Probabilistic Method for forecasting of electric load profiles

Probabilistische Methoden zur elektrischen Lastprognose
Master-Thesis von Andreas Wieland aus Mainz
Tag der Einreichung:

1. Gutachten: Prof. Dr. Gerhard Neumann
2. Gutachten: Prof. Dr. Jan Peters

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Department Computer Science
Institut Autonomous Systems Labs

Probabilistic Method for forecasting of electric load profiles
Probabilistische Methoden zur elektrischen Lastprognose

Vorgelegte Master-Thesis von Andreas Wieland aus Mainz

1. Gutachten: Prof. Dr. Gerhard Neumann
2. Gutachten: Prof. Dr. Jan Peters

Tag der Einreichung:

Erklärung zur Master-Thesis

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 2. November 2015

(Andreas Wieland)

Contents

Zusammenfassung

In Zuge der Energiewende gewinnen Intelligente Energiemanagementsysteme (EMS) zunehmend an Bedeutung. Diese koordinieren bereits heute den Einsatz von flexiblen einsetzbaren lokalen elektrischen Erzeugern und Verbrauchern wie Blockheitzkraftwerken, Brennstoffzellen und Wärmepumpen mit der volatilen Wind- und PV- Erzeugung. In derartigen Systemen spielen prognosegestützte vorausschauende Algorithmen eine große Rolle, weswegen die möglichst genaue Vorhersage der nicht steuerbaren Energieerzeugung bzw. des -verbrauchs eine große Herausforderung darstellt.

Während die Vorhersage der Photovoltaik- und der Windkrafterzeugung bereits mit verschiedensten Methoden aus der Statistik und dem Feld des "Machine Learning" erforscht wurden, ist die Vorhersage von elektrischen Lastprofilen noch nicht ausgereift.

Heute wird die Lastprognose oft in Form von Persistenzmodellen realisiert, in welchen die gleichen Daten eines vergangenes Tages für die Vorhersage eines neuen Tages benutzt wird. Ein Problem dieser "Similar Day"-Methode ist, dass auf von der alten Aufzeichnung abweichende Dynamiken nicht reagiert werden kann. Aus diesem Grund wurden einige andere Anstätze wie neuronale Netzwerke, Support Vector Maschines oder autoregressive Modelle erprobt.

In dieser Arbeit wird eine neue Methode mit dem Namen "Probabilistic Movement Primitives" für die Vorhersage von "short-term" Lastprofilen (STLF) vorgeschlagen, was gemeinhin einen Zeitraum zwischen ein paar Stunden bis zu einem Tag umfassen kann. Dieser neue Ansatz stammt ursprünglich aus der Robotik und wird dort unter anderem zur Berechnung von Regelparametern benutzt. Diese neue Methode beruht auf gewichteten Basisfunktionen, aus welchen hierarchische Wahrscheinlichkeitsmodelle erstellt werden. Auf Basis dieser Modelle kann in Anbetracht des bisherigen Tagesverlaufs die wahrscheinlichste Prognose berechnet werden.

Ein wichtiger Aspekt dieser neuen Methode ist die Möglichkeit mit ihr, eine Aussage über die Varianz der Prognose zu treffen. Dies gibt dem Benutzer ein Maß für die Genauigkeit der Vorhersage. Zudem kann die Prognose mit geringem Rechenaufwand mit aktuelleren Daten zu einem beliebigen Zeitpunkt aktualisiert werden. Dies führt zu einer graduellen Verbesserung der der Vorhersagegenauigkeit.

Abstract

In the transformation process of the carbon hydrate based energy system to a more sustainable state relying predominantly on renewable energies, intelligent energy management systems (EMS) become increasingly more important. These systems already coordinate flexible generators, consumers and the volatile production of wind and solar power. Forecast-based anticipatory optimization algorithms play an important part in modern EMS. These often rely on an accurate forecast of the non-controllable consumers and generators.

While the forecast for solar and wind production has been explored with different approaches from statistical to machine learning methods, the forecast of electric load profiles is still in the fledging stages.

To date, in most cases the load forecasting relies on persistence models. One key problem of this approach is that it cannot cope with dynamics deviating from the historical records. For this reason different approaches have been explored (e.g., neuronal networks, support vector machines, and auto regressive models).

This thesis will apply a new method called "Probabilistic Movement Primitives" (ProMP) to the forecast of "short-term" load profiles (STLF). The ProMP approach has been developed in robotics and is used for the derivation of control parameters. It employs weighted Gaussian basis functions for representing the historic data and creates a hierarchic distribution model on top of them. With input data from the very recent past, a forecast for the remainder of the current day can be calculated.

One important aspect of this new method is the possibility to calculate the variance of the forecast, which provides a measure of accuracy. Furthermore new forecast based on updated records can be calculated with very little computational overhead, this leads to a gradual improvement of the accuracy of the forecast.

1 Introduction

The International Energy Agency estimates that renewable energy generators are going to surpass coal as the top source of energy by 2035 [1]. However, introducing volatile generators like solar plants or wind turbines to the grid increases the need for intelligent energy management systems (EMS) to coordinate consumer and producer load. Coordination is necessary because energy from wind or solar farms can not be dispatched in the classical sense [2] and has to be synchronized with classical energy producers like coal plants to avoid energy surplus. Additionally, if the producers are not able to produce enough power in time, high fees have to be paid [3]. Consequentially, new technologies have been investigated to tackle the problems. Examples for intelligent EMS are cyber-physical energy systems [4] and model predictive control [5]. Both of these technologies need accurate forecasts of electric load profiles to ensure optimal usage and safety of the energy grid.

In the last decade, a lot of research has addressed the problem of forecasting solar and wind energy generation. The cause of this development is the increased attractiveness of these energy generation methods [6]. Various different approaches have been proposed which can be categorized into the following categories: physical, statistical, machine learning, and hybrid approaches [7]. Physical approaches try to describe the forecast by applying a physical model. An example is the forecast of solar energy output by measuring cloud movements [8]. Another example is the forecast of solar energy by calculating a radiance model [9]. Statistical approaches base their forecast solely on past data. *Zhang et al.* created a "similar day"-model by using data from three different locations around the world [10]. Approaches based on machine learning algorithms have been rising in popularity lately. Examples for these algorithms are artificial neural networks [11, 12], support vector machines [13] or Gaussian processes [14, 15]. Furthermore, a lot of research has been dedicated to hybrid methods [16, 17] which combine the formerly mentioned approaches to increase accuracy by increasing complexity. However, these methods are prone to overfitting and may not yield optimal results [18, 19, 20].

The research of forecasting consumer loads can be dated back to 1966 [21, 22]. Before the liberalization of the energy market, the most prominent use case for energy load forecasting has been the determination of the optimal and secure schedule for energy production [22]. After 1982, the first energy market liberalization took place in Chile, hence, a new use case emerged: in a non-monopolistic market situation the estimation of the electricity prize had become interesting [23].

Another use case for electric load forecasting, lies in demand-sight management systems where the forecast is used for short-term decision making [24] by system operators for the optimal utilization of the power generators and power stations [25]. This especially important in developing countries which try to use their limited electricity more efficiently [26]. Demand sight management systems often use short-term load forecasts (STLF) which have a time range from several hours to a day, other categories for forecasting electric consumer loads are very-short-term, mid-term and long-term [27]. While very-

short-term forecasting is relatively new, it only takes a look at a minute to minute time frame [28]. For years mid-term forecast has been used for scheduling maintenance and system improvements of the grid [29] and has a time frame of up to a month. If the time interval exceeds months, the forecast is called long-term, these forecasts are valuable for system operators to estimate the need for expansion [30, 31], especially in fast developing countries like China [32]. Mid-term and long-term forecasts suffer from the accumulation of forecast errors over time [33], hence they normally only forecast peak loads. Other methods in most publications forecast load curves or electric load profiles. Electric load profiles are time series which means that they are sequences of random numbers based on consecutive measurements taken at equal time intervals [23]. An overview of all methods and their time range can be seen in table 2.

	very-short-term	short-term	mid-term	long-term
time horizont	5min - 1h	1h - 24h	24h - weeks	months - years
forecasted fomat	load profiles	load profiles	peak loads	peak loads

Table 2: An overview of the different types of electric load forecasting, their time range and forecast format

In this work a new approach for short-term forecasting of electric load profiles will be proposed. It is based on *Probabilistic Movement Primitives* by *Paraschos et al.* [34]. This approach uses probability density functions to represent the data. Electric load profiles are described as a set of weighted Gaussian basis functions and then turned into a Gaussian probability function. By inducing a Gaussian prior distribution to the probability function, a small hierarchical Bayesian network is constructed which can be used to calculate forecasts for electric load profiles. Since this new approach is based on Gaussian probability functions, methods of probability theory can be used. Therefore, it is possible to increase the accuracy of the forecast over the course of a day by conditioning on data, which only recently and partly has become available. Another advantage is that a covariance of the forecast can be calculated which can be used as a measurement for the forecast accuracy.

This thesis is structured as follows: in section 2 an overview over the recent research and the related methods will be given. The novel approach for forecasting electric load profiles will be discussed in section 3. In section 4, the evaluation of the novel approach will be presented, followed by an outlook in section 5.

2 Related Work

In general, the forecast of electric load profiles can be classified as a regression problem. Regression problems are common problems in computer science where a system is supplied with input data and returns output data. The task is to look for the relationship between input and output data. This process is called regression analysis. In case of electric load profiles there is an additional difficulty because the data is highly nonlinear [35] and they incorporate a seasonal component which influences the load over the day. This seasonal component can be partly contributed to heating required on colder days or need for ventilation on warmer days. That is why it can be separated into summertime, wintertime and two transition periods in between [23].

Many different approaches have been applied to electric load forecasting of uncontrollable consumer loads. The simplest approach is, to use a static normalized load profile to estimate the consume of households, which is still common practice in Germany [36], although it is prone to uncertainties from

geographical differences [37]. Similar to electric load forecast for producer loads, the forecast for consumer loads can also be divided into the categories: statistical approach, machine learning, and hybrid approaches. The simplest statistical approach is the usage of a *similar-day* model [23] which is used in this thesis as a baseline method. Further information can be found in section 2.1. Another type of statistical approach used for load forecasting are autoregressive models where the current load is a linear aggregate of previous values [38]. Electric load profiles are nonlinear, therefore, a lot of research went into the use of artificial neural networks for forecasting [39, 40] because their capabilities in modeling nonlinear data is great. However, the configuration and the choice of input variables are difficult [41] and they come with a high computational cost for the training phase [42]. This forces the user to limit the input data which results in the forecast becoming less accurate [43].

In contrast to the neural network approach of minimizing the training error, support vector machines try to minimize the upper bound of the generalization error [44]. If the input data can be represented by scalar products, the *kernel trick*, proposed by *Vapnik et al.* [45] can be used. The *kernel trick* replaces the scalar products with a kernel function $k(x, x')$ [46], which can represent in different forms and allows support vector machines to be used with nonlinear data [47]. As a result, many people tried to use them for the forecast of electric load profiles [33, 48, 49]. One of the problems of support vector machines is that the determination of their hyper parameters is difficult [50].

More recently, relevance vector machines (RVM), also known as sparse Bayesian learning [51], have been used for forecasting electric load profiles [52, 53]. These models have a number of benefits compared to support vector machines: their forecasts are probabilistic and they use fewer kernel functions which leads to a fewer number of hyper parameters [54] and faster performance [47]. Probabilistic outputs are preferred because this way they inherit measurement for certainty [42]. However, unlike support vector machines, which have to solve a convex optimization problem, the inherent optimization problem of RVMs is not convex and, as a result, harder to solve [47].

Other, less popular methods are, for example, weighted regression models [55], fuzzy methods which are more like a classification approach [26], kernel regression [56] which are similar to fuzzy methods [57]. There are also hybrid methods involving fuzzy predictor for Hidden Markov models [58, 59] or neural networks as predictor in conjunction with Hidden Markov Models. The reason why a predictor is often used is that the forecast with Hidden Markov models is not straight forward because the forecast would be done by interpolating neighboring states [60]. However, the error-correction of Markov models is superior to other methods [61] which is why they have been used for the automatic generation of general load profiles [62] and the long-term forecast of electric load profiles [63]. Furthermore, there have been alternative attempts including multi-region load forecasts using an ensemble of different methods [64] and approaches which split the day into hours, to forecast the individual hours independently of each other [65]. A similar approach to the later [66], has shown superiority in a competition at *Puget Sound Power and Light Company* in a controlled environment [67].

For the remainder of this chapter, the most relevant methods are explained in more detail. In section 3, the novel approach with *Probabilistic Movement Primitives* will be introduced.

2.1 Persistence Models

A persistence model is the simplest statistical model possible [23]. The data which has been recorded in the past is used again for the forecast. The idea behind such models is that there is a correlation between time slots and, as a result, the difference between time slots should be small.

The major benefits of this approach are that the implementation is fairly simple and that these methods only depend on the chosen time intervals between recorded data and the expected recurrence for the forecast. If the data has a circular correlation between time intervals, these methods can become quite accurate and can be used as a benchmark for more sophisticated approaches [23].

After analyzing the industrial data set used in this thesis (further described in section 4.1), it has become obvious that the same weekdays in a season have similar profiles. Furthermore, profiles of weekdays are similar to each other but the profiles of weekdays are significantly different to days of the weekend. For example, load profiles of week days have a low load at the start of the day, increasing over the day and decreasing at the end of the day. The peak loads vary but generally are between 07:00 am and 12:00 am, this is shown in fig. 3. Contrary to that are weekends and national holidays where the load stays mainly the same over the day. This observation can be explained by the universal working hours as most people work during the week at the middle of the day until evening [68]. This behavior can also be seen in fig. 2 which shows the synthetic standard load profiles for average industrial load profiles $G0$ for Germany by the *VDEW (BDEW since 2007)* [69, 70].

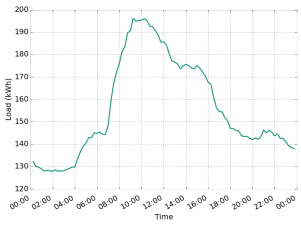

Figure 1: Mean load profile for all weekdays of all seasons of the industrial data sets

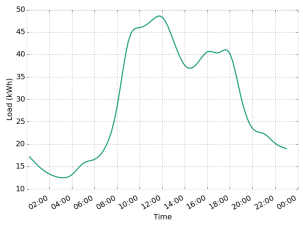

Figure 2: Mean industrial load profile $G0$ for all weekdays of all seasons

Contrary to industrial electric load profiles have electric load profiles of household their peak at the end of the day, which is typical the time after work [68]. This can also be seen in the standard load profile for average household load profiles $H0$ for Germany by *VDEW* [70] in fig. 4. The household data highly depends on the behavior and number of the inhabitants, which will be further highlighted in section 4.1.2.

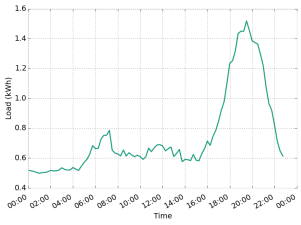

Figure 3: Mean load profile for all weekdays of all seasons of the household data set with two kids and two two workers

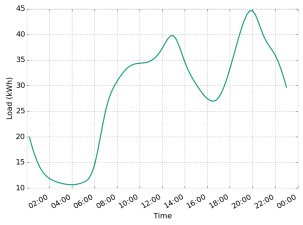

Figure 4: Mean household load profile $H0$ for all weekdays of all seasons

2.1.1 Advanced Persistence Model

The forecast of the persistence model can be improved by using the mean for multiple past week days. This is explained by the notion that the mean of the data will significantly reduce the influence of outliers and focus on the core pattern behind the data [71]. Mathematically, this model can be described as

$$\boldsymbol{y}_d^* = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{y}_{di} \tag{1}$$

where $\boldsymbol{y}_d^*$ represents the forecast, $\boldsymbol{y}_{di}$ are the past days with d being the day of the week and M being the number of past weeks.

2.2 Artificial Neural Networks

The human brain consists of millions and millions of interconnected neurons with the ability to communicate over these connections and create new connections [72], they are able to learn and excel at difficult tasks like pattern recognition. In computational science, neural networks are imitated by artificial neural networks (ANN). ANNs use layered simplified neurons called perceptrons to recreate the superior abilities in pattern recognition and other tasks.

Like their biological counterpart, ANNs consist of one or more layers of a multitude of perceptrons [42]. Each perceptrons consists of a nonlinear activation function $f\{.\}$ and a nonlinear basis function ϕ, with the output of the activation function connected to the next layer of perceptrons. Consequentially, each basis function is in "*itself a nonlinear function of a linear combination of the inputs, where the coefficients in the linear combination are adaptive parameters*" [47, page 227]. The coefficients are the weights w for each connection which influcence how much the output of a perceptron depends on each incoming connection, resulting in

$$y(\boldsymbol{x},\boldsymbol{w}) = f\left\{\sum_{i}^{M} w_i\phi_i(\boldsymbol{x})\right\}. \tag{2}$$

Here, $\phi_i(x)$ is the nonlinear basis function for input i and M is the number of inputs for this layer. For each perceptron, the input can be constructed as follows for D input variables:

$$a_j = \sum_{i=1}^{D} w_{ji}x_i + w_{j0}. \tag{3}$$

Activations a which are transformed by the activation function to the output $z_j = f(a_j)$ of the next layer [47]. The most common activation functions are the sigmoid or the hyperbolic tangent function [47]:

$$\mathrm{s}(x) = \frac{1}{1+e^{-x}} \qquad\qquad \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{4}$$

For the optimization of the weights, the ANN is trained on a set of N input vectors $\boldsymbol{x}_n$ with $n \in [1,N]$ and their corresponding *targets* $\boldsymbol{t}_n$ by minimizing the error function $J(w)$[47],

$$J(\boldsymbol{w}) = \frac{1}{2}\sum_{n=1}^{N} \|\boldsymbol{y}(\boldsymbol{x}_n, \boldsymbol{w}) - \boldsymbol{t}_n\|^2 . \tag{5}$$

The training phase for each ANN consists of two phases. The first part is the evaluation of the derivative of the error function with respect to the weights and the second part is the adjustment of the weights. The original algorithm from *Rumelhart et al.* is called *Error Backpropagation* [73] and uses gradient descent for the second stage. However, many other approaches are possible too[47].

The great advantage of neural networks lies in the diversity of functions which can be used as activation and basis function for each perceptron and their hidden layers. The hidden layers are the layers of perceptrons in between the first (input) layer and the last (output) layer. ANNs with hidden layers and no direct acyclic connections are called feedforward neural networks or multilayer perceptrons (MLP). An example for a small ANN with one hidden layer is sketched in fig. 5. It can be shown that an MLP can approximate any function with any desired level of accuracy [42].

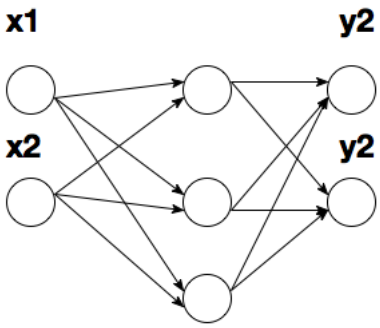

Figure 5: Example for a small ANN with one hidden layer which has three perceptrons. The input and output layer only consist of two perceptrons each.

In this thesis the *Neuronal Network Toolbox* for *MATLAB* from *MathWorks* has been used [74].

2.2.1 Error Backpropagation

In the following section the error backpropagation algorithm is described, taken from *Murphy* [42]. First the weights are Initialized randomly. After that all the activations a_n and the corresponding results of activation function $z = f(a_n)$ have to be calculated for all layers. The calculation is done in respect to the error function, aNother way to express the error function from eq. (5) is as follows:

$$J(\theta) = -\sum_{n}^{N}\sum_{k}^{K}\left(y^*_{nk} - y_{nk}\right)^2, \tag{6}$$

where N is the number of training data inputs, K the number of outputs and θ the weight matrices for the different layers[42]. Going backward through the network, the error $\boldsymbol{\delta}_n = \boldsymbol{y}^*_n - \boldsymbol{y}_n$ is calculated first

for the output layer and consequential for the hidden layers. The error has to be passed "down" from one layer to the next, hence the name of the algorithm. Afterwards the gradient can be calculate with,

$$\nabla_\theta J(\theta) = \sum_n \left[\boldsymbol{\delta}_n^v \boldsymbol{x}_n, \boldsymbol{\delta}_n^w \boldsymbol{z}_n \right] \tag{7}$$

where $\boldsymbol{\delta}_n^v$ is the error from the first layer, $\boldsymbol{\delta}_n^w$ is the error of the last layer and $\boldsymbol{z}_n$ is the hidden layer. There are other ways to calculate the derivative and different possible error functions [47].

The simplest way to update the weights for each layer is by using gradient descent [47] with,

$$\boldsymbol{w}^{(\tau+1)} = \boldsymbol{w}^\tau - \eta \nabla_\theta J(\theta) \tag{8}$$

with the *learning rate* $\eta > 0$ and τ being the number of the current layer.

2.2.2 Data Representation

The data representation in the ANN model for the training data and test data is similar to representation used for the Gaussian Process, which can be seen in section 2.3.2. This way of representing the data is different to the typical way of taking time as input and the load as output. Instead of using the time as an input vector, the loads have been chosen as the input and the output are the loads shifted by a certain distance d. The result is that the inputs load for a certain time y_t have a unknown relationship to the loads from time y_{t+d}. Additionally a history for each load is added so that each load for time y_t has a relation to the prior loads y_{t-i} where $i \in [1, H]$ and H defines the size of the history. The resulting correlation of a dataset with size $T + d$ can be described as:

$$\begin{gathered} \boldsymbol{X} \sim \boldsymbol{y} \\ \begin{pmatrix} y_{t-H} & y_{t-H+1} & \cdots & y_t \\ y_{t-H+1} & y_{t-H+2} & \cdots & y_{t+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{T-H} & y_{T-H+1} & \cdots & y_T \end{pmatrix} \sim \begin{pmatrix} y_{t+d} \\ y_{t+d+1} \\ \vdots \\ y_{T+d} \end{pmatrix}. \end{gathered} \tag{9}$$

The reason why this representation was chosen is that the electric energy used to a later date directly depends on the load which is currently used, sketched in fig. 6. The forecast can be done by using a $\boldsymbol{X}$ with the $T = d - 1$ where $\boldsymbol{y}$ is an unknown electric load profile.

2.3 Gaussian Processes

Gaussian Process (GP) is a state of the art model for regression. First introduced by *Matheron* [75, 76] as *kriging* based on the thesis by *Danie G. Krige* [77], it has later been summarized and reinvented by *Carl Edward Rasmussen and Chris Williams* for machine learning [46]. In this thesis, the toolbox "GPStuff", provided by *Jarno Vanhatalo and Jaakko Riihimäki* [78], has been used.

The foundation of a GP is the inference over functions $p(f|\boldsymbol{X}, \boldsymbol{y})$, where f is an unknown function with the relationship $y_i = f(x_i)$. Most algorithms try to simplify it by inferring over a set of parameters $p(\theta|\boldsymbol{X}, \boldsymbol{y})$. The way Gaussian Processes work can be seen as a more Bayesian alternative to kernel methods [42]. The forecast $\boldsymbol{y}_*$ for new inputs $\boldsymbol{x}_*$ are computed by:

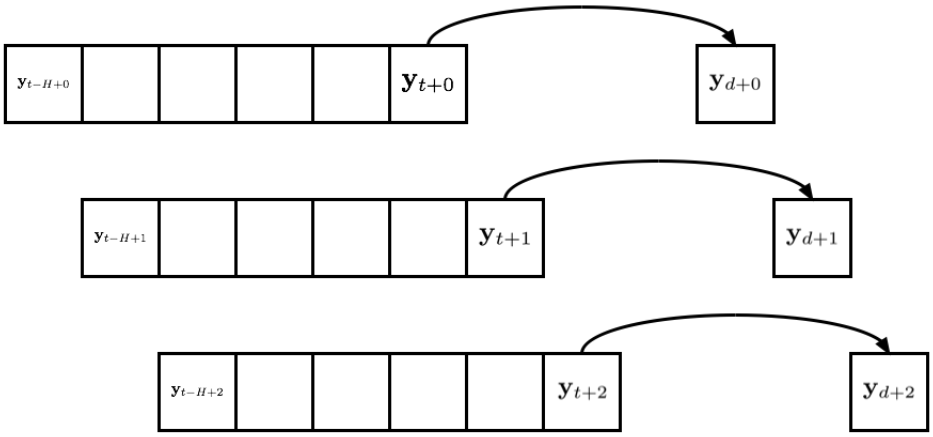

Figure 6: The data representation sketched with a forecast window of three and a history size of five.

$$p(\boldsymbol{y}_*|\boldsymbol{x}_*,\boldsymbol{X},\boldsymbol{y}) = \int p(\boldsymbol{y}_*|\boldsymbol{f},\boldsymbol{x}_*)p(f|\boldsymbol{X},\boldsymbol{y})\mathrm{d}\boldsymbol{f}. \tag{10}$$

The unknown function is represented by a GP which has been defined as "*a collection of random variables, of any finite number of which have a joint Gaussian distribution*" [46, page 13]. As such they are completely defined by a mean function $m(\boldsymbol{x})$ and a covariance function $K(\boldsymbol{x},\boldsymbol{x}')$:

$$\boldsymbol{f}(\boldsymbol{x}) \sim \mathrm{GP}\left(m(\boldsymbol{x}), K(\boldsymbol{x},\boldsymbol{x}')\right) \tag{11}$$
$$m(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x})] \tag{12}$$
$$K(\boldsymbol{x},\boldsymbol{x}') = \mathbb{E}\left[(f(\boldsymbol{x})-m(\boldsymbol{x}))(f(\boldsymbol{x}')-m(\boldsymbol{x}'))^T\right]. \tag{13}$$

The mean function is set to $m(\boldsymbol{x}) = 0$, since GPs are flexible enough to model the mean arbitrarily [42]. For the covariance function the squared exponential function, also known as Gaussian kernel or RBF kernel [42], will be used:

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j) = \sigma^2_{\mathrm{sexp}} \exp\left(-\frac{1}{2}\sum_{k=1}^{d}\frac{\left(x_{i,k}-x_{j,k}\right)}{l_k^2}\right). \tag{14}$$

where l_k is the length scale and defines the correlation between the points in the k dimension, σ^2 is the magnitude and defines the variability of the GPs Covariance function [79]. To forecast the output $\boldsymbol{f}_*$ the joint distribution over the training data $D = \{(\boldsymbol{x}_i,\boldsymbol{y}_i), i \in [1,N]\}$ and the test data $\boldsymbol{X}_*$ have to be calculated, as follows:

$$\begin{pmatrix}\boldsymbol{f}\\ \boldsymbol{f}_*\end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix}\boldsymbol{K} & \boldsymbol{K}_*\\ \boldsymbol{K}_*^T & \boldsymbol{K}_{**}\end{pmatrix}\right) \tag{15}$$

where $\boldsymbol{K} = K(\boldsymbol{x},\boldsymbol{x})$ has the size NxN, $\boldsymbol{K}_* = K(\boldsymbol{x},\boldsymbol{x}_*)$ has the size NxN_*, and $\boldsymbol{K}_{**} = K(\boldsymbol{x}_*,\boldsymbol{x}_*)$ has the size N_*xN_*. The posterior can be calculated by conditioning,

$$p(\boldsymbol{f}_*|\boldsymbol{x}_*, \boldsymbol{x}, \boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}_*|0, \boldsymbol{\Sigma}_*) \tag{16}$$

$$\boldsymbol{\mu}_* = \boldsymbol{K}_*^T \boldsymbol{K}^{-1} \boldsymbol{y} \tag{17}$$

$$\boldsymbol{\Sigma}_* = \boldsymbol{K}_{**} - \boldsymbol{K}_*^T \boldsymbol{K}^{-1} \boldsymbol{K}_*. \tag{18}$$

2.3.1 Sparse Gaussian Processes

Since the calculation of $\boldsymbol{K}^{-1}$ is computational expensive and numerically unstable [42], it is advised to use Cholesky decomposition [46]. However, this would still have a computational time need of $O(N^3)$. It is obvious that time could be saved using fewer training examples which is why the sparse model *Fully independent conditional* (FIC) [80] has been developed. Sparse models for Gaussian Processes work by introducing *inducing variables* u_i with $i \in [1, m]$ and $m < N$. The corresponding input locations are indicated by $\boldsymbol{X}_u$. The prior over the functions from eq. (10) changes to the following inducing conditional:

$$p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{y}) \approx q(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{X}_u, \boldsymbol{y}) = \int q(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{X}_u, \boldsymbol{u}, \boldsymbol{y}) p(\boldsymbol{u}|\boldsymbol{X}_u, \boldsymbol{y}) \mathrm{du} \tag{19}$$

where the latent variables are assumed to be conditionally independent, given u. This leads to

$$\begin{aligned} q(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{X}_u, \boldsymbol{u}, \boldsymbol{y}) &= \prod_i^m q_i(\boldsymbol{f}_i|\boldsymbol{X}, \boldsymbol{X}_u, \boldsymbol{u}, \boldsymbol{y}) \\ &= \mathcal{N}(\boldsymbol{f}|\boldsymbol{K}_{f,u}\boldsymbol{K}_{u,u}^{-1}u, \mathrm{diag}(\boldsymbol{K}_{f,f} - \boldsymbol{K}_{f,u}\boldsymbol{K}_{u,u}^{-1}\boldsymbol{K}_{u,f})) \\ &= \mathcal{N}(\boldsymbol{f}|\boldsymbol{K}_{f,u}\boldsymbol{K}_{u,u}^{-1}u, \boldsymbol{\Lambda})). \end{aligned} \tag{20}$$

By inducing a zero-mean Gaussian prior over $\boldsymbol{u}$ with $p(\boldsymbol{u}|\boldsymbol{X}_u, \boldsymbol{y}) \sim \mathcal{N}(0, \boldsymbol{K}_{u,u})$, the approximation from eq. (19) simplifies to:

$$q(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{X}_u, \boldsymbol{y}) = \mathcal{N}(\boldsymbol{f}|0, \boldsymbol{K}_{f,u}\boldsymbol{K}_{u,u}^{-1}\boldsymbol{K}_{u,f} + \boldsymbol{\Lambda}) \tag{21}$$

This can be calculated by using Sherman-Morrison-Woodbury formula [42]. The matrix multiplication will take most of the time which results in a computational time of $O(m^2 N)$ [79].

2.3.2 Electric Load Forecast with Gaussian Processes

The data representation for the Gaussian Process is the same one which has been used for the artificial neural network, from section 2.2.2. With each load y_t having a presumed relationship to the future load y_{t+d}, where d describes the forecast horizon and each load having a relationship to the historic data $y_{t-i}|i \in [1, H]$, which is sketched in eq. (9). The Gaussian Process will learn the correlation between loads of different times from the training data. The forecast can be done by inputing a vector,

$$\boldsymbol{x}_{t-d}^* = \begin{pmatrix} y_{t+H-d} & y_{t+H-d+1} & \cdots & y_{t-d} \end{pmatrix} \tag{22}$$

into eq. (10) with the learned prior distribution by eq. (21) and a likelihood function. There are plenty possible likelihood functions, each describes the ration of the noise on the input data [46]. If a Gaussian

likelihood function $p\left(\boldsymbol{y}|\boldsymbol{f},\sigma_{\text{lik}}^2\boldsymbol{I}\right)$ is used with a Gaussian prior distribution the marginalization can be done in closed form [46] otherwise the marginalization has to be approximated. The approximation can be done with Laplace approximation or *Markov Chain Monte Carlo* [79]. The output of the GP for a single $\boldsymbol{x}_{t-d}^*$ vector is the forecast of the load y_d^*, if the input vector is expanded up to the first unknown point y_d^*,

$$\boldsymbol{x}^* = \begin{pmatrix} \boldsymbol{x}_{t-d}^* \\ \boldsymbol{x}_{t-d+1}^* \\ \vdots \\ \boldsymbol{x}_{d-1}^* \end{pmatrix} \tag{23}$$

a forecast for the vector $\boldsymbol{y}^*$ becomes possible.

2.4 Hidden Markov Models

Markov models or Markov Chains are Bayesian networks in which each probability complies with the Markovian property or Markov assumption [42]. The Markovian property is defined as follows: "*Future probabilistic behavior is independent of its history at the present state* "[81, page 433] which means that in a Markov Chain, each state only dependents on the direct neighboring state and the system can be characterized by the initial distribution over states $p(x_1 = i)$ and the transition matrix $p(x_t = j|x_{t-1} = i)$ [42]. Hidden Markov models (HMM) are an extension of Markov Chains, where the Markov Chain is defined as the hidden states $z_t \in 1,\ldots,K$. In addition an observation model $x_t \in 1,\ldots,T$ is introduced [42]. The probability for an observation $\boldsymbol{x}_t$ can be defined as $p(\boldsymbol{x}_t|z_t)$. It is important to note that, the Markovian property only holds for the hidden layer, but not for the observation layer [42]. For all observed x_t and latent variables z_t, the joint probability can be expressed by

$$p\left(\boldsymbol{z}_{1:T},\boldsymbol{x}_{1:T}\right) = p(\boldsymbol{z}_{1:T})p(\boldsymbol{x}_{1:T}|\boldsymbol{z}_{1:T}) = \left[p(z_1)\prod_{t=2}^{T}p(z_t|p(z_{t-1})\right]\left[\prod_{t}^{T}p(\boldsymbol{x}_t|z_t)\right] \tag{24}$$

HMMs have been first introduced in the 1960s [82] and are, for example, used for speech recognition [83], DNA sequencing [60], or for different financial problems [84]. Most of the time, their hidden layer is estimate which is assumed to represent a predetermined model [42]. The estimation is performed by inferring over the observations $p(z_t|\boldsymbol{x}_{1:t})$ for online and $p(z_t|\boldsymbol{x}_{1:T})$ for offline learning [42]. However, HMM can also be used for the forecast of time series by forecasting $p(z_{t+h}|\boldsymbol{x}_{1:t})$ for $h > 0$ [85]. Another way to use HMMs for forecasting is by computing the class-conditional densities and feeding them into various classifiers [58] or by using the class-conditional in conjunction with Bayesian weight regression [86].

In many cases, the HMM is trained by using the Baum-Welch algorithm. *Jianhua et al.* for example, used it to forecast mid-term electricity prices with historic prices and electric loads as input [87]. The Baum-Welch algorithm is similar to the Expectation Maximization algorithm [42] which will be described in section 3.1.3.

If the hidden states are continuous, an HMM becomes a state space model (SSM) and if all probabilities are Gaussian, the SSM becomes a linear-Gaussian state space model (LG-SSM), also known as linear dynamical system (LDS)[42]. The Inference over such an LDS is done by using the Kalman filter algorithm which calculates the marginal posterior distribution $p(\boldsymbol{z}_t|\boldsymbol{y}_{1:t},\boldsymbol{0:t}) = \mathbb{N}(\boldsymbol{z}_t|\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$ in closed form [42]. This algorithm can be applied to forecast time series [42]. As a result, has been used for

"short-term" forecasting electric load profiles [88] and "very-short-term" load forecasting [89]. Other hybrid approaches are Kalman filters in conjunction with a Fuzzy predictor to forecast electric load profiles [90]. Another way to use the Kalman filter algorithm is, for training an ANN to forecast electric load profiles [91]. The great advantages of HMMs are that they are built upon a strong statistical foundation and are computationally efficient [60].

3 Probabilistic Movement Primitives for Electric Load Forecasting

In this section, a new approach to electric load forecasting will be described. *Probabilistic Movement Primitives* (ProMP) is a newly suggested method for Robot Control and a similar approach to *Dynamic Movement Primitives* (DMP) [92]. Here the idea is to show the robot how to do a certain task, and the robot derives its control laws from the observation. However, contrary to DMP, which uses second-order dynamical systems, ProMP uses a probabilistic approach [93]. In the following chapter, the new algorithm will be explained and it will be explained how apply it to electric load forecasting.

3.1 Probabilistic Trajectory Representation

ProMP uses a hierarchical Bayesian network or multi-level model to describe the data [42]. In the first step, the training data has to be transformed into *feature space* where each profile is represented in terms of fit parameters. This means applying a fixed nonlinear mapping on the training data, which maps the high dimensional data on the lower dimensional *feature space* [47]. The mapping will be realized in terms of using Gaussian basis functions

$$\phi_i(z_j) = \exp\left(-\frac{(z_j - c_i)^2}{2h}\right), \qquad \Phi_{ij} = \frac{\phi_i(z_j)}{\sum_i^N \phi_i(z_j)}, \tag{25}$$

where $z \in [0,1]$ represents the time steps of the day, c_i corresponds to the centers of the Gaussian function and h is defined as the step size or width of the basis function [34]. On the right hand side, the basis matrix, holding the normalized Gaussian basis functions, is defined. An example for the Gaussian basis functions and the resulting basis matrix is displayed in fig. 7 and fig. 8.

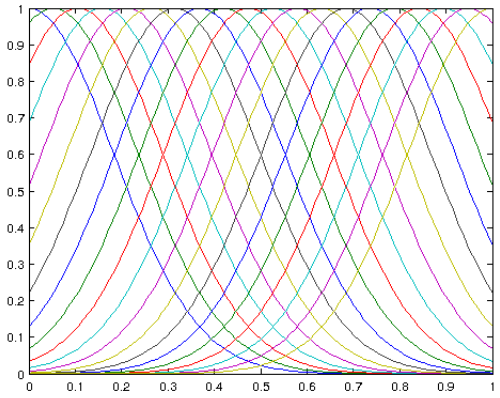

Figure 7: Gaussian basis functions

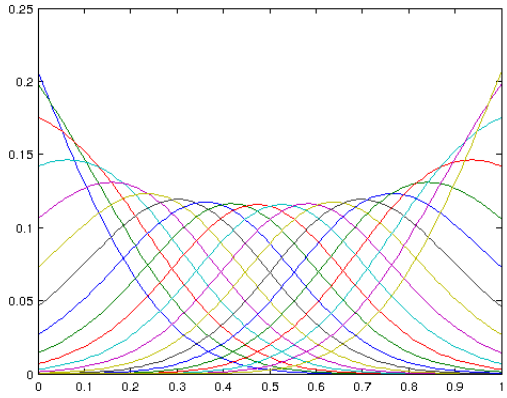

Figure 8: Normalized Gaussian basis functions

Introducing a weight vector $\boldsymbol{w}$ allows to compactly represent a load profile $\boldsymbol{y}$ of a single day as

$$\boldsymbol{y} = \boldsymbol{\Phi}^T \boldsymbol{w} + \boldsymbol{\epsilon}, \tag{26}$$

where $\epsilon \sim \mathcal{N}(0, \boldsymbol{\Sigma}_y)$ represents the Gaussian noise which is independent and identically distributed. The accuracy of the compact representation depends on the number of basis functions as shown in fig. 9.

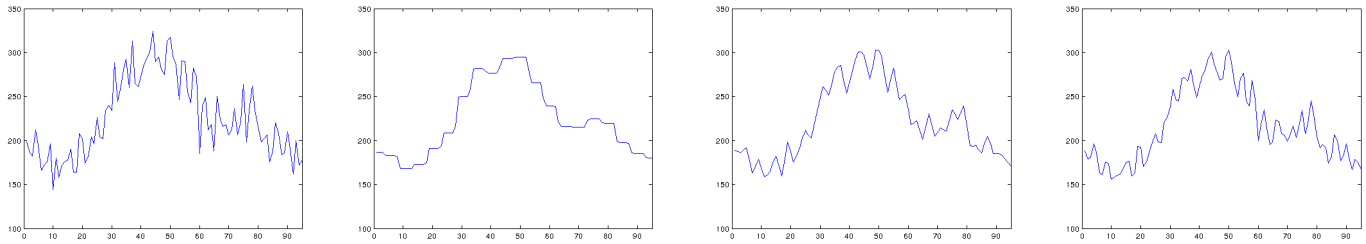

Figure 9: From left to right, a sample electric load profile and its compact representation respectively in terms of 20, 50, and 100 basis functions

Since the compact representation from eq. (26) is a linear function, it is possible to derive a general linear model [42] of the form:

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{w}) &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{\Phi}^T\boldsymbol{w}, \boldsymbol{\Sigma}_y) \\ &= \frac{1}{|2\pi\boldsymbol{\Sigma}_y|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\left(\boldsymbol{y}-\boldsymbol{\Phi}^T\boldsymbol{w}\right)^T \boldsymbol{\Sigma}_y^{-1}\left(\boldsymbol{y}-\boldsymbol{\Phi}^T\boldsymbol{w}\right)\right). \end{aligned} \tag{27}$$

Maximizing the above expression yields the most likely representation in *feature space* for the observed profile. In order to forecast a new profile, the weights have to be calculated for all observations. This results in N weight vectors, representing N single days. To formalize a distribution over the weights, the *sample mean* and the *sample covariance* for all time steps have to be calculated, as follows:

$$\mu_j = \frac{1}{N}\left(\sum_{i=1}^{N} x_{ij}\right), \qquad \Sigma_{ij} = \frac{1}{N-1}\sum_{i=1}^{N}\left(w_{ij}-\mu_j\right)\left(w_{ij}-\mu_k\right). \tag{28}$$

In the following sections we use θ to denote the tuple of the expectation vector $\boldsymbol{\mu}_w$ and the covariance matrix $\boldsymbol{\Sigma}_w$. With these parameters it is possible to formalize a distribution over the weights:

$$p(\boldsymbol{w}|\theta) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w). \tag{29}$$

This distribution makes it possible to calculate a forecast of an electric load profile $\boldsymbol{y}^*$ by marginalizing over the distribution of the weights,

$$\begin{aligned} p(\boldsymbol{y}^*|\theta) &= \int p(\boldsymbol{y}^*|\boldsymbol{w})p(\boldsymbol{w}|\theta)d\boldsymbol{w} \\ &= \int \mathcal{N}(\boldsymbol{y}^*|\boldsymbol{\Phi}^T\boldsymbol{w}, \boldsymbol{\Sigma}_{y^*})\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w)d\boldsymbol{w} \\ &= \mathcal{N}(\boldsymbol{y}^*|\boldsymbol{\Phi}^T\boldsymbol{\mu}_w, \boldsymbol{\Phi}^T\boldsymbol{\Sigma}_w\boldsymbol{\Phi} + \boldsymbol{\Sigma}_{y^*}) \\ &= \frac{1}{|2\pi\boldsymbol{\Phi}^T\boldsymbol{\Sigma}_w\boldsymbol{\Phi} + \boldsymbol{\Sigma}_{y^*}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\Phi}^T\boldsymbol{\mu}_w)^T(\boldsymbol{\Phi}^T\boldsymbol{\Sigma}_w\boldsymbol{\Phi} + \boldsymbol{\Sigma}_{y^*})^{-1}(\boldsymbol{y}-\boldsymbol{\Phi}^T\boldsymbol{\mu}_w)\right\}. \end{aligned} \tag{30}$$

The weights $\boldsymbol{w}$ can be learned by Ridge Regression, see section 3.1.2. Given multiple demonstrations, the parameters θ are derived from the learned weights. It is now possible to create a general forecast

which only depends on the parameters learned from the old data by maximizing the probability for the forecast which leads to

$$\boldsymbol{y}^* = \boldsymbol{\Phi}\boldsymbol{\mu}_w. \tag{31}$$

The standard deviation of the forecast σ^* can be calculated with

$$\boldsymbol{\sigma}^* = \sqrt{\operatorname{diag}\left(\boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi} + \boldsymbol{\Sigma}_{y^*}\right)}. \tag{32}$$

3.1.1 Conditioned Forecast

Over the course of a day, the forecast may be updated. In that case, the incorporation of the accumulated data until the forecast is needed would be preferred. Formulating this mathematically means that at time $t \in [0, 1)$ the new, only partly available, observation $\boldsymbol{y}_p^*$ has been recorded. In fact, it is possible to formulate a distribution over the partly observed data

$$p(\boldsymbol{y}_p^* | \boldsymbol{w}) = \mathcal{N}(\boldsymbol{y}_p^* | \boldsymbol{\Phi}_p^T \boldsymbol{w}, \boldsymbol{\Sigma}_{y_p^*}), \tag{33}$$

which will be fed into the hierarchical Bayesian network to reduce the uncertainty introduced by the distribution over the weights from eq. (29). The corresponding covariance to the partly observed data $\boldsymbol{y}_p$ is detonated by $\boldsymbol{\Sigma}_p$. By using Bayes Theorem, it is possible to reformulate the distribution over the weights, from eq. (29), to take the new data into account

$$p(\boldsymbol{w} | \boldsymbol{y}_p) \propto \mathcal{N}(\boldsymbol{y}_p | \boldsymbol{\Phi}_{0:t}^T \boldsymbol{w}, \boldsymbol{\Sigma}_p) p(\boldsymbol{w}) \tag{34}$$

and calculate a forecast $\boldsymbol{y}^*$ which is conditioned on the new data for the current day by marginalizing over the weights

$$p(\boldsymbol{y}^* | \theta) = \int \mathcal{N}\left(\boldsymbol{y}^* | \boldsymbol{\Phi}_t^T \boldsymbol{w}, \boldsymbol{\Sigma}_{y^*}\right) p(\boldsymbol{w} | \boldsymbol{y}_p) \mathrm{dw}. \tag{35}$$

Since the conditioning is done on a Gaussian distributions, the resulting distribution is also Gaussian. Therefore, the new parameters $\theta^* = \{\boldsymbol{\mu}_w^*, \boldsymbol{\Sigma}_w^*\}$ of the weight distribution can be calculated in closed form as

$$\boldsymbol{\mu}_w^* = \boldsymbol{\mu}_w + \boldsymbol{\Sigma}_w \boldsymbol{\Phi}_t (\boldsymbol{\Sigma}_p + \boldsymbol{\Phi}_t^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}_t)^{-1} (\boldsymbol{y}_p - \boldsymbol{\Phi}_t^T \boldsymbol{\mu}_w), \tag{36}$$

$$\boldsymbol{\Sigma}_w^* = \boldsymbol{\Sigma}_w - \boldsymbol{\Sigma}_w \boldsymbol{\Phi}_t (\boldsymbol{\Sigma}_p + \boldsymbol{\Phi}_t^T \boldsymbol{\Sigma}_w \boldsymbol{\Phi}_t)^{-1} \boldsymbol{\Phi}_t^T \boldsymbol{\Sigma}_w, \tag{37}$$

where $\boldsymbol{\Sigma}_p$ is the desired precision on the data, especially the fit on the observed data points $\boldsymbol{y}_p$ of the current day. As a result a new forecast, conditioned on the new data, can be calculated:

$$p(\boldsymbol{y}^* | \theta) = \mathcal{N}(\boldsymbol{y}^* | \boldsymbol{\Phi}_t^T \boldsymbol{\mu}_w^*, \boldsymbol{\Phi}_t^T \boldsymbol{\Sigma}_w^* \boldsymbol{\Phi}_t + \boldsymbol{\Sigma}_y^*). \tag{38}$$

By maximizing eq. (38), a forecast based on the partial data $\boldsymbol{y}_p$ can be calculated

$$\boldsymbol{y}^* = \boldsymbol{\Phi}\boldsymbol{\mu}_w^* \tag{39}$$

$$\boldsymbol{\sigma}^* = \sqrt{\operatorname{diag}\left(\boldsymbol{\Phi}^T \boldsymbol{\Sigma}_w^* \boldsymbol{\Phi} + \boldsymbol{\Sigma}_{y^*}\right)}. \tag{40}$$

The recalculation does not require much computational effort because all computations are in closed form. The result of conditioning is that, as day goes by, the forecast $\boldsymbol{y}^*$ increases its accuracy by reducing the uncertainty from eq. (33).

3.1.2 Ridge Regression

Ridge regression [94], or sometimes called Tikhonov regularization [95, 96], is a more Bayesian approach to regression compared to linear regression. The difference to linear regression lies in the added prior distribution over the weights, which reduces the overfitting tendency of linear regression [47].

In linear regression, the maximum likelihood estimation (MLE) or least squares estimation is done over the probability

$$p(y|\boldsymbol{x},\theta) = \mathcal{N}\left(y|\boldsymbol{w}^T\boldsymbol{x},\sigma^2\right), \tag{41}$$

which defines a linear model. A non-linear relationship of input to output data can be achieved by using *basis function expansion* with $\phi(\boldsymbol{x}) = [1, x, x^2, \ldots, x^d]$ [42] .

$$p(y|\boldsymbol{x},\theta) = \mathcal{N}\left(y|\boldsymbol{w}\,\phi(\boldsymbol{x}),\sigma^2\right), \tag{42}$$

MLE maximizes the logarithmic likelihood of the distribution and results in the following solution:

$$\boldsymbol{w} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \tag{43}$$

Ridge regression adds a distribution over the weights $p(\boldsymbol{w}|\alpha) = \mathcal{N}\left(\boldsymbol{w}|0,\alpha^2\boldsymbol{I}\right)$ to the eq. (41), expanding linear regression as follows:

$$\begin{aligned} p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})p(\boldsymbol{w}) \\ p(\boldsymbol{w}|\boldsymbol{X},\boldsymbol{y}) &\propto \mathcal{N}(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{w})\mathcal{N}(\boldsymbol{w}|0,\alpha^2\boldsymbol{I}) \end{aligned} \tag{44}$$

By maximizing the posterior distribution, a new estimation of $\boldsymbol{w}$ is possible [42]:

$$\arg\max_{\boldsymbol{w}} \sum_{i=1}^{N} \mathcal{N}\left(y_i|w_0 + \boldsymbol{w}^T x_i,\sigma^2\right) + \sum_{j=1}^{D} \ln \mathcal{N}\left(w_j|0,\alpha^2\right) \tag{45}$$

Solving eq. (45) in prospect to the weights $\boldsymbol{w}$ results in:

$$\boldsymbol{w} = \left(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{\Phi}\boldsymbol{y} \tag{46}$$

where λ corresponds to $\frac{\sigma^2}{\alpha^2}$ and λ is a term to penalize complexity [42]. The result is a more robust form of regression, since $(\boldsymbol{\Phi}\boldsymbol{\Phi}^T + \lambda\boldsymbol{I})$ is better invertible than linear regression counterpart $\boldsymbol{X}\boldsymbol{X}^T$ from eq. (43) [42].

3.1.3 Expectation Maximization

In real world applications, the completeness of the electric load profiles cannot always be guaranteed. The reason for that is that there are some common problems with the logging system. Maintenance, for example, can disrupt the chronology of the observation. With regard to the possibility of incomplete data, the need to look for another way to create the weights and the distribution over the weights, instead of Ridge Regression from section 3.1.2, to compensate for the loss becomes apparent. Expectation Maximization is a way to cope with missing data and will be used here to calculate the parameters $\boldsymbol{\mu}_w$ and $\boldsymbol{\Sigma}_w$. The basics for the implementation of the algorithm have been provided by *M. Ewerton*'s thesis [97].

The objective of the Expectation Maximization algorithm is to calculate the mean and covariance of $p(\boldsymbol{w})$, where $\boldsymbol{w}$ is a hidden vector-valued variable:

$$\prod_i p(\boldsymbol{y}_i|\theta) = \prod_i \sum_w p(\boldsymbol{y}_i|\boldsymbol{w}, \boldsymbol{\Sigma}_y) p(\boldsymbol{w}), \tag{47}$$

where $p(\boldsymbol{y}_i|\boldsymbol{w}, \boldsymbol{\Sigma}_y)$ is a Gaussian distribution over a training example and θ is, again, defined as the tuple of parameters $\theta = \{\boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w\}$. The initial estimation of θ can be done by using Ridge Regression, from section 3.1.2, resulting in

$$\theta_0 = \{\boldsymbol{\mu}_{w0}, \boldsymbol{\Sigma}_{w0}\}. \tag{48}$$

The Expectation Maximization algorithm gets its name from the two steps necessary to calculate eq. (47), which are sub sequential iterated until convergence is achieved. In the Expectation Step (E-Step) "'*the missing data are estimated given the observed data and current estimate of the model parameters*"'[98, page 5]. This is done by computing the so-called *expected complete data log likelihood* $Q(\theta, \theta_0)$. In the Maximization Step (M-Step), the new parameters θ are estimated by maximizing the $Q(\theta, \theta_0)$ function.

Expectation Step

In ProMP the *expected complete data log likelihood* has the form:

$$\begin{aligned}
Q(\theta, \theta_0) &= \sum_i \mathbb{E}_{\theta_0}(\ln(p_\theta(\boldsymbol{y}_i|\boldsymbol{w})|\boldsymbol{y} = \boldsymbol{y}_i)) \\
&= \sum_i \ln\left(\sqrt{\frac{1}{|2\pi\sigma^2\boldsymbol{I}|}}\right) \\
&- \frac{1}{2}\sum_i \left\{\boldsymbol{y}_i^T(\sigma^2\boldsymbol{I})^{-1}\boldsymbol{y}_i - 2\boldsymbol{y}_i^T(\sigma^2\boldsymbol{I})^{-1}\boldsymbol{\Phi}\boldsymbol{\mu}_{wi} + \boldsymbol{\mu}_{wi}^T\boldsymbol{\Phi}^T(\sigma^2\boldsymbol{I})^{-1}\boldsymbol{\Phi}\boldsymbol{\mu}_{wi} + Tr\left[\boldsymbol{\Phi}^T(\sigma^2\boldsymbol{I})^{-1}\boldsymbol{\Phi}\boldsymbol{\Sigma}_{wi}\right]\right\} \\
&+ \sum_i \ln\left(\sqrt{\frac{1}{|2\pi\boldsymbol{\Sigma}_w|}}\right) \\
&- \frac{1}{2}\sum_i \left\{(\boldsymbol{\mu}_{wi} - \boldsymbol{\mu}_w)^T\boldsymbol{\Sigma}_w^{-1}(\boldsymbol{\mu}_{wi} - \boldsymbol{\mu}_w) + Tr\left[\boldsymbol{\Sigma}_w^{-1}\boldsymbol{\Sigma}_{wi}\right]\right\}.
\end{aligned} \tag{49}$$

For the complete derivation of the equations, the reader should referred to [97].

Maximization Step

New estimations for the parameters are calculated by maximizing eq. (49), the new parameters are

$$\boldsymbol{\mu}_w^* = \frac{\sum_i \boldsymbol{\mu}_{wi}}{M}, \qquad \boldsymbol{\Sigma}_w^* = \frac{\boldsymbol{E}^T\boldsymbol{E} + \sum_i \boldsymbol{\Sigma}_{wi}}{M}. \tag{50}$$

where M is the number of training examples and the remaining variable are defined as follows:

$$\boldsymbol{\Sigma}_{wi} = \left(\boldsymbol{\Phi}^T(\sigma^2\boldsymbol{I})^{-1}\boldsymbol{\Phi} + \boldsymbol{\Sigma}_{w0}^{-1}\right)^{-1} \tag{51}$$

$$\boldsymbol{\mu}_{wi} = \boldsymbol{\Sigma}_{wi}\left(\boldsymbol{\Phi}^T(\sigma^2\boldsymbol{I})^{-1}\boldsymbol{\Phi} + \boldsymbol{\Sigma}_{w0}^{-1}\boldsymbol{\mu}_{w0}\right) \tag{52}$$

$$\boldsymbol{e}_i = \boldsymbol{\mu}_{wi} - \boldsymbol{\mu}_w, \qquad \boldsymbol{E} = \begin{pmatrix} \boldsymbol{e}_1^T \\ \boldsymbol{e}_2^T \\ \vdots \\ \boldsymbol{e}_M^T \end{pmatrix} \tag{53}$$

The parameters calculated this way can be used as in section 3.1 to calculate the unconditioned forecast of electric load profiles, with eq. (30) or the conditioned forecast of electric load profiles, from section 3.1.1 with eq. (38).

3.2 Clustering for ProMP

So far it has been implied that the data for each day is looking similar, with a peak at noon. However, this may not always be the case, because consumer loads vary greatly from weekdays to weekends or national holidays. This can be seen in section 3.2, which shows the Monday in week 22 from 2013 and section 3.2 shows the Thursday of the same week. After taking a look at section 3.2, which shows the electric load profile of the Saturday from the same week, their similarity becomes obvious. That is the case, because the Thursday has been a national holiday. These Figures are provided by a light industrial facility, further explained in section 4.1.

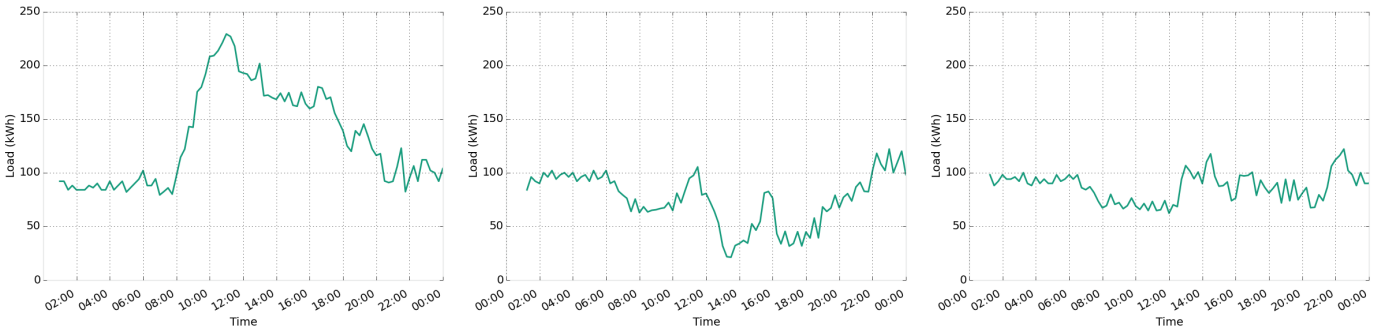

Figure 10: From left to right, the electric load profiles for: Monday the 27 in May 2013, Thursday the 30 in May 2013 and Saturday the 1 in June 2013

A workaround for the problem of the difference between weekdays and days of the weekend is to imply a clustering of the data. A naive approach would be that the system only learns the weights from

the same weekdays as the day of the forecast.

Nevertheless, seasonal components are still an issue. A shorter learning period ensures that seasonal influences are minimal. However, this means that the training data is drastically limited, which can result in a bad forecast. Another problem, is the identification of national holidays without predetermining them before hand.

These problems can be approached by using the combination of K-Means algorithm and Expectation Maximization algorithm. The result of K-Means is a predetermined number of clusters with mixed Gaussian probabilities [47]. By using Expectation Maximization, the Gaussian probability of each cluster can be calculated and the most probable cluster can be determined. By conditioning on the most probable cluster, a forecast based solely on the data entries of the most probable cluster is possible. This approach aims at identifying the difference between holidays, weekends and weekdays in runtime and in *weight space*.

3.2.1 K-Means Clustering

K-Means clustering goes back to the idea of *Steinhaus* in 1955 [99] and has first been formalized by *Lyod* in 1982 [100]. The bottom line of the algorithm is the partition of the training dataset with M entries into k smaller sections, with $k \ll M$. In each subsection (cluster) all entries have the same distance to the center of the cluster $\boldsymbol{\mu}_i$, named the cluster mean. The distance to the cluster mean is calculated by the following equation, which is sometimes called the distortion measure [47]:

$$J = \sum_{i=1}^{k} \sum_{\boldsymbol{x}_j \in S_i} \| \boldsymbol{x}_j - \boldsymbol{\mu}_i \|^2, \tag{54}$$

where $\boldsymbol{x}_j$ describes each observation with $j \in [0, M]$ and S_i, with $i \in [0, k]$, is one of the calculated clusters. The algorithm by *Lloyd* [100] proceeds as follows :

Initialization

- Choose random values for $\boldsymbol{\mu}_0$ to $\boldsymbol{\mu}_k$ from the data.

Assignment

- For each $\boldsymbol{x}_j$ calculate the distance to $\boldsymbol{\mu}_i$ and assign each $\boldsymbol{x}_j$ to the corresponding S_i where the distance is minimal.

Update

- For each S_i calculate $\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{\boldsymbol{x}_j \in S_i} \boldsymbol{x}_j$ and recalculate J with eq. (54).

Assignment step and update step are repeated until the difference of J from one iteration to the next falls below a predetermined threshold [47].

3.2.2 Expectation Maximization for Clustering

The result of clustering is a mixed Gaussian distribution [47]. To calculate the probabilities for each cluster and to redefine the distribution over the weights, the Expectation Maximization Algorithm is used.

The algorithm is defined similarly to the Expectation Maximization from section 3.1.3.

The probabilities of each cluster $p(k)$ are defined as $\alpha_k = \frac{n_k}{n}$, where n is the number observations and n_k is the number of observations assigned to cluster k. The clusters are a priori calculated by using K-Means from section 3.2.1, as well as the covariance matrix $\boldsymbol{\Sigma}_k$ and mean vector μ_k for each cluster. With these initial parameters, the Expectation Maximization Algorithm can be used in the following way.

Expectation Step

In the Expectation Step the probabilities of each cluster k, given the weight vector w_i, are calculated, as follows:

$$p(k|\boldsymbol{w}_i) = r_{ik} = \frac{\mathcal{N}(\boldsymbol{w}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\alpha_k}{\sum_j^K \alpha_j \mathcal{N}(\boldsymbol{w}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}, \tag{55}$$

where K is the number of clusters and r_{ik} are called the *responsibilities* [47].

Maximization Step

The Maximization Step is used to update the parameters:

$$n_k = \sum_{i=1}^{n} r_{ik}, \qquad a_k = \frac{n_k}{n} \tag{56}$$

$$\mu_k = \frac{\sum_i^n r_{ik}\boldsymbol{w}_i}{n_k}, \qquad \boldsymbol{\Sigma}_k = \frac{1}{n_k}\left(\sum_i^n (\boldsymbol{w}_i - \boldsymbol{\mu}_k)(\boldsymbol{w}_i - \boldsymbol{\mu}_k)^T\right). \tag{57}$$

Afterwards, the probability $p(\boldsymbol{w})$ can be calculated,

$$\begin{aligned} p(\boldsymbol{w}) &= \prod_i \sum_k p(k)p(\boldsymbol{w}|k) \\ &= \prod_i \sum_k \alpha_k \mathcal{N}(\boldsymbol{w}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \end{aligned} \tag{58}$$

Inference

With the prior distribution over the weights calculated by Expectation Maximization, it is possible to calculate the most probable cluster k^*, given the partial observation $\boldsymbol{y}_p$ from section 3.1.1, using Bayes Theorem [47] as follows:

$$\begin{aligned} p(k|\boldsymbol{y}_p) &\propto p(\boldsymbol{y}_p|k)p(k), \\ p(k|\boldsymbol{y}_p) &\propto p(\boldsymbol{y}_p|k)\alpha_k \end{aligned} \tag{59}$$

where $p(\boldsymbol{y}_p|k)$ can be calculated with $p(\boldsymbol{w}|k) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ from eq. (58) and $p(\boldsymbol{y}_p|\boldsymbol{w})$ from eq. (33) by

$$\begin{aligned} p(\boldsymbol{y}_p|k) &= \int p(\boldsymbol{y}_p|\boldsymbol{w})p(\boldsymbol{w}|k)\mathrm{d}\boldsymbol{w} \\ &= \int \mathcal{N}(y_t^*|\boldsymbol{\Phi}_t^T w, \boldsymbol{\Sigma}_y^*)\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\mathrm{d}\boldsymbol{w}\,. \end{aligned} \tag{60}$$

With the most probable cluster calculated by $k^* = \arg\max\left(p(k|\boldsymbol{y}_p)\right)$, it is possible to condition over the cluster k^* for the partial observation $\boldsymbol{y}_p$ and to calculate the most probable forecast $\boldsymbol{y}^*_{k^*}$:

$$p(\boldsymbol{w}|k^*, \boldsymbol{y}_p) \propto \mathcal{N}(\boldsymbol{y}_p|\boldsymbol{\Phi}_t^T w, \boldsymbol{\Sigma}^*_y)\mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_{k^*}\boldsymbol{\Sigma}_{k^*}) \tag{61}$$

$$\begin{aligned} p(\boldsymbol{y}^*_{k^*}\theta, k^*) &= \int p(\boldsymbol{y}^*_{k^*}|\boldsymbol{w})p(\boldsymbol{w}|k^*, \boldsymbol{y}_p)\mathrm{d}\boldsymbol{w} \\ &= \mathcal{N}\left(\boldsymbol{y}^*_{k^*}|\boldsymbol{\Phi}^T\boldsymbol{\mu}_{k^*}, \boldsymbol{\Phi}^T\boldsymbol{\Sigma}_{k^*}\boldsymbol{\Phi} + \boldsymbol{\Sigma}^*_y\right). \end{aligned} \tag{62}$$

These equation can be calculated in closed form like it has been shown before in eq. (36) and eq. (37). The result is the probability for the forecast as it has been defined in eq. (38) and can be used to calculate a more accurate forecast. The calculation is the same as in eq. (31):

$$\boldsymbol{y}^*_{k^*} = \boldsymbol{\Phi}\boldsymbol{\mu}_{k^*} \tag{63}$$

$$\boldsymbol{\sigma}^*_{k^*} = \sqrt{\mathrm{diag}\left(\boldsymbol{\Phi}^T\boldsymbol{\Sigma}_{k^*}\boldsymbol{\Phi} + \boldsymbol{\Sigma}_{y^*}\right)}. \tag{64}$$

4 Evaluations

In this section, the origin of the data sets will be described, followed by a section describing how the different approaches are optimized. Next is the definition of the evaluation criteria and the definition of the validation methods. This chapter is closed by the comparison of the different methods for electric load forecasting.

4.1 Data Sets

For the validation of the different forecasting methods, each method has been used on all time series of the different data sets. These data sets can be categorized into two groups, group one is aggregated load for a light industrial facility. The second group consists of load profiles for different types of households. For the sake of comparison the data has been normalized to equidistant time steps. The distance between between each data point is fifteen minutes. If the data has been available in a more finely graduated form, the mean for multiple time steps has been calculated and taken as new data point.

Despite indications of a positive correlation between electric consumption and air temperature [101], only the load will be used as a input variable for each approach. This is in contrast to a survey from 2001 done by *Hippert et al.* [102], which discovered that most load forecasting methods use additionally weather forecasts or an ensemble of them [103] as input variables. The reason why weather forecasts have not been used as additional input is that, they introduce further uncertainty, which can influence the forecast negatively [104] and are highly dependent on the location [64]. As a result each tested method is independent of the location and only depends of the historic data of the data set.

4.1.1 Industrial Data Set

This data set contains the consumed load data for the years of 2013 and 2014 recorded by the operator of the light industrial park in Kitzingen in Germany and has been provided as part of the research project Intellan with the Fraunhofer Institute for solar energy systems [105]. This light industrial park, shown

in fig. 11, consists of a multitude of buildings with different purposes and is inhabited by different types of companies, additionally many buildings have solar panels installed on the roof.

For each time step the true load value L_{true} is calculated by the sum of the loads for L_i of the different facilities M, as follows

Figure 11: The light industrial park in Kitzingen in Germany from which the industrial data set has been provided

$$L_{true} = \sum_{i}^{M} L_i - \sum_{i}^{M} PV_i + L_{bought} \tag{65}$$

where PV is the photo-voltaic load produced by the different solar plants on top of the roofs and L_{bought} is the load obtained from the grid. The industrial data for both years is highly similar, besides having different peaks, which can be seen in the overall mean in fig. 12 and in the mean of all the weekdays fig. 13. Although, the weekends look different, the data for 2014 looks similar to the data for a weekday, which suggests that work is done on the weekends, as seen in fig. 14. An overview of the industrial data set can be found in table 3.

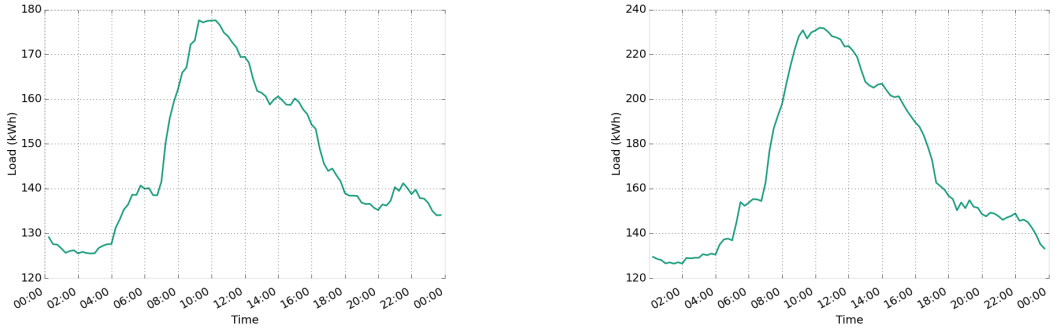

Figure 12: Average electric load profile, on the left for 2013 and on the right for 2014

Year	Minimal Load	Mean Load	Maximal Load
2013	−320.4800	147.2188	576.0000
2014	−558.0000	170.8113	461.0000

Table 3: An overview of industrial data set

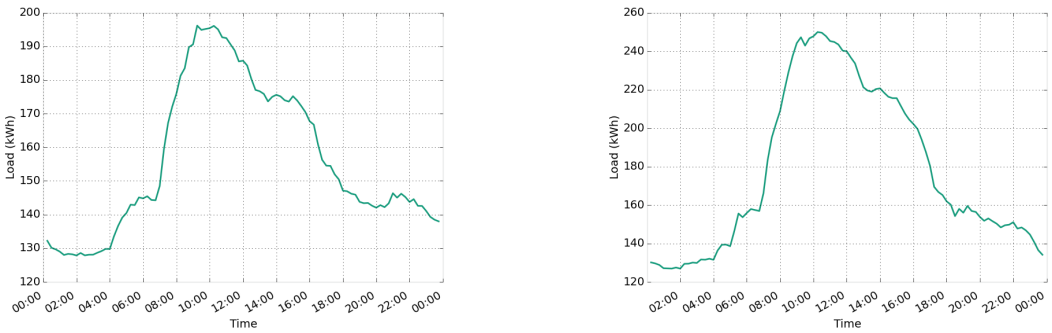

Figure 13: Average electric load profile for weekdays, on the left for 2013 and on the right for 2014

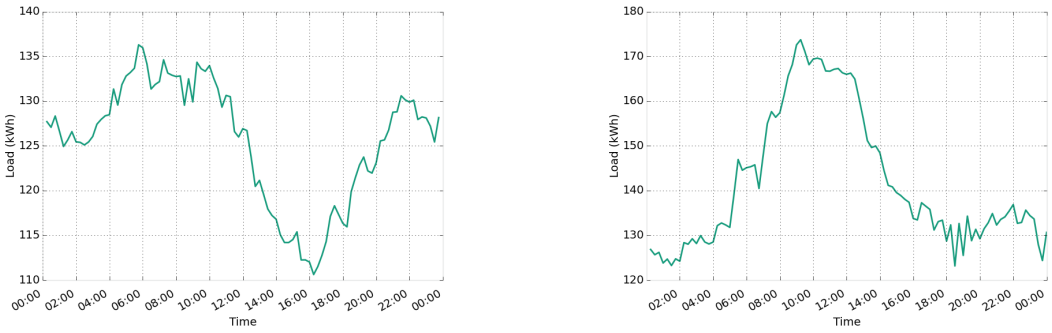

Figure 14: Average electric load profile for weekends, on the left for 2013 and on the right for 2014

4.1.2 Household Data Set

This data set contains the load data of different sized households for the year 2011. The data set includes: a very economical household of two workers and two kids with an annual sum of 2219 kWh (normal usage per capita is 2972 kWh [106]), a synthetic electric load profile of a four person household with an annual sum of 4647 kWh, a household with two kids and one worker with an annual sum of 5600 kWh and a household with two kids and two workers with an annual sum of 6500 kWh. By comparing the annual sums of the different households, it becomes obvious that the load is highly dependent on the number of inhabitants and the their behavior. As a result, the load profiles across the household look highly different, which is shown in figs. 15 to 17, and table 4.

Annual Sum	Minimal Load	Mean Load	Maximal Load
2219 kWh	0.0286	0.2533	4.8190
4647 kWh	0.0600	0.5308	7.0867
5600 kWh	−0.2600	0.6490	8.4280
6500 kWh	0.1827	0.7407	8.4827

Table 4: An overview of household data set

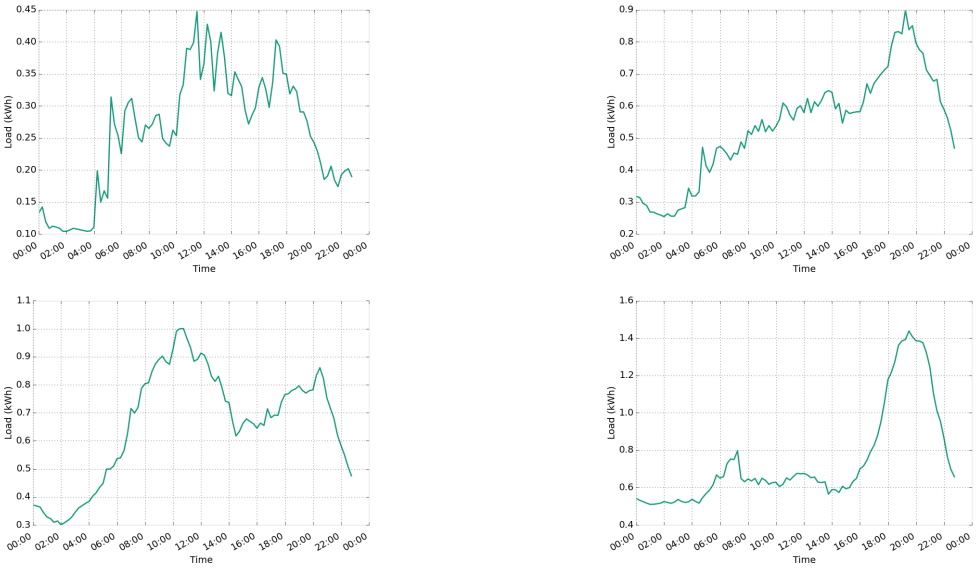

Figure 15: Mean of all electric load profiles for all households and all days. From left to right: household with 2219 kWh, household with 4647 kWh, household with 5600 kWh, household with 6500 kWh

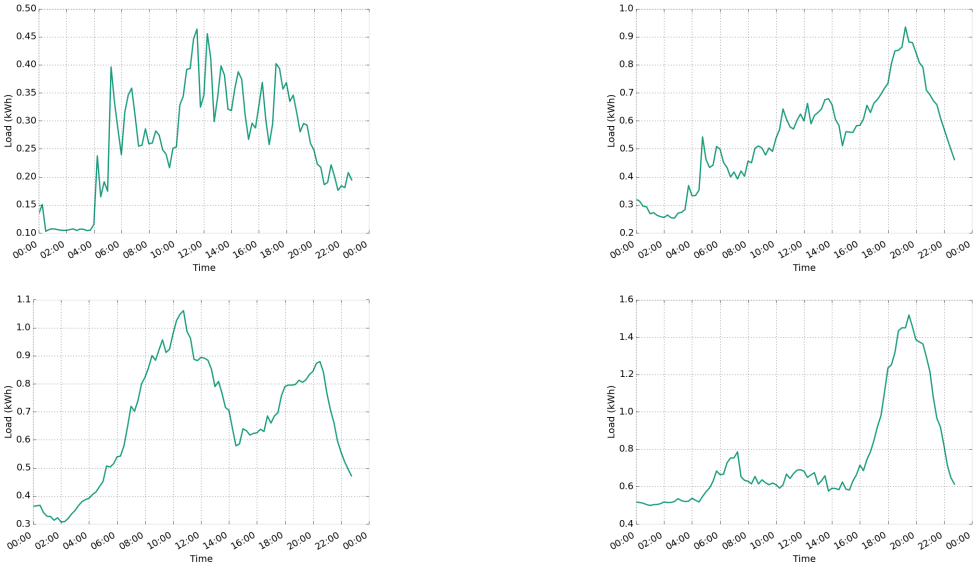

Figure 16: Mean of all electric load profiles for all households and all weekdays. From left to right: household with 2219 kWh, household with 4647 kWh, household with 5600 kWh, household with 6500 kWh

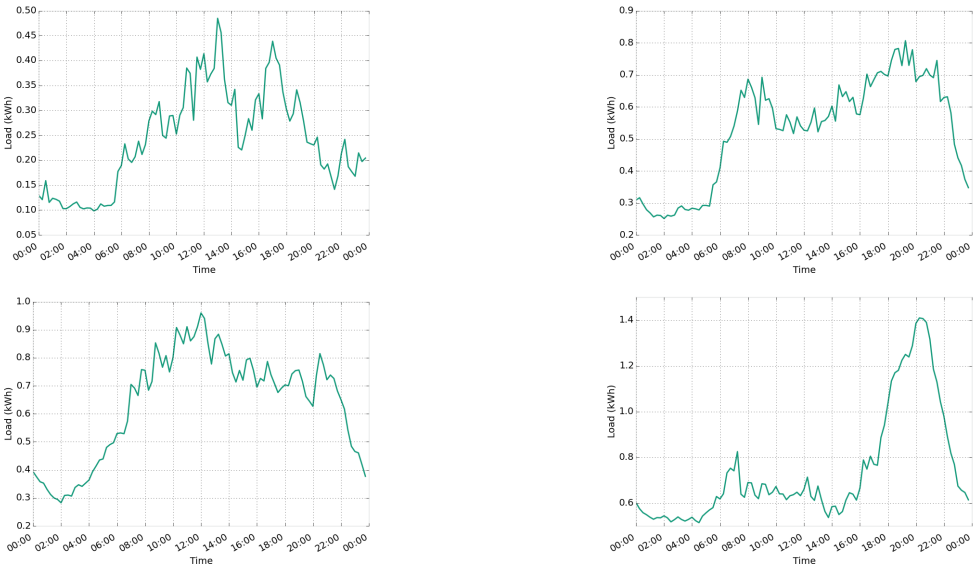

Figure 17: Mean of all electric load profiles for all households and all weekends. From left to right: household with $2219\,\mathrm{kWh}$, household with $4647\,\mathrm{kWh}$, household with $5600\,\mathrm{kWh}$, household with $6500\,\mathrm{kWh}$

4.1.3 Handling Outliers

Outliers are points in the data which are faulty. The reason for corrupted data points are plentiful: from maintenance on the logging system to software faults everything could be possible.

Most outliers can be easily detected because the data for these does not make sense. In the electric load profiles from the industrial data set, shown in fig. 18, are loads which are smaller than zero. This is not possible because the solar production load has been removed, as explained in section 4.1.1. Profiles with such outliers have been removed from the data set. Additionally, load profiles with only few values have been removed too. There are other outliers which can not be so easily detected and would need an expert to look at the data, these have not been removed [23]. An overview of the data without outliers can be seen in table 6.

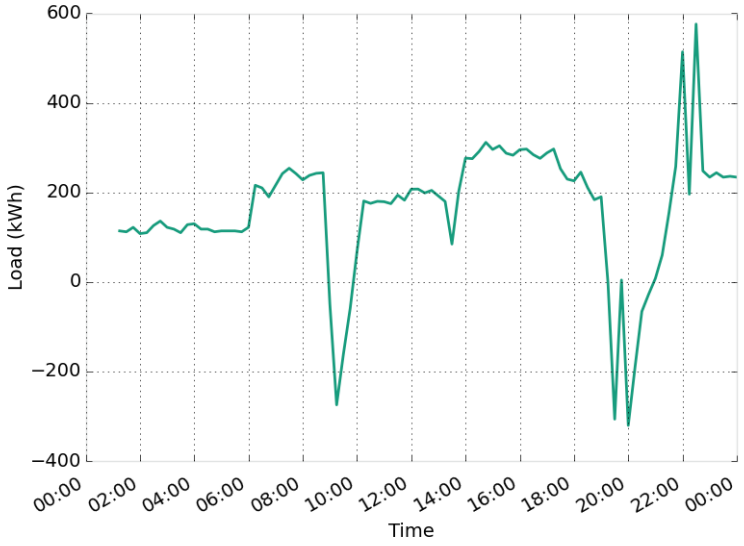

Figure 18: Outlier from Industrial Data Set

Data set	Minimal Load	Mean Load	Maximal Load
Industrial 2013	0.000	147.8001	386.0000
Industrial 2014	0.000	172.2946	461.0000
Household 2219 kWh	0.0286	0.2571	4.8190
Household 4647 kWh	0.0600	0.5373	7.0867
Household 5600 kWh	−0.2600	0.6586	8.4280
Household 6500 kWh	0.1827	0.7476	8.4827

Table 6: An overview of household data set

4.2 Parameter Optimization

In this section, the different parameters for each approach are highlighted and the optimization of these parameters will be explained.

4.2.1 Parameter Optimization for Probabilistic Movement Primitives

For the optimization of the hyperparameters for ProMP, the library from "'*Laboratoire de Recherche en Informatique*"' for Covariance Matrix Adaptation Evolution Strategy (CMAES) has been used. The library is based and developed by *Nikolaus Hansen* [107]. CMAES is a blackbox optimizer provided in different programming languages for example *Java*, *C++* or *MATLAB*. The optimizer is based on evolution algorithmic [107] and is used to find the optimal hyperparameters in respect to the RMSE explained in section 4.3. The hyperparameters are:

h
: which is the width of each basis function, see eq. (25).

N
: which is the total number of used basis functions and influences the dimensional reduction, which is explained in section 3.1.

λ
: which is the complexity penalty parameter for Ridge regression from section 3.1.2 and helps with the inversion when calculating the weights.

Σ_p
: which is the desired precision on the current data, see eq. (38), it was chosen as $\sigma_p \boldsymbol{I}$.

k
: if clustering has been used, will k define the number of clusters used, see section 3.2.1.

The optimization process has been started with 24 basis functions, each representing an hour of the day and the other parameters have been set to one. The number of cluster centers has not been found by optimization, instead it was set to 14 to automatically differentiate summer and winter days. The resulting values of the optimization for the hyperparameters can be seen in table 7.

Parameter Name	Parameter value
h	8.522×10^{-5}
N	48
λ	0.041
σ_p	0.952
k	14

Table 7: The optimal parameter found by the *CMAES* library

4.2.2 Parameter Optimization for Persistence Model

The normal persistence model has no parameters, the advanced *similar day*-model has only one parameter M. The parameter M represents the number of weeks included in the calculation of the mean over

the historic data, see section 2.1.1. This parameter has been optimized by hand and has been set to the value of $M = 3$. This means that up to three weeks are taken into account for the calculation of the forecast of the Persistence model.

4.2.3 Parameter Optimization for Artificial Neuronal Network

The parameters for the artificial neuronal network can be automatically be optimized by *MATLAB*. To optimize the parameters, the trainings data set is randomly split by *MATLAB* into three parts: 70% training, 15% validation, 15% testing. The ANN is trained by a more robust variation of gradient descent, which is called *Scaled Conjugate Gradient*, and is a algorithm made for batch optimization [47]. The performance of the ANN is automatically tested on the test part by calculating the *Mean Squared Error* $= \sum_i \frac{1}{n} \left(y_i^* - y_i\right)^2$. If the ANN can be validated six times, the training phase is finished.

Epochs
: the number of maximal training rounds done.

Neurons
: the number of maximal neurons are used for the hidden layer.

History Size
: the amount of loads are in relation to the actual load value, see section 2.2.2.

Forecast window
: the length between the actual load value and the future load value, see section 2.2.2.

The resulting values are can be seen in table 8 and the resulting artificial neuronal network created by *MATLAB* can be seen in fig. 19.

Epochs	max neurons	history size	forecast window
1000	10	90	4

Table 8: The optimal parameters found by hand

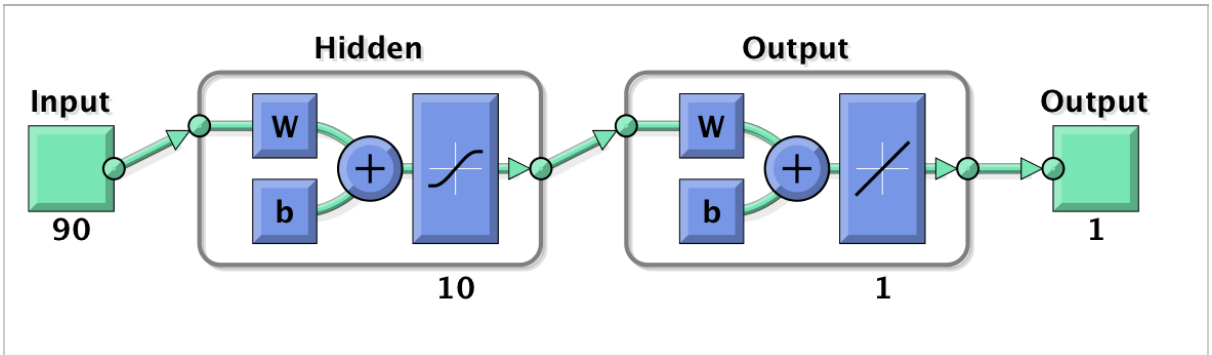

Figure 19: Sketch of the resulting neuronal network generated by *MATLAB*

4.2.4 Parameter Optimization for Gaussian Process

The *GPStuff* library, used in this thesis, comes with an optimization function. This function can optimize the GP with respect to the likelihood, the covariance, the inducing parameters or a combination of these. For the evaluation of the GP, it has been optimized with respect to the covariance and the likelihood. The optimization is done by *maximum a posterior* in cooperation with gradient descent. The most probable parameters θ from the underlying function $\boldsymbol{f}$ are deduced by the following calculation:

$$\begin{aligned} < \theta_{\max}, \phi_{\max} > &= \arg\max_{\theta,\phi} p\left(\{\boldsymbol{X}, \boldsymbol{y}\} \,|\phi, \theta\right) \\ &= \arg\max_{\theta,\phi} \int p\left(\boldsymbol{y}|\boldsymbol{f}, \phi\right) p\left(\boldsymbol{f}|\boldsymbol{X}, \theta\right) \mathrm{d}\boldsymbol{f}, \end{aligned} \tag{66}$$

where the prior distribution $p\left(\boldsymbol{f}|\boldsymbol{X}, \theta\right)$ is the distribution from eq. (19) and the likelihood distribution is a Gaussian likelihood of the form $p\left(\boldsymbol{y}|\boldsymbol{f}, \sigma_{\text{lik}}^2 \boldsymbol{I}\right)$. The likelihood distribution describes how the training data deviates from the predicted underlying function [46]. As a result the parameters θ are

σ_{lik}
: which describes the noise variance of the training inputs [79].

σ_{sexp}
: which defines the variability of the GP [46].

l_k
: which describes the correlation of the input variables between each other [79].

4.3 Evaluation Criteria

The performance of the different approaches to forecasting electric load profiles will be assessed by means of three criteria. The *root mean squared error* (RMSE) is the most common evaluation criterion for regression [46] and is calculated by:

$$\text{RMSE}_i = \sqrt{\frac{\sum_{t=1}^{n} (\boldsymbol{y}_i^* - \boldsymbol{y}_i)^2}{n}} \tag{67}$$

where $\boldsymbol{y}_i^*$ is the forecast, $\boldsymbol{y}_i$ is the actual value for that day and n is length of that day. Additionally, for the forecast done to a later time of the day the RMSE_i will be calculated for the remaining time of the day, without the time steps already available to the forecast algorithm. To compare the results of the cross validation, the sum of RMSE_i for each forecast and the mean over all RMSE_i for each forecast is calculated.

One of the problems with RMSE is that it is prone to outliers [108] and not ratio-scaled, which means that the calculated RMSE is only comparable to data drawn from the same data set [18]. To compare the approaches across different data sets the *mean absolute percentage error* (MAPE) is calculated. MAPE_i for day i is defined as:

$$\text{MAPE}_i = \frac{1}{n} \sum_{t=1}^{n} \frac{\left|100\left(\boldsymbol{y}_i - \boldsymbol{y}_i^*\right)\right|}{\boldsymbol{y}_i}, \tag{68}$$

where the variables are defined as before in eq. (67). The last evaluation criterion is, the calculation of the Symmetric Mean Absolute Percentage Error (SMAPE), which overcomes the tendency of MAPE to put a heavier penalty on positive error [108]. SMAPE_i for day i is defined as follows:

$$\text{SMAPE}_i = \frac{1}{n}\sum_{t=1}^{n}\frac{200\left|y_i - y_i^*\right|}{y_i + y_i^*} \tag{69}$$

In order to assess the performance of the different approaches with respect to the adjusting dynamics of the current day profile, new forecasts are generated at every hour from midnight to noon. This results in eleven forecasts for each day and eleven values for each evaluation criteria for each day. To compare the performance of each approach over all test days, the average for each time step is calculated. Additionally, the overall performance is evaluated by calculating the average over all time steps. Thus, a single value for each evaluation criteria for each data set is derived.

4.4 Cross Validation

To evaluate the performance of the different approaches, each approach will be validated by using the popular method of *k-fold* cross validation [42]. In *k-fold* cross validation, the data is randomly split into K *folds*, and each approach uses $K-1$ *fold* as training samples and the last *fold* as test sample. For each entry in the test sample, the evaluation criteria from section 4.3 are calculated. This procedure has to be done for each *fold*. For this thesis, two different ways to *fold* the data will be used, which are: $\frac{2}{3}$ cross validation, where three *folds* are used and *leave-one out* cross validation. *Leave-one out* cross (LOOC) validation is the fully exhaustive case of *k-fold* cross validation, where the data set with N entries is split into N folds and $N-1$ *folds* are used for training and only one data entry is used for testing, iterated over each entry. However, *leave-one out* cross validation is computationally very taxing, as a result, only the persistence model and the ProMP approach will be validated this way.

4.5 Comparison

In this section a comparison is made using the validation methods from section 4.4 and the evaluation criteria from section 4.3. Afterwards an general overview of the performance of each approach with respect to each evaluation criterion will be given. Followed by a conclusion at the end this section. For a complete list of the performance with respect to the different evaluation criteria, the reader is referred to section 5.

4.5.1 K-Means Evaluation

Some days have been chosen to highlight the performance of the K-Mean extension. These days are taken from the 2013 industrial data set. The ninth day of the industrial data set has been a Wednesday and can be seen in fig. 20. In table 17, it can be seen that overall error for the $\frac{2}{3}$ cross validation of the ProMP approach with Clustering has the second best value with MAPE and the best result with SMAPE, see table 16. In fig. 21, the performance forecast can be seen over the course of the day and the clustering in relation to the error can be seen in table 9. It becomes obvious that the cluster with the maximal likelihood is cluster one, other days in this cluster are for example: Tuesday the 01/08/13, Wednesday the 01/16/13 or Thursday the 01/31/13, shown in table 47. These days are all weekdays and days which are not national holidays. The first two assignments to cluster five are bad because cluster five includes mostly Sundays and Saturdays. However, these assignments are not important because most

days start similar, which can be concluded from the low error for these assignments and has also been discussed in section 4.1.1. The Assignment to the same cluster at 7:00 o'clock is wrong and results in a penalty in performance.

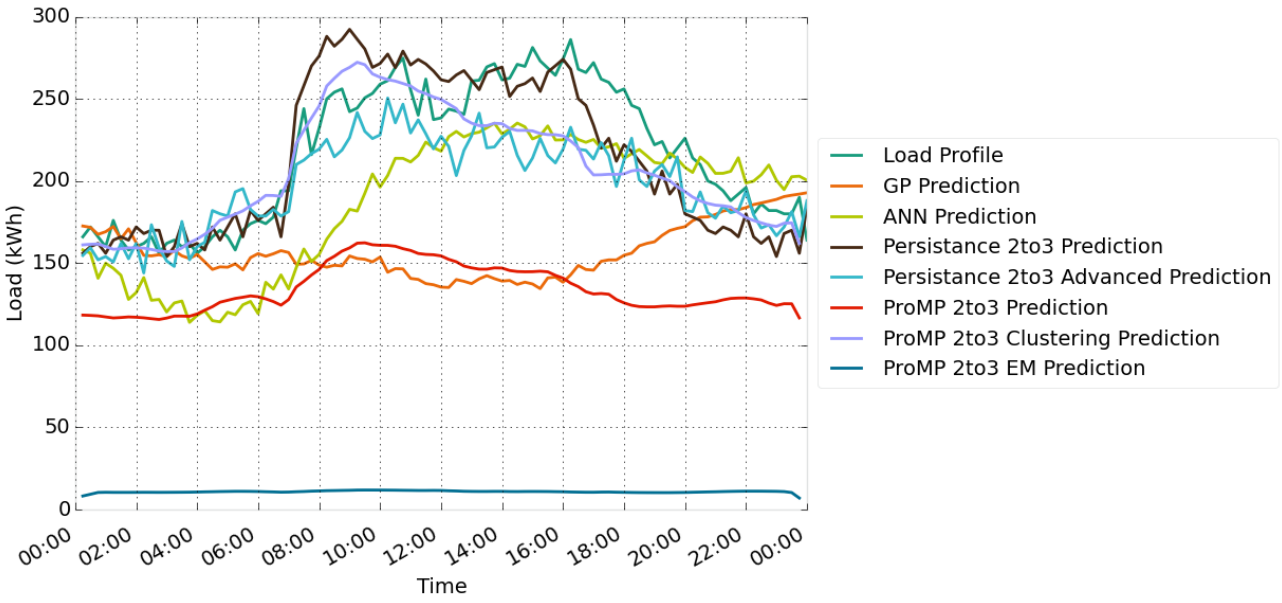

Figure 20: Load forecasting with all approaches for the ninth day in 2013 of the industrial data set

Time	RMSE	MAPE	SMAPE	Cluster
0:00	25.154	8.973	8.447	5
1:00	30.846	11.628	10.880	5
2:00	24.232	8.573	8.147	1
3:00	24.552	9.442	9.148	1
4:00	14.964	5.362	5.308	10
5:00	14.373	4.928	4.867	1
6:00	15.192	5.268	5.159	1
7:00	61.717	31.244	25.429	5
8:00	16.804	5.845	5.880	1
9:00	22.459	8.390	7.968	1
10:00	21.649	7.641	7.287	1
11:00	31.852	11.854	10.946	1

Table 9: The performance of the ProMP approach with K-Means extension with respect to the evaluation criteria

4.5.2 RMSE Comparison

The study of the RMSE shows that ProMPs beat the other approaches over the course of a day because it becomes gradually more accurate, while the difference in the industrial data set is quite obvious, see tables 27 and 28. The difference in the household data is small, see tables 23 to 26. This can be contributed to the different scale of the two data sets. The different scale of the data set makes it more difficult to compare the performance if only the RMSE is taken into account. However, a closer look

Figure 21: Load forecasting with ProMP and K-Means for the ninth day of the industrial data set. The black bar indicates the time the forecast is calculated. The times are: at 0:00 o'clock, at 4:00 o'clock, at 8:00 o'clock and at 12:00 o'clock

at the comparison of the household data shows that, while the other approaches struggle to keep the RMSE low, the ProMPs hold or decrease their RMSE, for example table 23. The rise of the RMSE for most of the methods is cause by the division by a decreasing number of points. The reason for a constant error in persistence approaches lie in their design, as it has been discussed in section 2.1, their basis is that, there is a similarity between electric load profiles but they will not be the same. In this context, the data of the household with an annual sum of 5600 kWh has to be mentioned in table 24, since there the mean RMSE for both persistence Models stay at their initial value. Around 27 for advanced persistence and 40 for the normal persistence model in LOOC and around 30 and 40 in $\frac{2}{3}$ cross validation.

The difference in scale becomes more apparent after taking a look at the sum over all RMSE, where the superiority of ProMPs becomes smaller. Especially in the second data set for 2014, ProMPs are beaten by the advanced persistence Model(sum of RMSE 13700 to 12878 in LOOC) . This can be contributed to the fact that RMSE is prone to outliers [18] and the mean over the RMSEs significantly reduces their influence [71]. As it has been discussed in section 4.1.3, the data sets are flawed especially the industrial data set. The ProMP Model with K-Means extension is performing especially bad, which can be seen in table 28 and table 27, both showing the means of the RMSE for the two years. This bad behavior can also be seen in tables 27 and 34, which display the sum for the RMSE. A reason for this behavior could be the determination of the right clusters, a solution could be the increase in clusters.

The household data is presumed to be less flawed, which results in better forecasts of ProMP and for the artificial neuronal network. As a result the ANN becomes competitive in this data sets, see tables 25 and 26. This could mean that the ANN has been overfitted by the optimization process while using the industrial data set. It is worth mentioning that results of the ANN and of the ProMP Model with the K-Means extension stay the same over the course of the day, while the pure ProMP approach and the ProMP Model with the Expectation Maximization extension, start inaccurate but gradually increase their accuracy.

In the overall performance comparison, see table 18, it becomes apparent that the average error for $\frac{2}{3}$ cross validation has the lowest value with ProMP for the household datasets. The extension of the ProMP approach with K-means further decreases the error. For the industrial data, the normal ProMP approach delivers lower errors then the Persistence approach and the ANN but does not beat the advanced Persistence model. Similar results can be seen in LOOC validation, where ProMP has in average lower errors than the normal Persistence approach but does not beat the advanced Persistence approach for the industrial datasets. For every Household data set ProMP delivers lower errors.

4.5.3 MAPE Comparison

The MAPE is not a perfect way to compare forecast and actual value with one another since the computation will become unstable if the actual value is near zero. As it can be seen in section 5, some values becomes infinity because the actual value to forecast has been zero or near zero.

However, the algorithm becomes more accurate over the course of a day, for example, in table 17 which shows the results of the LOOC validation and the results for the $\frac{2}{3}$ cross validation. The data has been produced by the ninth day of the first industrial data set (01/09/13), which is a Wednesday and has been further discussed in section 4.5.1.

The addition of Expectation Maximization to the ProMP Method leads to a error of 1905 % if no data of the current day has been seen. The overall performance can be enhanced if data of the current day is available and the forecast without data is discarded, see table 16. This behavior stays the same for most of the industrial dataset, which highlights another problem with MAPE: there is a lower bound but no upper bound. For the household data set, the Expectation Maximization extension behaves normal, with now catastrophic result for the forecast at midnight. However, it becomes obvious that all algorithms have trouble with the forecast of household data. Especially the persistence Model without the Mean extension. For the household with the highest annual sum, the ProMP with the K-Means extension behaves really well, while the normal ProMP does not work so well, which can be seen in table 35.

4.5.4 SMAPE Comparison

In the calculation of the SMAPE, the value is not divided by zero if the actual load value is zero. As a result the values for the SMAPE never become infinity and the is more robust near zeros [108]. However, as with the MAPE there is no upper bound, resulting in huge error values for the ProMP Method with the Expectation Maximization extension, which can be seen in table 46 and table 45. However, they are significantly smaller compared to the values of the MAPE. Still, this means that the ProMP Model with Expectation Maximization, in order to work well, needs data from the actual day to function. The superiority of the advanced persistence model is not significant in the industrial data if the SMAPE is taken into account.

Considering the household data, it becomes apparent that each approach is struggling with the data of the household with the smallest annual sum of 2219 kWh. This is similar to the comparison of the MAPE. However, the SMAPE makes it it possible to compare the Household with annual sum of 4647 kWh and the household with 5600 kWh because the values do not become infinite. For both households, the one with 4647 kWh and the one with 5600 kWh are the values of each forecast method similar. Only the ProMP approach with K-Means extension has a short lead,as can be seen in tables 42 and 43.

In the overall comparison in table 22 it is shown that ProMP with the K-Means extension is the best choice for the Households with 5600 kWh and 6500 kWh in the other cases the ProMP approaches suffers from the bad initial forecast. Overall only the advanced persistence model beats ProMP in the other data sets.

4.5.5 Comparison of the Computational Resourcefulness

In this section, the computational time, that is needed to train the model and to forecast a new electric load profile, is discussed. The calculation for the computational comparison has been carried out in *MATLAB* on a Linux computer, running a Ubuntu distribution, with 8GB of ram, an i7 with four cores and with a clock speed of 3.4 GHz.

Persistence Model

The simplicity of the *similar day* approach shines in this category, because the computational time needed for a forecast is minimal. If the used data structure to store the electric load profiles is a list, the time for the computations are static. The most time consuming step in the advanced version of the persistence model is the calculation of the mean.

Artificial Neural Network

The most time is consumed in the training phase of the artificial neural network which depends on the chosen number of neurons and the desired error on the training data. Traversing the artificial neural network for a new forecast is fairly quick, which is one of the advantages of neuronal networks, as can be seen in table 11.

Gaussian Process

The Gaussian process has a long training time, which highly depends on the chosen fit on the training data. Although a sparse model has been chosen, the computation can easily take 20 or more minutes for the training alone. The forecast phase begins after the training, where depending on the size of the forecast window, the forecast has to be repeated until the end of the day. For the chosen resolution of 15 minute time steps and the chosen forecast window, see section 4.2.4, the forecast has to be repeated 24 times for the complete day and correspondingly less if partial data has been available. The result for the

	total time
training	4.268 s ∼ 9.874 s
forecast	0.023 s ∼ 0.087 s

Table 11: Overview of the computational time need by the artificial neuronal network to optimize and predict one electric load profile

test day can be seen in table 13. The huge computational costs are the reason why the forecast over the course of a day could not be finished with the GP approach. The forecast of a complete day already takes up to 90 minutes and from comparing sample days it has been deduced that there is no benefit with this approach. As a result has forecast for each day for each data set, each containing a year, been declared not feasible.

	total time
optimization	1805.382 s
forecast	226.592 s

Table 13: Overview of the computational time need by the Gaussian Process to optimize and predict one electric load profile

Probabilistic Movement Primitives

The computational time for the basic ProMPs is low. The inversion for ridge regression taking the highest toll. The Expectation Maximization algorithm depends highly on the random initial variables that have been chosen. The K-Means algorithm highly depends on the number of cluster centers chosen, the data has been produced with 14 clusters in mind. With 28 centers, the time for k-means rises to 0.102s

	total time
ridge regression	0.142 s ∼ 0.214 s
unconditioned forecast	0 s
conditioned forecast	0.008 s
Expectation Maximization	11.52 s ∼ 82.835 s
K-means	0.093 s ∼ 0.037 s
EM for clustering	1.533 s ∼ 1.685 s

Table 15: Overview of the computational time need by ProMPs to optimize and predict one electric load profile including the time needed for additional Algorithms

4.6 Conlusion

ProMPs are able to forecast time series and can therefore be used to forecast electric load profiles. This new approach outperforms the other competing approaches in many cases and has similar results if the competition is not beaten. The different validation approaches showed the diversity and flexibility of the approach. The $\frac{2}{3}$ validation, for example, applied a faulty training data basis upon the contesters. Since the folds are drawn randomly, the flawless continuous of the training examples can not be guaranteed. Therefore, the days in the training sample are not equidistant. Consequently, the persistence model has been worse in these tests. While ProMPs, especially with the Expectation Maximization expansion, did not suffer as much. However, it became apparent that the quality of the forecast highly depends on the historic data. It hast also been shown that household data is more difficult to forecast, since the differences in the evaluation criteria are far wider than the difference in scale.

The extensions of the Probabilistic Movement Primitives have shown their advantages, the Expectation Maximization algorithm achieved better results in the $\frac{2}{3}$ cross validation. The reason for that is, that the algorithm suspects the data to be incomplete and tries to estimate the incomplete data. The other major extension of Probabilistic Movement Primitives has been the addition of K-Means algorithm to calculate the clusters of data beforehand. In smaller test on the industrial data the algorithm clusters similar days together, for example weekends and national holidays. Due to this property, K-Means has been applied to the training data in order to filter out seasonal influences. In some tests, (see section 4.5.3) the Probabilistic Movement Primitives, with the extension of K-Means, even achieved the highest scores. Difficulties with K-Means have been to determine the right amount of cluster centers. Too many would lead to overfitting and too few to worse results as without K-Means.

As it has been shown, the advantages are a probabilistic approach to electric load forecasting, which is still fast to compute and beats both complex methods, namely the artificial neural network and the Gaussian process, which has been explained in section 4.5. Even with added complexity like K-Means or Expectation Maximization, the computational time that is needed can still be considered low. It has to be mentioned that, while Gaussian processes are producing good results, the computational demand is too high for real world applications. Even with sparse methods, the time need is not competitive.

ANNs are still popular in the science community. They produce good results, while being straight forward to implement and usable. However, it became obvious that automatic optimization does not work so well in every circumstance. As a result, their portability from one use case to the next is limited. Shown by their not consistent performance across the different data sets.

The persistence model has been shown to be a competitive idea for electric load forecasting, since the electric load profiles are highly repetitive. The proneness to outliers from this model can be compensated by using the mean over historic data.

5 Future Work

From the Comparison in section 4.5, it becomes obvious that the forecast with Probabilistic Movement Primitives works and beats most of the competing methods, the artificial neuronal network, the persistence model and the advanced persistence model. The comparison with the GP has not been applicable in the fullest, since the computation time has been too high and the calculated outcomes have not been promising. However, there is still room to improve the forecast. Possible improvements could be achieved with *Seasonal Decomposition* to remove the seasonal completely [23]. Another possible way to improve the forecast would the use of *exponential smoothing* to smooth out the data before applying Probabilistic Movement Primitives [23].

There are different possibilities to proceed with the research of using ProMPs for electric load forecasting. The current approach can be enhanced to forecast broader time frames which results in the forecast of mid-term and long-term electric load profiles. Additionally, the current procedure can be changed to forecast peak loads. Another possibility would be to increase the time resolution, which would enable *very-short-term* electric load forecasting and maybe smooth out some of the spikes in the industrial profiles. However, that would increase the chance of outliers by faulty recordings [23]. By adding weather forecast, temperature or humidity forecast to the input data, the forecast could be enhanced. However, it would also introduce further uncertainties and remove the independence of the model from the location it is designed for, as weather forecasts have a high correlation with the place they are drawn from [23]. Another possibility is to take a look at forecasting of solar and wind power plants but the high influence of the weather could make this approach more difficult. However, the profiles for solar and wind plants should be more linear as influential weather phenomenons do not happen suddenly.

In general, electric load forecasts are used for decision making, by intelligent energy management systems, see section 1 and section 2, these base their decisions on the forecast. A more application-oriented approach would be to use the historic electric load and forecast the control laws for the energy management system directly. This would result in a reinforcement learning approach, which has been shown to be superior in comparison to the currently used model predictive control if used with stochastic and noisy data [109].

Since the liberation of energy markets around the world, the prediction of electric prices became interesting for consumers and producers alike [23]. The direct forecast of electrical energy prices [110], in respect to the historic load, can be another possibility for further research on ProMPs. Since probabilistic outputs deliver a natural form of risk management, the approaches may be transferred to different fields of application, such as the financial market. Research on ProMPs on the financial market may lead to new insights concerning their use. Other applications to apply ProMPs to are other types of time series forecast problems, such as forecasting stock exchanges [111].

Appendices

RMSE		
Approach	Mean Over All	Mean Without First Element
ProMP for LOOC	46.901	43.649
Persistance for LOOC	23.963	24.114
Persistance Advanced for LOOC	33.486	33.823
ProMP for 2/3	53.289	50.000
ProMP with EM for 2/3	91.173	80.279
ProMP with Clustering for 2/3	25.316	25.331
Persistance for 2/3	23.963	24.114
Persistance Advanced for 2/3	32.169	23.963
ANN for 2/3	37.840	37.555
GP for 2/3	106.199	108.568

MAPE		
Approach	Mean Over All	Mean Without First Element
ProMP for LOOC	16.066	13.578
Persistance for LOOC	9.539	9.629
Persistance Advanced for LOOC	13.776	13.970
ProMP for 2/3	26.175	22.969
ProMP with EM for 2/3	206.054	51.559
ProMP with Clustering for 2/3	9.929	10.016
Persistance for 2/3	9.539	9.629
Persistance Advanced for 2/3	12.812	9.539
ANN for 2/3	17.166	16.808
GP for 2/3	73.694	76.509

SMAPE		
Approach	Mean Over All	Mean Without First Element
ProMP for LOOC	18.507	16.440
Persistance for LOOC	9.242	9.322
Persistance Advanced for LOOC	12.638	12.801
ProMP for 2/3	22.004	19.828
ProMP with EM for 2/3	48.118	36.091
ProMP with Clustering for 2/3	9.122	9.184
Persistance for 2/3	9.242	9.322
Persistance Advanced for 2/3	11.826	9.242
ANN for 2/3	15.445	15.164
GP for 2/3	49.687	51.320

Table 16: The average results for the ninth day of the first industrial data set

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	43.435	8.550	11.648
1:00	18.106	8.716	12.040
2:00	12.693	8.895	12.330
3:00	13.834	9.088	12.780
4:00	12.716	9.348	13.180
5:00	10.097	9.684	13.270
6:00	16.660	9.958	13.522
7:00	20.709	10.169	13.946
8:00	10.418	9.993	14.587
9:00	10.858	9.812	15.190
10:00	12.277	9.874	16.011
11:00	10.993	10.384	16.810

$\frac{2}{3}$ cross validation			
Time	ProMP Clustering	ProMP EM	ProMP
0:00	8.973	1905.505	61.435
1:00	11.628	63.045	18.959
2:00	8.573	67.992	16.232
3:00	9.442	72.495	16.788
4:00	5.362	77.815	27.313
5:00	4.928	80.936	26.829
6:00	5.268	78.694	26.990
7:00	31.244	56.523	28.117
8:00	5.845	24.840	15.834
9:00	8.390	15.961	29.342
10:00	7.641	14.563	18.492
11:00	11.854	14.282	27.768

$\frac{2}{3}$ cross validation				
Time	GP	ANN	Persistence	Advanced Persistence
0:00	42.737	21.101	11.356	8.550
1:00	63.346	20.402	11.542	8.716
2:00	66.013	20.388	11.807	8.895
3:00	68.874	20.018	12.062	9.088
4:00	72.007	19.390	12.307	9.348
5:00	75.090	18.288	12.651	9.684
6:00	78.570	17.507	12.833	9.958
7:00	81.924	16.677	13.283	10.169
8:00	83.462	14.980	13.714	9.993
9:00	84.168	13.289	13.740	9.812
10:00	84.635	12.406	14.084	9.874
11:00	83.507	11.548	14.370	10.384

Table 17: The detailed MAPE results for the ninth day of the first industrial data set

$\frac{2}{3}$ Cross Validation			
Approach	ANN	Persistence	Persistence Advanced
Industrial Dataset 2013	41.919	38.719	31.260
Industrial Dataset 2014	59.327	49.718	40.009
Household 2219 kWh	0.384	0.523	0.435
Household 4647 kWh	0.452	0.580	0.495
Household 5600 kWh	0.648	0.899	0.758
Household 6500 kWh	0.530	0.651	0.550

$\frac{2}{3}$ Cross Validation			
Approach	ProMP	ProMP Clustering	ProMP EM
Industrial Dataset 2013	35.785	47.730	53.931
Industrial Dataset 2014	50.462	62.755	78.614
Household 2219 kWh	0.389	0.384	0.389
Household 4647 kWh	0.454	0.448	0.456
Household 5600 kWh	0.657	0.657	0.648
Household 6500 kWh	0.519	0.506	0.514

LOOC Validation			
Approach	Persistence	Persistence Advanced	ProMP
Industrial Dataset 2013	38.729	27.724	35.360
Industrial Dataset 2014	52.278	37.168	49.672
Household 2219 kWh	0.519	0.436	0.389
Household 4647 kWh	0.575	0.479	0.454
Household 5600 kWh	0.886	0.741	0.658
Household 6500 kWh	0.643	0.544	0.520

Table 18: Overall performance comparison for RMSE for all data sets

$\frac{2}{3}$ Cross Validation			
Approach	ANN	Persistence	Persistence Advanced
Industrial Dataset 2013	∞	24.025	17.865
Industrial Dataset 2014	∞	∞	17.905
Household 2219 kWh	88.616	149.521	105.056
Household 4647 kWh	61.090	70.508	57.530
Household 5600 kWh	65.774	$-\infty$	71.279
Household 6500 kWh	46.572	46.307	41.923

$\frac{2}{3}$ Cross Validation			
Approach	ProMP	ProMP Clustering	ProMP EM
Industrial Dataset 2013	22.366	27.862	142.975
Industrial Dataset 2014	27.689	25.964	220.022
Household 2219 kWh	80.920	80.673	83.208
Household 4647 kWh	53.306	56.313	52.874
Household 5600 kWh	63.325	68.398	67.330
Household 6500 kWh	40.009	40.610	39.622

LOOC Validation			
Approach	Persistence	Persistence Advanced	ProMP
, Industrial Dataset 2013	∞	15.500	22.086
Industrial Dataset 2014	∞	16.362	23.991
Household 2219 kWh	146.316	105.957	81.040
Household 4647 kWh	71.992	55.250	53.273
Household 5600 kWh	$-\infty$	70.333	62.686
Household 6500 kWh	46.284	40.850	40.448

Table 21: Overall performance comparison for MAPE for all data sets

$\frac{2}{3}$ Cross Validation			
Approach	ANN	Persistence	Persistence Advanced
Industrial Dataset 2013	24.690	21.856	17.855
Industrial Dataset 2014	26.843	21.700	17.734
Household 2219 kWh	70.935	66.796	65.992
Household 4647 kWh	51.579	49.743	48.256
Household 5600 kWh	53.526	55.451	54.979
Household 6500 kWh	38.601	35.723	34.643

$\frac{2}{3}$ Cross Validation			
Approach	ProMP	ProMP Clustering	ProMP EM
Industrial Dataset 2013	19.684	29.469	38.831
Industrial Dataset 2014	22.996	27.582	43.658
Household 2219 kWh	73.658	71.028	71.424
Household 4647 kWh	50.246	48.317	49.753
Household 5600 kWh	56.516	48.174	50.467
Household 6500 kWh	35.939	32.854	34.834

LOOC Validation			
Approach	Persistence	Persistence Advanced	ProMP
Industrial Dataset 2013	21.720	15.875	20.944
Industrial Dataset 2014	22.765	16.331	22.781
Household 2219 kWh	65.887	65.779	73.031
Household 4647 kWh	49.808	46.604	49.842
Household 5600 kWh	54.978	53.473	55.703
Household 6500 kWh	35.860	34.469	36.186

Table 22: Overall performance comparison for SMAPE for all data sets

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	0.466	0.575	0.486
1:00	0.474	0.584	0.494
2:00	0.484	0.596	0.504
3:00	0.495	0.609	0.516
4:00	0.507	0.624	0.528
5:00	0.520	0.639	0.541
6:00	0.532	0.654	0.553
7:00	0.541	0.668	0.565
8:00	0.546	0.676	0.572
9:00	0.553	0.686	0.581
10:00	0.559	0.695	0.589
11:00	0.568	0.703	0.596

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	0.465	0.460	0.454
1:00	0.473	0.468	0.463
2:00	0.483	0.478	0.470
3:00	0.494	0.489	0.481
4:00	0.506	0.501	0.493
5:00	0.518	0.513	0.505
6:00	0.530	0.526	0.517
7:00	0.539	0.534	0.523
8:00	0.543	0.537	0.526
9:00	0.551	0.546	0.535
10:00	0.558	0.553	0.546
11:00	0.567	0.561	0.559

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	0.497	0.582	0.491
1:00	0.484	0.592	0.499
2:00	0.493	0.604	0.509
3:00	0.503	0.618	0.521
4:00	0.514	0.632	0.534
5:00	0.526	0.648	0.547
6:00	0.537	0.664	0.559
7:00	0.547	0.677	0.571
8:00	0.553	0.685	0.578
9:00	0.562	0.696	0.588
10:00	0.569	0.705	0.596
11:00	0.575	0.713	0.603

Table 23: The mean RMSE results for the household with 6500 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	0.615	0.827	0.692
1:00	0.620	0.837	0.699
2:00	0.632	0.852	0.712
3:00	0.646	0.870	0.727
4:00	0.661	0.889	0.743
5:00	0.676	0.909	0.760
6:00	0.688	0.927	0.775
7:00	0.695	0.938	0.784
8:00	0.694	0.933	0.779
9:00	0.678	0.914	0.764
10:00	0.656	0.887	0.743
11:00	0.630	0.845	0.711

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	0.615	0.606	0.605
1:00	0.620	0.612	0.608
2:00	0.632	0.623	0.623
3:00	0.646	0.637	0.637
4:00	0.660	0.651	0.654
5:00	0.676	0.666	0.669
6:00	0.688	0.678	0.682
7:00	0.694	0.685	0.695
8:00	0.693	0.687	0.696
9:00	0.677	0.671	0.682
10:00	0.656	0.647	0.676
11:00	0.628	0.615	0.650

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	0.618	0.836	0.707
1:00	0.618	0.847	0.715
2:00	0.626	0.862	0.727
3:00	0.637	0.881	0.743
4:00	0.649	0.900	0.759
5:00	0.662	0.920	0.776
6:00	0.674	0.938	0.791
7:00	0.681	0.951	0.802
8:00	0.677	0.948	0.800
9:00	0.664	0.931	0.786
10:00	0.647	0.905	0.764
11:00	0.618	0.867	0.732

Table 24: The mean RMSE results for the household with 5600 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	0.417	0.527	0.438
1:00	0.423	0.536	0.446
2:00	0.432	0.548	0.456
3:00	0.442	0.560	0.466
4:00	0.451	0.572	0.476
5:00	0.459	0.583	0.484
6:00	0.466	0.591	0.492
7:00	0.471	0.597	0.497
8:00	0.473	0.601	0.500
9:00	0.471	0.598	0.498
10:00	0.470	0.598	0.498
11:00	0.467	0.596	0.495

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	0.417	0.417	0.412
1:00	0.423	0.424	0.419
2:00	0.432	0.434	0.428
3:00	0.442	0.444	0.438
4:00	0.451	0.453	0.447
5:00	0.459	0.461	0.454
6:00	0.466	0.468	0.461
7:00	0.471	0.474	0.465
8:00	0.474	0.477	0.467
9:00	0.472	0.474	0.465
10:00	0.472	0.474	0.464
11:00	0.469	0.472	0.460

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	0.424	0.531	0.452
1:00	0.426	0.540	0.460
2:00	0.433	0.551	0.470
3:00	0.441	0.564	0.481
4:00	0.448	0.576	0.490
5:00	0.455	0.587	0.499
6:00	0.462	0.597	0.507
7:00	0.466	0.605	0.513
8:00	0.468	0.607	0.517
9:00	0.467	0.604	0.515
10:00	0.467	0.604	0.516
11:00	0.463	0.599	0.513

Table 25: The mean RMSE results for the household with 4647 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	0.359	0.478	0.401
1:00	0.364	0.485	0.407
2:00	0.372	0.496	0.416
3:00	0.381	0.507	0.425
4:00	0.390	0.519	0.436
5:00	0.399	0.532	0.446
6:00	0.404	0.539	0.453
7:00	0.402	0.538	0.453
8:00	0.403	0.541	0.455
9:00	0.400	0.537	0.450
10:00	0.401	0.537	0.450
11:00	0.391	0.525	0.438

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	0.359	0.358	0.354
1:00	0.364	0.364	0.360
2:00	0.372	0.372	0.368
3:00	0.381	0.380	0.376
4:00	0.390	0.390	0.386
5:00	0.400	0.399	0.395
6:00	0.404	0.404	0.399
7:00	0.402	0.402	0.398
8:00	0.403	0.403	0.399
9:00	0.400	0.400	0.395
10:00	0.401	0.400	0.397
11:00	0.392	0.391	0.387

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	0.358	0.482	0.401
1:00	0.363	0.490	0.407
2:00	0.370	0.500	0.416
3:00	0.377	0.512	0.426
4:00	0.385	0.524	0.436
5:00	0.393	0.537	0.447
6:00	0.397	0.544	0.453
7:00	0.396	0.543	0.451
8:00	0.396	0.545	0.452
9:00	0.393	0.540	0.447
10:00	0.393	0.539	0.447
11:00	0.383	0.527	0.437

Table 26: The mean RMSE results for the household with 2219 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	65.480	48.226	34.537
1:00	48.774	49.088	35.092
2:00	51.203	49.986	35.675
3:00	51.939	50.867	36.233
4:00	57.382	51.880	36.872
5:00	59.196	52.944	37.559
6:00	52.767	54.102	38.289
7:00	55.981	55.222	38.996
8:00	41.409	55.798	39.396
9:00	37.295	55.133	39.011
10:00	35.930	53.407	37.978
11:00	38.703	50.686	36.381

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	64.812	171.537	74.141
1:00	50.435	68.963	55.062
2:00	52.405	72.126	59.832
3:00	52.990	74.985	59.977
4:00	56.729	76.902	60.705
5:00	61.053	78.056	61.984
6:00	52.800	78.198	63.736
7:00	58.064	78.027	63.092
8:00	41.531	73.962	58.931
9:00	38.834	66.261	59.663
10:00	36.703	56.400	65.714
11:00	39.181	47.955	70.223

$\frac{2}{3}$ cross validation				
Time	ANN	GP	Persistence	Advanced Persistence
0:00	56.187	117.531	45.922	37.187
1:00	56.241	112.917	46.707	37.780
2:00	57.339	113.104	47.535	38.387
3:00	58.454	113.170	48.334	38.999
4:00	59.663	112.341	49.241	39.683
5:00	60.943	109.225	50.206	40.418
6:00	62.144	103.181	51.240	41.180
7:00	63.384	94.935	52.269	41.924
8:00	63.505	89.595	52.856	42.362
9:00	61.401	84.862	52.426	41.998
10:00	58.089	93.321	51.005	40.917
11:00	54.569	83.159	48.878	39.275

Table 27: The mean RMSE results for the Industrial data set of 2014

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	57.861	36.496	26.639
1:00	34.666	36.987	26.886
2:00	35.264	37.539	27.176
3:00	36.468	38.032	27.422
4:00	38.454	38.676	27.732
5:00	37.426	39.305	28.048
6:00	40.026	39.917	28.330
7:00	38.059	40.472	28.550
8:00	29.516	40.658	28.576
9:00	26.232	40.007	28.306
10:00	24.837	38.975	27.815
11:00	25.513	37.688	27.209

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	57.593	140.461	64.952
1:00	34.699	55.446	39.517
2:00	35.169	57.979	44.390
3:00	37.209	59.729	48.260
4:00	39.326	61.237	49.807
5:00	37.925	60.711	48.757
6:00	39.919	58.084	49.596
7:00	39.592	50.760	46.013
8:00	29.170	32.559	45.013
9:00	26.810	24.657	45.592
10:00	25.607	22.853	43.995
11:00	26.402	22.700	46.872

$\frac{2}{3}$ cross validation				
Time	ANN	GP	Persistence	Advanced Persistence
0:00	41.568	44.858	36.523	29.997
1:00	39.636	43.344	37.001	30.294
2:00	40.403	44.255	37.538	30.635
3:00	41.153	45.217	38.045	30.937
4:00	41.981	46.233	38.641	31.302
5:00	42.755	48.350	39.255	31.668
6:00	43.465	49.329	39.845	31.994
7:00	44.114	50.221	40.378	32.277
8:00	44.164	49.608	40.585	32.341
9:00	43.016	48.318	39.992	31.946
10:00	41.318	46.983	39.015	31.254
11:00	39.456	45.342	37.805	30.470

Table 28: The mean RMSE results for the Industrial data set of 2013

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	170.607	210.528	177.990
1:00	173.591	213.857	180.854
2:00	177.200	218.095	184.489
3:00	181.228	223.035	188.688
4:00	185.624	228.389	193.233
5:00	190.302	233.931	197.940
6:00	194.830	239.419	202.544
7:00	198.109	244.419	206.705
8:00	199.676	247.513	209.336
9:00	202.267	251.189	212.691
10:00	204.579	254.466	215.557
11:00	207.977	257.138	218.299

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	170.136	168.270	166.132
1:00	173.119	171.276	169.346
2:00	176.707	174.871	172.106
3:00	180.696	178.886	175.986
4:00	185.069	183.235	180.262
5:00	189.736	187.848	184.790
6:00	194.143	192.483	189.133
7:00	197.273	195.305	191.520
8:00	198.780	196.560	192.655
9:00	201.614	199.679	195.900
10:00	204.099	202.287	199.709
11:00	207.430	205.465	204.761

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	181.729	212.833	179.751
1:00	177.149	216.524	182.708
2:00	180.463	220.994	186.424
3:00	184.152	226.017	190.671
4:00	188.110	231.440	195.271
5:00	192.373	237.139	200.044
6:00	196.611	242.867	204.726
7:00	200.141	247.760	208.870
8:00	202.484	250.665	211.450
9:00	205.654	254.772	215.067
10:00	208.148	257.897	218.069
11:00	210.358	260.896	220.875

Table 29: The $\sum$RMSE results for the household with 6500 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	225.091	302.543	253.095
1:00	227.044	306.250	256.006
2:00	231.429	311.924	260.546
3:00	236.402	318.536	266.096
4:00	241.744	325.513	271.981
5:00	247.406	332.833	278.205
6:00	251.873	339.212	283.532
7:00	254.498	343.458	286.891
8:00	253.893	341.595	285.254
9:00	248.238	334.417	279.672
10:00	240.247	324.708	271.765
11:00	230.471	309.303	260.061

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	225.179	221.834	221.352
1:00	227.010	224.039	222.634
2:00	231.400	228.095	228.020
3:00	236.359	232.960	233.246
4:00	241.720	238.186	239.441
5:00	247.398	243.694	245.030
6:00	251.884	248.163	249.656
7:00	254.081	250.705	254.428
8:00	253.500	251.520	254.685
9:00	247.899	245.606	249.693
10:00	239.925	236.628	247.460
11:00	229.746	225.000	238.009

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	226.272	305.870	258.855
1:00	226.284	309.901	261.664
2:00	229.295	315.512	266.156
3:00	233.115	322.287	271.799
4:00	237.441	329.337	277.830
5:00	242.230	336.652	284.148
6:00	246.620	343.189	289.631
7:00	249.412	347.888	293.469
8:00	247.785	347.112	292.625
9:00	242.958	340.794	287.650
10:00	236.718	331.288	279.614
11:00	226.196	317.464	267.810

Table 30: The $\sum$RMSE results for the household with 5600 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	152.663	192.811	160.416
1:00	154.868	196.189	163.235
2:00	158.213	200.427	166.764
3:00	161.884	205.018	170.590
4:00	165.161	209.316	174.112
5:00	167.962	213.225	177.271
6:00	170.520	216.486	179.964
7:00	172.318	218.382	181.746
8:00	173.273	219.828	183.071
9:00	172.493	218.721	182.149
10:00	172.159	218.867	182.179
11:00	170.956	218.180	180.994

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	152.715	152.718	150.712
1:00	154.898	155.241	153.194
2:00	158.239	158.672	156.555
3:00	161.908	162.403	160.221
4:00	165.186	165.716	163.441
5:00	168.023	168.667	166.165
6:00	170.648	171.304	168.634
7:00	172.568	173.332	170.288
8:00	173.614	174.455	170.947
9:00	172.883	173.371	170.064
10:00	172.602	173.658	169.673
11:00	171.619	172.853	168.182

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	155.050	194.258	165.462
1:00	156.007	197.591	168.313
2:00	158.574	201.792	171.940
3:00	161.390	206.393	175.884
4:00	164.026	210.711	179.480
5:00	166.547	214.726	182.766
6:00	169.035	218.448	185.729
7:00	170.617	221.275	187.926
8:00	171.428	222.246	189.242
9:00	170.990	221.168	188.549
10:00	171.052	220.928	188.892
11:00	169.421	219.096	187.779

Table 31: The $\sum$RMSE results for the household with 4647 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	131.244	174.794	146.697
1:00	133.231	177.555	148.992
2:00	136.159	181.374	152.213
3:00	139.349	185.528	155.716
4:00	142.820	190.038	159.515
5:00	146.196	194.606	163.341
6:00	147.782	197.115	165.685
7:00	147.058	197.010	165.629
8:00	147.444	198.018	166.355
9:00	146.304	196.383	164.720
10:00	146.749	196.509	164.604
11:00	143.238	192.212	160.289

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	131.309	131.060	129.710
1:00	133.278	133.062	131.664
2:00	136.203	136.001	134.577
3:00	139.387	139.213	137.749
4:00	142.851	142.706	141.200
5:00	146.224	146.105	144.537
6:00	147.805	147.796	146.043
7:00	147.062	147.274	145.619
8:00	147.400	147.581	146.037
9:00	146.252	146.447	144.725
10:00	146.748	146.378	145.139
11:00	143.448	143.054	141.675

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	131.127	176.387	146.828
1:00	132.749	179.270	149.122
2:00	135.276	183.071	152.353
3:00	138.005	187.226	155.855
4:00	140.922	191.783	159.660
5:00	143.895	196.395	163.461
6:00	145.439	199.123	165.764
7:00	144.855	198.711	165.227
8:00	145.069	199.300	165.569
9:00	143.762	197.802	163.730
10:00	143.712	197.300	163.749
11:00	140.079	192.788	159.865

Table 32: The $\sum$RMSE results for the household with 2219 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	23 179.999	17 072.079	12 226.085
1:00	17 265.916	17 377.141	12 422.536
2:00	18 125.782	17 694.868	12 628.787
3:00	18 386.511	18 006.945	12 826.621
4:00	20 313.322	18 365.485	13 052.649
5:00	20 955.552	18 742.173	13 295.995
6:00	18 679.491	19 152.201	13 554.243
7:00	19 817.446	19 548.657	13 804.610
8:00	14 658.656	19 752.654	13 946.190
9:00	13 202.470	19 516.997	13 809.918
10:00	12 719.267	18 905.937	13 444.280
11:00	13 700.725	17 942.712	12 878.740

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	22 943.436	60 723.989	26 245.885
1:00	17 853.958	24 413.068	19 491.955
2:00	18 551.270	25 532.607	21 180.505
3:00	18 758.456	26 544.813	21 231.909
4:00	20 082.132	27 223.275	21 489.730
5:00	21 612.678	27 631.971	21 942.237
6:00	18 691.339	27 681.943	22 562.543
7:00	20 554.594	27 621.455	22 334.505
8:00	14 702.107	26 182.724	20 861.444
9:00	13 747.401	23 456.567	21 120.712
10:00	12 993.000	19 965.602	23 262.593
11:00	13 870.214	16 975.991	24 858.975

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	19 890.181	16 256.482	13 164.186
1:00	19 909.149	16 534.351	13 374.169
2:00	20 298.183	16 827.234	13 589.125
3:00	20 692.568	17 110.282	13 805.584
4:00	21 120.743	17 431.200	14 047.850
5:00	21 573.782	17 773.028	14 307.964
6:00	21 999.112	18 139.121	14 577.765
7:00	22 437.818	18 503.056	14 841.185
8:00	22 480.889	18 711.189	14 996.232
9:00	21 736.130	18 558.893	14 867.195
10:00	20 563.631	18 055.670	14 484.524
11:00	19 317.539	17 302.822	13 903.186

Table 33: The $\sum$RMSE results for the industrial dataset of 2014

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	19 961.916	12 591.083	9190.337
1:00	11 959.769	12 760.354	9275.716
2:00	12 165.966	12 950.928	9375.793
3:00	12 581.372	13 120.883	9460.424
4:00	13 266.630	13 343.083	9567.527
5:00	12 912.059	13 560.170	9676.446
6:00	13 809.056	13 771.492	9773.932
7:00	13 130.240	13 962.833	9849.778
8:00	10 183.114	14 027.173	9858.658
9:00	9050.164	13 802.367	9765.525
10:00	8568.896	13 446.327	9596.190
11:00	8802.085	13 002.393	9387.265

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	19 869.665	48 459.030	22 408.360
1:00	11 971.061	19 128.831	13 633.480
2:00	12 133.419	20 002.892	15 314.597
3:00	12 836.984	20 606.507	16 649.871
4:00	13 567.611	21 126.715	17 183.433
5:00	13 084.062	20 945.196	16 821.194
6:00	13 771.938	20 038.859	17 110.626
7:00	13 659.200	17 512.125	15 874.348
8:00	10 063.492	11 232.775	15 529.538
9:00	9249.561	8506.837	15 729.209
10:00	8834.345	7884.312	15 178.233
11:00	9108.658	7831.395	16 170.969

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	14 341.007	12 600.283	10 349.016
1:00	13 674.270	12 765.178	10 451.352
2:00	13 939.072	12 950.725	10 568.951
3:00	14 197.928	13 125.557	10 673.388
4:00	14 483.376	13 331.216	10 799.341
5:00	14 750.438	13 542.809	10 925.580
6:00	14 995.560	13 746.660	11 037.885
7:00	15 219.256	13 930.504	11 135.400
8:00	15 236.604	14 001.903	11 157.536
9:00	14 840.685	13 797.237	11 021.462
10:00	14 254.599	13 460.091	10 782.667
11:00	13 612.414	13 042.766	10 512.158

Table 34: The $\sum$RMSE results for the industrial dataset of 2013

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	35.075	39.371	35.059
1:00	35.887	40.247	35.791
2:00	36.973	41.405	36.792
3:00	38.190	42.822	38.031
4:00	39.595	44.376	39.410
5:00	41.091	45.971	40.825
6:00	42.338	47.561	42.173
7:00	43.100	48.994	43.308
8:00	43.281	49.997	43.957
9:00	43.169	50.774	44.470
10:00	43.011	51.592	44.956
11:00	43.659	52.303	45.431

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	59.361	34.196	35.224
1:00	58.846	35.023	36.119
2:00	60.050	36.089	36.901
3:00	61.375	37.324	38.110
4:00	62.961	38.718	39.501
5:00	64.840	40.183	41.008
6:00	65.977	41.430	42.287
7:00	66.157	42.262	42.965
8:00	66.556	42.441	43.176
9:00	65.702	42.532	43.571
10:00	64.533	42.425	43.825
11:00	63.541	42.839	44.635

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	44.226	39.513	35.905
1:00	41.000	40.369	36.673
2:00	41.766	41.506	37.721
3:00	42.705	42.894	39.000
4:00	44.920	44.429	40.427
5:00	46.084	46.025	41.891
6:00	47.310	47.581	43.265
7:00	48.958	48.991	44.445
8:00	49.465	49.939	45.101
9:00	50.233	50.753	45.745
10:00	50.817	51.500	46.230
11:00	51.378	52.184	46.669

Table 35: The mean MAPE results for the household with 6500 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	58.643	-∞	65.219
1:00	58.244	-∞	65.664
2:00	59.446	-∞	66.757
3:00	60.775	-∞	68.221
4:00	62.338	-∞	69.801
5:00	64.157	-∞	71.675
6:00	65.248	-∞	73.070
7:00	65.606	-∞	73.875
8:00	65.969	-∞	73.678
9:00	65.062	-∞	72.905
10:00	63.817	-∞	72.133
11:00	62.930	-∞	70.991

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	59.361	61.964	64.131
1:00	58.846	61.935	63.477
2:00	60.050	63.128	64.849
3:00	61.375	64.713	66.356
4:00	62.961	66.518	68.182
5:00	64.840	68.644	70.371
6:00	65.977	70.033	71.755
7:00	66.157	71.138	72.181
8:00	66.556	75.308	64.688
9:00	65.702	71.849	72.838
10:00	64.533	69.207	71.673
11:00	63.541	63.522	70.275

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	62.466	-∞	65.916
1:00	61.953	-∞	66.405
2:00	62.900	-∞	67.517
3:00	63.657	-∞	69.005
4:00	65.135	-∞	70.703
5:00	66.691	-∞	72.652
6:00	68.032	-∞	74.046
7:00	68.794	-∞	74.822
8:00	68.671	-∞	74.806
9:00	68.038	-∞	74.213
10:00	67.187	-∞	73.225
11:00	65.764	-∞	72.043

Table 36: The mean MAPE results for the household with 5600 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	49.384	64.243	50.145
1:00	49.630	65.409	50.912
2:00	50.967	67.198	52.212
3:00	52.548	69.280	53.742
4:00	53.899	71.230	55.104
5:00	54.565	72.770	56.120
6:00	54.998	74.025	56.855
7:00	55.435	75.184	57.505
8:00	55.388	75.955	57.872
9:00	54.690	75.953	57.655
10:00	54.195	76.147	57.508
11:00	53.580	76.511	57.371

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	49.307	48.737	51.480
1:00	49.594	49.274	52.045
2:00	50.933	50.663	53.504
3:00	52.534	52.300	55.244
4:00	53.865	53.649	56.733
5:00	54.486	54.273	57.500
6:00	54.965	54.727	58.128
7:00	55.433	55.123	58.737
8:00	55.476	55.088	58.856
9:00	54.862	54.321	58.303
10:00	54.356	53.592	57.811
11:00	53.860	52.742	57.409

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	57.832	63.204	52.228
1:00	57.866	64.272	52.994
2:00	58.630	65.933	54.323
3:00	59.495	67.930	55.923
4:00	60.345	69.827	57.330
5:00	61.101	71.227	58.383
6:00	61.971	72.441	59.157
7:00	62.787	73.637	59.867
8:00	63.239	74.455	60.301
9:00	63.292	74.275	60.050
10:00	63.446	74.305	59.927
11:00	63.074	74.594	59.872

Table 37: The mean MAPE results for the household with 4647 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	75.165	130.756	95.322
1:00	76.224	133.567	97.249
2:00	78.044	137.381	99.972
3:00	80.063	141.492	102.934
4:00	82.390	146.178	106.304
5:00	83.933	150.350	109.328
6:00	84.058	152.840	111.060
7:00	83.502	153.444	111.395
8:00	83.081	153.775	111.665
9:00	82.433	152.507	110.047
10:00	82.284	152.682	109.136
11:00	81.305	150.824	107.067

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	74.872	76.367	74.471
1:00	75.982	77.728	75.719
2:00	77.824	79.794	77.644
3:00	79.833	82.020	79.730
4:00	82.165	84.598	82.129
5:00	83.712	86.301	83.727
6:00	83.939	86.638	83.993
7:00	83.367	86.070	83.532
8:00	83.063	85.709	81.517
9:00	82.440	84.907	82.273
10:00	82.419	84.713	82.174
11:00	81.424	83.648	81.169

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	83.242	133.143	94.476
1:00	84.314	136.037	96.385
2:00	85.871	140.068	99.097
3:00	87.608	144.231	101.991
4:00	89.666	149.107	105.352
5:00	91.229	153.419	108.249
6:00	91.524	156.067	110.133
7:00	90.706	157.092	110.440
8:00	90.538	158.153	110.522
9:00	89.970	156.745	108.802
10:00	90.009	156.370	108.417
11:00	88.712	153.823	106.811

Table 38: The mean MAPE results for the household with 2219 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	33.625	∞	15.126
1:00	20.553	∞	15.367
2:00	21.569	∞	15.614
3:00	22.193	24.543	15.847
4:00	26.465	25.183	16.115
5:00	31.238	25.878	16.420
6:00	14.818	26.694	16.783
7:00	42.961	27.425	17.100
8:00	19.318	27.692	17.270
9:00	17.133	27.404	17.203
10:00	17.610	26.653	16.955
11:00	20.411	25.494	16.544

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	32.822	2151.051	32.529
1:00	21.219	42.828	26.100
2:00	21.865	46.976	28.520
3:00	22.460	50.927	25.522
4:00	24.495	53.328	24.116
5:00	33.566	54.155	25.545
6:00	18.204	53.231	26.087
7:00	78.119	52.222	27.483
8:00	19.306	45.697	25.167
9:00	17.956	37.151	18.790
10:00	17.643	29.021	16.551
11:00	24.618	23.673	35.154

$\frac{2}{3}$ cross validation				
Time	ANN	GP	Persistence	Advanced Persistence
0:00	∞	∞	∞	16.551
1:00	27.064	43.496	∞	16.809
2:00	27.743	43.669	∞	17.071
3:00	28.364	43.748	23.109	17.330
4:00	29.066	43.777	23.652	17.631
5:00	29.804	43.556	24.248	17.976
6:00	30.360	43.089	24.943	18.356
7:00	30.963	42.496	25.596	18.705
8:00	30.720	42.319	25.893	18.907
9:00	29.406	42.200	25.735	18.842
10:00	27.799	47.280	25.151	18.567
11:00	26.190	46.460	24.273	18.100

Table 39: The mean MAPE results for the Industrial data set of 2014

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	37.010	∞	14.904
1:00	19.276	∞	15.014
2:00	19.555	∞	15.145
3:00	19.557	23.338	15.262
4:00	22.665	23.894	15.405
5:00	21.941	24.434	15.579
6:00	25.914	24.960	15.739
7:00	29.838	25.420	15.838
8:00	18.625	25.616	15.859
9:00	18.083	25.370	15.829
10:00	16.071	24.975	15.764
11:00	16.502	24.392	15.663

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	36.180	1316.042	44.484
1:00	19.057	43.853	22.263
2:00	19.449	47.945	27.444
3:00	32.264	50.862	28.748
4:00	23.316	53.402	28.452
5:00	23.787	52.334	25.214
6:00	21.801	48.518	31.143
7:00	25.984	39.816	20.742
8:00	17.404	21.468	27.047
9:00	16.381	14.913	26.129
10:00	15.596	13.340	26.097
11:00	17.176	13.206	26.577

$\frac{2}{3}$ cross validation				
Time	ANN	GP	Persistence	Advanced Persistence
0:00	∞	∞	22.002	17.048
1:00	23.338	18.272	22.389	17.209
2:00	24.006	18.887	22.818	17.394
3:00	24.676	19.507	23.275	17.578
4:00	25.386	20.189	23.787	17.786
5:00	26.025	21.349	24.311	18.011
6:00	26.610	21.965	24.828	18.214
7:00	27.112	22.527	25.270	18.357
8:00	27.078	22.121	25.447	18.409
9:00	26.256	21.458	25.190	18.308
10:00	25.214	20.863	24.761	18.134
11:00	24.137	20.167	24.225	17.932

Table 40: The mean MAPE results for the Industrial data set of 2013

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	31.246	31.077	29.870
1:00	32.036	31.696	30.460
2:00	33.001	32.518	31.261
3:00	34.055	33.529	32.251
4:00	35.264	34.627	33.347
5:00	36.566	35.738	34.470
6:00	37.658	36.822	35.513
7:00	38.343	37.757	36.368
8:00	38.616	38.379	36.864
9:00	38.784	38.838	37.272
10:00	38.967	39.403	37.732
11:00	39.695	39.942	38.218

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	31.092	30.095	28.569
1:00	31.880	30.882	29.372
2:00	32.840	31.809	30.052
3:00	33.880	32.843	30.954
4:00	35.073	34.013	32.002
5:00	36.362	35.264	33.144
6:00	37.430	36.336	34.064
7:00	38.050	36.842	34.393
8:00	38.232	36.889	34.364
9:00	38.435	37.245	34.877
10:00	38.637	37.514	35.697
11:00	39.362	38.276	36.760

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	37.223	31.064	30.008
1:00	35.682	31.675	30.624
2:00	36.246	32.479	31.453
3:00	36.914	33.458	32.452
4:00	37.609	34.526	33.560
5:00	38.490	35.636	34.692
6:00	39.349	36.688	35.716
7:00	39.911	37.563	36.556
8:00	40.240	38.094	36.958
9:00	40.497	38.612	37.428
10:00	40.516	39.142	37.883
11:00	40.534	39.744	38.388

Table 41: The mean SMAPE results for the household with 6500 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	51.989	52.192	50.284
1:00	52.055	52.508	50.655
2:00	53.002	53.099	51.332
3:00	54.247	53.884	52.220
4:00	55.278	54.413	53.120
5:00	56.818	55.396	54.279
6:00	57.934	56.242	55.171
7:00	58.447	56.901	55.709
8:00	58.583	57.035	55.619
9:00	57.767	56.723	55.135
10:00	56.667	56.146	54.518
11:00	55.644	55.196	53.630

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	52.846	47.755	44.737
1:00	52.837	47.809	44.847
2:00	53.769	48.249	45.433
3:00	54.969	48.981	46.524
4:00	56.109	49.727	47.838
5:00	57.688	50.955	48.973
6:00	58.800	51.695	49.252
7:00	59.265	51.844	49.743
8:00	59.512	52.321	49.986
9:00	58.559	52.324	49.699
10:00	57.431	52.293	50.698
11:00	56.405	51.654	50.364

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	54.466	52.746	51.657
1:00	53.112	53.058	52.032
2:00	52.758	53.585	52.667
3:00	52.416	54.367	53.576
4:00	52.536	54.885	54.568
5:00	53.089	55.891	55.797
6:00	53.719	56.716	56.730
7:00	54.018	57.291	57.239
8:00	54.183	57.385	57.213
9:00	54.148	57.174	56.770
10:00	54.039	56.572	56.176
11:00	53.830	55.741	55.324

Table 42: The mean SMAPE results for the household with 5600 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	45.558	45.470	42.592
1:00	46.053	46.080	43.168
2:00	47.305	47.069	44.161
3:00	48.755	48.258	45.338
4:00	50.121	49.422	46.427
5:00	50.863	50.290	47.179
6:00	51.369	50.925	47.737
7:00	51.889	51.550	48.264
8:00	52.061	51.994	48.617
9:00	51.725	52.090	48.584
10:00	51.445	52.214	48.596
11:00	50.966	52.332	48.584

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	45.807	45.427	44.040
1:00	46.306	46.128	44.746
2:00	47.561	47.394	45.974
3:00	49.038	48.866	47.407
4:00	50.421	50.232	48.708
5:00	51.179	50.784	49.355
6:00	51.715	51.143	49.790
7:00	52.313	51.693	50.241
8:00	52.587	51.758	50.338
9:00	52.287	51.199	50.038
10:00	52.036	51.271	49.780
11:00	51.698	51.142	49.387

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	52.006	45.684	44.074
1:00	51.202	46.195	44.639
2:00	51.128	47.083	45.642
3:00	51.074	48.223	46.863
4:00	51.056	49.335	47.949
5:00	51.105	50.158	48.732
6:00	51.406	50.843	49.357
7:00	51.859	51.544	49.942
8:00	52.039	51.973	50.372
9:00	52.059	52.005	50.462
10:00	52.065	51.921	50.522
11:00	51.944	51.955	50.522

Table 43: The mean SMAPE results for the household with 4647 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	67.664	62.013	61.065
1:00	68.771	62.749	61.961
2:00	70.331	63.766	63.191
3:00	71.971	64.794	64.490
4:00	73.875	66.051	66.006
5:00	75.283	67.066	67.185
6:00	75.468	67.639	67.902
7:00	75.217	67.646	67.954
8:00	75.099	67.656	67.990
9:00	74.589	67.390	67.616
10:00	74.591	67.219	67.375
11:00	73.514	66.652	66.616

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	68.141	66.047	65.644
1:00	69.310	67.073	66.842
2:00	70.916	68.622	68.422
3:00	72.570	70.249	70.037
4:00	74.497	72.136	71.910
5:00	75.914	73.509	73.204
6:00	76.169	73.823	73.291
7:00	75.825	73.760	73.170
8:00	75.707	73.572	73.172
9:00	75.241	73.295	72.815
10:00	75.327	72.913	71.998
11:00	74.282	72.093	71.827

$\frac{2}{3}$ cross validation			
Time	ANN	Persistence	Advanced Persistence
0:00	71.224	62.755	61.345
1:00	71.007	63.530	62.248
2:00	71.104	64.576	63.466
3:00	71.202	65.607	64.742
4:00	71.072	66.916	66.274
5:00	71.205	67.954	67.414
6:00	71.313	68.609	68.143
7:00	71.295	68.666	68.219
8:00	71.130	68.724	68.227
9:00	70.747	68.544	67.702
10:00	70.289	68.184	67.428
11:00	69.637	67.493	66.700

Table 44: The mean SMAPE results for the household with 2219 kWh

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	33.379	20.589	15.122
1:00	21.255	21.041	15.365
2:00	22.063	21.518	15.611
3:00	22.687	21.930	15.834
4:00	25.386	22.448	16.101
5:00	26.876	23.008	16.405
6:00	24.509	23.671	16.763
7:00	27.079	24.241	17.076
8:00	18.628	24.399	17.231
9:00	16.785	24.107	17.136
10:00	16.484	23.537	16.873
11:00	18.239	22.692	16.448

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	32.758	180.237	34.968
1:00	21.869	29.682	24.100
2:00	22.650	31.660	25.965
3:00	23.133	33.784	25.430
4:00	24.676	35.217	25.681
5:00	28.064	35.890	26.562
6:00	24.171	35.646	26.792
7:00	27.672	35.322	27.024
8:00	18.358	32.218	26.088
9:00	17.439	28.214	26.371
10:00	16.775	24.330	30.467
11:00	18.383	21.698	31.537

$\frac{2}{3}$ cross validation				
Time	ANN	GP	Persistence	Advanced Persistence
0:00	25.196	81.570	19.752	16.431
1:00	25.064	46.715	20.140	16.686
2:00	25.654	46.564	20.555	16.942
3:00	26.189	46.245	20.915	17.181
4:00	26.810	45.795	21.358	17.477
5:00	27.473	44.873	21.842	17.815
6:00	28.020	43.450	22.411	18.184
7:00	28.584	41.726	22.929	18.521
8:00	28.556	40.616	23.128	18.702
9:00	27.833	39.493	22.966	18.616
10:00	26.884	46.103	22.523	18.352
11:00	25.856	43.366	21.884	17.906

Table 45: The mean SMAPE results for the Industrial data set of 2014

LOOC			
Time	ProMP	Persistence	Advanced Persistence
0:00	36.218	20.101	15.256
1:00	19.865	20.420	15.378
2:00	20.189	20.783	15.521
3:00	21.566	21.122	15.638
4:00	22.497	21.567	15.790
5:00	21.641	22.005	15.971
6:00	23.554	22.424	16.140
7:00	22.756	22.755	16.246
8:00	17.097	22.841	16.260
9:00	15.268	22.593	16.207
10:00	14.631	22.249	16.114
11:00	16.043	21.786	15.981

$\frac{2}{3}$ cross validation			
Time	ProMP	ProMP EM	ProMP Clustering
0:00	36.075	169.572	40.734
1:00	19.680	31.076	21.642
2:00	20.031	33.209	24.487
3:00	21.672	34.954	26.949
4:00	22.989	36.532	27.682
5:00	21.980	36.378	27.490
6:00	23.148	34.647	28.375
7:00	23.466	30.060	25.942
8:00	16.789	18.869	25.202
9:00	15.548	14.250	30.321
10:00	15.016	13.217	27.403
11:00	−0.187	13.210	47.401

$\frac{2}{3}$ cross validation				
Time	ANN	GP	Persistence	Advanced Persistence
0:00	24.023	25.334	20.271	17.086
1:00	22.580	17.562	20.583	17.246
2:00	23.204	18.143	20.930	17.428
3:00	23.815	18.729	21.277	17.593
4:00	24.483	19.379	21.681	17.793
5:00	25.102	20.487	22.108	18.010
6:00	25.685	21.091	22.521	18.201
7:00	26.157	21.644	22.843	18.327
8:00	26.231	21.400	22.942	18.362
9:00	25.713	20.921	22.726	18.253
10:00	25.026	20.477	22.400	18.080
11:00	24.263	19.965	21.985	17.875

Table 46: The mean SMAPE results for the Industrial data set of 2013

Date	Day of the Week	Cluster	Day of the Year
20130107	Monday	2	7
20130108	Tuesday	1	8
20130110	Thursday	6	10
20130111	Friday	11	11
20130112	Saturday	5	12
20130113	Sunday	5	13
20130115	Tuesday	1	15
20130116	Wednesday	1	16
20130117	Thursday	1	17
20130118	Friday	11	18
20130119	Saturday	5	19
20130120	Sunday	5	20
20130125	Friday	11	25
20130127	Sunday	5	27
20130128	Monday	1	28
20130129	Tuesday	2	29
20130130	Wednesday	2	30
20130131	Thursday	1	31
20130201	Friday	11	32
20130203	Sunday	5	34
20130205	Tuesday	1	36
20130207	Thursday	11	38
20130211	Monday	1	42
20130212	Tuesday	1	43
20130214	Thursday	1	45
20130217	Sunday	5	48
20130219	Tuesday	2	50
20130220	Wednesday	2	51
20130221	Thursday	1	52
20130222	Friday	11	53
20130223	Saturday	5	54
20130225	Monday	1	56
20130227	Wednesday	11	58

Table 47: K-Means Result for January and February 2013 for the Industrial Dataset

Average results of MAPE for 0:00	
Dataset	Available GP
Industrial Dataset 2013	∞
Industrial Dataset 2014	∞
Household 2219 kWh	-
Household 4647 kWh	-
Household 5600 kWh	61.031
Household 6500 kWh	41.834

Average results of SMAPE for 0:00	
Dataset	Available GP
Industrial Dataset 2013	25.334
Industrial Dataset 2014	81.570
Household 2219 kWh	-
Household 4647 kWh	-
Household 5600 kWh	53.692
Household 6500 kWh	38.012

Average results of RMSE for 0:00	
Dataset	Available GP
Industrial Dataset 2013	44.858
Industrial Dataset 2014	117.531
Household 2219 kWh	-
Household 4647 kWh	-
Household 5600 kWh	0.631
Household 6500 kWh	0.483

Average results of $\sum$RMSE for 0:00	
Dataset	Available GP
Industrial Dataset 2013	15 475.965
Industrial Dataset 2014	40 901.856
Household 2219 kWh	-
Household 4647 kWh	-
Household 5600 kWh	224.326
Household 6500 kWh	176.890

Table 52: Available average Gaussian Process results for the 0:00 forecast

List of Tables

List of Figures

References

[1] International Energy Agency (IEA), *World Energy Outlook*. Paris: International Energy Agency, 2014 ed., 2014.

[2] A. Laouafi, M. Mordjaoui, and D. Dib, "One-hour ahead electric load and wind-solar power generation forecasting using artificial neural network," in *IREC2015 The Sixth International Renewable Energy Congress*, pp. 1–6, IEEE, mar 2015.

[3] S. S. Reddy, A. R. Abhyankar, and P. R. Bijwe, "Market clearing for a wind-thermal power system incorporating wind generation and load forecast uncertainties," in *2012 IEEE Power and Energy Society General Meeting*, pp. 1–8, IEEE, jul 2012.

[4] X. Sun, X. Wang, J. Wu, and Y. Liu, "Hierarchical sparse learning for load forecasting in cyber-physical energy systems," in *2013 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 533–538, IEEE, may 2013.

[5] G. Bruni, S. Cordiner, V. Mulone, V. Rocco, and F. Spagnolo, "A study on the energy management in domestic micro-grids based on Model Predictive Control strategies," *Energy Conversion and Management*, vol. 102, pp. 50–58, sep 2015.

[6] P. Nema, R. Nema, and S. Rangnekar, "A current and future state of art development of hybrid energy system using wind and PV-solar: A review," *Renewable and Sustainable Energy Reviews*, vol. 13, pp. 2096–2103, oct 2009.

[7] M. Abuella and B. Chowdhury, "Solar power probabilistic forecasting by using multiple linear regression analysis," in *SoutheastCon 2015*, pp. 1–5, IEEE, apr 2015.

[8] S. Cros, O. Liandrat, N. Sebastien, and N. Schmutz, "Extracting cloud motion vectors from satellite images for solar power forecasting," in *2014 IEEE Geoscience and Remote Sensing Symposium*, pp. 4123–4126, IEEE, jul 2014.

[9] M. Wittmann, H. Breitkreuz, M. Schroedter-Homscheidt, and M. Eck, "Case Studies on the Use of Solar Irradiance Forecast for Optimized Operation Strategies of Solar Thermal Power Plants," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 1, pp. 18–27, mar 2008.

[10] Y. Zhang, M. Beaudin, R. Taheri, H. Zareipour, and D. Wood, "Day-Ahead Power Output Forecasting for Small-Scale Solar Photovoltaic Electricity Generators," *IEEE Transactions on Smart Grid*, vol. 6, pp. 2253–2262, sep 2015.

[11] V. P. Singh, V. Vijay, M. Siddhartha Bhatt, and D. K. Chaturvedi, "Generalized neural network methodology for short term solar power forecasting," in *2013 13th International Conference on Environment and Electrical Engineering (EEEIC)*, pp. 58–62, IEEE, nov 2013.

[12] G. Li, J. Shi, and J. Zhou, "Bayesian adaptive combination of short-term wind speed forecasts from neural network models," *Renewable Energy*, vol. 36, pp. 352–359, jan 2011.

[13] P. Lauret, C. Voyant, T. Soubdhan, M. David, and P. Poggi, "A benchmarking of machine learning techniques for solar radiation forecasting in an insular context," *Solar Energy*, vol. 112, pp. 446–457, feb 2015.

[14] H. Mori and E. Kurata, "Application of Gaussian Process to wind speed forecasting for wind power generation," in *2008 IEEE International Conference on Sustainable Energy Technologies*, pp. 956–959, IEEE, nov 2008.

[15] S. Salcedo-Sanz, C. Casanova-Mateo, J. Munoz-Mari, and G. Camps-Valls, "Prediction of Daily Global Solar Irradiation Using Temporal Gaussian Processes," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, pp. 1936–1940, nov 2014.

[16] C. S. Ioakimidis, S. Lopez, K. N. Genikomsakis, P. Rycerski, and D. Simic, "Solar production forecasting based on irradiance forecasting using artificial neural networks," in *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*, pp. 8121–8126, IEEE, nov 2013.

[17] G. Sideratos and N. Hatziargyriou, "An Advanced Statistical Method for Wind Power Forecasting," *IEEE Transactions on Power Systems*, vol. 22, pp. 258–265, feb 2007.

[18] J. Armstrong and F. Collopy, "Error measures for generalizing about forecasting methods: Empirical comparisons," *International Journal of Forecasting*, vol. 8, pp. 69–80, jun 1992.

[19] Christian Höft, *Bewertung von Verfahren zur Prognose der elektrischen Last eine empirische Analyse*. PhD thesis, Technische Universität Dresden, 2004.

[20] J. W. Taylor, L. M. de Menezes, and P. E. McSharry, "A comparison of univariate methods for forecasting electricity demand up to a day ahead," *International Journal of Forecasting*, vol. 22, pp. 1–16, jan 2006.

[21] G. Heinemann, D. Nordmian, and E. Plant, "The Relationship Between Summer Weather and Summer Loads - A Regression Analysis," *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-85, pp. 1144–1154, nov 1966.

[22] H. Hahn, S. Meyer-Nieberg, and S. Pickl, "Electric load forecasting methods: Tools for decision making," *European Journal of Operational Research*, vol. 199, pp. 902–907, dec 2009.

[23] R. Weron, *Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach*. The Wiley Finance Series, Wiley, 2007.

[24] T. Hong, M. Gui, M. E. Baran, and H. L. Willis, "Modeling and forecasting hourly electric load by multiple linear regression with interactions," *IEEE PES General Meeting, PES 2010*, pp. 1–8, 2010.

[25] J. R. Cancelo, A. Espasa, and R. Grafe, "Forecasting the electricity load from one day to one week ahead for the Spanish system operator," *International Journal of Forecasting*, vol. 24, no. 4, pp. 588–602, 2008.

[26] S. Sachdeva and C. M. Verma, "Load Forecasting using Fuzzy Methods," in *2008 Joint International Conference on Power System Technology and IEEE Power India Conference*, pp. 1–4, IEEE, oct 2008.

[27] K. Metaxiotis, a. Kagiannas, D. Askounis, and J. Psarras, "Artificial intelligence in short term electric load forecasting: A state-of-the-art survey for the researcher," *Energy Conversion and Management*, vol. 44, no. 9, pp. 1525–1534, 2003.

[28] J. W. Taylor, "An evaluation of methods for very short-term load forecasting using minute-by-minute British data," *International Journal of Forecasting*, vol. 24, no. 4, pp. 645–658, 2008.

[29] M. Ghiassi, D. K. Zimbra, and H. Saidane, "Medium term system load forecasting with a dynamic artificial neural network model," *Electric Power Systems Research*, vol. 76, no. 5, pp. 302–316, 2006.

[30] T. Hong, J. Wilson, and J. Xie, "Long Term Probabilistic Load Forecasting and Normalization With Hourly Information," *IEEE Transactions on Smart Grid*, vol. 5, pp. 456–462, jan 2014.

[31] H. Willis and J. Northcote-Green, "Spatial electric load forecasting: A tutorial review," *Proceedings of the IEEE*, vol. 71, no. 2, pp. 232–253, 1983.

[32] Z. Zhang and S. Ye, "Long Term Load Forecasting and Recommendations for China Based on Support Vector Regression," in *2011 International Conference on Information Management, Innovation Management and Industrial Engineering*, vol. 3, pp. 597–602, IEEE, nov 2011.

[33] B.-j. Chen, M.-w. Chang, C.-j. Lin, and I. Engineering, "Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001," pp. 1–7, 2001.

[34] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, "Probabilistic Movement Primitives," in *Advances in Neural Information Processing Systems 26* (C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, eds.), pp. 2616–2624, Curran Associates, Inc., 2013.

[35] Li qiang Hou, Shan lin Yang, Xiao jia Wang, and Hui zhou Liu, "Load forecasting based on weighted kernel partial least squares algorithm in smart grid," in *IET International Conference on Information Science and Control Engineering 2012 (ICISCE 2012)*, pp. 2.47–2.47, Institution of Engineering and Technology, 2012.

[36] Vdew, "Umsetzung der Analytischen Lastprofilverfahren - Step-by-step," tech. rep., 2000.

[37] N. Lübke, A. Holst, and R. Tolzmann, "Entwicklung eines synthetischen Jahreslastprofils für Haushaltsabnahme aus Elt.-Versorgungsnetzen," *12. Symposium Maritime Elektrotechnik, Elektronik und Informationstechnik*, p. 6, 2007.

[38] G. Mbamalu and M. El-Hawary, "Load forecasting via suboptimal seasonal autoregressive models and iteratively reweighted least squares estimation," *IEEE Transactions on Power Systems*, vol. 8, no. 1, pp. 343–348, 1993.

[39] D. Park, M. El-Sharkawi, R. Marks, L. Atlas, and M. Damborg, "Electric load forecasting using an artificial neural network," *IEEE Transactions on Power Systems*, vol. 6, no. 2, pp. 442–449, 1991.

[40] P. Gonzalez and J. Zamarreno, "Prediction of hourly energy consumption in buildings based on a feedback artificial neural network," *Energy and Buildings*, vol. 37, pp. 595–601, jun 2005.

[41] H. S. Hippert and J. W. Taylor, "An evaluation of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting.," *Neural networks : the official journal of the International Neural Network Society*, vol. 23, pp. 386–95, apr 2010.

[42] K. P. Murphy, *Machine Learning A Probabilistic Perspective*. Cambridge, Mass.: MIT Press, 2012.

[43] M. Ghofrani, M. Hassanzadeh, M. Etezadi-Amoli, and M. S. Fadali, "Smart meter based short-term load forecasting for residential customers," *NAPS 2011 - 43rd North American Power Symposium*, pp. 13–17, 2011.

[44] W. Hong, "Electric load forecasting by support vector model," *Applied Mathematical Modelling*, vol. 33, pp. 2444–2454, may 2009.

[45] V. N. V. Bernhard E. Boser, Isabelle M. Guyon, "A Training Algorithm for Optimal Margin Classifiers," *ACM Workshop*, no. 5, pp. 144 – 152, 1992.

[46] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning.*, vol. 14. MIT Press, 2006.

[47] Bishop Christopher M., *Pattern recognition and machine learning*, vol. 4 of *Information science and statistics*. Springer, 2006.

[48] P. F. Pai and W. C. Hong, "Support vector machines with simulated annealing algorithms in electricity load forecasting," *Energy Conversion and Management*, vol. 46, pp. 2669–2688, oct 2005.

[49] M. Espinoza, J. a. K. Suykens, R. Belmans, and B. De Moor, "Electric Load Forecasting using Kernel-Based Modeling for Nonlinear System identification," *IEEE Control Systems*, vol. 27, no. 5, pp. 43–57, 2007.

[50] V. H. Ferreira and A. P. Alves da Silva, "Automatic Kernel Based Models for Short Term Load Forecasting," in *2009 15th International Conference on Intelligent System Applications to Power Systems*, pp. 1–6, IEEE, nov 2009.

[51] D. Yang, H. Li, G. D. Peterson, and Z. Zhang, "Joint electrical load modeling and forecasting based on sparse Bayesian Learning for the smart grid," in *2011 45th Annual Conference on Information Sciences and Systems*, pp. 1–6, IEEE, mar 2011.

[52] L. Niu, J. Zhao, and M. Liu, "Application of relevance vector regression model based on sparse bayesian learning to long-term electricity demand forecasting," in *2009 International Conference on Mechatronics and Automation*, pp. 2363–2367, IEEE, aug 2009.

[53] H. Mori and A. Takahashi, "Hybrid intelligent method of relevant vector machine and regression tree for probabilistic load forecasting," in *2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies*, pp. 1–8, IEEE, dec 2011.

[54] Q. Duan, J.-g. Zhao, L. Niu, and K. Luo, "Regression Based on Sparse Bayesian Learning and the Applications in Electric Systems," in *2008 Fourth International Conference on Natural Computation*, vol. 1, pp. 106–110, IEEE, 2008.

[55] H. Cho, Y. Goude, X. Brossat, and Q. Yao, "Modeling and Forecasting Daily Electricity Load Curves: A Hybrid Approach," *Journal of the American Statistical Association*, vol. 108, pp. 7–21, mar 2013.

[56] O. Kramer, B. Satzger, and J. Lässig, "Power prediction in smart grids with evolutionary local kernel regression," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (M. Grana Romay, Manuel and Corchado, Emilio and Garcia Sebastian, ed.), vol. 6076 LNAI, pp. 262–269, Springer Berlin Heidelberg, 2010.

[57] D. Salsburg, *The Lady Tasting Tea: How Statistics Revolutionized Science in the Twentieth Century*. 2002.

[58] M. Teixeira and G. Zaverucha, "Fuzzy Markov predictor in electric load forecasting," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, vol. 3, pp. 2416–2421, IEEE, 2002.

[59] Jianrong Jia and Dongxiao Niu, "Application of improved gray Markov model in power load forecasting," in *2008 Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies*, pp. 1488–1492, IEEE, apr 2008.

[60] D.-x. Niu, B.-e. Kou, and Y.-y. Zhang, "Mid-long Term Load Forecasting Using Hidden Markov Model," in *Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on*, vol. 3, pp. 481–483, IEEE, nov 2009.

[61] Dongxiao Niu, Hui Shi, Jianqing Li, and Cong Xu, "Research on power load forecasting based on combined model of Markov and BP neural networks," in *2010 8th World Congress on Intelligent Control and Automation*, pp. 4372–4375, IEEE, jul 2010.

[62] W. Labeeuw and G. Deconinck, "Residential Electrical Load Model Based on Mixture Model Clustering and Markov Models," *IEEE Transactions on Industrial Informatics*, vol. 9, pp. 1561–1569, aug 2013.

[63] C. Dicembrino and G. Trovato, "Structural breaks, price and income elasticity and forecast of the monthly italian electricity demand," in *2013 10th International Conference on the European Energy Market (EEM)*, pp. 1–12, IEEE, may 2013.

[64] S. Fan, Y. K. Wu, W. J. Lee, and C. Y. Lee, "Comparative study on load forecasting technologies for different geographical distributed loads," in *IEEE Power and Energy Society General Meeting*, pp. 1–8, IEEE, jul 2011.

[65] L. J. Soares and M. C. Medeiros, "Modeling and forecasting short-term electricity load: A comparison of methods with an application to Brazilian data," *International Journal of Forecasting*, vol. 24, no. 4, pp. 630–644, 2008.

[66] L. J. Soares and L. R. Souza, "Forecasting electricity demand using generalized long memory," *International Journal of Forecasting*, vol. 22, no. 1, pp. 17–28, 2006.

[67] R. Ramanathan, R. Engle, C. W. Granger, F. Vahid-Araghi, and C. Brace, "Short-run forecasts of electricity loads and peaks," *International Journal of Forecasting*, vol. 13, pp. 161–174, jun 1997.

[68] J. C. Messenger, S. Lee, D. McCann, and J. C. Messenger, *Working time around the world: Trends in working hours, laws, and policies in a global comparative perspective*. Routledge, 2007.

[69] C. Fünfgeld and R. Tiedemann, "Anwendung der Repräsentativen VDEW-Lastprofile step by step," *VDEW Materialien*, vol. M-05/2000, p. 34, 2000.

[70] T. Hartmann, *Lastprofile für unterbrechbare Verbrauchseinrichtungen - Praxisleitfaden*. Frankfurt am Main: VWEW-Energieverl., 1 ed., 2003.

[71] M. Sachs, *Wahrscheinlichkeitsrechnung und Statistik : für Ingenieurstudenten an Fachhochschulen*. München; Wien: Fachbuchverl. Leipzig im Carl-Hanser-Verl., 3 ed., 2009.

[72] R. Thompson, *Das Gehirn: Von der Nervenzelle zur Verhaltenssteuerung*. Spektrum Akademischer Verlag, 3 ed., 2010.

[73] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation," in *Parallel Distributed Processing* (D. E. Rumelhart and R. McClelland, eds.), ch. 8, p. 49, Cambridge, Mass.: MIT Press, sep 1986.

[74] MathWorks, "Neuronal Network Toolbox."

[75] G. Matheron, "Principles of geostatistics," *Economic Geology*, vol. 58, no. 8, pp. 1246–1266, 1963.

[76] G. Matheron, "The Intrinsic Random Functions and Their Applications," *Advances in Applied Probability*, vol. 5, no. 3, pp. 439–468, 1973.

[77] D. G. Krige, *A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand*. PhD thesis, 1952.

[78] J. Vanhatalo and J. Riihimäki, "GPstuff: Bayesian modeling with Gaussian processes," *Journal of Machine Learning Research*, vol. 14, pp. 1175–1179, 2013.

[79] J. Vanhatalo, J. Riihimäki, J. Hartikainen, P. Jylänki, V. Tolvanen, and A. Vehtari, "Bayesian Modeling with Gaussian Processes using the GPstuff Toolbox," 2012.

[80] E. Snelson and Z. Ghahramani, "Sparse Gaussian Processes using Pseudo-inputs," in *Advances in Neural Information Processing Systems 18* (Y. Weiss, B. Schölkopf, and J. C. Platt, eds.), pp. 1257–1264, MIT Press, 2006.

[81] H. V. Haghi and S. M. M. Tafreshi, "Modeling and Forecasting of Energy Prices using Non-stationary Markov Models versus Stationary Hybrid Models including a Survey of all Methods," in *2007 IEEE Canada Electrical Power Conference*, pp. 429–434, IEEE, oct 2007.

[82] L. E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *The Annals of Mathematical Statistics*, vol. 37, pp. 1554–1563, dec 1966.

[83] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[84] O. Wu, T. Liu, B. Huang, and F. Forbes, "Predicting Electricity Pool Prices Using Hidden Markov Models," *IFAC-PapersOnLine*, vol. 48, no. 8, pp. 343–348, 2015.

[85] A. M. Fraser, *Hidden Markov models and dynamic systems*. SIAM, jan 2008.

[86] P. Binh, T. Loan, and T. Khoa, "Application of Hidden Markov and Bayes for Demand Forecasting," in *2006 IEEE Power India Conference*, pp. 549–553, IEEE, 2006.

[87] Jianhua Zhang, Jingyue Wang, Rui Wang, and Guolian Hou, "Forecasting next-day electricity prices with Hidden Markov Models," in *2010 5th IEEE Conference on Industrial Electronics and Applications*, pp. 1736–1740, IEEE, jun 2010.

[88] M. Gastaldi, R. Lamedica, and A. Nardecchia, "Short-term forecasting of municipal load through a Kalman filtering based approach," in *IEEE PES Power Systems Conference and Exposition, 2004.*, pp. 687–692, IEEE, 2004.

[89] D. Trudnowski, W. McReynolds, and J. Johnson, "Real-time very short-term load prediction for power-system automatic generation control," *IEEE Transactions on Control Systems Technology*, vol. 9, pp. 254–260, mar 2001.

[90] H. Al-Hamadi and S. Soliman, "Fuzzy short-term electric load forecasting using Kalman filter," *IEE Proceedings - Generation, Transmission and Distribution*, vol. 153, no. 2, p. 217, 2006.

[91] E. Sanchez, A. Alanis, and J. Rico, "Electric load demand prediction using neural network trained by Kalman filtering," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, vol. 4, pp. 2771–2775, IEEE, 2004.

[92] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning Attractor Landscapes for Learning Motor Primitives," in *Advances in Neural Information Processing Systems 15* (S. Becker, S. Thrun, and K. Obermayer, eds.), pp. 1547–1554, MIT Press, 2003.

[93] A. Paraschos, G. Neumann, and J. Peters, "A Probabilistic Approach to Robot Trajectory Generation," in *Proceedings of the International Conference on Humanoid Robots (HUMANOIDS)*, 2013.

[94] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, vol. 12, no. 1, pp. 80–86, 1970.

[95] Tikhonov A., "Solution of Incorrectly Formulated Problems and the Regularization Method," *Soviet Math. Dokl.*, vol. 5, p. 1035/1038, 1963.

[96] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*. Mathematics and Its Applications, Springer, 1996.

[97] M. A. S. Ewerton, *Modeling Human-Robot Interaction with Probabilistic Movement Representations*. PhD thesis, Technische Universität Darmstadt, 2014.

[98] S. Borman, "The Expectation Maximization Algorithm A short tutorial," 2004.

[99] H. Steinhaus, "Sur la division des corp materiels en parties," *Bull. Acad. Polon. Sci*, vol. 1, pp. 801–804, nov 1955.

[100] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, mar 1982.

[101] A. Misiorek and R. Weron, "Application of external variables to increase accuracy of system load forecasts," in *Proceedings of APE05 Conference*, 2005.

[102] H. Hippert, C. Pedreira, and R. Souza, "Neural networks for short-term load forecasting: a review and evaluation," *IEEE Transactions on Power Systems*, vol. 16, no. 1, pp. 44–55, 2001.

[103] J. W. Taylor, "Using Weather Ensemble Predictions in Electricity Demand Forecasting Using Weather Ensemble Predictions in Electricity Demand Forecasting," *International Journal of Forecasting*, vol. 19, no. 0, pp. 57–70, 2003.

[104] N. Amjady, "Short-term bus load forecasting of power systems by a new hybrid method," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 333–341, 2007.

[105] M. Mierau, D. Noeren, and F. Becker, "Potential der ladung von elektrofahrzeugen durch photovoltaikenergie im privathaushalt," tech. rep., Fraunhofer ISE, Freiburg, 2014.

[106] IEA, "Key World Energy Statistics, Key World Energy Statistics 2014," 2014.

[107] N. Hansen, "The {CMA} evolution strategy: a comparing review," in *Towards a new evolutionary computation. Advances on estimation of distribution algorithms* (J. A. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, eds.), pp. 75–102, Springer, 2006.

[108] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, pp. 679–688, oct 2006.

[109] D. Ernst, M. Glavic, F. Capitanescu, and L. Wehenkel, "Reinforcement learning versus model predictive control: a comparison on a power system problem.," *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 39, no. 2, pp. 517–529, 2009.

[110] N. Amjady and A. Daraeepour, "Mixed price and load forecasting of electricity markets by a new iterative prediction method," *Electric Power Systems Research*, vol. 79, no. 9, pp. 1329–1336, 2009.

[111] M. Mojaddady, M. Nabi, and S. Khadivi, "Stock Market Prediction using Twin Gaussian Process Regression," *Old.Shahed.Ac.Ir*, no. August, 2015.