it compatible to regression problems. The base model of the ensemble is the Fast Incremental Model Trees with Drift Detection (FIMT-DD) algorithm, which was introduced by Ikonomovska et al. in 2006. FIMT-DD is a state-of-the-art online decision tree that is used for regression. The tree is constructed as follows: when a new instance arrives, FIMT-DD traverses the instance to a terminal node and updates the necessary statistics for this node. Then the algorithm checks if the splitting criterion is satisfied, in order to decide on whether this node should be further expanded. To predict the target value of an instance, FIMT-DD calculates a weighted average of the instance’s attributes. FIMT-DD has also the ability to detect concept drift and adjust to non-stationary environments. When concept drift is detected, FIMT-DD grows subtrees in order to replace parts of the tree that are not relevant for the new concept.

Due to measurement or communication errors, the data from a DH system can often contain missing values and outliers. We implement a mechanism that handles missing values and outliers in an online fashion, and incorporate it to the ML algorithm. Missing values for the temperature and the heat load are filled in by the average value of measurements from the previous 6 hours, since the most recent values are more relevant and can approximate the missing values to a sufficient extent. Outlier detection is carried out through a statistical approach. The algorithm computes the mean (μ) and standard deviation (σ) of the 50 most recent values for the heat load, and then a threshold is used in order to determine whether a value for the heat load is an outlier. A value x is considered an outlier if it satisfies one of the following conditions:

\[ x < \mu - 3\sigma \]
\[ x > \mu + 6\sigma \]

These thresholds were decided after parameter tuning. Outliers are not discarded. Instead, the algorithm removes the outlier value and imputes a new value
with the same technique that is used for imputing missing values.

Figure 3 Hourly heat load measurements with respect to outdoor temperature for a residential building

6.2 Data collection

The operational data consist of hourly measurements from 26 building substations in a part of Karlshamn DH network in the south of Sweden. The data was collected from 01-10-2013 to 31-03-2014. The nodes comprise residential buildings, commercial buildings and a school. Figure 3 shows the heat load for a residential building with respect to the outdoor temperature. For a given temperature, the higher values of the heat load occur during the day due to the increased consumption of hot tap water. For a commercial building (Figure 4) there is a higher variation since during the night and weekends there is a limited number
of people inside the building as well as the ventilation system does not operate at full capacity (Frederiksen & Werner, 2013).

To answer research questions 3 and 4, we use 9 benchmark datasets for regression and the operational data from the DH system. Table 2 presents the 10 datasets.

6.3 Software Platform

We conduct experiments using two software applications, namely Waikato Environment for Knowledge Analysis (WEKA) (Remco et al., 2013) and Massive Online Analysis (MOA) (Bifet & Kirkby, 2009). WEKA and MOA are open-source, written in Java and compatible with each other. They are widely used.
for ML tasks such as classification, regression and clustering. WEKA mainly contains batch algorithms, whereas MOA is a framework for online learning from data streams. Both applications can be used for data preprocessing and running experiments either through a Graphical User Interface or an API. The API is also used for developing new algorithms.

### 6.4 Experimental design

An experiment is a very common research method in Computer Science. It involves the manipulation of independent variables in order to discover the effect on a dependent variable. The combination of different conditions of the independent variables are called treatments. The experimental units are the recipients of experimental treatments. In the context of this work, the experimental units are the hourly measurements of the DH data, which constitute the instances of the dataset. The dataset contains 4344 instances. Each instance is characterized by the following attributes: time, weekday (true or false) and outdoor temperature.

For research questions 3 and 4, the experiments include also 9 more benchmark datasets for regression. The type of experimental design that is used in this study is the quasi experiment (Reichardt, 2009). The main aspect of the quasi experiments is that there is no random assignment of the experimental units to the different conditions of the independent variable. Instead, the investigators can manipulate how the experimental units are assigned to the different conditions of the independent variable. In this study, we use the same instances to test each condition of the independent variable. This design is called repeated measures, and results in a reliable comparison between different algorithms, thus ensuring internal validity (Ellis, 1999). The reason is that we evaluate all the algorithms
on the same instances and therefore we can determine which algorithm performs better for a given dataset. The evaluation method that is used in all the experiments is the prequen tial evaluation, i.e. each instance of the dataset is used to test the algorithm, and then the same instance is used to update the algorithm’s model. The instances arrive successively from the data stream.

The primary aim of the experiments is to evaluate the performance of the proposed algorithm in terms of the error in predicting the heat load. For this purpose, we use two of the most common evaluation metrics for regression, namely Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE), which are defined as follows

\[
MAE = \frac{\sum_{i=1}^{N} |p_i - y_i|}{N}
\]

\[
MAPE = \frac{\sum_{i=1}^{N} \frac{|p_i - y_i|}{p_i}}{N} \times 100
\]

where \(p_i\) is the actual value, \(y_i\) is the predicted value and \(N\) is the number of data points.

With respect to heat load forecasting, the MAE represents the average magnitude of the difference between the actual and predicted heat load in terms of kW. The MAPE is a relative metric, i.e. it does not depend on the scale of the values, and it is used to express the average error of a predictive model as a percentage.

The performance of the algorithm, in terms of MAE and MAPE, is compared with the performance of two other online algorithms, namely AMRules (Almeida, Ferreira, & Gama, 2013) and IBLStreams (Shaker & Hüllermeier, 2012).

All predictions are made 36 hours ahead.

We evaluate the proposed algorithm on 100 runs, using two approaches for aggregating the heat load. In relation to the second research question, we define the following null and alternative hypotheses:

\(H_{01}\): Using different approaches for aggregating the heat load in DH systems does not have a significant effect on the MAE of a decision-tree based algorithm.

\(H_{11}\): Using different approaches for aggregating the heat load in DH systems has a significant impact on the MAE of a decision-tree based algorithm.

The dependent variable of these hypotheses is the MAE of the proposed algorithm, whereas the independent variable is the different approaches that can be used for aggregating the heat load in DH systems.

Regarding the experiments for research questions 3 and 4, we compute the average of 10 runs. Before each run, we randomize the sequence of the instances.
Chapter 6. Method

The null and alternative hypotheses for research question 3 are:

\[ H_{02} : \text{Using different ensemble techniques has no significant effect on the MAE of a decision tree.} \]
\[ H_{12} : \text{Using different ensemble techniques has a significant effect on the MAE of a decision tree.} \]

The dependent variable of these hypotheses is the MAE of the FIMT-DD decision tree, whereas the independent variable is the different ensemble techniques. The two conditions of the independent variable are online bagging and online boosting. The number of trees in the ensemble for the two techniques is selected after parameter tuning for values 20, 30, 40, 50, 60, 70.

The null and alternative hypotheses for research question 4 are:

\[ H_{03} : \text{Increasing the weight of new instances has no significant effect on the MAE of a decision tree.} \]
\[ H_{13} : \text{Increasing the weight of new instances has a significant effect on the MAE of a decision tree.} \]

The dependent variable of these hypotheses is the MAE of the FIMT-DD decision tree, whereas the independent variable is the different incremental learning techniques. The two conditions of the independent variable are equal weight of new instances and increasing weight of new instances. The step for increasing the weight of the instances is selected after parameter tuning for values 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01.

6.5 Validity Threats

- **Internal validity**

  Internal validity refers to the extent to which changes in the dependent variable are caused by the independent variable (Kirk, 1982). To ensure internal validity in our experiments, we compare the algorithms by evaluating them on the same datasets, keeping also a fixed parameter configuration for the algorithms throughout the experiments. Moreover, noisy or corrupted data can be a threat to internal validity. For that reason, we perform data preprocessing to clean the data, e.g. fill in missing values, identify and remove outliers.

- **External validity**

  External validity refers to the extent to which we can generalize the experimental results (Kirk, 1982). Given that the operational data are collected
from a network that includes different types of buildings (residential, commercial as well as a school) and that the measurements span six months (October- April), we can be confident that the results can be generalized for bigger networks. Regarding research questions 3 and 4, we use 10 datasets (including the dataset from the DH system), and therefore we cannot generalize our results for every dataset. To mitigate this threat, we perform statistical tests to ensure statistical significance of our results.

- **Construct validity**

Construct validity refers to the extent to which a test measures what it is supposed to be measuring (Austin, Boyle, & Lualhati, 1998). In this study, we evaluate algorithms and techniques in terms of prediction error. The evaluation metrics that are used to characterize the prediction error are MAE and MAPE. These are two of the most commonly used statistics for evaluating the prediction of a regression model.

- **Statistical Conclusion validity**

Statistical Conclusion validity refers to the extent to which the conclusions we reach about the relationships between the variables are correct (Martin, Cohen, & Champion, 2013). To increase statistical conclusion validity, we ensure that our data are reliable. We choose 9 benchmark datasets that have been previously used in research studies. Moreover, the operational data are provided by an energy company (NODA Intelligent Systems AB). After conducting the experiments, we perform suitable statistical tests to ensure the statistical significance of our findings.
Chapter 7

Results

The predictive ability of the proposed algorithm is evaluated by conducting experiments for the two approaches of data aggregation. Approach 1 creates one model for each node and aggregates the predictions of all the nodes. Approach 2 aggregates first the heat load of all the nodes and then creates one model to predict the aggregated heat load. Figures 5 and 6 illustrate the predicted values, difference (actual - predicted), and evaluation metrics for Approaches 1 and 2 respectively. Table 3 shows the MAE and MAPE (using the aggregation Approach 2) for the proposed algorithm as well as two other online algorithms.

Regarding research questions 3 and 4, Tables 4 and 5 show the average of MAE and MAPE for the 10 runs of FIMT-DD, FIMT-DD (Bagging), FIMT-DD (Boosting) and FIMT-DD (Increasing Weight).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MAE</th>
<th>MAPE(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FIMT-DD (Bagging)</td>
<td>AMRules</td>
</tr>
<tr>
<td>DH data</td>
<td>78.714</td>
<td>182.562</td>
</tr>
</tbody>
</table>

Table 3 MAE and MAPE for FIMT-DD (using Bagging), Adaptive Model Rules (AMRules) and Instance Based Learner on Streams (IBLStreams)
## Chapter 7. Results

### Table 4 MAE and MAPE for FIMT-DD and FIMT-DD (Increasing Weight)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MAE</th>
<th>MAPE(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FIMT-DD (Increasing Weight)</td>
<td>FIMT-DD (Increasing Weight)</td>
</tr>
<tr>
<td>DH data</td>
<td>141.392</td>
<td>117.176</td>
</tr>
<tr>
<td>kinematics</td>
<td>0.13360</td>
<td>0.13213</td>
</tr>
<tr>
<td>bank8fm</td>
<td>0.03969</td>
<td>0.03115</td>
</tr>
<tr>
<td>2dplanes</td>
<td>1.02565</td>
<td>1.27906</td>
</tr>
<tr>
<td>fried</td>
<td>1.68793</td>
<td>1.64085</td>
</tr>
<tr>
<td>puma8nh</td>
<td>3.02209</td>
<td>2.88393</td>
</tr>
<tr>
<td>ailerons</td>
<td>3.12E-04</td>
<td>3.01E-04</td>
</tr>
<tr>
<td>puma32H</td>
<td>0.02206</td>
<td>0.01979</td>
</tr>
<tr>
<td>bank32NH</td>
<td>0.06921</td>
<td>0.05454</td>
</tr>
<tr>
<td>elevators</td>
<td>0.00454</td>
<td>0.00468</td>
</tr>
</tbody>
</table>

### Table 5 MAE and MAPE for FIMT-DD, FIMT-DD (Bagging), FIMT-DD (Boosting)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MAE</th>
<th>MAPE(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FIMT-DD (Bagging)</td>
<td>FIMT-DD (Boosting)</td>
</tr>
<tr>
<td>DH data</td>
<td>141.392</td>
<td>94.6270</td>
</tr>
<tr>
<td>kinematics</td>
<td>0.13360</td>
<td>0.12660</td>
</tr>
<tr>
<td>bank8fm</td>
<td>0.03969</td>
<td>0.03006</td>
</tr>
<tr>
<td>2dplanes</td>
<td>1.02565</td>
<td>1.11643</td>
</tr>
<tr>
<td>fried</td>
<td>1.68793</td>
<td>1.60432</td>
</tr>
<tr>
<td>puma8nh</td>
<td>3.02209</td>
<td>2.81384</td>
</tr>
<tr>
<td>ailerons</td>
<td>3.12E-04</td>
<td>3.13E-04</td>
</tr>
<tr>
<td>puma32H</td>
<td>0.02206</td>
<td>0.01924</td>
</tr>
<tr>
<td>bank32NH</td>
<td>0.06921</td>
<td>0.04812</td>
</tr>
<tr>
<td>elevators</td>
<td>0.00454</td>
<td>0.00455</td>
</tr>
</tbody>
</table>

Table 4 MAE and MAPE for FIMT-DD and FIMT-DD (Increasing Weight)

Table 5 MAE and MAPE for FIMT-DD, FIMT-DD (Bagging), FIMT-DD (Boosting)
Figure 5 Predictions and evaluation metrics for Approach 1
Figure 6 Predictions and evaluation metrics for Approach 2
The experimental results show that the proposed online algorithm has a MAPE of 4.77% which is lower than or at least equal to the mean percentage error of state-of-the-art approaches that have been proposed for heat load forecasting in DH systems. However, a direct comparison is not possible due to different operational data, configurations, and experimental design. The results are even more important if we take into account that online learning has an inherent disadvantage in comparison to traditional ML algorithms. This disadvantage pertains to the fact that traditional ML algorithms are fully trained with all the available data before the evaluation, whereas in online algorithms the training and the evaluation evolve simultaneously as new instances arrive from the data stream. The results also show that the proposed algorithm has lower error than two other online algorithms (AMRules and IBLStreams).

Regarding the experiments for research questions 3 and 4, we perform statistical analysis to test the significance of the results. Since each data point in the samples comes from a different dataset, we cannot assume a Gaussian distribution for the population. Hence, the appropriate test is the non-parametric Wilcoxon signed-rank test (Walpole, Myers, Myers, & Ye, 1993). To test the hypothesis that is related to research question 3, we run the two-sided Wilcoxon test using a confidence level of 0.05 and the result is a p-value of 0.009766. The p-value is the probability of obtaining a test statistic as extreme as the one that was observed, assuming that the null hypothesis is true. Since the probability of 0.97% is lower than the 5% of the confidence level, it means that the difference between the two techniques is not very likely given that the null hypothesis is true, and therefore we reject the null hypothesis. To test the hypothesis that is related to research question 4, we run the two-sided Wilcoxon test using a confidence level of 0.05 and the result is a p-value of 0.1055. Since the p-value is higher than the confidence level, we fail to reject the null hypothesis.

Another aspect of the experiments on the operational data is related to the propagation of the error with regard to different ways of data aggregation. Table 6 shows the mean and standard deviation of the MAE and MAPE for the two approaches for the 100 runs of the experiments. In addition, Table 6 shows the results of the Student T-Test which is performed to determine whether there is
a significant difference in the prediction error between the two approaches. The p-value is considerably lower than the significance level (0.05). This means that the error is significantly lower when we aggregate first the heat load of the 26 nodes and then build one model to make the predictions. The use of individual models for each node leads to higher error.

<table>
<thead>
<tr>
<th></th>
<th>Approach 1</th>
<th>Approach 2</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>85.063</td>
<td>78.714</td>
<td>22E-18</td>
</tr>
<tr>
<td></td>
<td>(0.019)</td>
<td>(0.007)</td>
<td></td>
</tr>
<tr>
<td>MAPE (%)</td>
<td>5.135</td>
<td>4.770</td>
<td>22E-18</td>
</tr>
<tr>
<td></td>
<td>(0.002)</td>
<td>(0.001)</td>
<td></td>
</tr>
</tbody>
</table>

Table 6 Mean, Standard Deviation and statistical analysis of 100 runs for Approaches 1 and 2

Finally, a key feature of the proposed model is its ability to handle missing values and outliers, which increases the robustness of the model to noise and measurement errors.
Chapter 9

Conclusions and Future Work

This work investigates the potential benefits of the application of online learning to the DH domain. We present an ensemble of decision trees that is able to capture the nonlinear, stochastic, non-stationary behaviour of the heat load. Experimental results on operational data show that the model possesses a strong predictive ability.

The model is memory-efficient, since it does not require the storage of data in the memory. Each instance is discarded after it updates the model. Another important feature of the algorithm is the ability to learn incrementally and process massive amounts of data in real time. Due to all the aforementioned advantages, online learning is increasingly used in real-world applications in order to process and analyse high volumes of data in real time. This work provides a foundation for further use of related ML algorithms in the domain of DH systems.

9.1 Practical Impact

An important aspect of this study is the potential benefits for both the industry and the society. Heat load forecasting assists in effective planning and management of DH systems. Thus, heat supplies can reduce operational costs and increase their turnover when selling to the power market the produced electricity from the CHP plants. Furthermore, heat load forecasting can lead to major environmental benefits. Heat suppliers can plan more effectively the base production and possibly avoid the activation of environmentally unsound peak-load boilers. This environmental aspect is critical for European countries on the grounds that the European Union has set environmental targets in 2020 that include a 20% reduction of greenhouse gas emissions below 1990 levels.

9.2 Future Work

One possible direction to extend this work is the implementation of a more sophisticated technique to handle missing data and outliers. Measurement errors occur frequently in real-world applications and therefore there is a need for a
robust technique that will handle missing data and outliers more effectively. This technique has to take into account the different type of buildings that exist in a DH network. The behaviour of the heat load varies according to the type of building and therefore an optimal solution cannot be achieved when the same approach is used for all types of buildings. A possible solution to this problem is the implementation of an automated parameter tuning technique that will tune parameters and thresholds according to the behaviour of the heat load for each building.

Another challenge for future work is to use meta-learning for parameter tuning and algorithm selection. Meta-learning is a technique that considers various features of a given dataset - such as number of instances, number of attributes, correlation between attributes - and selects an algorithm with a specific parameter configuration that is expected to perform better on that particular dataset. Thus, meta-learning automates the time-consuming task of finding the most suitable algorithm from a plethora of existing algorithms, and for that reason it is increasingly used in the ML domain.
References


