



DEGREE PROJECT IN MATHEMATICS,  
SECOND CYCLE, 30 CREDITS  
*STOCKHOLM, SWEDEN 2016*

# **Predicting Hourly Residential Energy Consumption using Random Forest and Support Vector Regression**

An Analysis of the Impact of Household  
Clustering on the Performance Accuracy

**WILLIAM HEDÉN**



# Predicting Hourly Residential Energy Consumption using Random Forest and Support Vector Regression

An Analysis of the Impact of Household Clustering on the Performance Accuracy

W I L L I A M   H E D É N

Master's Thesis in Mathematical Statistics (30 ECTS credits)  
Master Programme in Applied and Computational Mathematics (120 credits)  
Royal Institute of Technology year 2016  
Supervisor at Watty: Anders Huss  
Supervisor at KTH: Timo Koski  
Examiner: Timo Koski

TRITA-MAT-E 2016: 20  
ISRN-KTH/MAT/E--16/20-SE

Royal Institute of Technology  
*SCI School of Engineering Sciences*

**KTH** SCI  
SE-100 44 Stockholm, Sweden

URL: [www.kth.se/sci](http://www.kth.se/sci)



## Abstract

The recent increase of smart meters in the residential sector has lead to large available datasets. The electricity consumption of individual households can be accessed in close to real time, and allows both the demand and supply side to extract valuable information for efficient energy management. Predicting electricity consumption should help utilities improve planning generation and demand side management, however this is not a trivial task as consumption at the individual household level is highly irregular.

In this thesis the problem of improving load forecasting is addressed using two machine learning methods, Support Vector Machines for regression (SVR) and Random Forest. For a customer base consisting of 187 households in Austin, Texas, predictions are made on three spatial scales: (1) individual household level (2) aggregate level (3) clusters of similar households according to their daily consumption profile. Results indicate that using Random Forest with  $K = 32$  clusters yields the most accurate results in terms of the coefficient of variation. In an attempt to improve the aggregate model, it was shown that by adding features describing the clusters' historic load, the performance of the aggregate model was improved using Random Forest with information added based on the grouping into  $K = 3$  clusters. The extended aggregate model did not outperform the cluster-based models.

The work has been carried out at the Swedish company Watty. Watty performs energy disaggregation and management, allowing the energy usage of entire homes to be diagnosed in detail.



## Sammanfattning

Den senaste tidens ökning av smarta elmätare inom bostadssektorn medför att vi har tillgång till stora mängder data. Hushållens totala elkonsumption är tillgänglig i nära realtid, vilket tillåter både tillgångssidan och efterfrågesidan att nyttja informationen för effektiv energihantering. Att förutsäga elförbrukningen bör hjälpa elbolag att förbättra planering för elproduktion och hantering av efterfrågesidan. Dock är detta inte en trivial uppgift, då elkonsumtionen på individuell husnivå är mycket oregelbunden.

Denna masteruppsats föreslår att använda två välkända maskininlärningsalgoritmer för att lösa problemet med att förbättra lastprognoser, och dessa är Support Vector Machines för regression (SVR) och Random Forest. För en kundbas bestående av 187 hushåll i Austin, Texas, gör vi prognoser baserat på tre tillvägagångssätt: (1) enskilda hushåll (2) aggregerad nivå (3) kluster av liknande hushåll enligt deras dagliga förbrukningsprofil. Resultaten visar att Random Forest med  $K = 32$  kluster ger de mest precisa resultaten i termer av variationskoefficienten. I ett försök att förbättra den aggregerade modellen visade det sig att genom att lägga till ytterligare prediktionsvariabler som beskriver klustrens historiska last, kunde precisionen förbättras genom att använda Random Forest med information från  $K = 3$  olika kluster. Den förbättrade aggregerade modellen presterade inte bättre jämfört med de klusterbaserade modellerna.

Arbetet har utförts vid det svenska företaget Watty. Watty utför energidisaggregering och energihantering, vilket gör att bostäders energianvändning kan analyseras i detalj.



# Acknowledgments

I want to extend a great thank you to Anders Huss and the team at Watty for giving me the opportunity to investigate the subject of this thesis, and to my supervisor at KTH, Timo Koski, for steering me in the right direction in times of need. I want to thank my mother Ann-Louise Kinder, my stepfather Johan Kinder, and my late father Per Hedén, who encouraged me to seek a higher education and always supported me in my choices. To Bianca, my outstanding life-companion and best friend, whom listened endlessly to my complaints and came with many helpful advice - thank you for always being there for me, you are irreplaceable.



# Machine Learning Terminology

Dataset	A set of instances with a description of the attributes of the instances.
Training set	The partition of the dataset used to train the model and fit its internal parameters.
Test set	The partition of the dataset, disjoint from the training set, used to test the generalization accuracy of the model.
Feature, attribute, predictor	A variable believed to influence the outcome of the prediction.
Feature vector	A list of features describing an instance from the dataset.
Target, true value	The variable that the model seeks to predict.
Classifier, regressor	A trained machine learning algorithm that takes as input a feature vector and returns either a label (classifier) or a continuous value (regressor).
Accuracy (error)	The rate of correct (incorrect) predictions made by the model, measured by a predefined function.
Cross-validation	A method for testing accuracy of a classifier (regressor) where the data is divided into $k$ folds of near equal size. The classifier (regressor) is trained on $k - 1$ folds, leaving one fold out to test on. The process is repeated $k$ times. The accuracy of the classifier (regressor) is the average accuracy for the $k$ folds.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Objectives . . . . .	3
1.2	Scope and Limitations . . . . .	3
1.3	Disposition . . . . .	4
<b>2</b>	<b>Previous Work</b>	<b>5</b>
2.1	Short-Term Load Forecasting . . . . .	5
2.2	Conventional Methods . . . . .	6
2.3	Machine Learning Methods . . . . .	6
2.4	Method Proposal . . . . .	7
<b>3</b>	<b>Mathematical Background</b>	<b>8</b>
3.1	Support Vector Regression . . . . .	8
3.1.1	The Basic Idea . . . . .	8
3.1.2	Dual Formulation . . . . .	10
3.1.3	Kernels . . . . .	12
3.2	Decision Tree Learning . . . . .	13
3.2.1	Regression Trees . . . . .	14
3.2.2	Random Forests . . . . .	16
3.3	K-Means Clustering . . . . .	17
<b>4</b>	<b>Model Formulation</b>	<b>20</b>
4.1	Pecan Street Dataset . . . . .	20
4.2	Clustering . . . . .	21
4.3	Validation . . . . .	22
4.3.1	Performance Metrics . . . . .	22
4.3.2	Benchmarking . . . . .	23
4.4	Feature Selection . . . . .	24
4.4.1	Historical Load . . . . .	24
4.4.2	Calendar Features . . . . .	25
4.4.3	Temperatures . . . . .	26
4.4.4	Summary . . . . .	27
4.5	Hyperparameter Optimization . . . . .	27

<b>5</b>	<b>Results</b>	<b>31</b>
5.1	Forecasting on Pecan Street Dataset . . . . .	31
5.1.1	Forecasting: Individual Households . . . . .	33
5.1.2	Forecasting: Aggregate Level . . . . .	35
5.1.3	Forecasting: Cluster-based . . . . .	37
5.1.4	Analysis . . . . .	39
5.2	Accounting for Cluster Characteristics in Aggregate Forecasting . . . . .	40
5.2.1	Analysis . . . . .	42
5.3	Discussion . . . . .	42
<b>6</b>	<b>Conclusion</b>	<b>44</b>
6.1	Follow-up of Objectives . . . . .	44
6.2	Concluding Remarks . . . . .	46
	<b>Bibliography</b>	<b>48</b>
<b>A</b>	<b>Mercer’s Theorem</b>	<b>51</b>
<b>B</b>	<b>Optimizing the Leaf Node Value of a Regression Tree</b>	<b>52</b>
<b>C</b>	<b>Distribution of Households with K-Means Clustering</b>	<b>53</b>
<b>D</b>	<b>Hyperparameter Search for Individual Households</b>	<b>55</b>

# Chapter 1

## Introduction

Residential buildings constitute 25% of the Swedish electricity consumption [1]. Similarly in urban areas like Stockholm, the residential electricity consumption amounts to 30% of the total electricity consumption [2]. Undoubtedly there are numerous economic and environmental benefits to streamlining the energy usage and reducing the overall consumption. Three important targets are driving the development of the energy industry in Europe, namely the European Union 20-20-20 goals [3]:

- 20% cut in greenhouse gas emissions from 1990 levels
- 20% of energy coming from renewable resources
- 20% improvement in energy efficiency

To meet future demands of reduced greenhouse gas emissions, an increased amount of renewable energy and improved energy efficiency, changes must take place in the electric power system.

Alongside the introduction of the EU 20-20-20 goals the term "smart grid" has materialized. A first step in the development towards a smart grid is gaining a deeper understanding of the value chain in the electric power system. By retrieving more information practitioners seek to automate the process of delivering electricity from the electric utilities to the home and in turn reduce peak demand, operations and management costs [4]. To allow for a transition into the smart grid the European Union has imposed an obligation of a 80% roll-out of smart meters by 2020 [5].

The smart meter allows reading of the electricity consumption of individual households in close to real time. Collecting and analyzing smart meter data provides valuable knowledge about the user behavior, and is interesting both from a demand (consumer) and supply (electric utilities) perspective. For the consumers it allows efficient use of energy and identification of "energy guzzlers", whereas for the electric utilities there are economic benefits in developing accurate methods for predicting future energy usage. It

has been shown that a 1% reduction in the average forecast error can save hundreds of thousands or even millions of dollars for an electric utility [6].

Traditionally, residential energy demand estimates have been made at a regional or local level where consumption profiles are publicly available [7]. Electric utilities and other service offering companies within the energy industry can forecast the energy usage of a city like Stockholm using the consumption profile of the city, and multiply by their share of customers to arrive at the desired prediction. Contrarily, in a sensor based approach consumption profiles are gathered from smart electricity meters for all individual customers. It is relevant to investigate if a sensor based model with predictions made on the individual household level can predict the energy consumption more precisely than a regional or local model, due to being able to account for each individual household's consumption pattern and characteristics.

Combined with a selection of the spatial granularity, energy consumption predictions are often made for varying time horizons. Load forecasting is classified depending on the duration of the time horizon. Predictions from 15 minutes up to one day ahead fall into the category of short-term load forecasting (STLF), from one day to a year to medium-term load forecasting (MTLF) and between a year up to ten years ahead to long-term load forecasting (LTLF). Residential buildings and measurements drawn from a smart meter are examples of small-scale systems that display high variability in the load dynamics and impose strict requirements on the STLF modeling tools.

The purpose of this thesis is to investigate the impact of clustering on the prediction accuracy of future energy consumption for a customer base consisting of numerous households in an urban area. By making hourly single-step forecasts on three spatial data scales, namely individual households, clusters of similar households and on an aggregate level, we anticipate to recognize how traditional forecasting methods can be improved to meet the demands of the smart grid. Additionally, the research within sensor based forecasting is expanded by investigating state-of-the-art algorithms (Support Vector Regression) together with previously uncommon algorithms (Random Forest) in the work of residential energy consumption predictions. Clustering is performed with the K-Means Clustering algorithm. The terms energy consumption, energy demand and load are used interchangeably throughout the thesis.

## 1.1 Objectives

This section outlines the main objectives of the thesis, given in a relatively general manner. Towards the end of the thesis, in Section 6.1, each objective will be reviewed and related to the results.

- 1. Analyze historical residential energy consumption data**

Given historical time series of energy consumption from individual households, conduct a statistical analysis to highlight the behavior of the individual series and the dependency between different series.

- 2. Review and select appropriate mathematical models**

Review and evaluate mathematical models, emphasizing their ability to capture load dynamics and successfully forecast the future load.

- 3. Experiment with clustering and cluster sizes**

Conduct experiments to assess the impact of clustering and the cluster size on the accuracy of the load forecast.

- 4. Discuss implications on short-term load forecasting**

Given results of the above points, discuss strengths and weaknesses of outlined models and propose adjustments to allow for future studies to advance research within STLF.

## 1.2 Scope and Limitations

Used in this thesis is the Pecan Street Data Set [8], which is part of a research project conducted in the Austin area in Texas, with the goal of solving the global water and energy challenge. As such, any conclusions made may only be viable for the studied population in Austin and extensions to other geographical areas must be taken with care. With that said, numerous studies are currently performed on STLF with different model setups on different datasets. The ability to generalize grows as practitioners test developed model setups on new datasets in a spectrum of geographical areas.

The models used for predicting the future load are Support Vector Regression and Random Forest, where the underlying reason for selecting each is established in Section 2.4. To assess the quality of the models, a baseline model is created that serves as a benchmark. In the interest of determining the impact of clustering similar households before developing a forecast model, K-Means Clustering is used to cluster households based on the average daily load profile of each household. All calculations have been performed using `Python 2.7`, with base algorithm implementations from `scikit-learn`<sup>1</sup>.

---

<sup>1</sup><http://scikit-learn.org/stable/>

### 1.3 Disposition

The rest of this thesis is structured as follows. In Chapter 2 related work is reviewed and the current state-of-the-art assessed. In Chapter 3 the mathematical background is presented, laying the groundwork for the models outlined in Chapter 4. Chapter 5 presents the results of the forecasting and evaluates the impact of the number of clusters on the prediction accuracy. Chapter 6 discusses the impact of the results, where future work is also suggested. Section 6.2 concludes the thesis. Appendices are referred to where needed.

## Chapter 2

# Previous Work

This chapter provides an overview of previous work completed within short-term load forecasting. A brief introduction and a historic note is given in Section 2.1. Section 2.2 describes the assortment of traditional statistical methods used for STLF. Selected work is presented in Section 2.3, performed within both the commercial and residential building sector. Recent contributions to the field of residential load forecasting are presented, many of which have given rise to the study undertaken in this thesis. Finally, in Section 2.4, the algorithms adopted in this report are proposed.

### 2.1 Short-Term Load Forecasting

To make predictions, an analysis is performed on the load signal which is a time series. The goal is to impose a relationship between future and past samples. Then, a model can estimate the future evolution of the load signal in terms of its history and commonly some exogenous variables believed to influence the future load. In the first energy prediction contest, the Great Energy Predictor Shootout (GEPS), organized by the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) in 1993-1994, participants were asked to predict the hourly energy consumption of a commercial building. The winning contestant [9] developed a sensor based model using a machine learning algorithm that relied to a small extent on domain knowledge about the commercial building whose energy was predicted. Following the popularity and success of GEPS, more attention has been given to the field of short-term load forecasting. The following sections explore the various methods applied to STLF where forecasting approaches are commonly classified to conventional statistical methods and machine learning methods.

## 2.2 Conventional Methods

Statistical methods are white box models: the internal structure of the model is well known which allows for interpretation and understanding of the process. In a prediction setting, a regression analysis is applied where the outputs of the model are explicitly related to the inputs through mathematical equations. The family of statistical white box models includes multiple linear regression [10–12], autoregressive moving average (ARMA) [13], autoregressive integrated moving average (ARIMA) [14] and Kalman filter [15]. Conventional methods are advantageous in that they are easily implemented and interpreted, however their disability to handle non-linearity in short-term load series instead encourages the use of machine learning methods.

## 2.3 Machine Learning Methods

Succeeding GEPS, considerable work has been compassed within machine learning in the field of commercial energy load forecasting. Machine learning methods are so-called black box models: the internal dynamic is at most times unknown and at best difficult to interpret. However, the ability of the methods to learn complex internal representations without human interference is a major advantage. Notable work includes that of Chae et al. [16] where Artificial Neural Networks (ANN) were used to make day-ahead forecasts of the electricity usage for a commercial building with a temporal resolution of 15 minutes, and Fu et al. [17] where a Support Vector Machine (SVM) with a RBF kernel was trained for each hour of the day to predict the next day electricity load of a public building in Shanghai.

The recent increase of smart meters in the residential sector has lead to large available datasets. Together with advancements within machine learning, more effort is being put into monitoring, analyzing, characterizing and forecasting the energy usage of individual households. Among the most commonly used and successful machine learning algorithms are Support Vector Machines [10, 18–21] and Artificial Neural Networks [10, 18], however cases of using Tree-based methods have also been seen [22, 23].

In [18] the consumption of a family of three living in Warsaw, Poland was forecast a day ahead by training a model for each hour of the day using SVM and a Multi-Layer Perceptron (MLP). Edwards et al. [10] compare a Linear Regression (LR) with several variations of SVM and Feed-Forward Networks (FFN), and assess the results against the recognized GEPS dataset as well as three households located in Tennessee, US. Results indicate that a Least-Squares SVM outperforms said methods and best predicts the future hourly residential load.

In [19] an extension is made to forecast the load of multi-family residential buildings rather than individual households. Jain et al. use a SVM with a RBF kernel to examine the impact of both the temporal and the spatial granularity on the prediction accuracy,

and come to the conclusion that making hourly forecasts on the "by floor" level grants the optimal prediction results, as opposed to modeling single units or the whole building. In what seems to be a consistent trend in forecasting with SVMs and MLPs, Humeau et al. [20, 21] explore the similarity in usage between different households and design a cluster-based algorithm where the prediction accuracy is assessed as a function of the number of clusters. In a setting with 782 different households located in Ireland, SVM obtains an optimal prediction accuracy using four clusters, whereas for MLP and LR the error increases with the number of clusters.

## 2.4 Method Proposal

In line with previous work [10, 18–21], and what has been assessed as the current state-of-the-art, SVM for regression, Support Vector Regression (SVR) is proposed as the first method used in the thesis. The radial basis function (RBF) is selected as the kernel function to use with SVR, due to its ability to generalize and function well with nonlinear datasets [24]. Furthermore, continuing the research of [22, 23], Random Forest for regression is suggested as a second method. Little attention has been paid to Random Forests, a method which holds a great advantage in being rather insensitive to hyperparameter values [25]. Random Forests are also less prone to overfitting due to their characteristic of being an ensemble of decision trees trained on different parts of the same training set.

## Chapter 3

# Mathematical Background

In this chapter the mathematical background underlying the work presented in the thesis is put forward. The reader is expected to be acquainted with some basic optimization and graph theory. Section 3.1 and Section 3.2 describes two well-known machine learning techniques used in a variety of applications, namely Support Vector Regression and Random Forest. Section 3.3 provides an overview of the K-Means Clustering algorithm, used to partition observations into K different clusters.

Let  $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\} \subset \mathbb{R}^N \times \mathbb{R}$  be a set of training data, where  $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{iN}]$  denotes the N-dimensional input *features* and  $y_i$  the corresponding outputs, often referred to as *labels*. To give an example,  $\mathbf{x}_i$  could be a set of characteristics of an apartment such as the number of square meters, number of rooms, what floor it is located on, if it is located in a major city and  $y_i$  the corresponding price of the apartment.

### 3.1 Support Vector Regression

In this section an overview is given of the ideas underlying Support Vector (SV) machines for function approximation, known as Support Vector Regression. SV Machines spring from the theory of Perceptrons developed by Rosenblatt in 1962 [26], further contextualized in 1986 [27] when the back-propagation algorithm was discovered, and the theory of Structural Risk Minimization promoted by the likes of Vapnik and Chervonenkis [28, 29].

#### 3.1.1 The Basic Idea

The idea of  $\epsilon$ -SV regression [30] is to find a function  $f(\mathbf{x})$  that has at most  $\epsilon$  deviation from the actual targets  $y_i$  for all available training data. At the same time  $f(\mathbf{x})$  has to be as simple as possible; while an overly complex function will account for all variations

in the training set and yield a small error, it will not generalize well to previously unseen data points. As tempting as it might be to achieve a close to zero error measure on the training data, this is only used to prepare the algorithm for the real test in predicting new compositions of input features. The idea of achieving an overly complex  $f(\mathbf{x})$  is known as *overfitting*, and will be discussed in further detail in Section 3.2.1.

To make the derivation pedagogical the case of linear functions  $f(\mathbf{x})$  is first described, after which the analysis can be extended to the nonlinear case. Consider a function:

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b, \quad \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R} \quad (3.1)$$

where  $\langle \cdot, \cdot \rangle$  denotes the inner product in  $\mathbb{R}^N$ ,  $\mathbf{w}$  the weights of the linear function and  $b$  the bias. To ensure that  $f(\mathbf{x})$  deviates at most  $\epsilon$  from  $y_i$  we impose the constraints:

$$y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq \epsilon \quad (3.2)$$

$$\langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \leq \epsilon \quad (3.3)$$

The reduction of the complexity of  $f(\mathbf{x})$  is often translated into increasing the *flatness* [31]. In the case of (3.1) this translates to finding a small  $\mathbf{w}$ , which can be achieved by minimizing the norm given by  $\|\mathbf{w}\|^2 = \langle \mathbf{w}, \mathbf{w} \rangle$ . Together with the constraints in (3.2) and (3.3) this forms a convex optimization problem:

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} && \frac{1}{2} \|\mathbf{w}\|^2 \\ & \text{subject to} && \begin{cases} y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq \epsilon \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \leq \epsilon \end{cases} \end{aligned} \quad (3.4)$$

Solving (3.4) rests on the assumption that all input patterns  $\mathbf{x}_i$  are estimated with  $\epsilon$  precision to  $y_i$ . In most real world scenarios, however, this is not achievable for a sufficiently small  $\epsilon$  since data can be noisy and contain several outliers. To allow for a minimal number of errors the "Soft Margin" loss function was developed by Bennett and Mangasarian [32] and implemented to SV machines by Vapnik and Cortes [33]. It suggests the introduction of slack variables  $\xi_i, \xi_i^*$  to make the optimization problem (3.4) feasible in situations where it would otherwise be infeasible. The final formulation is given:

$$\begin{aligned} & \underset{\mathbf{w}}{\text{minimize}} && \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \\ & \text{subject to} && \begin{cases} y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq \epsilon + \xi_i \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \quad (3.5)$$

The constant  $C > 0$  controls the penalty on deviations larger than  $\epsilon$  and serves as a trade-off between achieving flatness and minimizing the amount of errors. The role of the constant  $C$  is depicted in Figure 3.1 for illustration purposes. Only the points that lie outside the  $\epsilon$ -tube contribute to the cost associated with  $C \sum_{i=1}^n (\xi_i + \xi_i^*)$ . If  $C$  is

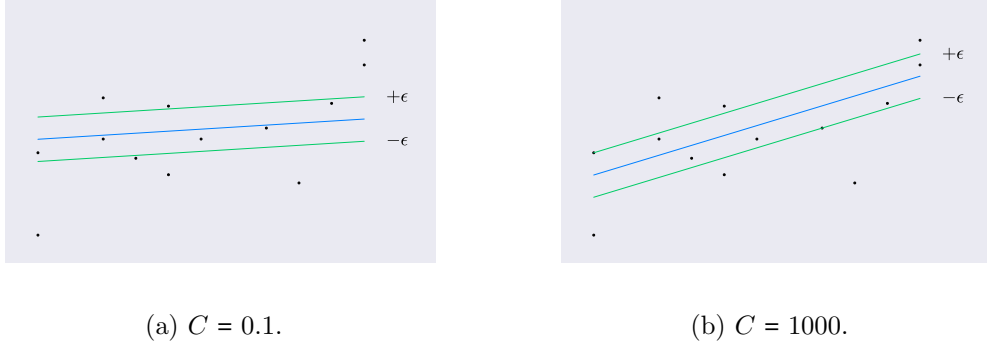


Figure 3.1: "Soft Margin" solution for a linear SV Machine.

large, the corresponding optimal  $f(\mathbf{x})$  will reduce the number of points lying outside the  $\epsilon$ -tube at the cost of the flatness of  $f(\mathbf{x})$ . Contrarily, a low  $C$  will focus on achieving flatness at the cost of falsely predicting some of the training data. While (3.5) presents a satisfying solution to the linear SV regression, the optimization problem can often be solved more efficiently in its dual formulation. The dual formulation is also key in that it will allow the transition to nonlinear prediction functions  $f(\mathbf{x})$  for the SV regression.

### 3.1.2 Dual Formulation

Going from the primal formulation of the optimization problem to its dual formulation involves the use of Lagrange multipliers. The Lagrange functional for (3.5) is:

$$\begin{aligned}
 L = & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) - \sum_{i=1}^n (\lambda_i \xi_i + \lambda_i^* \xi_i^*) \\
 & - \sum_{i=1}^n \alpha_i (\epsilon + \xi_i - y_i + \langle \mathbf{w}, \mathbf{x}_i \rangle + b) \\
 & - \sum_{i=1}^n \alpha_i^* (\epsilon + \xi_i^* + y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b)
 \end{aligned} \tag{3.6}$$

where  $\lambda_i, \lambda_i^*, \alpha_i, \alpha_i^* \geq 0$  are Lagrange multipliers. It is known (see for instance work by Mangasarian [34]) that the solution to (3.5) is given by the saddle point of the Lagrangian in (3.6), where the objective is to minimize  $\mathbf{w}, b, \xi_i, \xi_i^*$  while at the same time maximizing  $\lambda_i, \lambda_i^*, \alpha_i, \alpha_i^*$ . The point of minimum is obtained through the partial derivatives of  $L$ :

$$\frac{\partial L}{\partial b} = \sum_{i=1}^n (\alpha_i^* - \alpha_i) = 0 \quad (3.7)$$

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^n (\alpha_i - \alpha_i^*) \mathbf{x}_i = 0 \quad (3.8)$$

$$\frac{\partial L}{\partial \xi_i} = C - \lambda_i - \alpha_i = 0 \quad (3.9)$$

$$\frac{\partial L}{\partial \xi_i^*} = C - \lambda_i^* - \alpha_i^* = 0 \quad (3.10)$$

Inserting (3.7), (3.8), (3.9) and (3.10) to (3.6) yields:

$$\begin{aligned} L = & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ & - \epsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) \end{aligned}$$

From (3.9) and (3.10) one sees that  $\alpha_i = C - \lambda_i$  and since  $\lambda_i \geq 0$  the Lagrange multipliers  $\alpha_i, \alpha_i^*$  are bounded,  $\alpha_i, \alpha_i^* \in [0, C]$ . The dual optimization problem is given:

$$\begin{aligned} \underset{\alpha_i, \alpha_i^*}{\text{maximize}} \quad & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ & - \epsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) \\ \text{subject to} \quad & \begin{cases} \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases} \end{aligned} \quad (3.11)$$

In deriving (3.11) the dual variables  $\lambda_i, \lambda_i^*$  were eliminated. Additionally, through the partial derivative (3.8)  $\mathbf{w}$  is given, and consequently  $f(\mathbf{x})$  can be rewritten:

$$\mathbf{w} = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \mathbf{x}_i \implies f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = \sum_{i=1}^n (\alpha_i - \alpha_i^*) \langle \mathbf{x}_i, \mathbf{x} \rangle + b$$

What is remarkable is that  $\mathbf{w}$  is described as a linear combination of the training patterns  $\mathbf{x}_i$  weighed by the Lagrange multipliers  $\alpha_i, \alpha_i^*$ . As a result the complexity of the function  $f(\mathbf{x})$  is rather independent of the dimensionality of the input features, and instead rests on the number of nonzero  $\alpha_i, \alpha_i^*$ . Note also that the complete optimization problem is given by scalar multiplications and inner products between the input features, and that

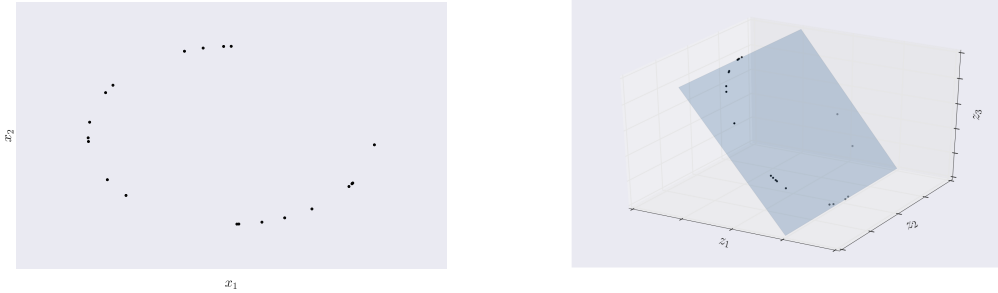
the prediction function  $f(\mathbf{x})$  can be evaluated without explicitly computing  $\mathbf{w}$ . With these observations in mind, the extension to a nonlinear predictor function  $f(\mathbf{x})$  can be made.

### 3.1.3 Kernels

Initially, to solve the problem of nonlinear data, the training features  $\mathbf{x}_i$  can be preprocessed by some map  $\Theta : \mathbb{R}^N \mapsto \mathbb{F}$  into feature space  $\mathbb{F}$  where the SV regression algorithm outlined in (3.11) can be applied. To give an example, consider the case depicted in Figure 3.2a. By utilizing the following map [30]:

$$\begin{aligned} \Theta : \mathbb{R}^2 &\mapsto \mathbb{R}^3 \\ (x_1, x_2) &\mapsto (z_1, z_2, z_3) = (x_1^2, \sqrt{2}x_1x_2, x_2^2) \end{aligned}$$

The training features  $\mathbf{x}_i$  can be approximated by a hyperplane, as seen in Figure 3.2b, which in the original two-dimensional space corresponds to an ellipse.



(a) Nonlinear input features  $\mathbf{x}_i$ .

(b) Mapping  $(x_1, x_2) \mapsto (x_1^2, \sqrt{2}x_1x_2, x_2^2)$ .

Figure 3.2: Example of nonlinear SV regression.

While the preprocessing serves as a satisfying approach to the aforementioned problem, other problems may quickly become computationally infeasible. From Cortes and Vapnik [33]: "To construct polynomial of degree 4 or 5 in a 200 dimensional space it may be necessary to construct hyperplanes in a billion dimensional feature space."

The breakthrough came in 1992 when Boser et al. [35] showed that instead of making a nonlinear transformation of the input patterns  $\mathbf{x}_i$  followed by dot products in feature space, two input patterns  $\mathbf{x}_i$  can first be compared in input space through some pre-defined metric before making a nonlinear transformation of the resulting comparison. Going back to the previous example, it can be shown that the inner product of the preprocessed features can be rewritten:

$$\langle (x_1^2, \sqrt{2}x_1x_2, x_2^2), (x_1'^2, \sqrt{2}x_1'x_2', x_2'^2) \rangle = \langle \mathbf{x}, \mathbf{x}' \rangle^2$$

The example shows that an explicit representation for the preprocessing map  $\Theta(\mathbf{x})$  is not necessary, as long as the rewritten inner product is a proper inner product. In Section 3.1.2 it was concluded that the SV regression algorithm depended only on inner products between the input features  $\mathbf{x}_i$ . This allows us to reformulate the problem without using  $\Theta(\mathbf{x})$  explicitly, through defining  $K(\mathbf{x}, \mathbf{x}') = \langle \Theta(\mathbf{x}), \Theta(\mathbf{x}') \rangle$ , where  $K(\mathbf{x}, \mathbf{x}')$  is known as a *kernel*. The nonlinear SV regression is given:

$$\begin{aligned} \underset{\alpha_i, \alpha_i^*}{\text{maximize}} \quad & -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) K(\mathbf{x}_i, \mathbf{x}_j) \\ & - \epsilon \sum_{i=1}^n (\alpha_i + \alpha_i^*) + \sum_{i=1}^n y_i (\alpha_i - \alpha_i^*) \\ \text{subject to} \quad & \begin{cases} \sum_{i=1}^n (\alpha_i - \alpha_i^*) = 0 \\ \alpha_i, \alpha_i^* \in [0, C] \end{cases} \end{aligned} \quad (3.12)$$

where  $f(\mathbf{x})$  is given by  $f(\mathbf{x}) = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(\mathbf{x}_i, \mathbf{x}) + b$ . The implications of using a kernel is that the optimization problem now seeks to determine the *flattest* possible  $f(\mathbf{x})$  in feature space rather than in input space.

The family of functions  $K(\mathbf{x}, \mathbf{x}')$  that corresponds to inner products in some feature space  $\mathbb{F}$  must obey Mercer's Theorem [36], outlined in Appendix A. Commonly used kernels include polynomial, sigmoid and radial basis function (RBF) kernels, refer to Table 3.1 for details.

Kernel Name	$K(\mathbf{x}, \mathbf{x}')$	Parameters
Polynomial	$(\langle \mathbf{x}, \mathbf{x}' \rangle + c)^p$	$p \in \mathbb{N}, c \geq 0$
Sigmoid	$\tanh(\gamma \langle \mathbf{x}, \mathbf{x}' \rangle + r)$	$\gamma > 0, r < 0$
RBF	$e^{-\gamma \ \mathbf{x} - \mathbf{x}'\ ^2}$	$\gamma > 0$

Table 3.1: List of commonly used kernel functions.

## 3.2 Decision Tree Learning

Random Forests stem from decision tree learning, a predictive modeling approach used in statistics and machine learning. Decision trees are a type of classifier or regressor that splits the training data into smaller subsets until a predefined criterion is met, and make a viable tool for visualizing decision making. Traditionally, decision trees have been created manually using human expertise and domain knowledge. As problems

grew more complex, several algorithms for automated rule extraction were developed as a solution. Some of the earliest papers on automated rule extraction for decision trees concentrate on classification [37, 38]. Perhaps one of the most influential references until this day remains the CART algorithm by Breiman et al. [39], short for Classification and Regression Trees. In the book the authors provide a thorough description of decision trees used for classification and regression respectively.

The rest of this section is structured as follows. In 3.2.1 decision trees used for regression will be explained in detail. Following the training and evaluation of a regression tree, 3.2.2 describes how an ensemble of regression trees can form what is known as a Random Forest.

### 3.2.1 Regression Trees

Regression trees are directed graphs, and like SVR, serve the purpose of approximating a function  $f(\mathbf{x})$  that minimizes the deviations from the true values  $y_i$ . However, in contrast with the wide spectrum of available functions for SVR defined by the kernel, function approximation provided by regression trees is highly non-smooth due to the additive nature of the model. Regression trees partition the input space to a set of regions and fit a constant value within each region that represents the approximation  $f(\mathbf{x})$ . The partitioning is represented by the shape of the regression tree, where each path from the root of the tree to a leaf node corresponds to a region. To get a prediction, the feature vector  $\mathbf{x}_i$  passes a series of logical tests at the inner nodes and subsequently progresses down different paths of the tree depending on the characteristics of  $\mathbf{x}_i$ . An example of a regression tree is given in Diagram 3.1.

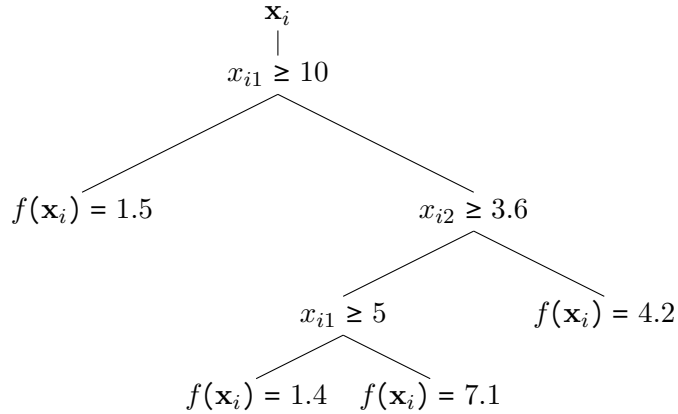


Diagram 3.1: Example of a regression tree in a setting with  $\mathbf{x}_i \in \mathbb{R}^2$ ,  $f(\mathbf{x}_i) \in \mathbb{R}$ .

The challenge lies in constructing the regression tree, that is, determining the logical tests and the constant values at the leaf nodes that minimize the deviations of the

approximation  $f(\mathbf{x})$  from the true values  $y_i$ , given by the mean squared error:

$$\frac{1}{N} \sum_{i=1}^N (y_i - f(\mathbf{x}_i))^2$$

$f(\mathbf{x})$  can only take on a finite number of values given by the leaf nodes  $l$ . Let  $c_l$  denote the constant belonging to leaf node  $l$ , then the error associated with a leaf node with prediction value  $c_l$  is given by:

$$\text{MSE}_l = \frac{1}{N_l} \sum_{y_i \in \mathfrak{D}_l} (y_i - c_l)^2$$

where  $N_l$  is the number of training samples in leaf node  $l$  and  $\mathfrak{D}_l$  a set containing the training samples in  $l$ . The error for the whole tree  $T$  is defined as a weighed average of the error in its leaves:

$$\text{MSE}_T = \frac{1}{N} \sum_{l \in T} \sum_{y_i \in \mathfrak{D}_l} (y_i - c_l)^2 \quad (3.13)$$

It has been shown that for regression trees based on least squares, the  $c_l$  that minimize the expected value of the mean squared error is the mean of the target variables  $y_i$ :

$$c_l = \frac{1}{N_l} \sum_{y_i \in \mathfrak{D}_l} y_i$$

A proof is given in Appendix B. The implications are that test samples ending up in leaf node  $l$  should be assigned the value  $c_l$  which is the average of the target values for the training samples belonging to  $l$ .

We are now in a position to build an optimal regression tree. The tree starts with a root node which contains all the training samples. The predicted value for any test sample is given by the average of all target values in the training data. The training data is split into two subsets if there exists a split that decreases the error given by (3.13). The best split is the one that maximizes the decrease in error, given by the difference in MSE between the tree with and without the split:

$$\Delta \text{MSE} = \text{MSE}_t - \frac{N_{t_l}}{N_t} \text{MSE}_{t_l} - \frac{N_{t_r}}{N_t} \text{MSE}_{t_r}$$

where  $t_l$  is the left child node of  $t$  and  $t_r$  the right child node containing  $N_{t_l}$  and  $N_{t_r}$  samples respectively. This is a greedy algorithm that searches exhaustively through all splitting criteria for all features in  $\mathbf{x}_i$  and selects the one with the largest  $\Delta \text{MSE}$ . The

tree is grown until no further splits can be done. As the tree grows, the training set is split into increasingly smaller samples. Intuitively, the tree will continue to split its nodes until there is only one sample in each leaf node, where the error is zero. Unsurprisingly, building such an overly large regression tree will not generalize well when testing with previously unseen samples. This phenomenon is known as *overfitting*.

To combat overfitting there are primarily two remedies available to regression trees. First, the tree can be *pre-pruned*, meaning that the tree is only grown until it reaches a certain length, or until the leaf nodes contain a predefined number of training samples. On the other hand, the tree can be *post-pruned*. In the second case the full tree is built at first, meaning that the leaf nodes each contain just one sample. Then, using a validation set, nodes are removed from the bottom of the tree if the accuracy is at least as good as the accuracy of the unpruned tree. Pre- and post-pruning are adequate tools for avoiding overfitting of decision trees, nonetheless in the pursuance of high-performing models decision trees lag behind. The next part extends the single decision tree to an ensemble of decision trees with the ambition of establishing a rigorous model.

### 3.2.2 Random Forests

In 1996 Breiman proposed a method to improve accuracy of decision trees, where several decision trees are generated using bootstrapped replicates  $\mathfrak{D}_B$  of the training set  $\mathfrak{D}$ . The size of  $\mathfrak{D}_B$  is the same as the training set size, but the samples are usually drawn with replacement. The prediction of a test sample is achieved by averaging the predictions of the bootstrapped decision trees. The method is known as *bagging*, coined from the term "bootstrap **agg**regating". Bagging was shown to outperform a single decision tree both in the case of classification and regression [40], the rationale being that averaging the prediction from several trees reduces the variance without changing the bias [41].

Continuing the study on bagging, Breiman [42] proposed an extension by only considering a random subset of the available predictors at each split when building the tree. The underlying reason is to decorrelate the trees: In a case where a dataset has one very strong predictor and several other moderately strong predictors, almost all of the bagged trees will use the strong predictor as the first splitting criterion, yielding very similar and hence correlated trees. The point of bagging is to reduce variance, and averaging highly correlated trees does not reduce variance as much as averaging uncorrelated trees does. Thus, when considering only a subset of predictors, the problem is overcome. Using bagging together with random subset selection yields what is known as a *Random Forest*.

The building of a Random Forest rests primarily on the selection of two hyperparameters: the number of trees in the forest and the number of features to consider when evaluating the best split. For each additional tree, a new bootstrapped dataset  $\mathfrak{D}_B$  is constructed from the training set  $\mathfrak{D}$ . As the number of trees in the forest increases, the more chance there is that trees have overlapping training sets. The advantage however is that more

votes are cast in the prediction process, decreasing the generalization error. It has been shown that as the number of trees increases, the accuracy approaches the theoretical limit of the forest [42]. The number of features considered at each split controls the variation between trees. By considering all features at each split every tree will select its global optimal feature, rendering similar trees. Lowering the number of features considered at each split increases the chance that the global optimal feature is left out of the subset of features tested. The ambition is to create a mixture of decision trees split in a variety of different ways, resulting in a range of predictions.

### 3.3 K-Means Clustering

Clustering refers to the method of grouping observations in a dataset into subgroups, or clusters, where observations that belong to the same cluster are more similar to each other compared to those in other clusters. The perception of similarity is often domain-specific, however common similarity measures include Euclidean distance, correlation-based distance and cosine similarity. Clustering can be very useful in that it allows finding subgroups within the dataset, and in relation to the current thesis this translates into finding similar households in terms of electrical consumption. In contrast to SVR and Random Forest, who are both supervised algorithms, clustering is an unsupervised problem. In the supervised case we are trying to infer a relationship between the features and the target variable by training the algorithm with samples where the target is known. In an unsupervised case we are also trying to discover a structure or relationship, however there is no true answer to how the clustering should be performed or what the correct answer is.

Among the best-known clustering methods is K-Means Clustering, also known as Lloyd's algorithm after Stuart Lloyd who proposed the algorithm in 1957. With K-Means Clustering the objective is to partition the dataset into a predefined number of clusters  $K$ . After the number  $K$  is chosen, each observation is assigned to exactly one of the  $K$  clusters. The best clustering is the one that minimizes the within-cluster dissimilarity, measured by the squared Euclidean distance. Given a set of observations  $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\} \subset \mathbb{R}^N$  and a set of  $K$  clusters  $\mathcal{C} = \{C_1, C_2, \dots, C_K\}$ , for cluster  $j$  the within-cluster dissimilarity is given by:

$$\sum_{\mathbf{x}_i \in C_j} ||\mathbf{x}_i - \boldsymbol{\mu}_j||^2$$

where  $\boldsymbol{\mu}_j = \frac{1}{N_j} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$  is the mean of the observations in  $C_j$  and  $N_j$  the number of observations belonging to  $C_j$ . The complete optimization problem that is K-Means Clustering is given by:

$$\underset{\mathfrak{C}}{\text{minimize}} \sum_{j=1}^K \sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2 \quad (3.14)$$

Solving 3.14 is computationally very difficult, since the observations of  $\mathfrak{X}$  can be grouped in a large amount of ways. Yet there exists a greedy algorithm for solving the optimization problem given by (3.14) that allows the finding of a local optimal solution. The algorithm, elegant in its simplicity, is given as follows where the superscript  $m$  indicates iteration  $m$  of the algorithm:

1. Given a predefined number of clusters  $K$ , randomly assign an initial set of cluster means  $\boldsymbol{\mu}_1^{(0)}, \boldsymbol{\mu}_2^{(0)}, \dots, \boldsymbol{\mu}_K^{(0)}$ .
2. Assign each observation  $\mathbf{x}_i$  to the cluster whose mean is most similar to the observation. Here this translates into finding the closest cluster mean since we are computing the squared Euclidean distance:

$$\text{cluster}(\mathbf{x}_i) = \underset{j}{\operatorname{argmin}} \{ \|\mathbf{x}_i - \boldsymbol{\mu}_j^{(m)}\|^2 \}$$

The complete set of observations belonging to cluster  $j$  are given by:

$$C_j^{(m)} = \{ \mathbf{x}_i : \|\mathbf{x}_i - \boldsymbol{\mu}_j^{(m)}\|^2 \leq \|\mathbf{x}_k - \boldsymbol{\mu}_j^{(m)}\|^2 \ \forall j, 1 \leq j \leq K \}$$

3. For each cluster  $K$  compute the cluster centroid, which is the new mean of the cluster:

$$\boldsymbol{\mu}_j^{(m+1)} = \frac{1}{N_j^{(m)}} \sum_{\mathbf{x}_i \in C_j^{(m)}} \mathbf{x}_i$$

4. Return to step 2 and iterate until the cluster assignments stop changing.

The algorithm is guaranteed to converge, however due to the arbitrary initialization of the cluster centroids not always to the desired result. Therefore, the algorithm is often run several times, after which the best result in terms of (3.14) can be chosen. Figure 3.3 shows one iteration of K-Means Clustering along with the updated assignments after one iteration of the algorithm. As can be seen, in spite of an initial random assignment of the clusters, the algorithm quickly converges to a desirable result.

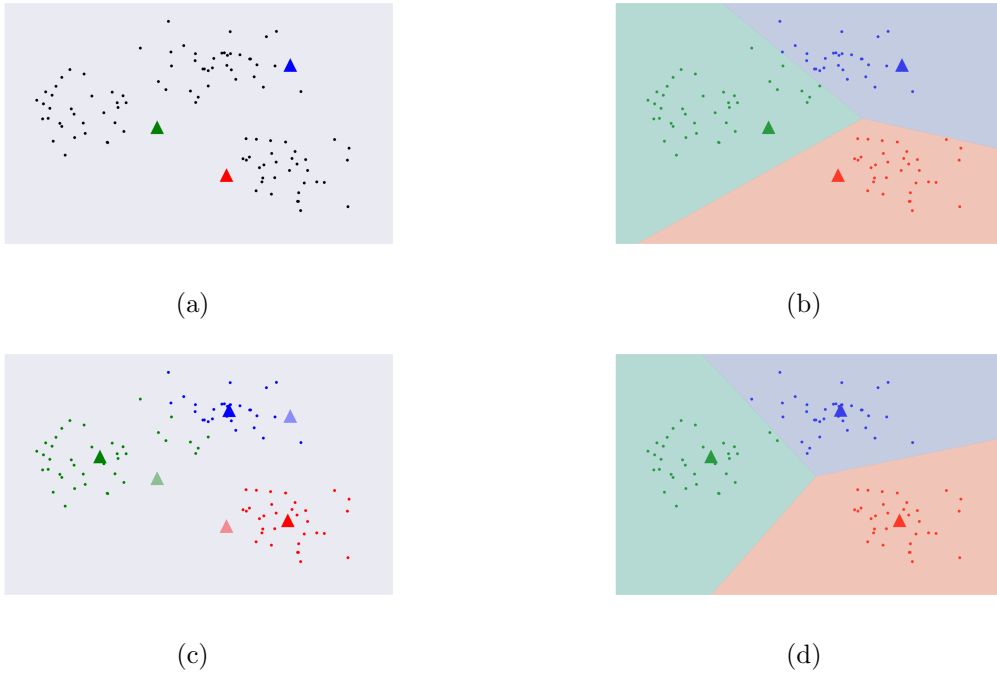


Figure 3.3: Example of the K-Means Clustering algorithm on Gaussian data using  $K = 3$  clusters. (a) The observations of the dataset are given by black dots. In the first step three cluster centroids are randomly generated, visualized by the triangles. (b) In the second step of the algorithm, each observation is assigned to the centroid to which it is closest. The partitions represent the Voronoi regions generated by the clusters. (c) After assigning the observations, the third step computes the new cluster centroids. The old centroids are shown in a transparent color. (d) The algorithm returns to the second step and again assigns observations to the centroid to which they are closest. As can be seen, several samples lying near the border between the green and blue Voronoi regions change cluster assignment. Therefore, the algorithm will continue to iterate until it reaches the final cluster assignments.

## Chapter 4

# Model Formulation

Chapter 4 builds on the theory presented in Chapter 3 and formulates the two forecasting models implemented in the thesis: Support Vector Regression with a RBF kernel and Random Forest. Section 4.1 introduces the Pecan Street dataset, part of an extensive research project in Texas, US. Section 4.2 describes how different households are grouped into clusters before training. Continuing, Section 4.3 describes the evaluation of the two models, before Section 4.4 gives an overview of the features included in each model. Section 4.5 ends the chapter, where the strategy for selecting the optimal hyperparameters is presented.

### 4.1 Pecan Street Dataset

The Pecan Street dataset consists of 1390 households<sup>1</sup>. For each household, the hourly average energy consumption is given in kilowatt-hours (kWh). Out of the 1390 households, 861 reside in Austin, Texas. The remaining households are from other cities in Texas, e.g. Houston and Dallas. To be able to cluster households and include exogenous features such as the outside temperature, only households in Austin are considered. Of the 861 households in Austin, 605 are still part of the Pecan Street program. To have a sufficient amount of data to work with, a requirement is set that rules out households that have less than 18 months worth of data. The requirement leaves 546 households in the dataset, 373 from which there is data available to download. The data was collected from 00:00 15 October 2014 to 06:00 16 April 2016.

For several households a few hourly measurements are missing, primarily caused by malfunctioning of the physical measurement devices. A rolling mean with a window size of six was used to fill missing data points if there existed at least one data point in the window. Households with any remaining missing measurements were left out from

---

<sup>1</sup>As of the descriptive metadata document, visited on 05.05.2016.

the data set. The final dataset consists of 187 households. An example of the energy consumption for one of the households is given in Figure 4.1.

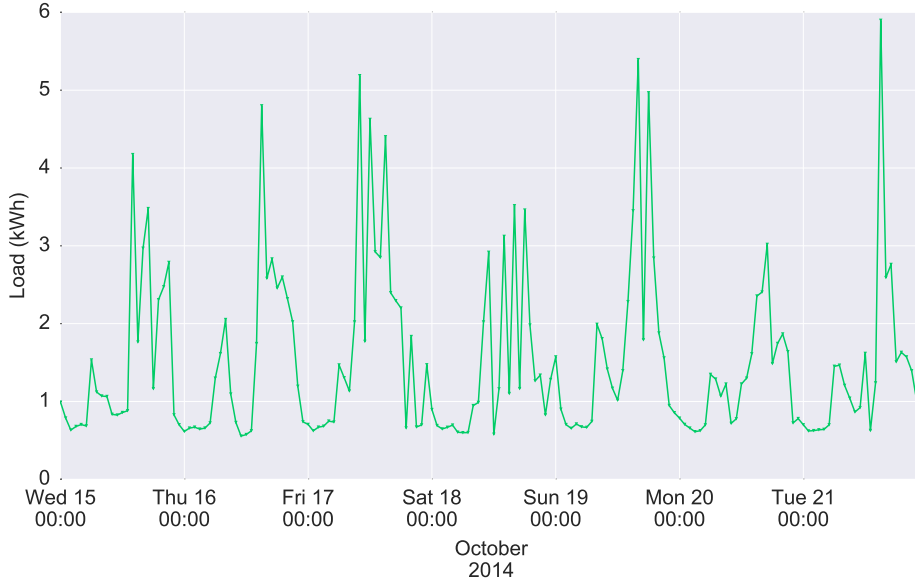


Figure 4.1: Hourly average energy consumption for household ID 77, measured over a period of one week.

## 4.2 Clustering

One of the main objectives of the thesis is investigating the impact of clustering on the prediction accuracy of the energy consumption for a customer base. The Pecan Street dataset provides 187 households to work with, and as a result there are primarily two intuitive ways in which to model the customer base: (1) Each household can be modeled independently which allows forecasting of individual households. This approach requires 187 models, one for each household, after which the forecasts of each household can be summed to get the aggregate consumption. (2) The households' energy consumption can be summed before developing a model. This is interpreted as modeling the behavior of the customer base as a whole and only requires a single model.

With the aforementioned modeling approaches, a third option rises in which similar households can be grouped together, leading to developing a model for each cluster of households. Each cluster is forecast separately, and the forecasts are aggregated to form the total forecast. The amount of clusters  $K$  to distribute the households into can be any number from one to the total number of households. Effectively, options (1) and (2) are

merely special cases of clustering the households using  $K = 187$  and  $K = 1$  respectively. In [20] Humeau et al. cluster households by considering the average consumption of each household in each hour. With this, every household is defined by a 24-dimensional vector that represents the consumption profile of the household. Using K-Means Clustering to group the different households is consequently performed in a 24-dimensional space. Clustering is tested using a number of clusters  $K \in \{1, 2, 3, 4, 6, 8, 16, 32, 64, 187\}$ , and for each cluster size the algorithm is run 10 times with different cluster initializations. The final clustering will be the best outcome of 10 runs in terms of (3.14). An example of the distribution of households among  $K = 16$  clusters is given in Figure 4.2. For the remaining distributions, see Appendix C.

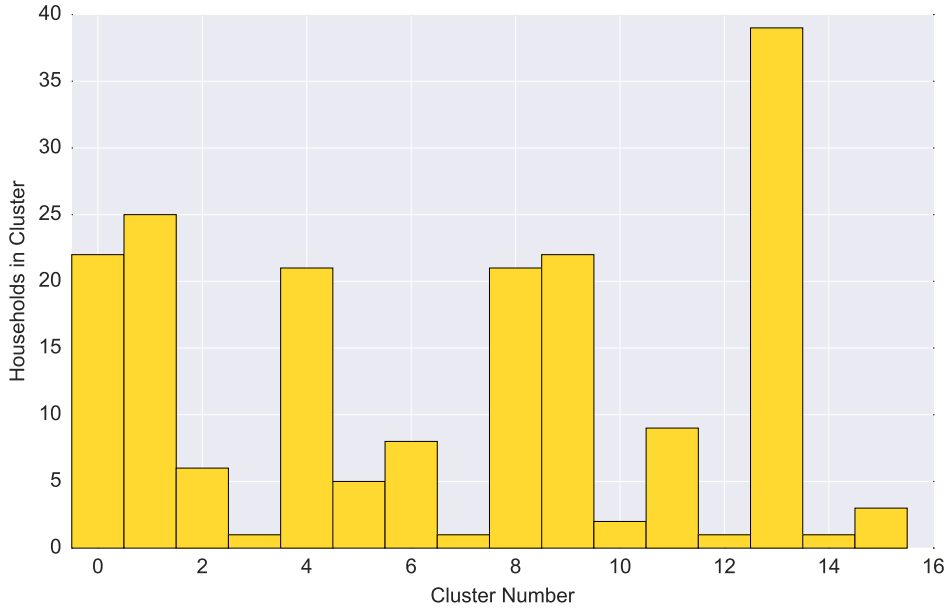


Figure 4.2: Distribution of households among  $K = 16$  clusters.

## 4.3 Validation

The following section deals with the evaluation of STLF algorithms, and provides an overview of the most commonly used error metrics along with a benchmark model.

### 4.3.1 Performance Metrics

In the literature there are primarily three metrics used in evaluating the performance of forecasting models, and these are: Coefficient of Variance (CV) [10, 16, 19], Mean Bias

Error (MBE) [10, 16, 17] and Mean Absolute Percentage Error (MAPE) [10, 21, 23]. The CV is defined as:

$$CV = \frac{\sigma}{\mu} = \frac{\sqrt{\frac{1}{N-1} \sum_{i=1}^N (y_i - f(\mathbf{x}_i))^2}}{\bar{y}}$$

where  $\sigma$  and  $\mu$  are the variance and mean respectively,  $f(\mathbf{x}_i)$  the predicted energy consumption,  $y_i$  the actual consumption and  $\bar{y}$  another notation for the average energy consumption. The CV measures to what extent the overall prediction varies with respect to the mean of the actual consumption, where a low CV value indicates that a model has precise forecasts. The CV can be seen as an extension of the Root Mean Squared Error (RMSE), which is maybe one of the most commonly used metrics in a regression setting. Continuing, the MBE measures how likely a model is to overestimate or underestimate the energy consumption. The MBE is given by:

$$MBE = \frac{\frac{1}{N-1} \sum_{i=1}^N (y_i - f(\mathbf{x}_i))}{\bar{y}}$$

An optimal MBE is achieved when the value is close to zero, corresponding to a model that neither is too conservative nor too aggressive in its predictions. The MAPE simply measures the percentage error in each measurement, and is commonly used due to its simplicity:

$$MAPE = \frac{1}{N} \frac{\sum_{i=1}^N |y_i - f(\mathbf{x}_i)|}{y_i}$$

In this study the overall performance of a model will be assessed using the CV metric. In situations where the CV metrics are similar, the MBE will serve as a tie breaker. In cases where none of the mentioned metrics can tell the models apart, the decision is based on the MAPE. The advantage of using as many as three metrics to assess the error of a model is that it gives a comprehensive view of each model, outlining possible flaws that would not be visible with only one metric.

### 4.3.2 Benchmarking

Along with the measurements of the performance of the models, defined by the three aforementioned metrics, a baseline model is created with the purpose of serving as a benchmark. The baseline predicts in every instance, that the energy consumption in the next hour will be equal to the currently observed energy consumption. Mathematically speaking, this is equal to the predictions:

$$f(\mathbf{x}_i) = y_{i-1}$$

given that the subscripts denote the timestep. The baseline prediction model will therefore be a delayed version of the actual energy consumption, with a one hour lag.

## 4.4 Feature Selection

A key component in building a rigorous machine learning model is selecting appropriate features to include in the model. Common practice includes using prior domain knowledge to select the features assumed to influence the target variable, however algorithms exist where the best set of features are selected from a large feature space.

### 4.4.1 Historical Load

With the electric load being a time series, we intuitively want to add historic loads as features. Specifically, the consumption from the last 48 hours should be of particular interest. Returning to Figure 4.1, for any single household the load series displays high volatility. In fact, the usage pattern seems rather chaotic and is not following any pattern in particular. Figure 4.3a and Figure 4.3b shows the autocorrelation for a randomly selected household and for the aggregation of all households. The autocorrelation is a measure of the correlation between values of a random process at different times, and is defined by:

$$\rho(s, t) = \frac{\mathbb{E}[(X_s - \mu_s)(X_t - \mu_t)]}{\sigma_s \sigma_t}$$

where  $X_i$  is the value given by the process or time series at time  $i$ ,  $\mu_i$  is the mean of the process at time  $i$  and  $\sigma_i$  is the standard deviation of the process at time  $i$ .

Made visible by the figures is that consumption is highly correlated to consumption the previous hour and the consumption from the hour before the last. Likewise for the aggregated load of all households, the consumption is strongly correlated with the consumption at the same hour the previous day and the day before the last. The conclusions are further reinforced by previous studies [10, 19], where considering the consumption of the last two hours has shown significant results in the case of hourly single-step predictions.

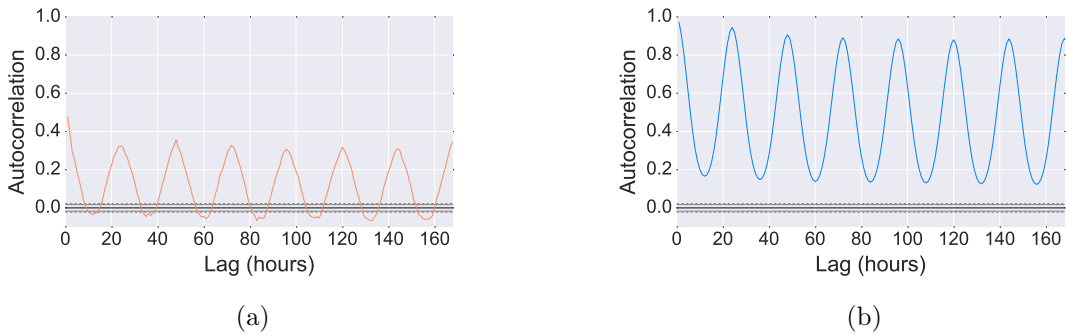


Figure 4.3: Autocorrelation over one week, where the horizontal lines correspond to the 95% and 99% confidence bands where the dashed line is the 99% confidence band, for (a) household ID 2532 (b) an aggregation of all household loads.

### 4.4.2 Calendar Features

Figure 4.4 shows the hourly consumption for one week, averaged over all weeks and households and is a representation of the load profile for the average household an average week. The weekdays are clearly distinguishable from the weekends, where the load peaks as people wake up and get ready for work, before dropping until people start to arrive home from work. For all days of the week the load peaks around 19:00, likewise every other hour of the day seems to follow a pattern. With these observations in mind, a binary feature is added indicating whether the current day is a weekday or a weekend alternatively a public holiday. Additionally, the current hour of the day is added as a feature. When adding features, thought should be given to how the features are represented in feature space. Two feature vectors that are similar should be close in the Euclidean sense in feature space, hence the feature vector with the current hour set to 23 should be close to the feature vector with current hour 00. The sine and cosine make a representation of the clock, which allows us to represent the current hour with the help of two features:

$$\text{CurrentHour}(h) = \left\{ \sin\left(\frac{2\pi h}{24}\right), \quad \cos\left(\frac{2\pi h}{24}\right) \right\}$$

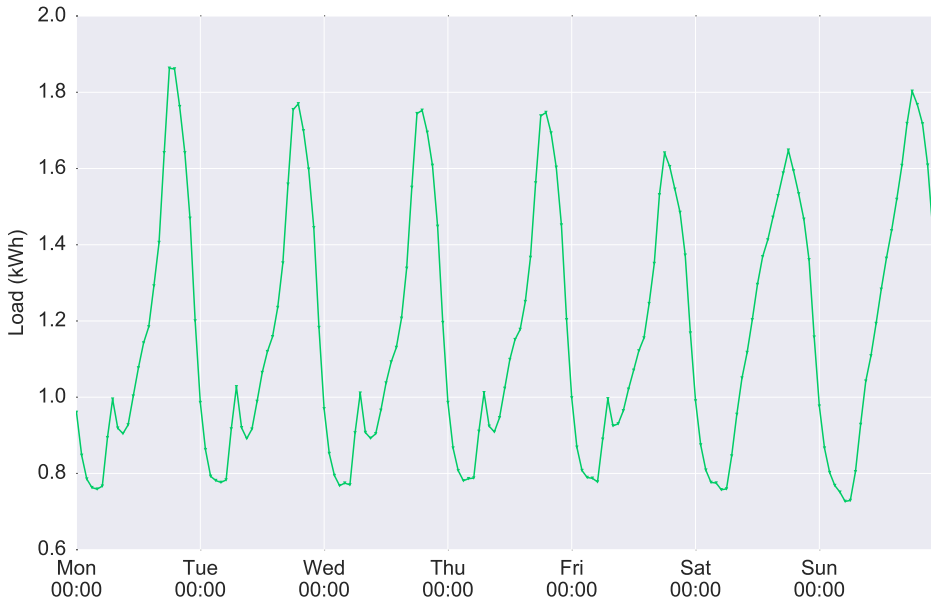


Figure 4.4: Weekly average energy consumption, averaged over all households.

### 4.4.3 Temperatures

It should come as no surprise that home electricity usage is highly correlated with the outside temperature. To test this hypothesis, temperature data for Austin, TX was gathered for the period of the dataset through the National Centers for Environmental Information (NCEI) [43]. The name of the weather station from which the data was gathered is Austin 33 NW, and Figure 4.5 shows the historic temperatures. A rolling mean with a window size of three was used to fill missing data points if there existed at least one data point in the window. For any remaining missing values, temperatures were taken from the nearby Austin-Bergstrom Intl Airport weather station.

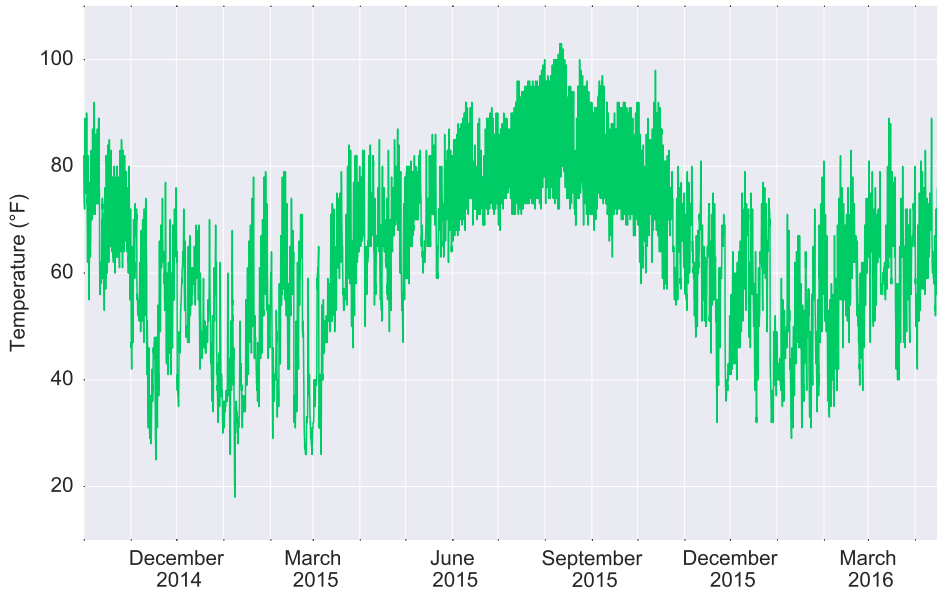


Figure 4.5: Hourly temperature in Austin, TX between November 2014 and May 2016.

Figure 4.6a shows that there is a correlation between the load and the corresponding temperature for a single household at 20:00 over the full period of the dataset. To further confirm the hypothesis Figure 4.6b displays an equal relationship, however the average load is taken for each date at 20:00 for all households. Summer days in Austin are characterized by high temperatures, and many stay inside during the hottest part of the day. The steady increase in usage as the temperature rises suggests that air conditioners are the impelling cause of the trend. With the aforementioned correlation in mind, the temperature is added as a feature. For simplicity, real temperatures have been used when predicting load instead of modeling the temperature and predicting its future evolution.

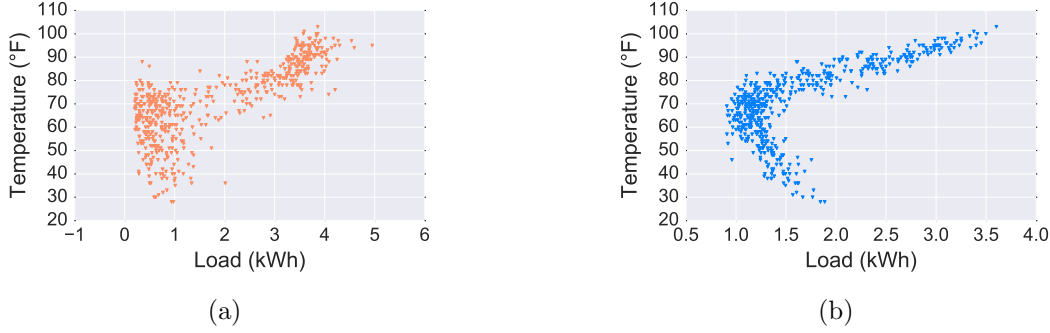


Figure 4.6: Consumption at 20:00 and the corresponding temperature (a) household ID 744 (b) averaged over all households.

#### 4.4.4 Summary

In the previous sections, prior domain knowledge has been a tool in hypothesizing feature candidates, after which an analysis has been made to confirm and lay out the exact features to be used. In summarizing, the feature vector to be used in all coming experiments is the following:

$$\mathbf{x}(t) = \left[ \text{Load}(t-1), \text{Load}(t-2), \text{Load}(t-24), \text{Load}(t-48), \right. \\ \left. \text{Weekday}(t), \sin\left(\frac{2\pi h}{24}\right), \cos\left(\frac{2\pi h}{24}\right), T(t) \right]$$

where  $\text{Load}(\cdot)$  is the electricity consumption at time  $\cdot$ ,  $\text{Weekday}(t)$  a binary variable indicating whether the current day is a working day or a weekend alternatively a public holiday,  $\sin(\frac{2\pi h}{24})$  the sine of the current hour,  $\cos(\frac{2\pi h}{24})$  the cosine of the current hour, and  $T(t)$  the temperature at time  $t$ . For each target  $y(t) = \text{Load}(t)$  there will be a corresponding eight-dimensional  $\mathbf{x}(t)$ , and it is with these eight features that the model aims to predict the future load.

## 4.5 Hyperparameter Optimization

In Section 3.1 and 3.2 the Support Vector Regression and Random Forest algorithms were outlined, both of which depend on several hyperparameters. A recapitulation of the hyperparameters is given in Table 4.1.

Model Name	Hyperparameter	Description
SVR	$C$	Penalty on errors in the approximation function
SVR	$\gamma$	Influence of a single training sample using a RBF kernel
Random Forest	$n_{trees}$	Number of trees in the forest
Random Forest	$n_{features}$	Number of features to consider at each split

Table 4.1: List of hyperparameters for SVR and Random Forest.

The selection of appropriate values for the hyperparameters is an important task, where depending on the dataset at hand, different values will result in the most robust model. The hyperparameter combinations  $\{C, \gamma\}$  and  $\{n_{trees}, n_{features}\}$  can be optimized by a cross-validated grid-search in hyperparameter space. First, a range of hyperparameter values are selected, after which a model is trained on each combination of the selected values. The performance of the model is assessed by performing k-fold cross-validation on the training set, and averaging the performance of the k folds. In the end, the set of parameters that results in the best performing model is selected.

In this thesis, 5-fold cross-validation is used to assess the accuracy of the hyperparameter combinations, since this is equal to 80% of the samples being used for training and 20% as a validation set. The model performance is assessed by the CV metric. For SVR, the parameter  $C$  is tested using a set of values  $\{1, 10, 10^2, 10^3\}$ , while  $\gamma$  is tested using a set of values  $\{2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 2^0\}$ . For the Random Forest, the parameter  $n_{trees}$  is tested using a set of values  $\{50, 100, 200, 300\}$ , whereas  $n_{features}$  is tested using a set of values  $\{1, 2, 3, 4, 5, 6, 7, 8\}$ .

An issue rises in that the computation time increases drastically for individual households when  $C \geq 1000$ , while at the same time  $\gamma \geq 0.5$ . The aggregated load curve does not suffer from the same issue as individual households: As the households are aggregated the load curve is smoothened out, in contrast to the load curves of the individual households who display a high amount of variability. Therefore SVR requires a longer period of time to solve the optimization problem for individual households. We see a particularly large computation time when the penalty on errors ( $C$ ) is large. Figure 4.7 displays CV on the validation set for SVR and Random Forest, and computation time for the set of parameter combinations  $\{C, \gamma\}$  for a individual household selected at random. In Appendix D, an equal analysis is done for four randomly selected households.

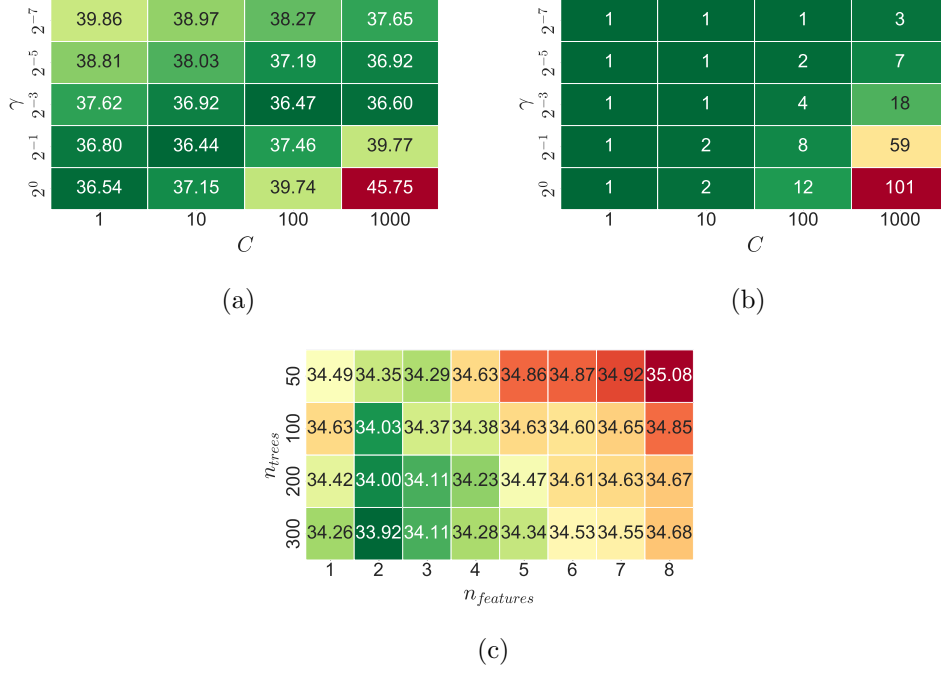


Figure 4.7: Household ID 946: (a) CV (%) for various SVR hyperparameter combinations  $\{C, \gamma\}$  with 5-fold cross-validation on the training data. (b) Computation time in seconds for various SVR hyperparameter combinations  $\{C, \gamma\}$  with 5-fold cross-validation on the training data. (c) CV (%) for various Random Forest hyperparameter combinations  $\{n_{trees}, n_{features}\}$  with 5-fold cross-validation on the training data.

What can be inferred is that the choice of hyperparameter combinations  $\{C, \gamma\}$  and  $\{n_{trees}, n_{features}\}$  does not result in significant variations in the CV metric, however it strongly affects the computation time. Based on these findings the hyperparameters are set to  $C = 100$  and  $\gamma = 2^{-3}$  for SVR and  $n_{trees} = 200$  and  $n_{features} = 2$  for Random Forest for all individual households.

On the contrary, for the aggregation of all households the choice of hyperparameters strongly affects the outcome of the predictions on the validation set. Figure 4.8 displays the CV on the validation set in addition to the computation time for the set of hyperparameter combinations. For the aggregation of all households, we let  $C = 1000$  and  $\gamma = 2^{-7}$  for SVR and  $n_{trees} = 300$  and  $n_{features} = 7$  for Random Forest. The same hyperparameter values are also used when clustering households beyond  $K = 1$  clusters, since intuitively for a small number of clusters the load curves should be smooth and display similar behavior to the aggregated case.

$\gamma$	$2^0$	$2^{-1}$	$2^{-3}$	$2^{-5}$	$2^{-7}$
$C$	1	10	100	1000	
	14.82	12.46	10.64	9.98	
	14.07	11.46	10.50	10.50	
	13.31	12.32	12.32	12.32	
	14.95	14.95	14.95	14.95	
	15.36	15.36	15.36	15.37	

(a)

$n_{trees}$	50	6.75	6.34	6.20	6.05	5.99	5.96	5.96	5.98
	100	6.68	6.25	6.11	6.01	5.97	5.92	5.93	5.96
	200	6.67	6.24	6.08	6.00	5.94	5.92	5.92	5.93
	300	6.65	6.23	6.06	5.99	5.94	5.92	5.91	5.92
		1	2	3	4	5	6	7	8
		$n_{features}$							

(b)

Figure 4.8: Aggregated households: (a) CV (%) for various SVR hyperparameter combinations  $\{C, \gamma\}$  with 5-fold cross-validation on the training data. (b) CV (%) for various Random Forest hyperparameter combinations  $\{n_{trees}, n_{features}\}$  with 5-fold cross-validation on the training data.

## Chapter 5

# Results

Chapter 5 presents the consumption forecasts of SVR and Random Forest. Section 5.1 introduces the general procedure of the forecast algorithm, before sections 5.1.1, 5.1.2 and 5.1.3 present the forecast results from forecasting individual households, the aggregate and a number of clusters consisting of similar households. Section 5.2 proposes an adjustment to the aggregate model, providing the model with more information to be able to capture cluster characteristics. Finally, Section 5.3 provides a discussion of the results.

### 5.1 Forecasting on Pecan Street Dataset

The Pecan Street dataset, consisting of 187 households, is split into a training and a test set where the training amounts to two thirds of the full dataset. The first year of data is used for training, leaving the last half year of data for testing. The training set is shuffled and normalized before training the models. In the following sections, the households are aggregated into a number of different clusters using K-Means Clustering, whereupon each cluster is trained using the aggregated training set of the households belonging to said cluster. After training, each cluster predicts its future energy consumption using the part of the test set belonging to that cluster, after which the predictions from each cluster are summed. The performance metrics are always calculated on the aggregate consumption, which effectively allows comparison of models based on a different number of underlying clusters  $K$ . The complete process is described in Algorithm 1. The baseline model is independent of the number of clusters, and the duplicate results showed for the baseline are merely a help for the reader to compare the performance of SVR and Random Forest with the performance of the baseline model.

---

**Algorithm 1** Forecast Process for SVM and Random Forest

---

**Require:**  $0 < K < \text{NumHouseholds}$ **do** Cluster households to K different clusters**for**  $i = 1$  **to**  $i = K$  **do** $Data_i(t) \leftarrow$  Aggregate load( $t$ ) of households belonging to cluster  $i$  $Data_i(t) \leftarrow$  Add features for all entries  $t$  $Data_i(t) \leftarrow$  Scale  $Data_i(t)$  to be  $[0, 1]$  $Train_i(t), Test_i(t) \leftarrow$  Split  $Data_i(t)$  into a training and a test set $Train_i(t) \leftarrow$  Shuffle the observations of the training set, i.e. create a random permutation**for** SVR **and** Random Forest **do**Train Model with training set  $Train_i(t)$ **end for** $f_i^{SVR}(t) \leftarrow$  Make predictions on  $Test_i(t)$  $f_i^{SVR}(t) \leftarrow$  Rescale predictions $f_i^{RandomForest}(t) \leftarrow$  Make predictions on  $Test_i(t)$  $f_i^{RandomForest}(t) \leftarrow$  Rescale predictions**end for** $f^{SVR}(t) \leftarrow \sum_{i=1}^K f_i^{SVR}(t)$  $f^{RandomForest}(t) \leftarrow \sum_{i=1}^K f_i^{RandomForest}(t)$ 

---

### 5.1.1 Forecasting: Individual Households

In the following section, a separate model is created for each household in the Pecan Street dataset, resulting in 187 models to be trained and evaluated both for SVR and Random Forest. Individual forecasting is achieved by setting  $K = 187$ , resulting in not performing any clustering at all on the dataset. Table 5.1 displays the performance of SVR, Random Forest and the baseline model, assessed by the metrics outlined in Section 4.3. Additionally the mean and the variance of the errors in the predictions,  $\epsilon_i = y_i - f(\mathbf{x}_i)$ , are given by  $\mu_\epsilon$  and  $\sigma_\epsilon$ . To give a graphical representation of the results, Figure 5.1 shows the predicted load curves in comparison with the true load curve over a sample period of one week. The reason for not displaying the full test set is that the period is too large to give a meaningful and understandable plot. Figures 5.2, 5.3 and 5.4 allow for an in-depth analysis of the predictions, showing a scatterplot of the predicted load versus the true load and a density plot of the resulting prediction errors  $\epsilon_i$  for each of the models.

	CV (%)	MBE (%)	MAPE (%)	$\mu_\epsilon$ (kWh)	$\sigma_\epsilon$ (kWh)
SVR	23.06	-18.80	23.14	-31.11	22.11
Random Forest	11.56	-2.28	9.24	-3.77	18.75
Baseline	12.22	-0.03	9.38	-0.05	20.22

Table 5.1: Performance metrics for  $K = 187$  clusters.

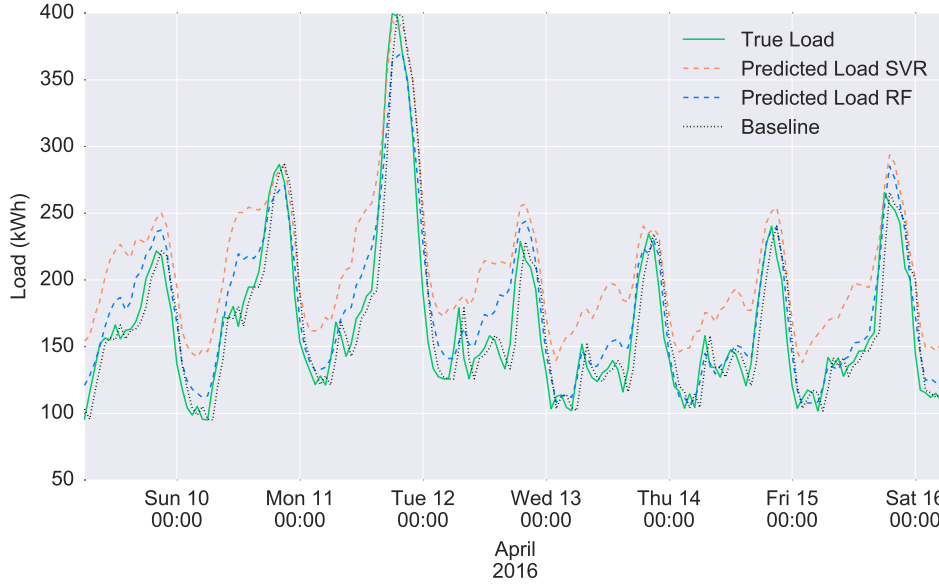


Figure 5.1: Hourly energy predictions for  $K = 187$  clusters over a sample period of one week.

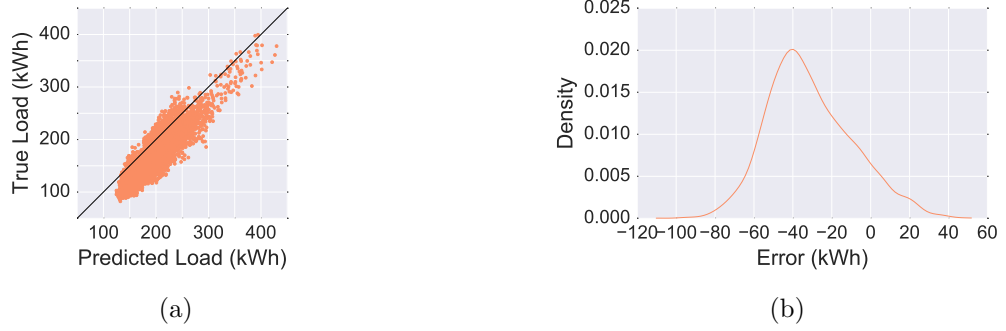


Figure 5.2: SVR with  $K = 187$  clusters: (a) Scatterplot of the predicted load versus the true load on the test set, where the black line indicates a perfect fit. (b) Density plot of the errors  $\epsilon_i = y_i - f(\mathbf{x}_i)$ .

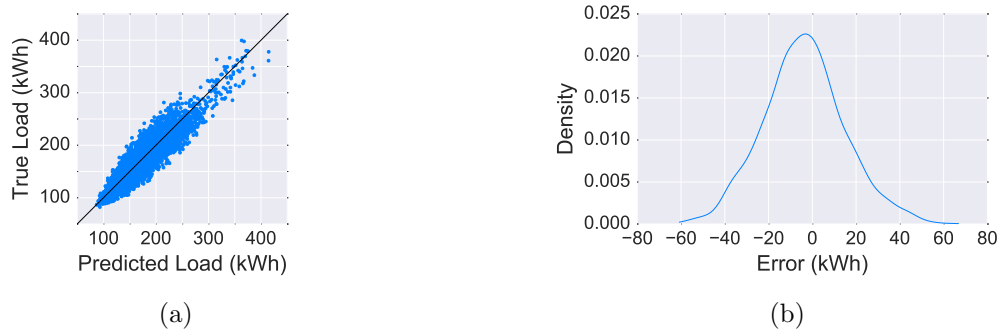


Figure 5.3: Random Forest with  $K = 187$  clusters: (a) Scatterplot of the predicted load versus the true load on the test set, where the black line indicates a perfect fit. (b) Density plot of the errors  $\epsilon_i = y_i - f(\mathbf{x}_i)$ .

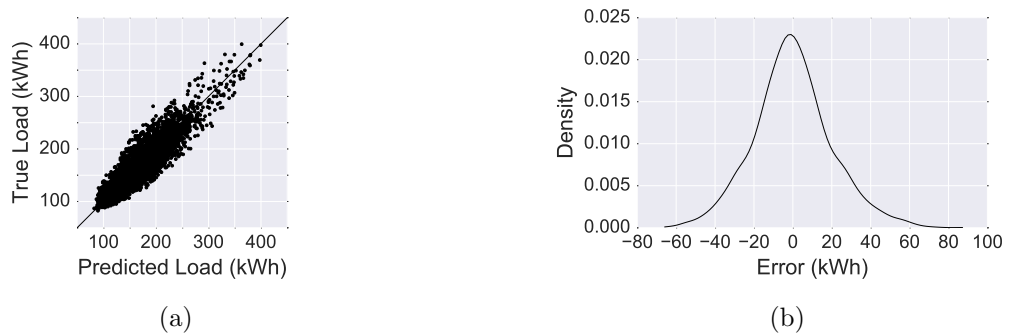


Figure 5.4: Baseline with  $K = 187$  clusters: (a) Scatterplot of the predicted load versus the true load on the test set, where the black line indicates a perfect fit. (b) Density plot of the errors  $\epsilon_i = y_i - f(\mathbf{x}_i)$ .

### 5.1.2 Forecasting: Aggregate Level

In the following section, the households are aggregated before training resulting in one single model, which is the equivalent of using  $K = 1$  clusters. Table 5.2 displays the performance of SVR, Random Forest and the baseline model, assessed by the metrics outlined in Section 4.3. Additionally the mean and the variance of the errors in the predictions,  $\epsilon_i = y_i - f(\mathbf{x}_i)$ , are given by  $\mu_\epsilon$  and  $\sigma_\epsilon$ . To give a graphical representation of the results, Figure 5.5 shows the predicted load curves in comparison with the true load curve over a sample period of one week. The reason for not displaying the full test set is that the period is too large to give a meaningful and understandable plot. Figures 5.6, 5.7 and 5.8 allow for an in-depth analysis of the predictions, showing a scatterplot of the predicted load versus the true load and a density plot of the resulting prediction errors  $\epsilon_i$  for each of the models.

	CV (%)	MBE (%)	MAPE (%)	$\mu_\epsilon$ (kWh)	$\sigma_\epsilon$ (kWh)
SVR	11.10	2.21	9.08	3.65	18.00
Random Forest	10.05	0.74	7.78	1.22	16.58
Baseline	12.22	-0.03	9.38	-0.05	20.22

Table 5.2: Performance metrics for  $K = 1$  clusters.

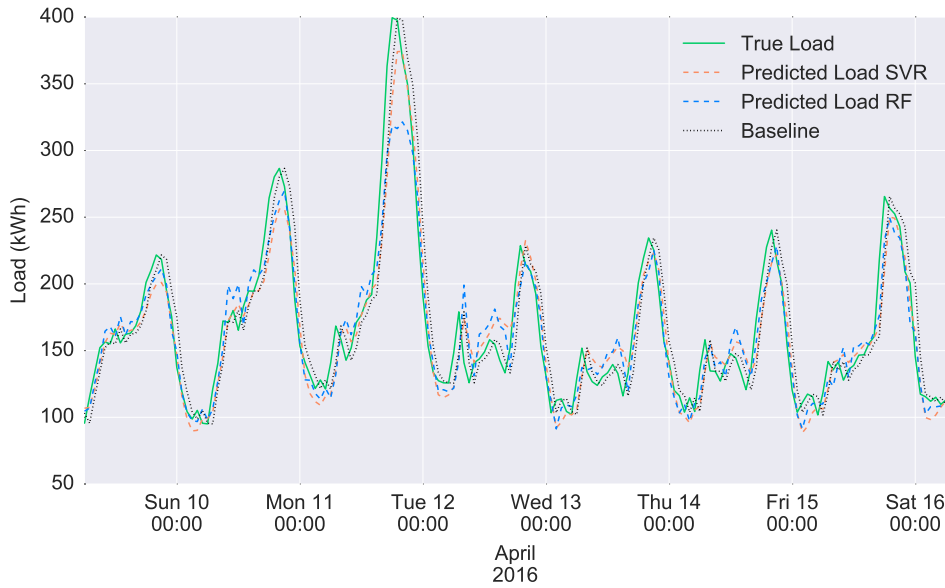


Figure 5.5: Hourly energy predictions for  $K = 1$  clusters over a sample period of one week.

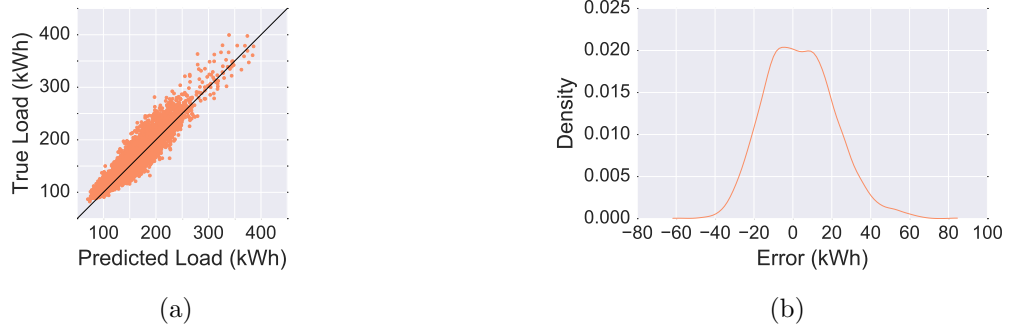


Figure 5.6: SVR with  $K = 1$  clusters: (a) Scatterplot of the predicted load versus the true load on the test set, where the black line indicates a perfect fit. (b) Density plot of the errors  $\epsilon_i = y_i - f(\mathbf{x}_i)$ .

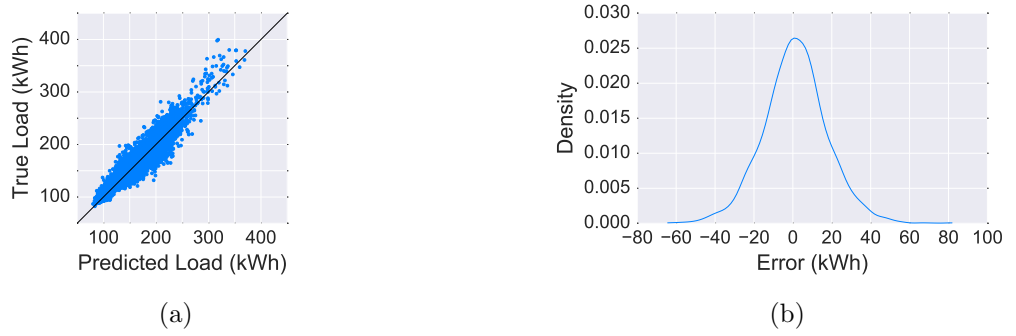


Figure 5.7: Random Forest with  $K = 1$  clusters: (a) Scatterplot of the predicted load versus the true load on the test set, where the black line indicates a perfect fit. (b) Density plot of the errors  $\epsilon_i = y_i - f(\mathbf{x}_i)$ .

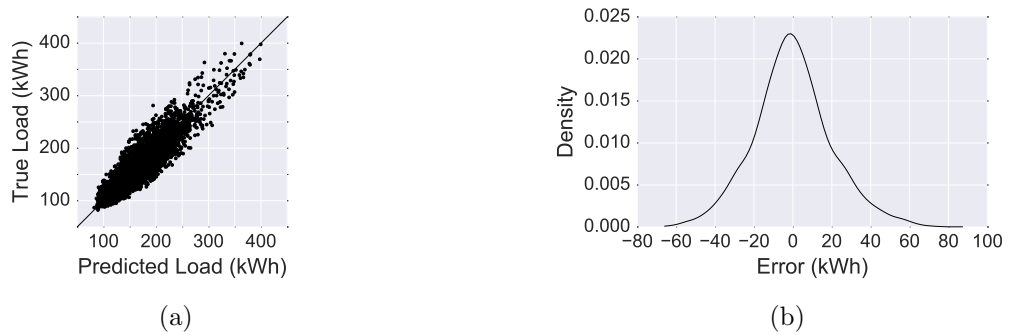


Figure 5.8: Baseline with  $K = 1$  clusters: (a) Scatterplot of the predicted load versus the true load on the test set, where the black line indicates a perfect fit. (b) Density plot of the errors  $\epsilon_i = y_i - f(\mathbf{x}_i)$ .

### 5.1.3 Forecasting: Cluster-based

The two previous sections, forecasting individual households and the aggregate, were both special cases of K-Means Clustering using  $K = 187$  and  $K = 1$  clusters respectively. In this section, a number of different clusters are explored in order to assess the optimal cluster size. Table 5.3 displays the performance of SVR, Random Forest and the baseline model, assessed by the metrics outlined in Section 4.3. Additionally the mean and the variance of the errors in the predictions,  $\epsilon_i = y_i - f(\mathbf{x}_i)$ , are given by  $\mu_\epsilon$  and  $\sigma_\epsilon$ . Figures 5.9 and 5.10 give a visual representation of the performance metrics as a function of the cluster size  $K$ .

K		CV (%)	MBE (%)	MAPE (%)	$\mu_\epsilon(\text{kWh})$	$\sigma_\epsilon(\text{kWh})$
	Baseline	12.22	-0.03	9.38	-0.05	20.22
1	SVR	11.10	2.21	9.08	3.65	18.00
	Random Forest	10.05	0.74	7.78	1.22	16.58
2	SVR	9.45	1.39	7.32	2.30	15.48
	Random Forest	9.75	0.75	7.60	1.25	16.09
3	SVR	9.41	1.60	7.28	2.65	15.34
	Random Forest	9.57	1.13	7.49	1.87	15.73
4	SVR	9.74	2.08	7.44	3.45	15.75
	Random Forest	9.76	2.44	7.44	4.04	15.63
6	SVR	9.49	-0.30	7.72	-0.50	15.70
	Random Forest	9.58	1.44	7.40	2.38	15.68
8	SVR	9.41	-0.66	7.62	-1.10	15.54
	Random Forest	9.33	1.30	7.17	2.15	15.29
16	SVR	9.31	-1.24	7.61	-2.06	15.26
	Random Forest	9.10	2.02	6.91	3.35	14.68
32	SVR	10.50	-3.70	9.25	-6.12	16.26
	Random Forest	8.89	1.43	6.77	2.37	14.52
64	SVR	13.27	-7.96	12.68	-13.17	17.57
	Random Forest	9.20	0.07	7.21	0.12	15.23
187	SVR	23.06	-18.80	23.14	-31.11	22.11
	Random Forest	11.56	-2.28	9.24	-3.77	18.75

Table 5.3: Performance metrics for all models and cluster sizes.

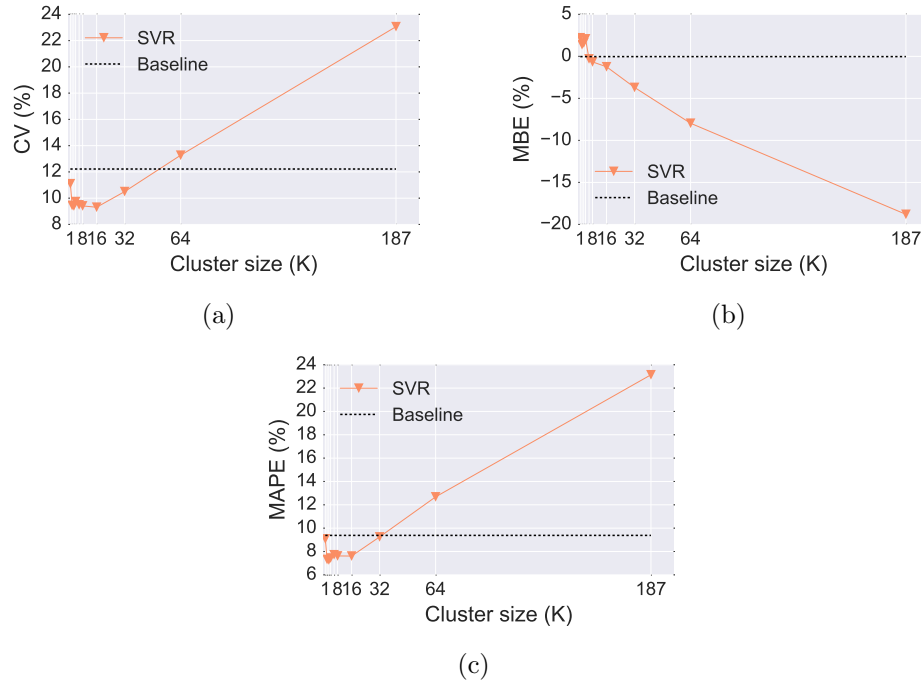


Figure 5.9: Performance metrics as a function of the cluster size  $K$  for SVR: (a) CV (%) (b) MBE (%) (c) MAPE (%). The baseline model is included as a benchmark.

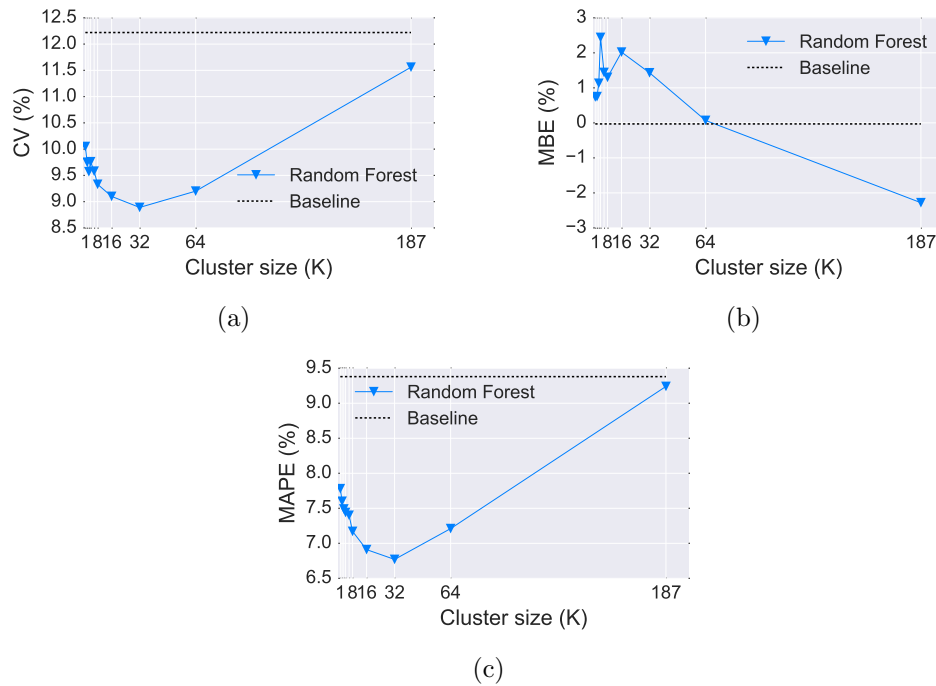


Figure 5.10: Performance metrics as a function of the cluster size  $K$  for Random Forest: (a) CV (%) (b) MBE (%) (c) MAPE (%). The baseline model is included as a benchmark.

#### 5.1.4 Analysis

From the results a general trend can be inferred: Random Forest shows a stable performance for all cluster sizes and outperforms the baseline for all tested cases. SVR on the other hand, while in some cases better than the Random Forest, diverges when the cluster size increases and is beaten by the baseline. The experiments were run several times with different seeds, where Random Forest showed no significant variations in the performance. Additionally, as the performance of SVR is independent of the randomness involved in the forecasting, any variations are omitted from the results.

For the approach of forecasting individual households, Random Forest yields the most desirable results, followed by the baseline model. SVR performs significantly worse than the aforementioned models in this case, as can be seen in Table 5.1. A possible explanation might be that SVR is quite sensitive to the choice of hyperparameters, and that using the same hyperparameter values for all different households might not be optimal. In addition to high CV and MAPE values, SVR also heavily overshoots in its predictions, as is seen in Figure 5.2 where the average prediction is off by over 30 kWh where the true values range from approximately 80 to 800 kWh. Random Forest also overshoots slightly in its predictions, however this model captures the trend in the consumption to a significantly higher degree.

Table 5.2 shows that forecasting on the aggregate level is preferably done with a Random Forest. Contrary to individual forecasting, SVR here outperforms the baseline model. Figures 5.6 and 5.7 along with Table 5.2 indicate that SVR is undershooting slightly, whereas Random Forest does not seem to either overshoot nor undershoot in its predictions.

The cluster-based forecasting approach yields results that indicate that for some number of clusters  $K$ , both SVR and Random Forest outperform the individual and aggregate approaches respectively. By looking at the CV value in Table 5.3, the optimal model is found using Random Forest with  $K = 32$  different clusters, followed by Random Forest with  $K = 16$  and  $K = 64$  clusters, after which SVR with  $K = 16$  clusters is chosen. The performance measured by the MAPE also indicates that Random Forest with  $K = 32$  clusters is preferred. By the MAPE, SVR obtains its optimal performance using  $K = 3$  clusters. The optimal Random Forests are for all mentioned cluster sizes  $K = 32, 16$  and  $64$  undershooting somewhat, whereas SVR with  $K = 16$  is slightly overshooting. While CV is used as the primary metric to determine the optimal model, the MBE can play an important role if a desired strategy is to either always be conservative and underpredict the consumption or to be confident and make sure the predictions are always capturing the load peaks.

## 5.2 Accounting for Cluster Characteristics in Aggregate Forecasting

A question rises whether or not the models are given enough information. In the introduction it was hypothesized that in theory individual forecasting should outperform aggregate forecasting since we are able to account for each individual's consumption pattern. To test this hypothesis, an alteration of the aggregate model is given where households are first grouped into a number of clusters  $K$ . Then, using the original eight features given in 4.4.4,  $\text{Load}(t-1)$ ,  $\text{Load}(t-2)$ ,  $\text{Load}(t-24)$ ,  $\text{Load}(t-48)$  are added as features to the aggregate model for each cluster  $K$ . Finally, the aggregate model is trained and predictions are made on the aggregate level. Effectively, the aggregate model makes predictions based on its eight original features along with four new features for each additional cluster  $K$ , resulting in a total of  $8 + 4K$  features.

Table 5.4 displays the performance of SVR, Random Forest and the baseline model, assessed by the metrics outlined in Section 4.3. Additionally the mean and the variance of the errors in the predictions,  $\epsilon_i = y_i - f(\mathbf{x}_i)$ , are given by  $\mu_\epsilon$  and  $\sigma_\epsilon$ . Figures 5.11 and 5.12 give a visual representation of the performance metrics as a function of the cluster size  $K$ .

K		CV (%)	MBE (%)	MAPE (%)	$\mu_\epsilon(\text{kWh})$	$\sigma_\epsilon(\text{kWh})$
	Baseline	12.22	-0.03	9.38	-0.05	20.22
1	SVR	11.10	2.21	9.08	3.65	18.00
	Random Forest	10.05	0.74	7.78	1.22	16.58
2	SVR	11.70	3.40	9.41	5.62	18.52
	Random Forest	9.59	1.60	7.34	2.64	15.65
3	SVR	11.32	2.19	8.92	3.63	18.38
	Random Forest	9.27	0.85	7.08	1.40	15.28
4	SVR	11.10	2.53	8.55	4.18	17.89
	Random Forest	9.53	1.90	7.12	3.14	15.46
6	SVR	12.39	4.62	9.39	7.65	19.02
	Random Forest	9.88	3.51	7.20	5.81	15.28
8	SVR	13.71	7.06	9.88	11.69	19.44
	Random Forest	11.26	6.35	8.09	10.50	15.39
16	SVR	16.30	11.18	11.55	18.51	19.63
	Random Forest	14.85	10.75	10.76	17.79	16.95
32	SVR	20.76	16.22	15.10	26.84	21.45
	Random Forest	21.51	18.02	17.02	29.82	19.44

Table 5.4: Performance metrics for aggregate models with information from  $K$  clusters.  $K = 1$  refers to the original aggregate model from Section 5.1.2.

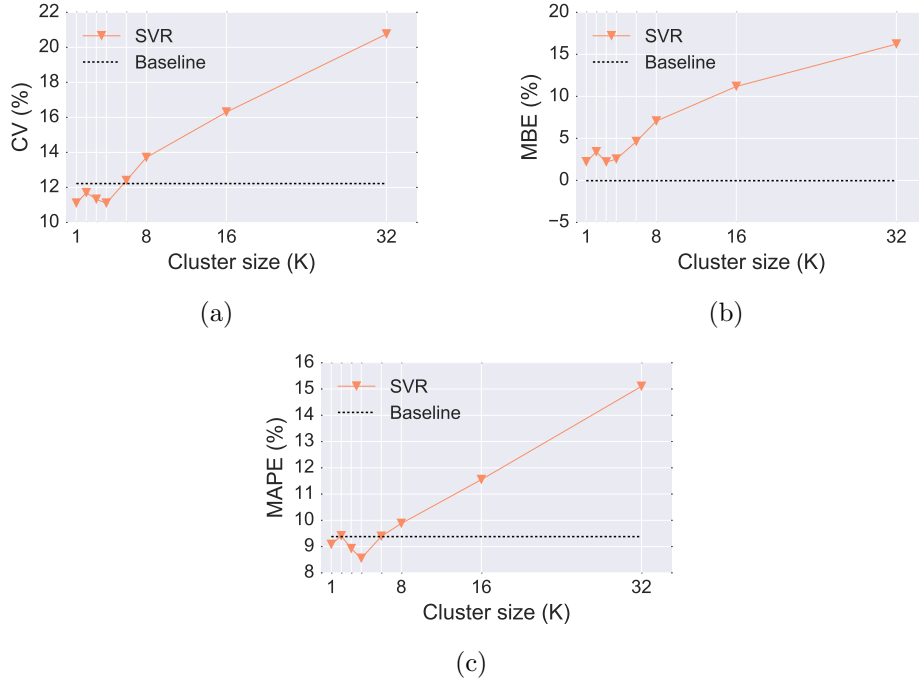


Figure 5.11: Performance metrics for the aggregate as a function of the information from  $K$  clusters for SVR: (a) CV (%) (b) MBE (%) (c) MAPE (%). The baseline model is included as a benchmark.  $K = 1$  refers to the original aggregate model from Section 5.1.2.

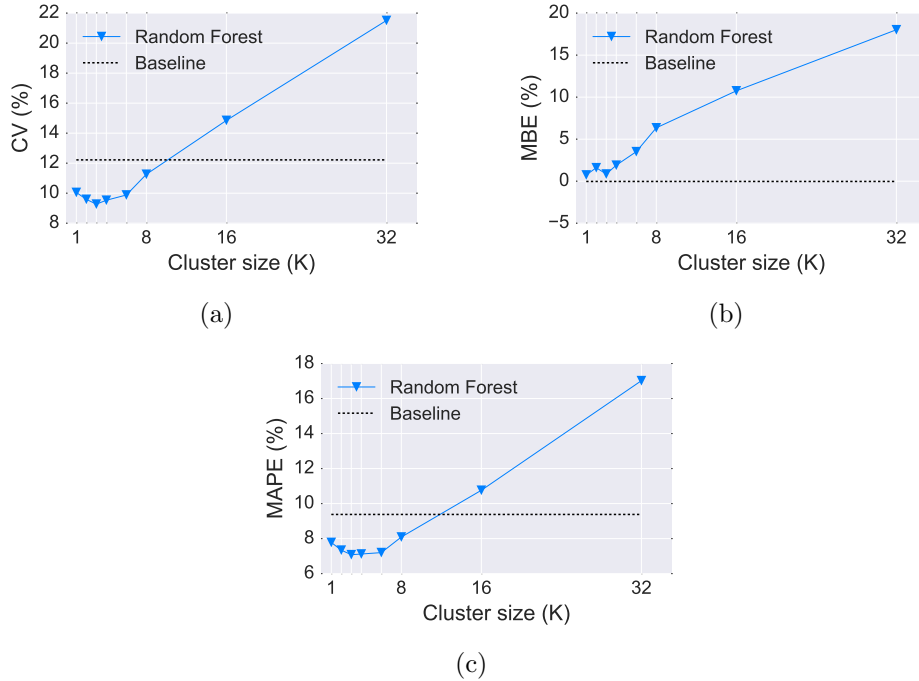


Figure 5.12: Performance metrics for the aggregate as a function of the information from  $K$  clusters for Random Forest: (a) CV (%) (b) MBE (%) (c) MAPE (%). The baseline model is included as a benchmark.  $K = 1$  refers to the original aggregate model from Section 5.1.2.

### 5.2.1 Analysis

The extension of the original aggregate model provides some interesting results. As is seen in Table 5.4, adding additional information in the form of extra features can provide a stronger model. Compared to the original aggregate model, which is the case with  $K = 1$ , Random Forest provides an improvement when historic load is added based on  $K = 3$  clusters. Increasing the number of clusters increases the CV value, up to  $K = 8$  clusters where the extension is worse than the original, and  $K = 16$  clusters where Random Forest is outperformed by the baseline. While the CV and MAPE is better for Random Forest with  $K = 3$  compared to the original case, the model's MBE is higher, indicating that it is undershooting to a larger extent than in the original case.

SVR shows no significant improvements when adding additional information, the CV value of the original model is matched when using  $K = 4$  clusters, however at the cost of an increased MBE. It is possible that SVR suffers from the previously described drawback, namely that the hyperparameters  $\{C, \gamma\}$  should be recalibrated with an exhaustive grid-search for each new model. This disadvantage is further discussed in the next section.

## 5.3 Discussion

The results were briefly commented in 5.1.4 and 5.2.1, however in what follows is a more thorough discussion of the outcome of the experiments, potential explanations for certain behavior and advantages of selecting specific model setups.

For the initial results on investigating the impact of the number of clusters on load forecasting, both SVR and Random Forest showed improved results when using a small number of clusters  $K$  compared to the aggregate case with  $K = 1$  clusters. As both Figure 5.9 and 5.10 display, the CV value reaches a minimum as the number of clusters increases from  $K = 1$ , before increasing and reaching a maximum at  $K = 187$ . The best performing model was found for Random Forest with  $K = 32$  clusters, and for SVR with  $K = 16$  clusters.

The best performing model, a Random Forest with  $K = 32$  clusters, indicates that grouping the households based on the daily load profile of each household is preferably done by dividing the 187 households into 32 different clusters. In Appendix C the grouping of households among the 32 clusters is showed, and what can be inferred is that there is primarily 10 large groups of typical households, followed by a number of very specialized households. Grouping by the average daily load profile of each household is definitely an interesting and intuitive way to cluster households, however using other representations to cluster households would be interesting to see how it affects the outcome of the predictions.

Although the behavior is similar for SVR and Random Forest, Random Forest shows

a much more stable performance. Where the difference in CV values depending on the cluster size differs only 3% for Random Forest, the maximum and minimum are separated by over 13% for SVR. In addition, SVR is outperformed by the baseline model as  $K \geq 64$ . As was mentioned briefly in previous sections, Random Forests are rather insensitive to the choice of hyperparameters which explains the stable performance. The large variations in model performance for SVR could be due to the fact that the hyperparameters  $\{C, \gamma\}$  were limited to only two different pairs of values, depending on whether the model was for individual households or a cluster of similar households. Even though the analysis in 4.5 suggested otherwise, SVR might be quite sensitive to parameter selection, especially considering when small errors from many households are added together resulting in an overall bad performance. To test the true potential of SVR, an exhaustive parameter search would have to be done for each of the models (in the case of individual load forecasting this results in 187 models having to find its optimal hyperparameters), requiring a vast amount of computational power.

In an attempt to provide the model with more information about the individual household's consumption, the load of several clusters was added to the aggregate model, resulting in a single model with  $8 + 4K$  features instead of the original eight,  $K$  being the number of clusters to group the households into. The rationale for taking this approach is to improve performance in terms of lowered CV values, but also to increase computational efficiency. In the original model setup, a model has to be trained for each cluster  $K$ , where we tested values from  $K = 1$  up to  $K = 187$ . Intuitively, it is more desirable having to handle a single model compared to training and optimizing 187 different models. Still, as households are aggregated we lose information about the individual households, which is the reason for wishing to add features that conserve the individual patterns while at the same time enabling training and forecasting on the aggregate level.

With this approach, the number of clusters used to provide additional information to the aggregate model had a significant impact on the outcome of the predictions. In contradiction with the optimal cluster-size for cluster-based forecasting, this new approach prefers grouping to a smaller number of clusters. Random Forest showed an improvement compared to the aggregate model, and the best model was attained when information was added based on the grouping into  $K = 3$  clusters, resulting in predictions based on 20 features. This result is also, although not an improvement, comparable with the outcome of the cluster-based forecasting. SVR showed no significant improvement when providing the model with more information, instead the results quickly became worse compared to the baseline model. An underlying reason for the decline of SVR could be similar to what was recently discussed, namely that the model might require an exhaustive parameter search. Another reason for the stagnating performance of Random Forest and SVR as more information is added could be that the number of features increases to a level where significant overfitting takes place. Adding information from  $K = 32$  clusters results in models with 136 features, compared to just having eight features originally.

## Chapter 6

# Conclusion

The last chapter discusses the most important findings of the thesis. Section 6.1 reviews the objectives presented in 1.1. Section 6.2 concludes the thesis.

### 6.1 Follow-up of Objectives

With the results of the load forecasting presented, the objectives that were set at the beginning of the thesis are reviewed. Four objectives were stated relatively general, describing statements that we hoped to be able to answer. Here, the results are connected with the objectives in an attempt to clarify the outcome of the thesis and structure the main findings.

#### 1. Analyze historical residential energy consumption data

*Given historical time series of energy consumption from individual households, conduct a statistical analysis to highlight the behavior of the individual series and the dependence between different series.*

Short-term load forecasting is a highly volatile and non-linear problem, as was indicated by the difficulties of producing accurate forecasts of the future energy consumption and visualized through example figures of individual households. However, we saw that by grouping households the load curve was smoothened out resulting in a somewhat more recurring pattern. The advantages of grouping households in different ways was also presented in the prediction accuracy, indicating that a relationship exists between different households. As the number of families in the customer base increases the aggregate should become easier to predict from a mathematical perspective, since the load curve smoothes out. Several factors were shown to correlate with the load, in particular the load was shown to be periodic following the rather routine lifestyle of many households. The addition of exogenous features such as the outside temperature provides a valuable insight

suggesting that the consumption of a household is based on a number of factors. Including more exogenous features, may it be social or economic, would yield an interesting experiment in which we could learn more about the underlying factors for different consumption patterns.

## **2. Review and select appropriate mathematical models**

*Review and evaluate mathematical models, emphasizing their ability to capture load dynamics and successfully forecast the future load.*

A thorough and extensive study was made on existing literature in the field of short-term load forecasting, both within the commercial and residential sector. The studies presented a range of methods and approaches used for STLF, however in our interest to take a data-driven approach, we narrowed in on machine learning methods to be able to handle the non-linearity that presents itself in STLF series. Within machine learning, some methods clearly stood out and in particular Support Vector Machines for regression and Artificial Neural Networks had shown satisfying results, with the first being superior. In addition a Random Forest was suggested to test a previously unseen algorithm. Based on the performed experiments, Random Forest provided successful forecasts and was to a large extent more accurate and stable compared to SVR. The advantages of using a Random Forest are manifold: The algorithm is rather independent of the choice of hyperparameters as was shown in an experiment, which removes the need for an exhaustive parameter search and reduces computational strain. Due to the structure of the algorithm, it is exempt from a high degree of overfitting which is a common issue for many algorithms. The structure also allows Random Forests to scale well with large datasets and high-dimensional feature vectors.

## **3. Experiment with clustering and cluster sizes**

*Conduct experiments to assess the impact of clustering and the cluster size on the accuracy of the load forecast.*

Clustering was performed to group similar households by looking at the average daily load profile of households. The selected cluster feature provides an intuitive way of grouping households since the average daily load profile represents the consumption a typical day for every family. Therefore, similar households should display similar average daily load profiles. A question rises whether or not this is the best way to cluster households. While we can be certain that on average the grouped households show a similar load, behavior of extremities such as peaks or longer periods of absence due to holidays could differ vastly. In much the same way the strategy of an electric utility affects if we prefer an over- or underpredicting model, the cluster feature could be set depending on the preferences of the utility. In the scope of this thesis, clustering based on the average daily load profile provided favorable results that outperformed the models where no clustering was undertaken. We found that Random Forest and SVR yielded the most

accurate results with respect to the CV value when clustering with  $K = 32$  and  $K = 16$  clusters respectively. In an attempt to reduce the number of models and see if the aggregate model could be improved by adding more features, historic load from clusters was added as features to the aggregate model. While the extended aggregate model did not provide improvements compared the cluster-based models, it improved the results compared to the original aggregate model using Random Forest with information added based on the grouping into  $K = 3$  clusters. The extended aggregate model indicates that, rather than performing cluster-based forecasting, by adding more information in terms of features, we can create a single strong predictive model which has the benefit of reducing computation time.

#### 4. Discuss implications on short-term load forecasting

*Given results of the above points, discuss strengths and weaknesses of outlined models and propose adjustments to allow for future studies to advance research within STLf.*

It was shown that households can be grouped into a number of typical consumption patterns, which effectively improved the prediction accuracy compared to traditional methods. The introduction of Random Forests for load forecasting showed impressive results and stable performance across all different approaches taken, and is a method that should be tested to a larger extent in future work. A major disappointment was the accuracy of SVR, however as has been discussed in detail, it could be due to the decision that was made to omit an exhaustive hyperparameter search due to the lack of computational power. The results in this thesis have demonstrated the effectiveness of a cluster-based approach, however to further research within STLf future work could include using algorithms to select the best features from a large feature space (**Feature selection**), implement ANNs and compare to the findings of this thesis (**ANNs**), test other cluster features and methods (**Cluster features and methods**), experiment with new datasets to assess if the results are similar (**New datasets**), extend the single-step forecasts to multi-step forecasts (**Multi-step forecasts**) and analyze the impact of the size of the customer base on the prediction accuracy (**Size of customer base**).

## 6.2 Concluding Remarks

In this thesis a model was developed for predicting the hourly residential energy consumption of a customer base consisting of 187 households in Austin, Texas. To arrive at the aggregate predictions of the customer base, three different approaches were taken to building the model. First, each households was modeled independently, yielding a total number of 187 models whose predictions were aggregated to form the total prediction. Secondly, the households were aggregated and a single model was developed, treating the customer base as a single unit. A third option consisted of grouping similar

households based on the average daily load profile of each household. Comparing the approaches, using both Random Forest and Support Vector Regression, showed that the most accurate results were achieved using Random Forest and clustering households into 32 different clusters. In an attempt to improve the aggregate model, it was shown that by adding features describing the clusters' historic load, the performance of the aggregate model was improved using Random Forest with information added based on the grouping into  $K = 3$  clusters. The extended aggregate model did not outperform the cluster-based models.

Deciding the characteristics of a short-term load forecasting model in a real world scenario will be heavily influenced by the requirements of the utility, where factors such as the risk appetite, supply requirements and finances will play an important role. It is without doubt a topic that will grow more important as the smart grid develops, where all parties involved should reap the environmental and economic benefits of progressing short-term load forecasting.

# Bibliography

- [1] Statistiska Centralbyrån. *Tillförsel och användning av el 2001–2014 (GWh)*. (Swedish) [*The supply and use of electricity 2001–2014 (GWh)*]. 2016. URL: [http://www.scb.se/sv/\\_Hitta-statistik/Statistik-efter-amne/Energi/Tillforsel-och-anvandning-av-energi/Arlig-energistatistik-el-gas-och-fjarrvarme/6314/6321/24270/](http://www.scb.se/sv/_Hitta-statistik/Statistik-efter-amne/Energi/Tillforsel-och-anvandning-av-energi/Arlig-energistatistik-el-gas-och-fjarrvarme/6314/6321/24270/) (visited on 04/19/2016).
- [2] Statistiska Centralbyrån. *Slutanvändning (MWh), efter län och kommun, förbrukarkategori samt bränsletyp. År 2009 - 2014*. (Swedish) [*End use (MWh), by county and municipality, consumer category and fuel type. Years 2009 - 2014*]. 2016. URL: [http://www.statistikdatabasen.scb.se/pxweb/sv/ssd/START\\_\\_EN\\_\\_EN0203/SlutAnvSektor/?rxid=336d4c1c-aba7-4a47-b61d-4144dd5ca29a#](http://www.statistikdatabasen.scb.se/pxweb/sv/ssd/START__EN__EN0203/SlutAnvSektor/?rxid=336d4c1c-aba7-4a47-b61d-4144dd5ca29a#) (visited on 04/20/2016).
- [3] European Commission. *2020 climate & energy package*. 2014. URL: [http://ec.europa.eu/clima/policies/strategies/2020/index\\_en.htm](http://ec.europa.eu/clima/policies/strategies/2020/index_en.htm) (visited on 04/20/2016).
- [4] U.S. Department of Energy. *What is the Smart Grid?* 2016. URL: [https://www.smartgrid.gov/the\\_smart\\_grid/smart\\_grid.html](https://www.smartgrid.gov/the_smart_grid/smart_grid.html) (visited on 02/24/2016).
- [5] European Commission. *Commission Recommendation of 9 March 2012 on preparations for the roll-out of smart metering systems*. 2013. URL: <http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32012H0148&from=SV> (visited on 04/20/2016).
- [6] B. F. Hobbs et al. “Analysis of unit commitment of improved load forecasts”. In: *IEEE Transactions on Power Systems* 14 (1999), pp. 1342–1348.
- [7] Svenska Kraftnät. *Förbrukningsprofiler*. (Swedish) [*Consumption Profiles*]. 2016. URL: <https://mimer.svk.se/ConsumptionProfile/Submit> (visited on 04/19/2016).
- [8] Pecan Street Inc. Dataport. 2016. URL: <https://dataport.pecanstreet.org> (visited on 04/25/2016).
- [9] D. J. C. Mackay. “Bayesian Non-linear Modelling for the Prediction Competition”. In: *ASHRAE Trans* 100 (1994), pp. 1053–1062.
- [10] R. E. Edwards, J. New, and L. E. Parker. “Predicting future hourly residential electrical consumption: A machine learning case study”. In: *Energy and Buildings* 49 (2012), pp. 591–603.

- 
- [11] A. Boiron, S. Lo, and A. Marot. “Predicting Future Energy Consumption”. In: *Stanford University* (2012).
  - [12] S.I. Hill et al. “The impact on energy consumption of daylight saving clock changes”. In: *Energy Policy* 38.9 (2010), pp. 4955–4965.
  - [13] S.S. Pappas et al. “Electricity demand loads modeling using AutoRegressive Moving Average (ARMA) models”. In: *Energy* 33.9 (2008), pp. 1353–1360.
  - [14] A. Veit et al. “Household Electricity Demand Forecasting - Benchmarking State-of-the-Art Methods”. In: *Proceedings of the 5th International Conference on Future Energy Systems e-Energy '14* (2014), pp. 233–234.
  - [15] H. Takeda, Y. Tamura, and S. Sato. “Using the ensemble Kalman filter for electricity load forecasting and analysis”. In: *Energy* 104 (2016), pp. 184–198.
  - [16] Y. T. Chae et al. “Artificial neural network model for forecasting sub-hourly electricity usage in commercial buildings”. In: *Energy and Buildings* 111 (2016), pp. 184–194.
  - [17] Y. Fu et al. “Using Support Vector Machine to Predict Next Day Electricity Load of Public Buildings with Sub-metering Devices”. In: *Procedia Engineering* 121 (2015), pp. 1016–1022.
  - [18] K. Gajowniczek and T. Zabkowski. “Short term electricity forecasting using individual smart meter data”. In: *Procedia Computer Science* 35 (2014), pp. 589–597.
  - [19] R. K. Jain et al. “Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy”. In: *Applied Energy* 123 (2014), pp. 168–178.
  - [20] S. Humeau et al. “Electricity Load Forecasting for Residential Customers: Exploiting Aggregation and Correlation between Households”. In: *Sustainable Internet and ICT for Sustainability* (2013), pp. 1–6.
  - [21] S. Humeau et al. “Individual, Aggregate, and Cluster-based Aggregate Forecasting of Residential Demand”. In: *EPFL* (2014).
  - [22] C. Lier. “Applying Machine Learning Techniques to Short Term Load Forecasting”. In: *University of Groningen* (2015).
  - [23] J. Yang and J. Stenzel. “Short-term load forecasting with increment regression tree”. In: *Electric Power Systems Research* 76 (2006), pp. 880–888.
  - [24] H. Zhao and F. Magoulès. “Feature Selection for Predicting Building Energy Consumption Based on Statistical Learning Method”. In: *Journal of Algorithms & Computational Technology* 6.1 (2012), pp. 59–77.
  - [25] G. Dudek. “Short-Term Load Forecasting Using Random Forests”. In: *Intelligent Systems'2014: Proceedings of the 7th IEEE International Conference Intelligent Systems IS'2014, Warsaw, Poland 2* (2015), pp. 821–828.
  - [26] F. Rosenblatt. *Principles of Neurodynamics*. Spartan Books, 1962.
  - [27] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning internal representations by backpropagating errors”. In: *Nature* 323 (1986), pp. 533–536.

- [28] V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, 1982.
- [29] V. Vapnik and A. Y. Chervonenkis. *Teoriya raspoznavaniya obrazov: Statisticheskie problemy obucheniya. (Russian) [Theory of pattern recognition: Statistical problems of learning]*. Nauka, 1974.
- [30] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.
- [31] A. J. Smola and B. Schölkopf. “A Tutorial on Support Vector Regression”. In: *Statistics and Computing* 14.3 (2004), pp. 199–222.
- [32] K. P. Bennett and O. L. Mangasarian. “Robust linear programming discrimination of two linearly inseparable sets”. In: *Optimization Methods and Software* 1 (1992), pp. 23–34.
- [33] C. Cortes and V. Vapnik. “Support-Vector Networks”. In: *Machine Learning* 20.3 (1995), pp. 273–297.
- [34] O. L. Mangasarian. *Nonlinear Programming*. McGraw-Hill, 1969.
- [35] B. E. Boser, I. Guyon, and V. Vapnik. “A training algorithm for optimal margin classifiers”. In: *Proceedings of the Fifth Annual Workshop of Computational Learning Theory, ACM Pittsburgh* 5 (1992), pp. 144–152.
- [36] J. Mercer. “Functions of positive and negative type and their connection with the theory of integral equations”. In: *Philosophical Transactions of the Royal Society, London* 209 (1909), pp. 415–446.
- [37] E. B. Hunt, J. Marin, and P. J. Stone. *Experiments in Induction*. Academic Press, 1966.
- [38] R. Quinlan. “Discovering Rules by Induction from Large Collections of Examples”. In: *Expert Systems in the Micro-electronic Age, Edinburgh University Press* (1979), pp. 168–201.
- [39] L. Breiman et al. *Classification and Regression Trees*. Wadsworth, 1984.
- [40] L. Breiman. “Bagging predictors”. In: *Machine Learning* 24.2 (1996), pp. 123–140.
- [41] J. H. Friedman. “On bias, variance, 0/1 - loss, and the curse-of-dimensionality”. In: *Department of Statistics and Stanford Linear Accelerator Center* (1996).
- [42] L. Breiman. “Random forests”. In: *Machine Learning* 45.1 (2001), pp. 5–32.
- [43] National Centers for Environmental Information. 2016. URL: <http://www.ncei.noaa.gov> (visited on 05/05/2016).

## Appendix A

# Mercer's Theorem

**Theorem 1.** *Let  $\mathbb{F}$  be a compact subset of  $\mathbb{R}^N$ . Suppose  $K : \mathbb{F} \times \mathbb{F} \rightarrow \mathbb{R}$  is a continuous and symmetric function, which is square-integrable in  $\mathbb{F} \times \mathbb{F}$  and satisfies:*

$$\int_{\mathbb{F} \times \mathbb{F}} K(\mathbf{x}, \mathbf{y}) f(\mathbf{x}) f(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0 \quad \forall f \in L_2(\mathbb{F})$$

*then there exist functions  $\phi_i : \mathbb{F} \rightarrow \mathbb{R}$  and numbers  $\lambda_i \geq 0$  such that*

$$K(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y}) \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{F}$$

## Appendix B

# Optimizing the Leaf Node Value of a Regression Tree

Let  $Y$  be a continuous random variable with probability density function  $f(y)$ . We want to minimize the expected value of the mean squared error with respect to  $c_l$ :

$$\begin{aligned}\epsilon &= \mathbb{E}[(Y - c_l)^2] \\ &= \int_{-\infty}^{+\infty} (y - c_l)^2 f(y) dy \\ &= \int_{-\infty}^{+\infty} (y^2 - 2yc_l + c_l^2) f(y) dy \\ &= \int_{-\infty}^{+\infty} y^2 f(y) dy - 2c_l \int_{-\infty}^{+\infty} y f(y) dy + c_l^2\end{aligned}$$

Since  $\int_{-\infty}^{+\infty} f(y) dy = 1$ . Minimize  $\epsilon$  with respect to  $c_l$ :

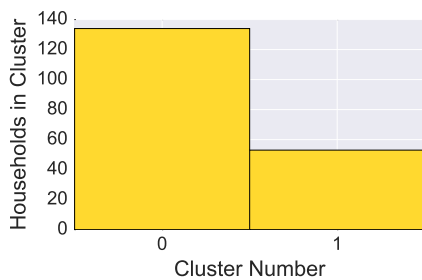
$$\begin{aligned}\frac{\partial \epsilon}{\partial c_l} &= 0 - 2 \int_{-\infty}^{+\infty} y f(y) dy + 2c_l = 0 \\ &\leftrightarrow \\ c_l &= \int_{-\infty}^{+\infty} y f(y) dy\end{aligned}$$

This is the definition of the expected value of  $Y$ , hence:

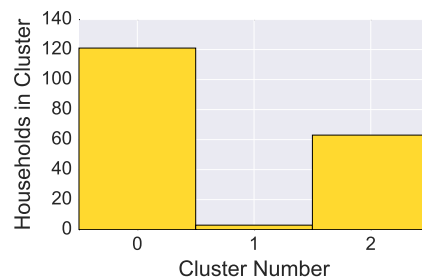
$$c_l = \mathbb{E}[Y] \quad \blacksquare$$

## Appendix C

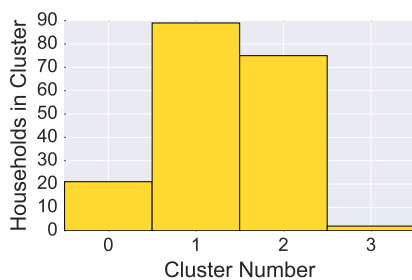
# Distribution of Households with K-Means Clustering



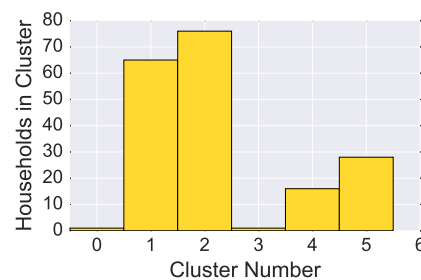
(a) Distribution of households among  $K = 2$  clusters.



(b) Distribution of households among  $K = 3$  clusters.



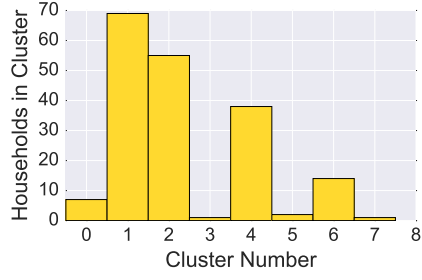
(c) Distribution of households among  $K = 4$  clusters.



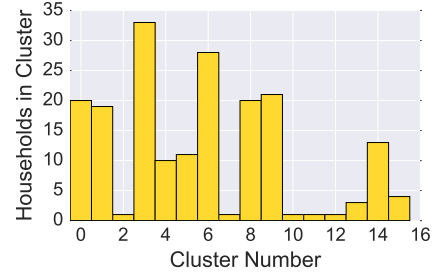
(d) Distribution of households among  $K = 6$  clusters.

## APPENDIX C. DISTRIBUTION OF HOUSEHOLDS WITH K-MEANS CLUSTERING

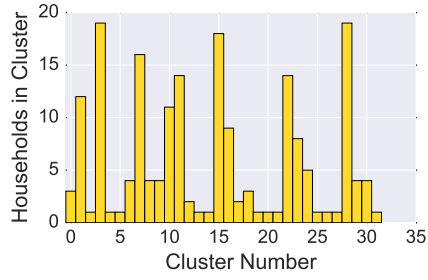
---



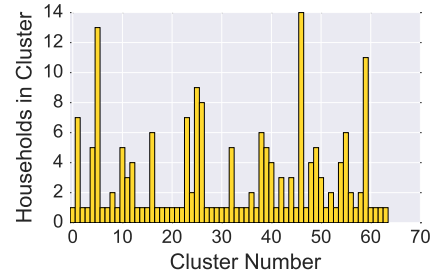
(e) Distribution of households among  $K = 8$  clusters.



(f) Distribution of households among  $K = 16$  clusters.



(g) Distribution of households among  $K = 32$  clusters.



(h) Distribution of households among  $K = 64$  clusters.

## Appendix D

# Hyperparameter Search for Individual Households

$\gamma$	1	10	100	1000
$2^0$	36.54	37.15	39.74	45.75
$2^{-1}$	36.80	36.44	37.46	39.77
$2^{-3}$	37.62	36.92	36.47	36.60
$2^{-5}$	38.81	38.03	37.19	36.92
$2^{-7}$	39.86	38.97	38.27	37.65

(a)

$\gamma$	1	10	100	1000
$2^0$	1	2	12	101
$2^{-1}$	1	2	8	59
$2^{-3}$	1	1	4	18
$2^{-5}$	1	1	2	7
$2^{-7}$	1	1	1	3

(b)

$n_{trees}$	1	2	3	4	5	6	7	8
50	34.49	34.35	34.29	34.63	34.86	34.87	34.92	35.08
100	34.63	34.03	34.37	34.38	34.63	34.60	34.65	34.85
200	34.42	34.00	34.11	34.23	34.47	34.61	34.63	34.67
300	34.26	33.92	34.11	34.28	34.34	34.53	34.55	34.68

(c)

Figure D.1: Household ID 946: (a) CV (%) for various SVR hyperparameter combinations  $\{C, \gamma\}$  with 5-fold cross-validation on the training data. (b) Computation time in seconds for various SVR hyperparameter combinations  $\{C, \gamma\}$  with 5-fold cross-validation on the training data. (c) CV (%) for various Random Forest hyperparameter combinations  $\{n_{trees}, n_{features}\}$  with 5-fold cross-validation on the training data.

## APPENDIX D. HYPERPARAMETER SEARCH FOR INDIVIDUAL HOUSEHOLDS

---

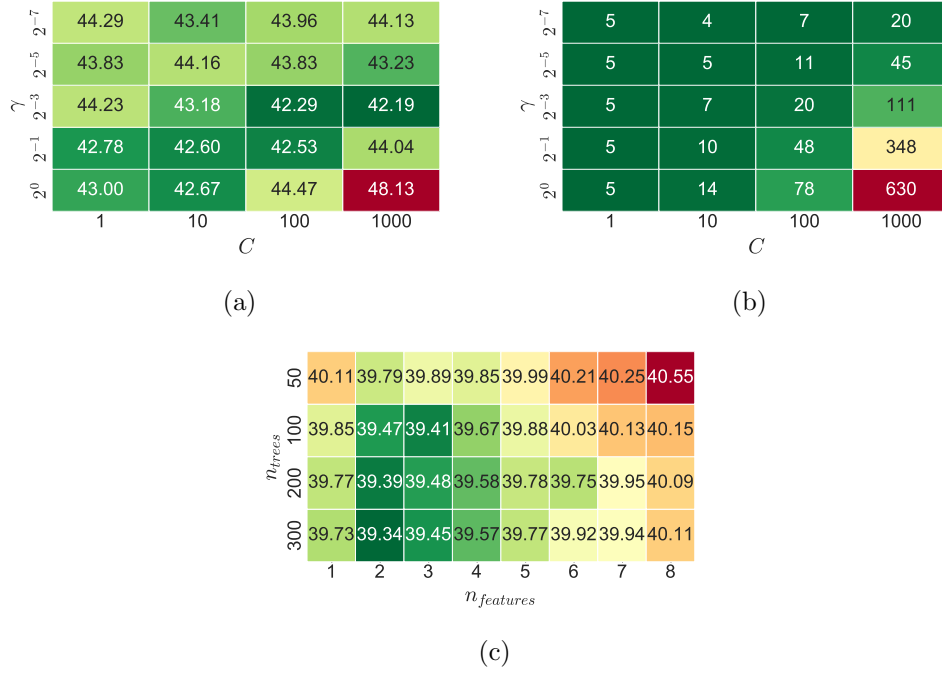


Figure D.2: Household ID 1718: (a) CV (%) for various SVR hyperparameter combinations  $\{C, \gamma\}$  with 5-fold cross-validation on the training data. (b) Computation time in seconds for various SVR hyperparameter combinations  $\{C, \gamma\}$  with 5-fold cross-validation on the training data. (c) CV (%) for various Random Forest hyperparameter combinations  $\{n_{trees}, n_{features}\}$  with 5-fold cross-validation on the training data.

$\gamma$	$2^0$	$2^{-1}$	$2^{-3}$	$2^{-5}$	$2^{-7}$
$C$	1	10	100	1000	
	36.84	35.62	35.05	34.26	
	35.70	34.88	34.06	33.92	
	34.68	34.56	34.43	34.95	
	35.69	35.19	36.03	37.42	
	36.01	36.03	37.57	42.12	

(a)

$\gamma$	$2^0$	$2^{-1}$	$2^{-3}$	$2^{-5}$	$2^{-7}$
$C$	1	10	100	1000	
	3	4	6	15	
	3	4	8	32	
	4	5	16	80	
	4	7	32	210	
	4	9	42	275	

(b)

$n_{trees}$	50	100	200	300					
$n_{features}$	1	2	3	4	5	6	7	8	
	30.46	29.71	29.61	29.60	29.63	29.69	29.76	30.07	
	30.28	29.43	29.40	29.24	29.40	29.52	29.70	29.76	
	30.27	29.44	29.24	29.19	29.33	29.38	29.54	29.67	
	30.27	29.36	29.23	29.29	29.27	29.44	29.44	29.56	

(c)

Figure D.3: Household ID 744: (a) CV (%) for various SVR hyperparameter combinations  $\{C, \gamma\}$  with 5-fold cross-validation on the training data. (b) Computation time in seconds for various SVR hyperparameter combinations  $\{C, \gamma\}$  with 5-fold cross-validation on the training data. (c) CV (%) for various Random Forest hyperparameter combinations  $\{n_{trees}, n_{features}\}$  with 5-fold cross-validation on the training data.

# APPENDIX D. HYPERPARAMETER SEARCH FOR INDIVIDUAL HOUSEHOLDS

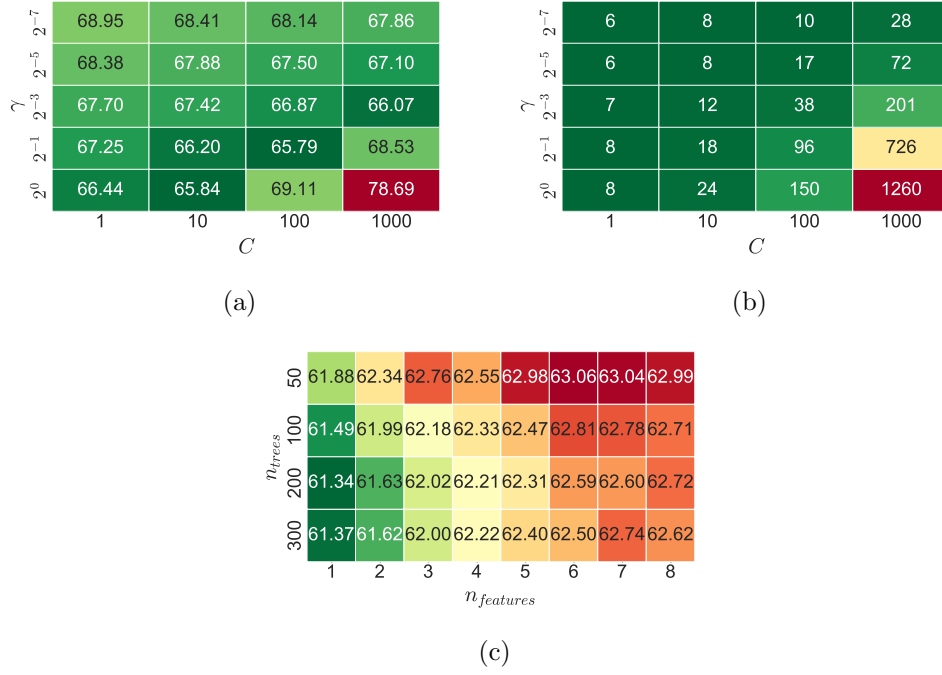


Figure D.4: Household ID 9921: (a) CV (%) for various SVR hyperparameter combinations  $\{C, \gamma\}$  with 5-fold cross-validation on the training data. (b) Computation time in seconds for various SVR hyperparameter combinations  $\{C, \gamma\}$  with 5-fold cross-validation on the training data. (c) CV (%) for various Random Forest hyperparameter combinations  $\{n_{trees}, n_{features}\}$  with 5-fold cross-validation on the training data.



TRITA -MAT-E 2016:20  
ISRN -KTH/MAT/E--16/20-SE