Chapter 2
Short Term Load Forecasting

Electrification of transport and heating, and the integration of low carbon technologies (LCT) is driving the need to know when and how much electricity is being consumed and generated by consumers. It is also important to know what external factors influence individual electricity demand.

Low voltage networks connect the end users through feeders and substations, and thus encompass diverse strata of society. Some feeders may be small with only a handful of households, while others may have over a hundred customers. Some low voltage networks include small-to-medium enterprises (SMES), or hospitals and schools, but others may be entirely residential. Furthermore, local feeders will also likely register usage from lighting in common areas of apartments or flats, street lighting and other street furniture such as traffic lights.

Moreover, the way that different households on the same feeder or substation use electricity may be drastically different. For example, load profiles of residential households will vary significantly depending on the size of their houses, occupancy, socio-demographic characteristics and lifestyle. Profiles will also depend on whether households have solar panels, overnight storage heating (OSH) or electric vehicles [1]. Thus, knowing how and when people use electricity in their homes and communities is a fundamental part of understanding how to effectively generate and distribute electrical energy.

In short term load forecasting, the aim is to estimate the load for the next half hour up to the next two weeks. For aggregated household demand, many different methods are proposed and tested (see e.g. Alfares and Nazeeruddin [2], Taylor and Espasa [3], Hong and Fan [4], etc.). Aggregating the data smooths it, therefore makes it easier to forecast. The individual level demand forecasting is more challenging and comes with higher errors, as shown in Singh et al. [5], Haben et al. [1]. The growth of literature on short term load forecasting at the individual level has started with the wider access to higher resolution data in the last two decades, and is still developing.

M. Jacob et al., *Forecasting and Assessing Risk of Individual Electricity Peaks*, Mathematics of Planet Earth,
https://doi.org/10.1007/978-3-030-28669-9_2

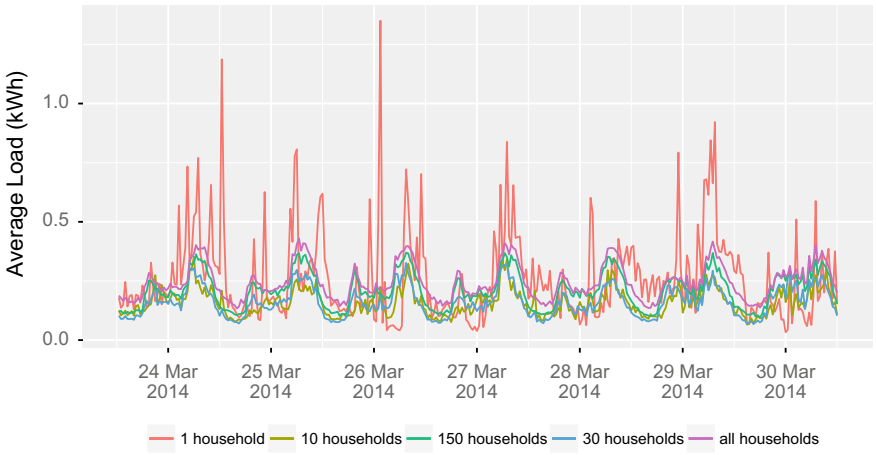

Fig. 2.1 Aggregations of different number of households

So, why is it not enough to look at electricity from an aggregated point of view? Firstly, aggregated load profiles may not reflect individual load profiles, as can be seen from the example in Fig. 2.1. Here, the load has been aggregated for different number of households in one week and the subsequent smoothing is evident. Not only are aggregated load profiles smoother, they also tend to have stronger seasonality and weather dependency than disaggregated load profiles [6]. Demand side response, which encompasses efforts to modify consumption patterns, can be better informed by forecasts which can predict irregular peaks. This is especially true with distributed generation and when both demand and supply become dynamic with LCT integration.

Secondly, as noted by Haben et al. [1], aggregations of individual load profiles do not consider network losses or other loads that are usually not monitored, such as traffic lights and street furniture. Having information of all load allows for better modelling and hence more efficient energy distribution and generation.

Thirdly, considering aggregated load profiles tells us little about individual households or businesses, who may benefit from having tailored energy pricing plans and contracts or need help for informed decision making regarding investment in batteries, photovoltaic and other LCT [7]. The enabling of these kinds of decision making processes is one of the central motivations of the research presented in this book. To do so, we want to consider forecasting methods from both statistics and machine learning literature, specifically the state of the art forecasts within different categories, at the time of writing, and compare them.

In the past, forecasting individual households and feeders was a challenge not just because new forecasting techniques were developing, but also because of the lack of access to a high quality data. The availability of smart meter data alleviates this hindrance and gives new opportunity to address this challenge.

Over the course of this chapter, we will consider several different forecasting algorithms stemming from various areas of mathematics. In Sect. 2.1, we consider the literature on different forecasts and discuss their strengths and weaknesses. Similarly, in Sect. 2.2, we will consider some popular ways of validating forecasts and discuss the merits, limitations and appropriateness of each. In the discussion in Sect. 2.3, we will motivate the choices of forecasts and error measures used for the case studies to be presented in Chap. 5.

2.1 Forecasts

Historically, forecasts have been generated to represent typical behaviour and thus have mostly relied on expectation values. Consequently, many popular algorithms in the literature, and in practice, are point load forecasts using averaging techniques [8] and indeed considering aggregations as mentioned above. Point load forecasts refer to forecasts which give a single, usually mean, value for the future load estimate. Regardless, the need to integrate LCT, market competition and electricity trading have brought about the need for probabilistic load forecasts which may include intervals, quantiles, or densities as noted by Hong and Fan [4] and Haben et al. [1]. In either point load forecasting or probabilistic load forecasting, many approaches exist and increasingly mixed approaches are being used to create hybrid profiles to better represent load with irregular peaks.

The challenge that we are interested in addressing in this chapter is the following: given past load (measured in kWh), we want to create a week-long forecast with the same time resolution as the data, for one or more households. While electricity is consumed continuously, we work with time-stamped, discrete load measurements denoted by y_t where $t = \{1, 2, \ldots, N\}$ denotes time and usually obtained from a smart meter.

In this section, we will review several forecasting algorithms. We will illustrate the analyses presented in this chapter in a case study Sect. 5.1, using the TVV endpoint monitor data described in Sect. 1.2.2.

2.1.1 Linear Regression

Different techniques based on linear regression have been widely used for both short term and long term load forecasting. They are very popular due to the simplicity and good performance in general. Regression is used to estimate the relationship between different factors or predictors and the variable we want to predict. Linear regression assumes that these relationships are linear and tries to find the optimal parameters (or weights) so that the prediction error is minimal. This enables us to easily introduce different kind of variables such as calendar variables, past load and temperature. The basic model for multiple linear regression (MLR) is given by

$$y_t = \boldsymbol{\beta}^T \boldsymbol{x}_t + \epsilon_t, \tag{2.1}$$

where y_t is the dependent variable at time t which is influenced by the p independent variables $\boldsymbol{x}_t = (1, x_{t1}, x_{t2}, \ldots, x_{tp})^T$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$ are the corresponding regression parameters. The random error term, ϵ_t, is assumed to be normally distributed with zero mean and constant variance $\sigma^2 > 0$, i.e. $\epsilon_t \mathcal{N}(0, \sigma^2)$. Also, $E(\epsilon_t \epsilon_s) = 0$, for $t \neq s$.

The dependent variable or series is the one we are interested in forecasting, whereas the $\boldsymbol{x}_t$ contains information about the factors influencing the load such as temperature or a special day.

As noted in the tutorial review by Hong and Fan [4], the regressions coefficients or parameters are usually estimated using ordinary least squares using the following formula:

$$\hat{\boldsymbol{\beta}} = \left(\sum_{t=1}^{n} \boldsymbol{x}_t \boldsymbol{x}_t^T \right)^{-1} \sum_{t=1}^{n} \boldsymbol{x}_t \boldsymbol{y}_t \tag{2.2}$$

The least squares estimator for $\boldsymbol{\beta}$ is unbiased, i.e., $\mathbb{E}[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$. We also make the connection that the least squares estimator for $\boldsymbol{\beta}$ is the same as the maximum likelihood estimator for $\boldsymbol{\beta}$ if the errors, ϵ_t, are assumed to be normally distributed.

This simple linear model is then basis for various forecasting methods. We start by listing several examples for aggregated load forecast. For example, Moghram and Rahman [9] explored MLR, amongst others, to obtain 24 h ahead hourly load forecast for a US utility whilst considering dry bulb[1] and dew point temperature[2] as well as wind speed. Two models were calibrated, one for winter and one for summer. The authors divided the day into unequal time zones which corresponded roughly to overnight, breakfast, before lunch, after lunch, evening and night time. It was found that dividing the day in this way resulted in a better fit than not dividing the day at all or dividing it equally. The authors also found significant correlations for temperature and wind speed when using the MLR model.

Charlton and Singleton [10] used MLR to create hourly load forecasts. The regression model considered temperature (up to the power of two), day number, and the multiplication of the two. The created model accounts for the short term effects of temperature on energy use, long term trends in energy use and the interaction between the two. Further refinements were introduced by incorporating smoothed temperature from different weather stations, removing outliers and treating national holidays such as Christmas as being different to regular holidays. Each addition resulted in reduction in errors.

[1] Dry bulb temperature is the temperature as measured when the thermometer is exposed to air but not to sunlight or moisture. It is associated with air temperature that is most often reported.

[2] Dew point temperature is the temperature that would be measured if relative humidity is 100% and all other variables are unchanged. Dew point temperature is always lower than the dry bulb temperature.

In a similar vein to the papers above, Alfares and Nazeeruddin [2] consider nine different forecasting algorithms in a bid to update the review on forecasting methods and noted the MLR approach to be one of the earliest. The aim was to forecast the power load at the Nova Scotia Power Corporation and thus pertained to aggregated load. They found the machine learning algorithms to be better overall. The vast literature surveyed in Alfares and Nazeeruddin [2], Hong and Fan [4] and many other reviews, show linear regression to be popular and reasonably competitive despite its simplicity.

While the most common use of regression is to estimate the mean value of the dependent variable, when the independent variables are fixed, it can be also used to estimate quantiles [1, 11].

The simple seasonal quantile regression model used in Haben and Giasemidis [11] was updated in Haben et al. [1] and applied to hourly load of feeders. Treating each half-hour and week day as separate time-series, the median quantile is estimated using the day of trial, three seasons (with sin and cos to model periodicity), a linear trend and then temperature is added using a cubic polynomial.

To find optimal coefficients for linear regression, one usually relies on ordinary least squares estimator. Depending on the structure of a problem, this can result in an ill-posed problem. Ridge regression is a commonly used method of regularisation of ill-posed problems in statistics. Suppose we wish to find an $\boldsymbol{x}$ such that $A\boldsymbol{x} = \boldsymbol{b}$, where A is a matrix and $\boldsymbol{x}$ and $\boldsymbol{b}$ are vectors. Then, the ordinary least squares estimation solution would be obtained by a minimisation of $||A\boldsymbol{x} - \boldsymbol{b}||_2$. However for an ill-posed problem, this solution may be over-fitted or under-fitted. To give preference to a solution with desirable properties, the regularisation term $||\Gamma\boldsymbol{x}||_2$ is added so that the minimisation is of $||A\boldsymbol{x} - \boldsymbol{b}||_2 + ||\Gamma\boldsymbol{x}||_2$. This gives the solution $\hat{\boldsymbol{x}} = \left(A^T A + \Gamma^T \Gamma\right)^{-1} A^T \boldsymbol{b}$.[3]

2.1.2 Time Series Based Algorithms

The key assumptions in classical MLR techniques is that the dependent variable, y_t, is influenced by independent predictor variables $\boldsymbol{x}_t$ and that the error terms are independent, normally distributed with mean zero and constant variance. However, these assumptions, particularly of independence, may not hold, especially when measurements of the same variable are made in time, say owing to periodic cycles in the natural world such as seasons or in our society such as weekly employment cycles or annual fiscal cycle. As such, ordinary least squares regression may not be appropriate to forecast time series. Since individual smart meter data may be treated as time series, we may borrow from the vast body of work that statistical models provide, which allow us to exploit some of the internal structure in the data. In this section, we will review the following time series methods: autoregressive (AR)

[3] In the Bayesian interpretation, simplistically this regularised solution is the most probable solution given the data and the prior distribution for $\boldsymbol{x}$ according to Bayes' Theorem.

models (and their extensions), exponential smoothing models and kernel density estimation (KDE) algorithms.

2.1.2.1 Autoregressive Models

Time series that stem from human behaviour usually have some temporal dependence based on our circadian rhythm. If past observations are very good indicators of future observations, the dependencies may render linear regressions techniques an inappropriate forecasting tool. In such cases, we may create forecasts based on autoregressive (AR) models. In an AR model of order p, denoted by AR(p), the load at time t, is a sum of a linear combination of the load at p previous times and a stochastic error term:

$$y_t = a + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t, \tag{2.3}$$

with a is a constant, ϕ_i are AR parameters to be estimated, p is the number of historical measurements used in the estimation and ϵ denotes the error term which is typically assumed to be independent with mean 0 and constant variance, σ^2. In a way, we can see some similarity between the MLR model and the AR model; in the MLR, load is dependent on external variables but in the AR model, load is a linear combination of previous values of load.

An example of using an AR model to estimate feeders' load is given in Haben et al. [1]. Here, the model is applied to residuals of load, $r_t = y_t - \mu_t$, where μ_t is an expected value of weekly load. The most obvious advantage of using the residuals is that we can define r_t in a such way that it can be assumed to be stationary. In addition, μ_t models typical weekly behaviour and thus changing the definition of μ_t allows the modeller to introduce seasonality or trends quite naturally and in various different ways, as opposed to the using load itself. In Haben et al. [1], the AR parameters were found using the Burg method.[4] Seasonality can be introduced by including it in the mean profile, μ_t.

Other examples of AR models and their modifications include Moghram and Rahman [9], Alfares and Nazeeruddin [2], Weron [12] and Taylor and McSharry [13], but most of these are studies with aggregated load profiles.

Since we expect that past load is quite informative in understanding future load, we expect that AR models will be quite competitive forecasts, especially when built to include trends and seasonality.

[4]The Burg method minimises least square errors in Eq. (2.3) and similar equation which replaces r_{t-i} with r_{t+i}.

2.1.2.2 Seasonal Autoregressive Integrated Moving Average—SARIMA Models

From their first appearance in the seminal Box & Jenkins book in 1970 (for the most recent edition see Box et al. [14]), autoregressive integrated moving average (ARIMA) time series models are widely used for analysis and forecasting in a wide range of applications. The time series y_t typically consists of trend, seasonal and irregular components. Instead of modelling each of the components separately, trend and seasonal are removed by differencing the data. The resulting time series is then treated as stationary (i.e. means, variances and other basic statistics remain unchanged over time). As we have seen in the previous section, AR models assume that the predicted value is a linear combination of most recent previous values plus a random noise term. Thus,

$$y_t = a + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t,$$

where a is a constant, ϕ are weights, p is a number of historical values considered and $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$. The moving average model (MA) assumes the predicted value to be the linear combination of the previous errors plus the expected value and a random noise term, giving

$$y_t = \mu + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t,$$

where μ is the expected value, θ are weights, q is the number of historical values considered, and $\epsilon_t \sim \mathcal{N}(0, \sigma^2)$. The main parameters of the model are p, d and q, where p is the number of previous values used in the auto-regressive part, d is the number of times we need to difference the data in order to be able to assume that is stationary, and q is the number of previous values used in the moving average part. When strong seasonality is observed in the data, a seasonal part, modelling repetitive seasonal behaviour, can be added to this model in a similar fashion, containing its own set of parameters P, D, Q. A SARMA model, seasonal autoregressive moving average model for 24 aggregated energy profiles is explored in Singh et al. [5] based on 6 s resolution data over a period of one year. Routine energy use is modelled with AR part and stochastic activities with MA part. A daily periodic pattern is captured within a seasonal model. The optimal parameters were determined as $p = 5$ and $q = 30$. The least error square minimisation was used, where the results with different parameter values were compared and the ones that minimised the error were picked up. Interestingly, SARMA not only outperformed other methods (support vector, least square support vector regression and artificial neural network with one hidden layer of ten nodes) regarding mean load prediction, but also regarding peak load prediction, resulting in smaller errors for peaks.

(S)ARMA and (S)ARIMA models can be extended using exogenous variables such as temperature, wind chill, special day and similar inputs. These are called

(S)ARIMAX or (S)ARMAX models, for example Singh et al. [5] gives the following ARMAX model

$$y_t = a + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t + \mu + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \sum_{i=1}^{r} \beta_i T_{t-i},$$

where βs are further parameters that represent the exogenous variables T_t, for instance the outdoor temperature at time t. Two simple algorithms **Last Week (LW)** and **Similar Day (SD)** can be seen as trivial (degenerate) examples of AR models with no error terms, and we will used them as benchmarks, when comparing different forecasting algorithms in Chap. 5. The Last Week (LW) forecast is a very simple forecast using the last week same half-hour load to predict the current one. Therefore, it can be seen as an AR model where $p = 1, d = 0, a = 0, \phi_1 = 1, \epsilon_t \equiv 0$.

The **Similar Day (SD)** forecast instead uses the average of n last weeks, same half-hour loads to predict the current one. Therefore, it can be seen as an AR model where $p = n, d = 0, a = 0, \phi_1, \ldots \phi_n = \frac{1}{n}, \epsilon_t \equiv 0$.

2.1.2.3 Exponential Smoothing Models

The simplest exponential smoothing model puts exponentially decreasing weights on past observations.

Suppose we have observations of the load starting from time $t = 1$, then the single/simple exponential smoothing model is given by

$$S_t = \alpha y_t + (1 - \alpha) S_{t-1}, \tag{2.4}$$

where $\alpha \in (0, 1)$, S_t is the output of the model at t and the estimate for the load at time $t + 1$. Since the future estimates of the load depend on past observations and estimates, it is necessary to specify S_1. One choice for $S_1 = y_1$, but this puts potentially unreasonable weight on early forecasts. One may set S_1 to be the mean of the first few values instead, to circumvent this issue. Regardless, the smaller the value of α, the more sensitive the forecast is to the initialisation.

In the single exponentially smoothed model, as α tends to zero, the forecast tends to be no better than the initial value. On the other hand, as α tends to 1, the forecast is no better than the most recent observation. For $\alpha = 1$, it becomes the LW forecast given in the previous section. The choice for α may be made by the forecaster, say from previous experience and expertise or it may chosen by minimising error functions such as a mean square error.

When the data contains a trend, a double exponential smoothing model is more suitable. This is done by having two exponential smoothing equations: the first on the overall data (2.5) and the second on the trend (2.6):

$$S_t = \alpha y_t + (1-\alpha)(S_{t-1} + b_{t-1}), \tag{2.5}$$
$$b_t = \beta(S_t - S_{t-1}) + (1-\beta)b_{t-1}, \tag{2.6}$$

where b_t is the smoothed estimate of the trend and all else remains the same. We now have a second smoothing parameter β that must also be estimated. Given the model in (2.5) and (2.6), the forecast for load at time $t+m$ is given by $y_{t+m} = S_t + mb_t$.

Of course, we know that electricity load profiles have daily, weekly and annual cycles. Taylor [15] considered the triple exponential smoothing model, also known as the Holt-Winters exponential smoothing model, to address the situation where there is not only trend, but intraweek and intraday seasonality. The annual cycle is ignored as it is not likely to be of importance for forecasts of up to a day ahead. Taylor [15] further improved this algorithm by adding an AR(1) model to account for correlated errors. This was found to improve forecast as when the triple exponential model with two multiplicative seasonality was used, the one step ahead errors still had large auto-correlations suggesting that the forecasts were not optimal. To compensate, the AR term was added to the model.

Arora and Taylor [16] and Haben et al. [1] used a similar model, though without trend, to forecast short term load forecast of individual feeders with additive intraday and intraweek seasonality. Haben et al. [1] found that the so called **Holt-Winters-Taylor (HWT)** triple exponential smoothing method that was first presented in Taylor [17] was one of their best performing algorithms regardless of whether the temperature was included or omitted.

A model is given by the following set of equations:

$$\begin{aligned} y_t &= S_{t-1} + d_{t-s_1} + w_{t-s_2} + \phi e_{t-1} + \epsilon_t, \\ e_t &= y_t - (S_{t-1} + d_{t-s_1} + w_{t-s_2}), \\ S_t &= S_{t-1} + \lambda e_t, \\ d_t &= d_{t-s_1} + \delta e_t, \\ w_t &= w_{t-s_2} + \omega e_t, \end{aligned} \tag{2.7}$$

where y_t is the load, S_t is the exponential smoothed variable often referred to as the level, w_t is the weekly seasonal index, d_t is the daily seasonal index, $s_2 = 168$, $s_1 = 24$ (as there are 336 half-hours in a week and 48 in a day), e_t is the one step ahead forecast error. The parameters λ, δ and ω are the smoothing parameters. This model has no trend, but it has intraweek and intraday seasonality. The above mention literature suggests that when an exponential model is applied, the one-step ahead errors have strong correlations that can be better modelled with an AR(1) model, which in (2.7) is done through the ϕ term. The k-step ahead forecast is then given by $S_t + w_{t-s_2+k} + \phi^k e_t$ from the forecast starting point t.

2.1.2.4 Kernel Density Estimation Methods

Next, we briefly consider the kernel density estimation (KDE), that is quite popular technique in time-series predictions, and has been used for load prediction frequently. The major advantage of KDE based forecasts is that they allow the estimation of the entire probability distribution. Thus, coming up with probabilistic load forecasts is straight forward and results are easy to interpret. Moreover, a point load forecast can be easily constructed, for example by taking the median. This flexibility and ease of interpretation make kernel density forecasts useful for decision making regarding energy trading and distribution or even demand side response. However, calculating entire probability density functions and tuning parameters can be computationally expensive as we will discuss shortly.

We divide the KDE methods into two broad categories, conditional and unconditional. In the first instance, the unconditional density is estimated using historical observations of the variable to be forecasted. In the second case, the density is conditioned on one or more external variables such as time of day or temperature. The simplest way to estimate the unconditional density using KDE is given in (2.8):

$$\hat{f}(l) = \sum_{i=1}^{t} K_{h_y}(y_i - l), \tag{2.8}$$

where $\{y_1, \ldots, L_t\}$ denotes historical load observations, $K_h(\cdot) = K(\cdot/h)/h$ denotes the kernel function, $h_L > 0$ is the bandwidth and $\hat{f}(y)$ is the local density estimate at point y which takes any value that the load can take. If instead, we want to estimate the conditional density, then:

$$\hat{f}(l|x) = \frac{\sum_{i=1}^{t} K_{h_x}(X_i - x) K_{h_L}(y_i - l)}{\sum_{i=1}^{y} K_{h_x}(X_i - x)}, \tag{2.9}$$

where $h_L > 0$ and $h_x > 0$ are bandwidths. KDE methods have been used for energy forecasting particularly in wind power forecasting but more recently Arora and Taylor [18] and Haben et al. [1] used both conditional and unconditional KDE methods to forecast load of individual low voltage load profiles. Arora and Taylor [18] found one of the best forecasts to be using KDE with intraday cycles as well as a smoothing parameter. However, Haben et al. [1] chose to exclude the smoothing parameter as its inclusion costs significant computational efforts. In general, the conditional KDE methods have higher computational cost. This is because the optimisation of the bandwidth is a nonlinear which is computationally expensive and the more variables on which the density is estimated, the more bandwidths must be estimated.

In the above discussion, we have omitted some details and challenges. Firstly, how are bandwidths estimated? One common method is to minimise the difference between the one step ahead forecast and the corresponding load. Secondly, both

Arora and Taylor [18] and Haben et al. [1] normalise load to be between 0 and 1. This has the advantage that forecast accuracy can be more easily discussed across different feeders and this also accelerates the optimisation problem. However, (2.8) applies when l can take any value and adjustments when the support of the density is finite. Arora and Taylor [18] adjust the bandwidth near the boundary whereas Haben et al. [1] do not explicitly discuss the correction undertaken. The choice of kernels in the estimation may also have some impact. The Gaussian kernel[5] was used in both of the papers discussed above but others may be used, for example Epanechnikov[6] or biweight[7] kernels.

2.1.3 Permutation Based Algorithms

Though the methods discussed in the above section are widely used forecasting tools, their performances on different individual smart meter data-sets vary. Some of the mentioned algorithms have smoothing properties and thus, they may be unsuitable when focusing on individual peak prediction. We now list several permutation-based algorithms that are all based on the idea that people do same things repeatedly, but in slightly different time periods. This is of relevance for modelling demand peaks.

2.1.3.1 Adjusted Average Forecast

One of the simple forecasts we mentioned before at the end of Sect. 2.1.2.1, Similar day (SD) forecast averages over the several previous values of load. For example, to predict a load on Thursday 6.30 pm, it will use the mean of several previous Thursdays 6.30 pm loads. But what happens if one of those Thursdays, a particular household is a bit early (or late) with their dinner? Their peak will move half an hour or hour earlier (or later). Averaging over all values will smooth the real peak, and the mistake will be penalised twice, once for predicting the peak and once for missing earlier (later) one. Haben et al. [19] introduced a new forecasting algorithm which iteratively updates a base forecast based on average of previous values (as the SD forecasting), but allows permutations within a specified time frame. We shall refer to it as the **Adjusted Average (AA)** forecast. The algorithm is given as follows:

(i) For each day of the week, suppose daily profiles $\boldsymbol{G}^{(k)}$ are available for past N weeks, where $k = 1, \ldots, N$. By convention, $\boldsymbol{G}^{(1)}$ is the most recent week.
(ii) A base profile, $\boldsymbol{F}^{(1)}$, is created whose components are defined by the median of corresponding past load.

[5] $K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$.
[6] $K(u) = \frac{3}{4}(1-u^2)$ for $|u| \leq 1$ and $K(u) = 0$ otherwise.
[7] $K(u) = \frac{15}{16}(1-u^2)^2$ for $|u| \leq 1$ and $K(u) = 0$ otherwise.

(iii) This baseline is updated iteratively in the following way. Suppose, at iteration k, we have $\boldsymbol{F}^{(k)}$ for $1 \leq k \leq N-1$, then $\boldsymbol{F}^{(k+1)}$ is obtained by setting $\boldsymbol{F}^{(k+1)} = \frac{1}{k+1}\left(\hat{\boldsymbol{G}}^{(k)} + k\boldsymbol{F}^{(k)}\right)$, where $\hat{\boldsymbol{G}}^{(k)} = \hat{P}\boldsymbol{G}^{(k)}$ with $\hat{P} \in \mathcal{P}$ being a permutation matrix s.t. $||\hat{P}\boldsymbol{G}^{(k)} - \boldsymbol{F}^{(k)}||_4 = \min_{P \in \mathcal{P}} ||P\boldsymbol{G}^{(k)} - \boldsymbol{F}^{(k)}||_4$. $\mathcal{P}$ is the set of restricted permutations i.e, for a chosen time window, ω, the load at half hour i can be associated to the load at half hour j if $|i-j| \leq \omega$.

(iv) The final forecast is then given by $\boldsymbol{F}^{(N)} = \frac{1}{N+1}\left(\sum_{k=1}^{N} \hat{\boldsymbol{G}}^{(k)} + \boldsymbol{F}^{(1)}\right)$.

In this way, the algorithm can permute values in some of historical profiles in order to find the smallest error between observed and predicted time series. This displacement in time can be reduced to an optimisation problem in bipartite graphs, **the minimum weight perfect matching in bipartite graphs**, [20], that can be solved in polynomial time.

A graph $G = (V, E)$ is bipartite if its vertices can be split into two classes, so that all edges are in between different classes. Two bipartite classes are given by observations y_t and forecasts f_t, respectively. Errors between observations and forecasts are used as weights on the edges between the two classes. Instead of focusing only at errors $e_t = y_t - f_t$ (i.e. solely considering the edges between y_t and f_t), differences between

$$y_t - f_{t-1}, y_t - f_{t+1}, y_t - f_{t-2}, y_t - f_{t+2}, \ldots, y_t - f_{t-\omega}, y_t - f_{t+\omega},$$

are also taken into account, for some plausible time-window ω. It seems reasonable not to allow, for instance, to swap morning and evening peaks, so ω should be kept small.

These differences are added as weights and some very large number is assigned as the weight for all the other possible edges between two classes, in order to stop permutations of points far away in time. Now, the perfect matching that minimises the sum of all weights, therefore allowing possibility of slightly early or late forecasted peaks to be matched to the observations without the double penalty is found. The minimum weighted perfect matching is solvable in polynomial time using the Hungarian algorithm Munkres [21], with a time complexity of $O(n(m + n\log n))$ for graphs with n nodes (usually equal to 2×48 for half-hourly daily time series and m edges ($\approx 2 \times n \times \omega$). It is important to notice that although each half-hour is considered separately for prediction, the whole daily time series is taken into account, as permutations will affect adjacent half-hours, so they need to be treated simultaneously.

2.1.3.2 Permutation Merge

Based on a similar idea, Permutation Merge (PM) algorithm presented in Charlton et al. [22] uses a faster optimisation—the minimisation of p-adjusted error (see

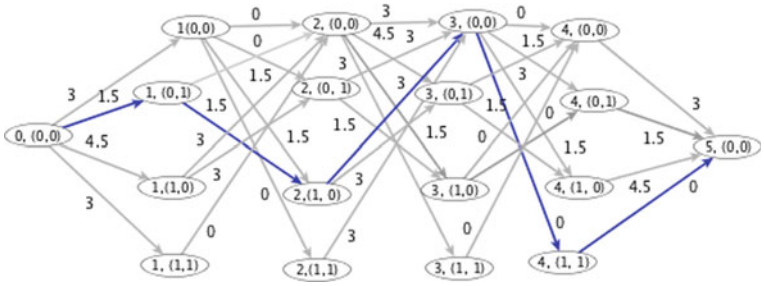

Fig. 2.2 An example of permutation merge

Sect. 2.2.2.2) to match peaks in several profiles simultaneously, based on finding a shortest path in a directed acyclic graph (a graph with directed edges and no cycles). Either Dijkstra algorithm or a topological sort can be used for that Schrijver [20].

Given the n previous profiles, the algorithm builds a directed acyclic graph between each point in time and its predecessors and successors inside a time-window ω, allowing for permutations of values in that window. The cost of each permutation is the difference of the two values that is caused by permutation. Then the minimum weighted path gives an 'averaged' profile with preserved peaks.

As the algorithms complexity is $\mathcal{O}(n\omega^N 4^{N\omega})$, where n is the number of historic profiles, N is the length of time series and ω is time window where permutations are allowed, only small ωs are computationally feasible. If we have two profiles of length five, $\boldsymbol{x} = [0, 0, 3, 0, 0]$ and $\boldsymbol{y} = [3, 0, 0, 0, 0]$ and $\omega = 1$, so we can permute only adjacent values, the constructed graph and the minimum length (weighted) path is given below on Fig. 2.2. As we have two profiles, and are trying to find two permutations that will give us the minimum difference with the median of those two profiles, in each times-step we have 4 possibilities: $(0, 0)$ means both profiles stay the same, $(0, 1)$ the first stays the same, the second is permuted, $(1, 0)$ the first is permuted, the second stays the same, and $(1, 1)$ means both are permuted. As we have to have perfect matching we have $n + 1 = 6$ layers in the graph, and some paths are not available. The solution gives us $[0, 3, 0, 0, 0]$ for both profiles.

2.1.3.3 Adjusted k-nearest Neighbours and Extensions

Valgaev et al. [23] combined the p-adjusted error from Sect. 2.2.2.2 and PM using the k-nearest neighbour (kNN) regression algorithm. The standard kNN algorithm starts by looking for a similar profile in historic data. This is usually done by computing Euclidean based distance between the profiles and returning k minimum distance ones. Then the arithmetic mean is computed and returned as a prediction. Here, instead of the Euclidean distance, the p-adjusted error is used, and instead of computing an arithmetic mean, permutation merge is used to compute adjusted mean. This approach is extended to *A*djusted feature aware k-nearest neighbour (AFkNN)

in Voß et al. [24] using external factors (temperature, bank holiday, day of week), with one difference. Instead of using adjusted error, the Gower distance

$$D_G(i, j) = \frac{1}{N} \sum_{i=1}^{N} \frac{|x_i^{(f)} - x_j^{(f)}|}{\max x^{(f)} - \min x^{(f)}},$$

is deployed. This is computationally demanding, but can result in better performance than PM in average as it has been shown in Voß et al. [24].

The advantage of permutation based algorithms, as mentioned above, is that these iterative permutations allow forecasted load profiles to look more like the observed load profiles. They are better able to replicate the irregular spikes than the more common averaging or regression based algorithms. However, some error measures such as those that will be discussed in Sect. 2.2.1, can doubly penalise peaky forecasts. Both Charlton et al. [22] and Haben et al. [19] demonstrate how a flat "average" forecast is only penalised once for missing the observed peak whereas if a peak is forecasted slightly shifted from when it actually it occurs, it will be penalised once for missing the peak and again for forecasting it where it was not observed.

2.1.4 Machine Learning Based Algorithms

Machine learning algorithms such as artificial neural networks and support vector machines have been remarkably successful when it comes to understanding power systems, particularly for high voltage systems [25, 26] or aggregated load [2, 9, 27]. The big advantage of machine learning techniques is that they can be quite flexible and are capable of handling complexity and non-linearity [12, 28].

However, the parameters such as weights and biases in a machine learning framework do not always have similarly accessible physical interpretations as in the statistical models discussed, Moreover, some machine learning algorithms such as those used for clustering do not include notions of confidence intervals [29]. Nonetheless, since they have such large scope within and outside of electricity forecasting and since we are mostly interested in point load forecasting in this book, we review two key methods within artificial neural networks, multi-layer perceptron and long short term memory network, and discuss support vector machines.

2.1.4.1 Artificial Neural Networks

Artificial Neural Networks (ANN) are designed to mimic the way the human mind processes information; they are composed of neurons or nodes which send and receive input through connections or edges. From the input node(s) to the output node(s), a neural network may have one or more hidden layers. The learning may be shallow i.e. the network has one or two hidden layers which allows for faster computation.

Or it may be deep, meaning it has many hidden layers. This then allows for more accurate forecasts, but at the cost of time and complexity. When there is a need to forecast many customers individually, computational and time efficiency is a practical requirement for everyday forecasting algorithms. Thus, shallow neural networks such as multi-layer perceptron (MLP) with one hidden layer tended to be used frequently [30–32].

2.1.4.2 Multi-layer Perceptron

MLP is an example of a feedforward ANN. This means that the network is acyclic, i.e. connections between nodes in the network do not form a cycle. MLP consist of three or more layers of nodes: the input layer, at least one hidden layer, and an output layer.

Nodes have an activation function which defines the output(s) of that node given the input(s). In MLP, activation functions tend to be non-linear with common choices being the rectifier ($f(x) = x^+ = \max(0, x)$), the hyperbolic tangent ($f(x) = \tanh(x)$), or the sigmoid function ($f(x) = 1/(1 + e^{-x})$) and the neural network is trained using a method known as backpropagation.

Briefly, it means that the gradient of the error function is calculated first at the output layer and then distributed back through the network taking into accounts the weights and biases associated with the edges and connections in the network. Gajowniczek and Ząbkowski [32] and Zufferey et al. [31] are two recent examples of the use of MLP to individual smart meter data both with one hidden layer.

Gajowniczek and Ząbkowski [32] had 49 perceptrons in the input layer, 38 perceptrons in the hidden layer and the 24 perceptrons in the output layer to coincide with hourly load forecasts. However, Gajowniczek and Ząbkowski [32] was tried on load profile on one household where many details such as occupant number, list of appliances were known. Of course, such information is not freely available.

Zufferey et al. [31], on the other hand, tested a MLP forecast on a larger number of households and found that 200 perceptrons in the hidden layer was a reasonable trade-off between accurate predictions and reasonable computation time. They also found that the inclusion of temperature had limited influence on forecast accuracy, which is similar to the findings of Haben et al. [1] using time-series methods.

2.1.4.3 Long-Short-Term-Memory

Even more recently, Kong et al. [33] used a type of recurrent neural network (RNN) known as the long short-term memory (LSTM) RNN.

These types of models have been successfully used in language translation and image captioning due to the their architecture; since this type of RNN have links pointing back (so they may contain directed cycles, unlike the neural networks discussed before), the decision they make at a past time step can have an impact on the decision made at a later time step.

In this way, they are able to pick up temporal correlation and learn trends that are associated with human behaviour better than traditional feed-forward neural networks. When compared to some naive forecasting algorithms such as the similar day forecast as well as some machine learning algorithms, Kong et al. [33] found that LSTM network was the best predictor for individual load profiles, although with relatively high errors (MAPE, see Sect. 2.2.1, was still about 44% in the best case scenario for individual houses).

The LSTM network that has been implemented in Kong et al. [33] has four inputs: (i) the energy demand from the K past time steps, (ii) time of day for each of the past K energy demand which is one of 48 to reflect half hours in a day, (iii) day of the week which is one of 7, (iv) a binary vector that is K long indicating whether the day is a holiday or not. Each of these are normalised. The energy is normalised using the min-max normalisation.[8]

The normalisation of the last three inputs is done using one hot encoder.[9] The LSTM network is designed with two hidden layers and 20 nodes in each hidden layer. The MAPE (see Sect. 2.2) is lowest for individual houses when LSTM network is used when $K = 12$ though admittedly the improvement is small and bigger improvements came from changing forecasting algorithms.

2.1.4.4 Support Vector Machines

Support Vector Machines (SVM) are another commonly used tool in machine learning, though usually associated with classification. As explained in Dunant and Zufferey [28], SVM classify input data by finding the virtual boundary that separates different clusters using characteristics which can be thought of the features. This virtual boundary is known as the hyper-plane. Of course, there may exist more than one hyper-plane, which may separate the data points. Thus the task of an SVM is to find the hyper-plane such that the distance to the closest point is at a maximum.

We may then use the SVM for regression (SVR) to forecast load as it has been done in Humeau et al. [27] and Vrablecová et al. [34]. In this case, instead of finding the function/hyper-plane that separates the data, the task of the SVR is to find the function that best approximates the actual observations with an error tolerance ϵ. For non-linear solutions, the SVR maps the input data into a higher dimensional feature space.

Humeau et al. [27] used both MLP and SVR to forecast load of single household and aggregate household using data from the Irish smart meter trials. The goal was to create an hour ahead forecast and 24 h ahead forecast. The features used included

[8] Suppose observations are denoted by $\boldsymbol{x}$. These can be normalised with respect to their minimum and maximum values as follows: $z = \frac{\boldsymbol{x} - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})}$.

[9] The one hot encoder maps each unique element in the original vector that is K long to a new vector that is also K long. The new vector has values 1 where the old vector contained the respective unique element and zeros otherwise. This is done for each unique element in the original vector.

hour of the day and day of the week in calendar variables and the load from the past three hours as well as temperature records in Dublin.

In order to deal with variability and to give an indication of the evolution of load, the authors also add load differences, i.e. $L_i - L_{i-1}$ and $L_i - 2L_{i-1} + L_{i-2}$. Humeau et al. [27] noticed that for individual households, both for the hour ahead and the 24 h ahead, the linear regression outperforms the SVR which the authors do not find surprising. The effectiveness lies in the aggregated load cases where the internal structure, which the SVR is designed to exploit, is clearer.

More recently, Vrablecová et al. [34] also used SVR method to forecast load from the Irish smart meter data. Many kernel functions, which map input data into higher dimensions, were tried. The best results were found using radial basis function kernels and it was noted that sigmoid, logistic and other nonparametric models had very poor results. For individual households, SVR was not found to be the best methods.

Thus, from the reviewed literature we conclude that, while SVR is a promising algorithm of forecasting aggregated load, the volatility in the data reduces its effectiveness when it comes to forecasting individual smart meter data.

2.2 Forecast Errors

As more and more forecasting algorithms become available, assessing how close the forecast is to the truth becomes increasingly important. However, there are many ways to assess the accuracy of a forecast depending on the application, depending on the definition of accuracy and depending on the need of the forecaster. Since one of the earlier comparisons of various forecasting algorithms by Willis and Northcote-Green [35], competitions and forums such as "M-competitions" and "GEFcom" have been used to bring researchers together to come up with new forecasting algorithms and assess their performance. As noted by Hyndman and Koehler [36] and Makridakis and Hibon [37], these competitions help set standards and recommendations for which error measures to use.

2.2.1 Point Error Measures

The **mean absolute percentage error (MAPE)** defined as

$$MAPE = \frac{100\%}{N} \sum_{i=1}^{N} \left| \frac{f_i - a_i}{a_i} \right|, \tag{2.10}$$

where $\boldsymbol{f} = (f_1, \ldots, f_N)$ is the forecast and $\boldsymbol{a} = (a_1, \ldots, a_N)$ is the actual (observed) load, is one of the most common error measures in load forecasting literature.

It is scale independent, so it can be used to compare different data-sets [36]. It is advantageous because it has been historically used and thus often forms a good benchmark. It is also simple and easily interpreted. However, it is not without flaws. As noted in Hyndman and Koehler [36] if $\exists\, i, a_i = 0$, MAPE is undefined. Furthermore, as a point error forecasts, it suffers from the double penalty effect which we shall explain later. In this book, we adopt a common adjustment that allows for data points to be zero and define the MAPE to be

$$MAPE = 100\% \frac{\sum_{i=1}^{N} |f_i - a_i|}{\sum_{i=1}^{N} a_i} \tag{2.11}$$

where $\boldsymbol{f} = (f_1, \ldots, f_N)$ is the forecast and $\boldsymbol{a} = (a_1, \ldots, a_N)$ is the actual (observed) load,

The **Mean Absolute Error (MAE)** is also similarly popular due to its simplicity, although it is scale dependent. We define it as follows

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |f_i - a_i|. \tag{2.12}$$

However, similar to the MAPE, the MAE also susceptible to doubly penalising peaky forecasts. As pointed out in Haben et al. [1], the scale-dependence of MAE can be mitigated by normalising it. In this way the authors were able to compare feeders of different sizes. In our case normalising step is not necessary as we compare different algorithms and look for an average over the fixed number of households.

Haben et al. [19] consider a p-norm error measure. We define it here by

$$E_p \equiv ||\boldsymbol{f} - \boldsymbol{a}||_p := \left(\sum_{i=1}^{N} |f_i - a_i|^p \right)^{\frac{1}{p}}, \tag{2.13}$$

where $p > 1$. In this book, as in Haben et al. [19], we take $p = 4$ as it allows larger errors to be penalised more and smaller errors to be penalised less. Thus, we will use E_4 in order to focus on peaks.

Lastly, we also consider the **Median Absolute Deviation (MAD)** which is defined the median of $|f_i - a_i|$ for $i = 1, \ldots, N$. The MAD is considered more robust with respect to outliers than other error measures.

2.2.2 Time Shifted Error Measures

In this section we present some alternatives to the standard error measures listed in Sect. 2.2.1. Suppose the actual profile has just one peak at time k and the forecast also has just one peak at time $k \pm ß$ where $i > 0$ is small (say maybe the next or previous time unit). The point error measures in Sect. 2.2.1 penalise the forecast twice: once for not having the peak at time k and a second time for having a peak at time $k + i$. A flat forecast (fixed value for all time) under these circumstances would have a lower error even though in practice it is not a good or useful to the forecaster. To deal with such problems, it would be intuitively beneficial to be able to associate a shifted peak to a load, including some penalty for a shift, as long as it is within some reasonable time-window. In this section we discuss two ways of comparing time series: dynamic time warping and permuted (so called adjusted) errors.

2.2.2.1 Dynamic Time Warping

Dynamic Time Warping (DTW) is one of the most common ways of comparing two time series not necessarily of equal length. It has been used in automatic speech recognition algorithms.

Dynamic Time Warping calculates an optimal match between two sequences given the some condition: suppose you have time series $\boldsymbol{x}$ and $\boldsymbol{y}$ with length N and M, respectively. Firstly, the index from X can match with more than one index of Y, and vice versa. The first indices must match with each other (although they can match with more than one) and last indices must match with each other.

Secondly, monotonicity condition is imposed, i.e. if there are two indices, $i < j$ for $\boldsymbol{x}$ and if i matches to some index l of $\boldsymbol{y}$, then j cannot match to an index k of $\boldsymbol{y}$ where $k < l$. One can make this explicit: an (N, M)-warping path is a sequence $p = (p1, \ldots, p_L)$ with $p_\ell = (n_\ell, m_\ell)$ for $\ell \in \{1, \ldots, L\}$, $n_\ell = \{1, \ldots, N\}$, and $m_\ell = \{1, \ldots, M\}$ satisfying: (i) boundary condition: $p_1 = (1, 1)$ and $p_L = (N, M)$, (ii) monotonicity condition: $n_1 \leq n_2 \leq \cdots \leq n_L$ and $m_1 \leq m_2 \leq \cdots \leq m_L$ and (iii) step size condition: $p_{\ell+1} - p_\ell \in \{(1, 0), (0, 1), (1, 1)\}$ for $\ell \in \{1, 2, \ldots, L - 1\}$.

This is useful to compare electricity load forecast with observation as it allows us to associate a forecast at different time points with the load and still give it credit for having given useful information. However, there are some obvious downsides; the time step in the forecast can be associated with more than one observation which is not ideal. Moreover, if we were to draw matches as lines, lines cannot cross. This means that if we associate the forecast at 7.30 am with observation 8 am, we cannot associate forecast at 8 am with the observation at 7.30 am.

2.2.2.2 Adjusted Error Measure

The adjusted error concept and algorithm to calculate it introduced by Haben et al. [19] addresses some of the issues. The idea of this permutation based error measure was used to create the forecasts discussed in Sect. 2.1.3. Given a forecast and an observation over a time period, both sequences have the same length. Indices are fully (or perfectly) matched. Each index in one sequence is matched only to one index in the other sequence and any permutations within some tolerance window are allowed. The error measure assumes that an observation has been *well enough* forecasted (in time) if both the forecast and the observation are within some time window ω. We define the **Adjusted p-norm Error (ApE)** by

$$\hat{E}_p^{\omega} = \min_{P \in \mathcal{P}} ||P\boldsymbol{f} - \boldsymbol{x}||_p, \tag{2.14}$$

where $\mathcal{P}$ again represents the set of restricted permutations. In Haben et al. [19], the minimisation is done using the Hungarian algorithm, but faster implementations are possible using Dijkstra's shortest path algorithm or topological sort as discussed by Charlton et al. [22].

While these may be intuitively suitable for the application at hand, they are computationally challenging and results are not easily conveyed to a non-specialist audience. The ApE also does not satisfy some properties of metrics In Voß et al. [38], a distance based on the adjusted p-norm error, the local permutation invariant (LPI) is formalised. Let $\mathcal{P}_n$ denote the set of $n \times n$ permutation matrices. Let $\mathcal{L}_n^{\omega} = \{P = (p_{ij} \in \mathcal{P}_n : p_{ij} = 1 \Rightarrow |i - j| \leq \omega\}$. Then the function $\delta : \mathbb{R}^n \times \mathbb{R}^n$, such that

$$\delta(x, y) = \min\{\|Px - y\| : P \in \mathcal{L}_n^{\omega}\}$$

is an LPI distance induced by the Euclidean norm $\|.\|$.

2.3 Discussion

Here, we have looked at several studies that review various load forecasting algorithms and how to assess and compare them. Clearly the literature on how to do this well for individual load profiles is an emerging field. Furthermore, only recently the studies regarding forecasting techniques for individual feeders/households comparing both machine learning algorithms and statistical models became available. This has been done in past for aggregated or high voltage systems [2, 9], but only recently for individual smart meter data.

Linear regression is widely used for prediction, on its own or in combination with other methods. As the energy is used by all throughout the day, and people mostly follow their daily routines, autoregressive models, including ARMA, ARIMA and ARIMAX, SARMA and SARIMAX models are popular and perform well in

predicting peaks. Also triple exponential smoothing models, such as Holt-Winters-Taylor with intraday, intraweek and trend components are good contenders, while kernel-density estimators less so for individual households data. As expected, they work better on higher resolutions or aggregated level, where the data is smoother.

Permutation-based methods are relatively recent development. They attempt to mitigate a 'double penalty' issue that standard errors penalise twice slight inaccuracies of predicting peaks earlier or later. Instead of taking a simple average across time-steps, with their adjust averaging they try to obtain a better 'mean sample', and therefore to take into account that although people mostly follow their daily routine, for different reasons their routine may shift slightly in time.

Finally, multi-layer perceptron and recurrent neural network appear to cope well with the volatility of individual profiles, but there is a balance of computational and time complexity and improvement, when comparing them with simpler, explainable and faster forecasts.

There are yet many problems to be solved, such as the question of which features are important factors in individual load forecasting. While in general, there is an agreement that the time of the day, the day of the week and season are important factors for prediction, temperature, which is an important predictor of aggregate level seems to be not very relevant for prediction, (except for households with electric storage heating), due to the natural diversity of profiles being higher than temperature influence.

We have also looked at the standard error measures in order to evaluate different forecasting algorithms. While percentage errors such as are widely used as being scale-free and using absolute values they allow for comparison across different data-sets, we discuss the limitations: a small adjustment allows MAPE to cope with time-series with zero values, but it still suffers from a double penalty problem—trivial, straight line mean forecasts can perform better than more realistic, but imperfect 'peaky' forecasts similarly to MAE. MAD error measure is introduced for error distributions that might be skewed, and 4-norm measure highlights peak errors. Alternatives that use time-shifting or permutations are also mentioned, as they can cope with a double penalty issue, but are currently computationally costly.

References

1. Haben, S., Giasemidis, G., Ziel, F., Arora, S.: Short term load forecasting and the effect of temperature at the low voltage level. Int. J. Forecast. (2018)
2. Alfares, H.K., Nazeeruddin, M.: Electric load forecasting: literature survey and classification of methods. Int. J. Syst. Sci. **33**(1), 23–34 (2002)
3. Taylor, J.W., Espasa, A.: Energy forecasting. Int. J. Forecast. **24**(4), 561–565 (2008)
4. Hong, T., Fan, S.: Probabilistic electric load forecasting: a tutorial review. Int. J. Forecast. **32**(3), 914–938 (2016)
5. Singh, R.P., Gao, P.X., Lizotte, D.J.: On hourly home peak load prediction. In: 2012 IEEE Third International Conference on Smart Grid Communications (SmartGridComm), pp. 163–168 (2012)

6. Taieb, S.B., Taylor, J., Hyndman, R.: Hierachichal probabilistic forecasting of electricity demand with smart meter data. Technical report, Department of Econometrics and Business Statistics, Monash University (2017)
7. Rowe, M., Yunusov, T., Haben, S., Holderbaum, W., Potter, B.: The real-time optimisation of dno owned storage devices on the lv network for peak reduction. Energies **7**(6), 3537–3560 (2014)
8. Gerwig, C.: Short term load forecasting for residential buildings—an extensive literature review. In: Neves-Silva, R., Jain, L.C., Howlett, R.J. (Eds.), Intelligent Decision Technologies, pp. 181–193. Springer International Publishing, Cham (2015)
9. Moghram, I., Rahman, S.: Analysis and evaluation of five short-term load forecasting techniques. IEEE Trans. Power Syst. **4**(4), 1484–1491 (1989)
10. Charlton, N., Singleton, C.: A refined parametric model for short term load forecasting. Int. J. Forecast. **30**(2), 364–368 (2014)
11. Haben, S., Giasemidis, G.: A hybrid model of kernel density estimation and quantile regression for gefcom2014 probabilistic load forecasting. Int. J. Forecast. **32**(3), 1017–1022 (2016)
12. Weron, R.: Modeling and Forecasting Electricity Loads and Prices: A Statistical Approach, vol. 403. Wiley (2007)
13. Taylor, J.W., McSharry, P.E.: Short-term load forecasting methods: an evaluation based on european data. IEEE Trans. Power Syst. **22**(4), 2213–2219 (2007)
14. Box, G., Jenkins, G., Reinsel, G.: Time Series Analysis: Forecasting and Control. Wiley Series in Probability and Statistics. Wiley (2008)
15. Taylor, J.W.: Short-term electricity demand forecasting using double seasonal exponential smoothing. J. Oper. Res. Soc. **54**(8), 799–805 (2003)
16. Arora, S., Taylor, J.W.: Short-term forecasting of anomalous load using rule-based triple seasonal methods. IEEE Trans. Power Syst. **28**(3), 3235–3242 (2013)
17. Taylor, J.W.: Triple seasonal methods for short-term electricity demand forecasting. Eur. J. Oper. Res. **204**(1), 139–152 (2010)
18. Arora, S., Taylor, J.W.: Forecasting electricity smart meter data using conditional kernel density estimation. Omega **59**, 47–59 (2016)
19. Haben, S., Ward, J., Greetham, D.V., Singleton, C., Grindrod, P.: A new error measure for forecasts of household-level, high resolution electrical energy consumption. Int. J. Forecast. **30**(2), 246–256 (2014)
20. Schrijver, A.: Combinatorial Optimization: Polyhedra and Efficiency. Springer, Berlin, Heidelberg (2002)
21. Munkres, J.: Algorithms for the assignment and transportation problems. J. Soc. Ind. Appl. Math. **5**, 32–38 (1957)
22. Charlton, N., Greetham, D.V., Singleton, C.: Graph-based algorithms for comparison and prediction of household-level energy use profiles. In: Intelligent Energy Systems (IWIES), 2013 IEEE International Workshop on, pp. 119–124. IEEE (2013)
23. Valgaev, O., Kupzog, F., Schmeck, H.: Designing k-nearest neighbors model for low voltage load forecasting. In: 2017 IEEE Power Energy Society General Meeting, pp. 1–5 (2017)
24. Voß, M., Haja, A., Albayrak, S.: Adjusted feature-aware k-nearest neighbors: Utilizing local permutation-based error for short-term residential building load forecasting. In: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), pp. 1–6 (2018)
25. Ekonomou, L.: Greek long-term energy consumption prediction using artificial neural networks. Energy **35**(2), 512–517 (2010)
26. Rudin, C., Waltz, D., Anderson, R.N., Boulanger, A., Salleb-Aouissi, A., Chow, M., Dutta, H., Gross, P.N., Huang, B., Ierome, S., et al.: Machine learning for the new york city power grid. IEEE Trans. Pattern Anal. Mach. Intell. **34**(2), 328 (2012)
27. Humeau, S., Wijaya, T.K., Vasirani, M., Aberer, K.: Electricity load forecasting for residential customers: exploiting aggregation and correlation between households. In: Sustainable Internet and ICT for Sustainability (SustainIT), 2013, pp. 1–6. IEEE (2013)

28. Dunant, J., Zufferey, T.: Investigation of forecasting techniques in distribution grids. Semester project of Power System Laboratory, ETHZ (2018)
29. O'Neil, C., Schutt, R.: Doing Data Science: Straight Talk From the Frontline. O'Reilly Media, Inc. (2013)
30. Wijaya, T.K., Vasirani, M., Humeau, S., Aberer, K.: Cluster-based aggregate forecasting for residential electricity demand using smart meter data. In: Big Data (Big Data), 2015 IEEE International Conference on, pp. 879–887. IEEE (2015)
31. Zufferey, T., Ulbig, A., Koch, S., Hug, G.: Forecasting of smart meter time series based on neural networks. In: International Workshop on Data Analytics for Renewable Energy Integration, pp. 10–21. Springer (2016)
32. Gajowniczek, K., Ząbkowski, T.: Short term electricity forecasting using individual smart meter data. Proced. Compu. Sci. **35**, 589–597 (2014)
33. Kong, W., Dong, Z.Y., Jia, Y., Hill, D.J., Xu, Y., Zhang, Y.: Short-term residential load forecasting based on lstm recurrent neural network. IEEE Trans. Smart Grid (2017)
34. Vrablecová, P., Ezzeddine, A.B., Rozinajová, V., Šárik, S., Sangaiah, A.K.: Smart grid load forecasting using online support vector regression. Comput. Electr. Eng. **65**, 102–117 (2018)
35. Willis, H.L., Northcote-Green, J.: Comparison tests of fourteen distribution load forecasting methods. IEEE Trans. Power Appar. Syst. **6**, 1190–1197 (1984)
36. Hyndman, R.J., Koehler, A.B.: Another look at measures of forecast accuracy. Int. J. Forecast. **22**(4), 679–688 (2006)
37. Makridakis, S., Hibon, M.: The m3-competition: results, conclusions and implications. Int. J. Forecast. **16**(4), 451–476 (2000)
38. Voß, M., Jain, B., Albayrak, S.: Subgradient methods for averaging household load profiles under local permutations. In: The 13th IEEE PowerTech 2019 (2019)