Day-Ahead Building Power Demand Forecasting in Smart Grids

Zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS

von der KIT-Fakultät für
Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie

genehmigte

DISSERTATION

von

Oleg Valgaev

Tag der mündlichen Prüfung:	13.12.2022
Referent:	Prof. Dr. Hartmut Schmeck
Korreferent:	Prof. Dr. Oliver Grothe

Abstract

In this dissertation, we propose a novel *day-ahead load forecasting method* that can be applied without manual setup on any building and is more accurate than currently existing methods for predicting low-voltage loads. Day-ahead predictions allow a smart grid to mitigate the volatility of decentralized renewable generators locally, by using demand flexibilities of the buildings located in the area. Historically, power system operators forecast low-voltage demand for the upcoming day using standard load profiles. While this basic method is effective for large consumer aggregations, it lacks accuracy when applied on smaller loads and the flexibility to consider modern energy equipment in the buildings. More advanced forecasting methods that exist for the high-voltage level, rely on manual fine-tuning and can be used only in singular cases. Our aim is to develop a method that can replace standard load profiles for predicting low-voltage loads on a *wide scale* – a method that can be applied on numerous individual buildings of different size and type without any explicit knowledge of them.

We formulate the wide-scale day-ahead load forecasting problem in low-voltage domain studying various loads and their characteristics. Considering time-series nature of the data and its nonstationarity, we combine nonparametric functional data analysis with the theory of statistical learning to introduce a univariate autoregressive *functional neighbor model* with corresponding forecasting algorithm. Additionally, we present an extension that allows to consider exogenous variables which can affect the consumption of a given building. We evaluate the model on an extensive, publicly available dataset of loads and use inferential statistics comparing our model to numerous references.

The main result of this work is a load forecaster that can be universally applied in a distribution grid using historic load measurements and, optionally, further inputs. Statistical analysis shows that our model can be expected to be significantly more accurate than standard load profiles and more sophisticated approaches based on classical time series analysis and machine learning. Even for the largest loads, our method can be expected to be at least 39% more accurate than standard load profiles that were designed to predict larger aggregations of end-consumers. Therefore, given mass adoption of smart meters, the proposed functional neighbor model can replace standard load profiles that were used in power systems since their inception. Improved accuracy and flexibility of the proposed method facilitates various smart grid applications that can increase the efficiency of the existing distribution system infrastructure and aid accommodating renewable energy generators.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Prof. Hartmut Schmeck from KIT Karlsruhe Institute of Technology and Dr. Friederich Kupzog from AIT Austrian Institute of Technology. Their mentorship, assistance, and inspiration have been invaluable throughout my doctoral journey. They have helped refine my ideas, offered useful advice, and encouraged me to strive for excellence.

Special thanks go to Prof. Oliver Grothe and Prof. Andreas Oberweis for co-examining this work, forming the thesis committee, and investing their time and effort.

I am also grateful to my colleagues at the Institute of Applied Informatics and Formal Description Methods of the KIT Karlsruhe Institute of Technology and the Energy Department of AIT Austrian Institute of Technology. Their constructive feedback and fruitful discussions have greatly contributed to this work. In particular, I want to thank Roman Schwalbe for his technical assistance and Andrea Reichenauer for her support with administrative tasks.

To my friends Dimitris, Mitchel, Alexandra, Marie, Oriol, Ricardo, Richard, Christina and Simon, thank you for being a source of motivation, comfort, and joy during this challenging time.

Finally, I dedicate this thesis to my mother, Elena Valgaeva, and her husband, Sigmund Schiretz. Their steadfast love and support have always guided me. They have shown me the importance of ambition, the need for perseverance, and the power of determination that were necessary to complete this thesis.

Financial Support

The doctoral research has been carried out in the context of an agreement on joint doctoral supervision between AIT Austrian Institute of Technology GmbH, and KIT Karlsruhe Institute of Technology.

The study was conducted as a part of Smart City Demo Aspern project funded within the program Smart Cities by the Austrian Climate and Energy Fund under project number 846141.

Contents

Part I

Introduction

Electric power grid is one of the largest interconnected systems on the planet. It currently undergoes a major transformation striving to reduce its ecological imprint and cover the ever-growing consumption more efficiently. What persists, is its fundamental operating principle of maintaining the equilibrium between energy supply and demand. Unlike other resources and despite considerable research efforts, electricity can be stored only to a limited extent. The balance has to be maintained through a robust control where supply and demand forecasts play an essential role.

The power system transformation affects, both, supply and demand side of the European grid. A rising share of renewable generation in the EU is partly referable to the installation of numerous *decentralized energy generators*. However, the existing power system infrastructure was originally designed for a steady and projectable supply by a relatively small number of large centralized power plants. On the demand side, the consumers become more divers following increasing adoption of batteries, electric mobility, and smart buildings capable of adjusting their consumption.

These developments impose new challenges on the distribution network infrastructure that hosts decentralized energy generators and a vast number of end-consumers. The distribution network was engineered for a unidirectional power flow from the bulk high-voltage transmission system to the low- and mid-voltage end-consumers. The possibility of congestion was minimized by design, given only basic monitoring. Following the transformation of the European power system, the resulting energy-flows will require active local control to avoid cost-prohibitive re-dimensioning of the existing infrastructure.

The concept of a smart grid aims to improve controllability of the distribution system and allows to use the existing infrastructure more efficiently. Installation of advanced information and communication technology allows system operators to anticipate local congestions on a daily basis. Predicting the generation and demand day-ahead and at the level of single buildings in a given area allows to take control measures widely discussed under the term *demand response* [Sia14].

We initiate our study motivating the development of a wide-scale day-ahead building load forecasting method – a method that can be applied on numerous individual buildings of different size and type without any explicit knowledge of them (Chapter 1). We place this work within the context of power engineering focusing on the smart grid notion and highlight several use-cases for such forecasting method (Chapter 2). Concluding the first part of the thesis, we highlight the main contributions and provide an outline of the research presented in this work (Chapter 3).

1 Motivation

Until recently, there was a limited interest in short-term[1] building load forecasting. Power engineers assumed the consumption to be uncontrollable and to follow a steady pattern defined in advance. Forecasting was done globally, at the high-voltage level, where only large aggregations of the consumers are considered given limited information about them. In the future, we will need to balance the increasing share of decentralized renewable energy supply *locally* – predicting building power demand in a given area.

The wide-scale introduction of smart metering is on the way and provides new possibilities for load forecasting in a distribution system. In 2024, 77% of the consumers in the EU will be equipped with a smart meter [ET20]. This device delivers a daily load curve of the consumer with a subhourly resolution. While in the past, only a general information about the consumer (e.g., type, annual consumption) was available to the power system operators, smart-meter data analysis is becoming a separate research field of which load forecasting is one of the main applications [WCHK18].

To accommodate decentralized energy generators and consider eminent changes in the nature of demand, power engineers need to develop and install new equipment and elaborate on the corresponding control mechanisms. Many of the proposed demand response concepts rely on a local load-forecast down to the level of single buildings [MK10]. However, currently existing models are inadequate for a *wide-scale application* – i.e., forecasting the disaggregated load of numerous individual buildings of different type and size without any manual setup. The problems discussed in this section motivate the development of a novel technique for predicting building loads on a wide scale using smart-meter data.

[1] While there is no clear definition for the short-term horizon, in the load forecasting literature, short-term is regarded as a horizon from one to several hours up to one to several days ahead [MTAR15, RK15].

1.1 Problem – Accuracy of the Forecast

At present, power engineers assume that the end-consumer load follows a predefined pattern named *standard load profile (SLP)* common for its general type (enterprise, residential, office, etc.) and scaled by its annual consumption [Bun82]. Such profiles represent a daily average load curve typical for the given region and consumer type. Using only basic data about the consumers, SLPs deliver accurate predictions in a traditional grid for large aggregations of loads.

However, there are several reasons why SLPs are defective in predicting the consumption of individual buildings. Though accurate for aggregations of hundreds of loads[2], they only rudimentary reflect the diversity and highly stochastic nature of the demand of a single building – especially one of a smaller size (Figure 1.1). At the same time, the general consumer type, for which an SLP is assigned, cannot always be unambiguously identified.

An inapt forecast can cause critical problems in the distribution system. A substantial prediction error can lead to a significant performance loss of predictive control. When applied to numerous buildings, a forecast improvement of a single per cent can lead to sizable cost savings [SSM16]. With increasing share of distributed generation and storage, a congestion can occur at any moment, which makes accurate short-term load forecasts fundamental for the operation of smart buildings and distribution systems.

1.2 Problem – Change in Load Characteristics

Building loads are becoming more intermittent and diverse. Until now, facilities were often thermally heated and consumed electricity according to a steady occupancy pattern modeled by few standard profiles. Following the EU decarbonization strategy, we increasingly see buildings equipped with electrical *heating, ventilation, and air-conditioning (HVAC)*, *photovoltaic (PV)* panels and storages [Eur18]. In the future, the load profile of a building will vary notably depending on the day and installed equipment.

Power demand of new and retrofitted facilities can often be linked to *exogenous variables* such as weather or a control signal. For instance, solar irradiation affects the net load of a building with a PV-panel. A dramatic temperature change can affect the daily curve given electrical HVAC-equipment. More generally, a *smart building* can adjust its net consumption following a control signal that is related to weather or an input from a demand response application. These variables need to be considered by the load model of such building.

[2] We can expect a 10% forecast error for aggregations that include over 400 low-voltage end-consumers [Ber00].

Surging installations of PV-panels, electrical HVAC and other energy equipment change the consumption characteristics at lower levels of aggregation. As a result, building loads are no longer reflected by few standard profiles. Increasingly, distribution system operators need a more versatile model that can consider the increasing building load diversity and possible dependency on external inputs.

1.3 Problem – Forecasting on Wide Scale

Forecasting building loads on a wide scale implies predicting the disaggregated power demand of numerous individual buildings without any explicit knowledge of them. This restricts the usage of many established time series forecasting techniques. A building load forecaster for a wide-scale application cannot rely on any manual setup and has to work with limited data. Moreover, the model has to reflect the diversity, volatility and nonstationarity of the time series representing electricity consumption of buildings connected to the distribution grid. Until now, standard profiles is the only approach used to predict the loads in a distribution system on a wide scale.

In contrast, short-term load forecasting at the transmission system level is considered a solved problem. There exist myriad models accurate within a single percent margin [FGSC19]. Naturally, a transmission system operator has only one model that needs to be set up and maintained. Such model, often considers numerous variables, years of historical data and relies on manual setup and fine-tuning.

Similar methods can also be accurate predicting intraday load of a specific building. Numerous applications of sophisticated time series and machine learning techniques have been used to do so [BZN+19]. As on the transmission system level, they are set up manually and often applied to larger buildings with a steady consumption pattern.

However, when the same models are applied to more volatile, nonstationary loads (e.g. smaller buildings, single homes and enterprises), they fail to reflect the volatility of consumption. As a result, even simple heuristic models including the SLPs can, still, be more accurate than sophisticated machine learning approaches [HGP15, VKS20]. Further, years of training data and different inputs might not be available for each building in the distribution system.

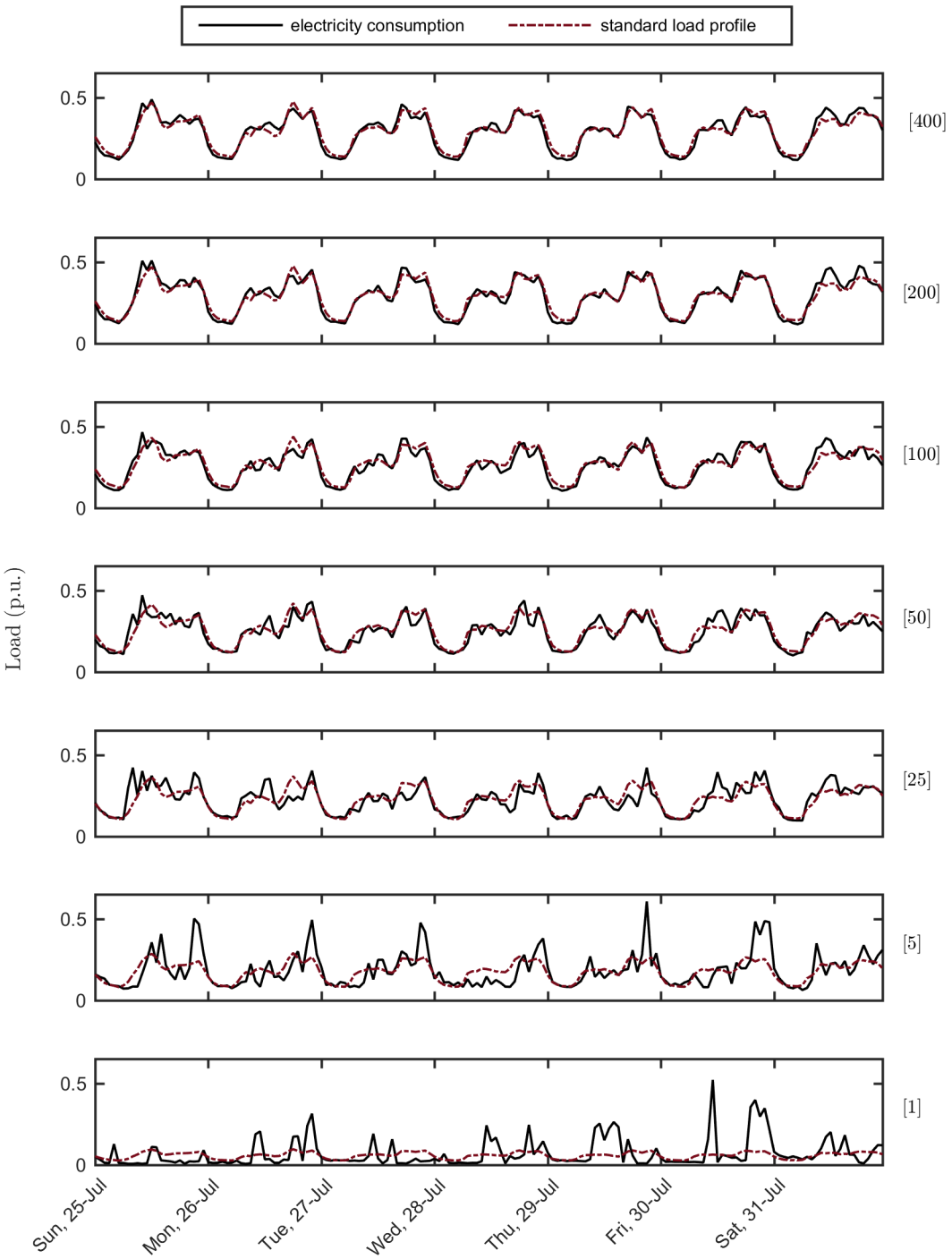

Figure 1.1: Electricity consumption at various levels of load aggregation. The number in parenthesis denotes the number of aggregated residential buildings taken from a public smart-meter dataset [Arc16]. Time series were normalized individually by their peak value. Observe that larger aggregations follow a steady pattern that is approximated well by a standard load profile. At the same time, the electricity consumption becomes more volatile with the decreasing aggregation size. For smaller aggregations, the standard load profile does not reflect the volatility of the power demand. For instance, the weekly pattern of single family home ([1]) is hard to identify and, consequently, the forecasting becomes more challenging.

2 Aim and Scope

This work aims to create a model for predicting day-ahead electrical load curves of buildings on a *wide scale*[1] using smart-meter data. The aforementioned problems motivate the development of a novel method for predicting *local loads*[2] in the electricity distribution system. The challenge is to develop a model that we can apply, with no manual configuration and adjustment, on a wide range of old and newly commissioned buildings. The model has to work with scarce training data, because abundant historical measurements might not be available for each building. An example of such model is the currently used standard load profiles approach that requires only minimal knowledge about the buildings. Accordingly, we formulate the following aim of our study.

Research aim. *Provide an alternative to the standard load profiles for the day-ahead forecasting of local loads on a wide scale.*

For this study, we assume that smart-meter data with hourly resolution is available for each building and focus on deterministic day-ahead load curve prediction for individual buildings and their aggregations. *Day-ahead load forecast* is the key component for operation of a smart grid – an electricity supply network that uses digital communication technology to detect and react to local changes in usage. Within the notion of demand response, there are different concepts that use smart-grid infrastructure and rely on a day-ahead prediction to improve power system efficiency. In this work, we focus on the following use cases.

[1] In this study, *wide-scale forecast* implies the prediction of numerous disaggregated loads of different type and size (Definition 7.2.1). Note that we only consider a disaggregation down to the level of an individual building. Further disaggregation to the level of single electrical devices common for nonintrusive load monitoring research field is beyond the scope of this thesis.

[2] In this study, we use the term *local load* for the loads connected to the distribution system and that include only few end-consumers located in the same geographical area such as buildings (Definition 7.2.2). Though local loads include, but are not limited to, buildings, the terms *local load* and *building load* are used interchangeably in this thesis.

2.1 Use Case – Building Energy Management

An accurate forecast is fundamental for the operation of a *building energy management system (BEMS)* and consumption optimization of the facility. Anticipating the load day-ahead allows to reduce energy costs by scheduling storage capacity [CZAS12, KCBG13, MDRH+18]. Further, such prediction allows to quantify the available flexibility that can be sold on a day-ahead market over an aggregator as it was often demonstrated [Asp, OLM+18]. On a global scale, scheduling of building load flexibilities facilitates far-reaching demand management concepts such as the usage of variable electricity prices [YOHS18].

2.2 Use Case – Virtual Power Plants

Smart grids, combined with market liberalization ongoing in the EU, allow new business models for the electricity market [NA16]. Those are often based on pooling smaller[3] consumers and producers into a single unit named *virtual power plant (VPP)*. The aggregated load flexibility of the resulting entity can be sold, among other options, on a day-ahead market [CSVM16]. The VPP-operators rely on the day-ahead predictions of the consumers and producers in the pool to optimize the cost of supply, anticipate energy purchases and estimate the amount of flexibility which they can monetize [NHG17, YFLL19].

2.3 Use Case – Microgrids and Energy Communities

Microgrid is another concept within the smart grid notion that relies on day-ahead forecasts [NL14]. Instead of maintaining the power balance globally, through a wholesale market, the energy can be managed within the local *energy community* [Dira, Dirb]. A *microgrid central controller* is responsible for maintaining the power balance, dispatching generation and load flexibility as well as managing the power exchange with the distribution grid [PKG14b, PKG14a]. At present, there exist numerous microgrids demonstrating the concept [MBC16]. For operational purposes, the controller has to forecast the output of decentralized energy generators and the load which often consists of the few buildings within the microgrid. Moreover, anticipating power surplus and load flexibility allows a microgrid to participate on the day-ahead market either directly or via a VPP [LXT16, YHAG19, FCDM+16]. Therefore, an accurate day-ahead forecast directly benefits microgrid operators and the energy communities.

[3] Load size can be expressed in terms of annual consumption and is linked to the number of aggregated end-consumers.

3 Research Overview

Adoption of smart metering provides new opportunities for predicting building power demand in distribution systems. Following the motivation and scope of this thesis, we formulate the research question:

Research question. *How can we use smart-meter data to predict day-ahead electricity consumption of individual buildings and their aggregations on a wide scale?*

With our research, we aim to provide an alternative to the standard load profiles that are currently used for day-ahead load forecasting in absence of any detailed knowledge about the buildings. Towards this objective, we provide three contributions to the field of power engineering and applied statistics as described in the next section. Moreover, we made several publications over the course of our study which we list subsequently. We conclude this chapter by providing an outline of the dissertation.

3.1 Contributions

In this work, we propose and validate a novel method for predicting the day-ahead load curves of buildings. Our method can be applied to any building without any manual setup requirements and is more accurate than currently existing techniques for predicting local loads. Moreover, our method can consider external inputs which we expect to affect the load of a given building. While developing the method that answers the research question, we make the following contributions.

Contribution 1. *Establish the wide-scale day-ahead local load forecasting as an area of research in distribution system operation.*

Motivated by the outlined use-cases, we formulate the wide-scale day-ahead local load forecasting problem – the problem of predicting the load curves for the upcoming day on numerous individual buildings using a before-the-meter approach. We provide a unified view on load forecasting in the distribution systems combining the perspectives of statistical learning theory, classical and functional time series analyses. Herewith, we provide a classification of the existing building load forecasting methods and their critical review placing them into the context of a wide-scale application.

Contribution 2. *Evaluate existing data-driven models in context of wide-scale day-ahead local load forecasting.*

We formulate and apply a methodology to evaluate and compare forecasting models for a wide-scale application to local loads. For evaluation, we use a scale-independent error notion specialized for quantifying the forecast accuracy for a volatile daily load curve, and we consider the error distribution with appropriate descriptive statistics. For model comparison, we apply inferential statistics with appropriate statistical tests. We demonstrate the methodology evaluating most common forecasting models on a public smart-meter dataset. To simulate a wide-scale application, we compute hundreds of thousands daily load forecasts with each model, predicting the power demand of individual buildings and their aggregations. The quantity of predictions allows us to draw statistically significant conclusions about several common data-driven models applied for wide-scale day-ahead local load forecasting.

Contribution 3. *Propose a novel forecasting methodology for predicting day-ahead building load curves on a wide scale.*

We combine nonparametric regression with functional data analysis creating a novel modular approach for data-driven multistep autoregressive prediction (*functional neighbor model*). We propose a corresponding forecasting algorithm that can be applied on a wide scale as it uses minimal data and requires no manual setup[1]. Moreover, we provide an extension that allows to consider exogenous variables. We evaluate our algorithm computing hundreds of thousands of daily forecasts and compare it to numerous reference models. The results indicate that our forecaster is significantly more accurate than the common reference models predicting building loads of any size. Even for the largest loads, our method can be expected to be at least 39% more accurate than standard load profiles that were designed to predict larger aggregations of end-consumers.

3.2 Publications

To some extent, the content of this thesis echoes our previously published work. In this section, we list the peer-reviewed publications that can be related to this thesis. These works were referenced accordingly in the text.

[1] The proposed algorithm requires three months of the most recent historical load measurements and requires neither any information about the building nor manual fine-tuning of the model parameters.

3.2.1 Journal Articles

- **[VKS20]** *Oleg Valgaev, Friederich Kupzog, and Hartmut Schmeck. "Adequacy of neural networks for wide-scale day-ahead load forecasts on buildings and distribution systems using smart-meter data." Energy Informatics 3.1 (2020): 1-17.* In this article, we investigated the usage of neural-network methodology for predicting the day-ahead load curves of local loads and evaluated its performance on a sample of single family homes and residential aggregations. The setup of the neural network architectures used as reference models in this thesis (Section 9.2.2.2) can be attributed to this article.
- **[VKS17a]** *Oleg Valgaev, Friederich Kupzog, and Hartmut Schmeck. "Building power demand forecasting using K-nearest neighbors model – practical application in Smart-City-Demo Aspern project." CIRED-Open Access Proceedings Journal 2017.1 (2017): 1601-1604.* In this article, we applied the K-nearest neighbors forecaster on the buildings from the Smart-City-Demo Aspern project, investigating the need to account for external inputs when predicting the load of smart buildings.
- **[VK16a]** *Oleg Valgaev and Friederich Kupzog. "Building power demand forecasting." it-Information Technology 58.1 (2016): 37-43.* In this article, we proposed the general forecasting methodology for power demand forecasting of smart buildings that considers scheduled demand response and the predicted PV-generation. However, over the course of our study, the initial proposal was substantially changed and restructured.

3.2.2 Conference contributions

- **[VKS17c]** *Oleg Valgaev, Friederich Kupzog, and Hartmut Schmeck. "Outlining ensemble K-nearest neighbors approach for low-voltage power demand forecasting." Proceedings of the Eighth International Conference on Future Energy Systems (e-Energy). ACM, 2017.* In this publication, we outlined the improvement of the merging step (Section 8.2.3) within the K-nearest neighbors forecasting methodology. In particular, we proposed the, so called, ensemble merger which we did not include into the current study but will investigate in more detail in our future research.
- **[VKS17b]** *Oleg Valgaev, Friederich Kupzog, and Hartmut Schmeck. "Designing K-nearest neighbors model for low-voltage load forecasting." 2017 IEEE Power & Energy Society General Meeting. IEEE, 2017.* In this publication, we investigated the usage of the k-fold cross-validation for model setup and the permutation merger discussed in Section 8.2.3.2.

- **[VKS16]** *Oleg Valgaev, Friedrich Kupzog, and Hartmut Schmeck. "Low-voltage power demand forecasting using K-nearest neighbors approach." 2016 IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia). IEEE, 2016.* In this publication, we applied the multivariate K-nearest neighbor approach and investigated the effect of the bandwidth choice for computing the day-ahead load forecasts on a small sample of local loads of different type and size.
- **[VK16b]** *Oleg Valgaev and Friederich Kupzog. "Building power demand forecasting using k-nearest neighbors model – initial approach." 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC). IEEE, 2016.* In our first publication, we proposed the usage of the multivariate K-nearest neighbor model for predicting the day-ahead loads of individual buildings.

3.3 Thesis Outline

The dissertation contains twelve consecutive chapters and is structured into four parts. After concluding the introduction, Part II locates our research on the intersection of power engineering and applied statistics. We describe and classify existing time series prediction approaches (Chapter 4) providing the necessary background for the critical review of the existing load forecasting models (Chapter 5). Moreover, we provide the necessary context on smart buildings and grids where these models are to be applied (Chapter 6). We dedicate the subsequent Part III to the methods we use for answering the research question. To formulate the wide-scale day-ahead local load forecasting problem (Chapter 7), we describe the characteristics of building loads and multistep forecasts. Additionally, we present a methodology for assessing forecast accuracy in a wide-scale application. Subsequently, we provide a solution for the forecasting problem developing a forecaster that is based on a novel *functional neighbor* approach for predicting day-ahead building load curves (Chapter 8). Finishing the methodological part, we describe how our model is evaluated together with the common reference models from the literature and detail the corresponding simulation of wide-scale building load forecasting (Chapter 9). Finally, Part IV presents (Chapter 10) and discusses (Chapter 11) the simulation results before concluding the dissertation (Chapter 12).

Part II

Background

Throughout the second part of the thesis, we provide a theoretical background for our study. In the most general sense, a time series is a sequence of data and accurately forecasting its future values can be a grand challenge depending on the application field. We begin this part with some formal definitions and an introduction to time series forecasting describing theoretical concepts and approaches for this task (Chapter 4). Next, we describe the applications to short-term load forecasting, drawing the line between different domains of a power system and focusing on building load forecasting for which we review the relevant literature (Chapter 5). We conclude this part by providing some context on smart buildings including those that provided a platform for our study (Chapter 6).

4 Time Series Forecasting

A time series is a sequenced collection of observations indexed by time. It can be defined and modeled using the notions of random variables and stochastic processes. Subsequently, we introduce the necessary statistical concepts and provide corresponding formal definitions[1]. Further in this chapter, we present various time series modeling approaches and describe some common forecasting methods that we relate to throughout this study.

Definition 4.0.1. *Probability space* is a triple $(\Omega, \mathcal{A}, \mathbb{P})$ where *sample space* Ω is a set of possible outcomes or *realizations* of an experiment, $\mathcal{A}$ is a set of subsets of Ω called *events* and $\mathbb{P}$ is a *probability measure*[2]

$$\begin{aligned} \mathbb{P}\colon \mathcal{A} &\to [0;1] \\ A &\mapsto \mathbb{P}\Big[A\Big] \end{aligned}$$

that assigns a probability to an event $A \in \mathcal{A}$.

Definition 4.0.2. *Random variable* $\mathbf{x}$ is a measurable function

$$\begin{aligned} \mathbf{x}\colon \Omega &\to \mathcal{S} \\ \omega &\mapsto \mathbf{x}(\omega) \end{aligned}$$

that assigns each realization in Ω to an element in a measurable space $\mathcal{S}$. Such an element is named *observation*.

A random variable $\mathbf{x}$ can be described by its expectation, variance and the probability density function. For now, we focus on the case of a real-valued, continuous variable (i.e., $\mathcal{S} = \mathbb{R}$) and define such a variable by writing $\mathbf{x} \in \mathbb{R}$ for simplicity.

---

[1] For our purposes, we adopt general concepts from the standard mathematical literature sometimes discarding the nuances that are not relevant in our application. For more a detailed explanation on statistics and probability theory see [Was04] and [HPS] where we source the definitions found in this chapter.

[2] In fact, $\mathcal{A}$ is a so called σ-*algebra* and $\mathbb{P}$ underlies some further restrictions that are not relevant in our case.

Definition 4.0.3. *Probability* of an event $\mathbf{x}(\omega) \leq a$ denoted as $\mathbb{P}\Big[\mathbf{x} \leq a\Big]$ can be described for all $x, a \in \mathbb{R}$ as

$$\mathbb{P}\Big[\mathbf{x} \leq a\Big] := \int_{-\infty}^{a} f_{\mathbf{x}}(x)dx, \tag{4.1}$$

where $\mathbf{f_x}\,(x)$ is the probability density function.

Definition 4.0.4. *Probability density function (PDF)* $\mathbf{f_x}\,(x)$ is a function that is given for a random variable $\mathbf{x}$ and satisfies the following

1. $\mathbf{f_x}\,(x) \geq 0$ for all x
2. $\int_{-\infty}^{\infty} \mathbf{f_x}\,(x)\,dx = 1$
3. $\mathbb{P}\Big[a \leq \mathbf{x} \leq b\Big] = \int_{a}^{b} \mathbf{f_x}\,(x)\,dx$ for any $a, b \in \mathbb{R}$.

Definition 4.0.5. *Expectation* of a random variable $\mathbf{x}$ is defined as the mean

$$\mathbb{E}\Big[\mathbf{x}\Big] := \mu_{\mathbf{x}} = \int_{-\infty}^{\infty} x\mathbf{f_x}\,(x)\,dx \tag{4.2}$$

of its unconditional PDF and is also often called *population mean*.

Definition 4.0.6. *Variance* of a random variable $\mathbf{x}$ is defined as

$$\sigma_{\mathbf{x}} := \mathbb{E}\Big[(\mathbf{x} - \mu_{\mathbf{x}})^2\Big] = \int_{-\infty}^{\infty} (x - \mu_{\mathbf{x}})^2\mathbf{f_x}\,(x)\,dx \tag{4.3}$$

and corresponds to the expected deviation from the population mean.

In our study, it is important to distinguish between population parameters describing a PDF and *sample statistics* that describe the given data.

Definition 4.0.7. *Sample mean* or *uniform average* of a sample $X = [x_1, \ldots, x_n]$ is calculated as

$$\text{Avg}[X] = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{4.4}$$

while the *sample variance* is calculated as

$$\text{Var}[X] = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \text{Avg}[X])^2 \tag{4.5}$$

Dependencies within sequential data are essential for time series. We will use the following concepts which describe the relationships between two (or more) random variables.

Definition 4.0.8. *Conditional probability density function (CPDF)* is defined for the random variables $\mathbf{x}$, $\mathbf{y}$ and observations $x, y \in \mathcal{S}$ as

$$\mathbf{f}_{\mathbf{x}|\mathbf{y}}(x|y) = \mathbb{P}\Big[\mathbf{x} = x \mid \mathbf{y} = y\Big] = \frac{\mathbb{P}\Big[\mathbf{x} = x, \mathbf{y} = y\Big]}{\mathbb{P}\Big[\mathbf{y} = y\Big]} = \frac{\mathbf{f}_{\mathbf{x},\mathbf{y}}(x, y)}{\mathbf{f}_{\mathbf{y}}(y)} \tag{4.6}$$

for $\mathbf{f}_{\mathbf{y}}(y) > 0$ and $\mathbf{f}_{\mathbf{x},\mathbf{y}}(x, y)$ being their *joint PDF*.

Definition 4.0.9. *Covariance* between random variable $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$\text{Cov}(\mathbf{x}, \mathbf{y}) := \sigma_{\mathbf{x},\mathbf{y}} = \mathbb{E}\Big[(\mathbf{x} - \mu_{\mathbf{x}})(\mathbf{y} - \mu_{\mathbf{y}})\Big]. \tag{4.7}$$

Definition 4.0.10. *Correlation* between random variables $\mathbf{x}$ and $\mathbf{y}$ is defined as

$$\text{Corr}(\mathbf{x}, \mathbf{y}) := \rho_{\mathbf{x},\mathbf{y}} = \frac{\sigma_{\mathbf{x},\mathbf{y}}}{\sqrt{\sigma_{\mathbf{x}}}\sqrt{\sigma_{\mathbf{y}}}}. \tag{4.8}$$

Definition 4.0.11. *Stochastic process* $\mathcal{Y} = \{y_t : t \in \mathcal{T}_0\}$ is a collection of random variables defined in a probability space $(\Omega, \mathcal{A}, \mathbb{P})$. Set $\mathcal{T}_0$ is called *index set* representing the time. A process is said to be *stationary*[3] if:

- $\mathbb{E}\Big[y(t)\Big] = \mu_{\mathcal{Y}}$
- $\text{Cov}(y(t), y(t-s)) = \gamma_s$

where $\mu_{\mathcal{Y}}$ is a constant for all t and the covariance γ_s between different observations only depends on their separation in time s and not on t.

The concepts introduced above allow us to formally define a time series and discuss notions from the *time series analysis* – a research area concerned with describing, summarizing and drawing conclusions from sequential data about the underlying stochastic process and creating its *model* for studying the dependencies and forecasting [BD91, Ham95].

Definition 4.0.12. *Time series* $\{y_t\}_{t \in \mathcal{T}_s}$ is a realization of a stochastic process $\mathcal{Y}$ with $\mathcal{T}_s \subseteq \mathcal{T}_0$ represented by a sequenced collection of observations indexed by time t.

The time series is said to be *discrete* if $\mathcal{T}_s$ is a discrete set of time points at which observations were made and we denote such time series with a subindex Y_t. Otherwise, a time series is said to be *continuous* or *functional* and we denote such time series as $Y(t)$.

[3] Statisticians distinguish between weakly and strict stationarity. In this work, however, we use the term stationary meaning weakly stationary in statistical sense as is often done in the literature [Ham95].

In practice, time series are often discrete and given by an array Y where its t'th element can be seen as an *observation* of a corresponding random variable y_t. Further on, unless specified differently, when using the attributes of a time series, such as stationarity, we relate to the attributes of the underlying stochastic process.

Process-driven modeling is a traditional approach to time series forecasting based on *time series analysis*[4]. This field of study includes various methods that were developed throughout the last century having numerous applications in econometrics, finance and others [BD91, Ham95]. The main focus of time series analysis lies on the autocorrelation within the data as well as statistical integrity and consistency[5] of the models. Stationarity is a central assumption for a large part of the theory, which might not be given for a real-world time series[6]. The aim of the analysis lies on transforming and decomposing the series into a deterministic and stochastic part such as it is done through *classical decomposition*

$$Y_t = T_t + S_t + I_t, \tag{4.9}$$

where T_t represents a deterministic *trend*, deterministic periodic function S_t describes patterns occurring at a fixed frequency named *seasonality* and *irregularity* I_t corresponds to a stochastic part of the time series that can be reasonably modeled as a realization of a stationary process.

The decomposition is usually done manually using the a priori knowledge about the process such as its physics, in an iterative procedure, where a researcher inspects various time series graphs[7]. The stochastic part I_t is often modeled applying a set of rules and heuristics known under *Box-Jenkins methodology* [BJRL15] resulting in an ARIMA-model that is used to forecast the Y_t as described later in the text[8].

Data-driven modeling is an alternative approach to predicting future values of Y_t that is based on a recently developed *theory of statistical learning* which has its main applications in the fields of machine learning and computer science [FHT08, Vap10]. This theory focuses on inferring the relation between an input and the output of an unknown system given a set of data. It includes various methods for *supervised learning* that allow to formulate and solve the forecasting task as a *regression problem* defined as follows.

[4] Process-driven forecast are also, sometimes, named *model-based* because they require an explicit time series model. In this study we purposely avoid such a name.

[5] See Definition 4.2.3.

[6] There is a considerable ongoing effort to model non stationary processes [CSB+15].

[7] There are some attempts to automatically estimate trend and seasonality. See [HA18] and references therein.

[8] With ARIMA and variants thereof being the most common, there is a myriad of other time series models many of which are described in detail in [BD91, Ham95].

Definition 4.0.13. *Regression Problem.* Let $X \in \mathbb{R}^q$ be a vector of random variables named *inputs*[9] and let $y \in \mathbb{R}$ be a random variable named *output*[10] related to X through a fixed but unknown CPDF $f_{y|X}(y|X)$. Given

- a set of functions $\mathbf{r}_\alpha(X)$ indexed by $\alpha \in \Lambda$, where Λ is a set of *parameters*
- a *training set* $\mathcal{T} := \{(X_i, y_i) \mid 1 \leq i \leq m\}$ where (X_i, y_i), for $i = 1, \ldots, m$, are the pairs of observations obtained according to their joint PDF $f_{X,y}(X, y)$
- *loss function* $L(y, \mathbf{r}_\alpha(X)) \geq 0$ for all X, y

and assuming that X and y are related through the *regression equation*

$$y = \mathbf{r}(X) + \epsilon, \tag{4.10}$$

where ϵ represents a random error term independent of X, estimate a *regression function*[11] $\mathbf{r}(X)$, minimizing the *expected prediction error (EPE)*[12]

$$\text{EPE}(\alpha) = \mathbb{E}\Big[L(y, \mathbf{r}_\alpha(X))\Big]. \tag{4.11}$$

In the equation (4.10), the systematic information that X provides about y is modeled by $\mathbf{r}(X)$. Since y might not only depend on X, error ϵ represents all the uncaptured influences that are independent of X (e.g., measurement errors). Hence, an observation y_i can be seen as a sum of a deterministic *model* $\mathbf{r}(X)$ and a stochastic *error* ϵ.

There are numerous *regression methods and techniques* to calculate a *regression estimate* $\hat{\mathbf{r}}(X)$ that is a *fit* of the true relation $\mathbf{r}(X)$. In fact, modern statistics extends the regression problem beyond *multivariate analysis (MVA)* to more abstract objects (e.g., continuous curves) [MA14, WCM16]. We provide the following general classification[13] of regression methods that we will explore in this study.

Definition 4.0.14. *Model Classification.* Let $\mathbf{r} : \mathbb{X} \to \mathbb{Y}$ be an operator between some spaces $\mathbb{X}, \mathbb{Y}$. A regression method (model) for the estimation of $\mathbf{r}$ describes a condition of the form

$$\mathbf{r} \in \mathcal{M} \subseteq \mathcal{Q}(\mathbb{X}, \mathbb{Y}), \tag{4.12}$$

where model *class* $\mathcal{M}$ is a subset of all operators $\mathcal{Q}(\mathbb{X}, \mathbb{Y})$ that map elements of $\mathbb{X}$ to $\mathbb{Y}$. We call the model (4.12) *parametric*[14] if $\mathcal{M}$ has a finite number of elements and

[9] Depending on the field, these are also called predictors, regressors, independent variables or features.
[10] Depending on the field, these are also called responses or dependent variables.
[11] Depending on the context, regression function is sometimes called *regression model* or simply *model*.
[12] Also named *risk* as in [Vap10].
[13] In accordance with the generalization provided in [FV03, FR11].
[14] We provide examples of parametric models in Section 4.1.

nonparametric[15] otherwise. Further, we call a model *univariate* if $\mathbb{X} = \mathbb{R}$, *multivariate* if $\mathbb{X} = \mathbb{R}^q$ and *functional* if $\mathbb{X}$ is an infinite-dimensional space of functions[16].

It might be difficult to draw the a sharp line between the approaches for time series forecasting since, in practice, we follow similar steps to, first, create a model and, then, use it for predicting future values of Y_t. We summarize these steps as follows.

1. *Model setup*: We consider time-series representations (e.g., various plots) and descriptive statistics to identify trends, seasonality and dependencies between the variables. Such insights help us to select an appropriate model type, and set its *hyperparameters*[17].
2. *Model training*: Once a model is selected, we use a training algorithm[18] on the available training data to estimate $\mathbf{r}$.
3. *Model evaluation*: Graphs, error notions and statistical tests are used to access model accuracy, determine the amount and type of information not captured by the model and validate the initially made assumptions.

In this context, process-driven modeling focuses on describing the underlying process that generates the data. Having an explicit process model, we can analytically forecast Y_t with various horizons and corresponding confidence bounds. Alternatively, data-driven modeling focuses on estimating the deterministic input-output relation that is valid for unseen data.

However, for both approaches, given a model $\hat{\mathbf{r}}$ and using the *squared error* loss

$$L(y, \mathbf{r}(X)) = (y - \mathbf{r}(X))^2, \tag{4.13}$$

we can be show that, for a given *input query* X^*, the best forecast $\hat{y}_t$ is

$$\hat{y}_t = \mathbf{r}(X) = \mathbb{E}\Big[y_t \mid X^*\Big]. \tag{4.14}$$

[15] We provide examples of nonparametric models in Section 4.2.

[16] If we have more than one functional predictor, such problems are some times called *multi-functional* and are yet to be researched [FV03].

[17] Parameters of a model that cannot be learned directly from the data and are often tuned manually.

[18] Depending on the context, this step can be called *model estimation.* as in Box-Jenkins methodology or *model fitting* as common in statistical literature. The corresponding algorithm is called accordingly *curve fitting*, estimation or *learning algorithm.*

This also applies for time series[19] after an *embedding step* where X is defined such that it considers up to p most recent observations of Y_t

$$X^* = [y_{t-1}, \ldots, y_{t-p}]. \tag{4.15}$$

In context of forecasting, the terms parametric and nonparametric are used to indicate that a *parametric model* assumes an existing global relationship between the input and output. *Nonparametric model* is the one that is estimated *locally* for the given X^*. Further in the text we contrast both model families and describe the most common techniques for each type. At the end of this chapter we describe functional regression – an alternative view on forecasting that we will apply in our study.

4.1 Parametric Regression

Parametric regression methods assume that X and y have a globally valid relation and $\mathbf{r}(X)$ has a predefined form fully described by a set of parameters. All parametric models have in common that the model has to be determined by learning those parameters from the historical data *before* predicting y_t for a given X^*.

To explicate the parametric forecasting approach, we use the most prominent example of *multiple linear regression (MLR)* model where $\mathbf{r}(X)$ is assumed to be a linear function of X, so that the output is modeled following (4.10) as

$$y = X^* W^{\mathrm{T}} + \epsilon, \tag{4.16}$$

where $X = [1, x_1, \ldots, x_p]$ is the input and $W = [w_0, \ldots, w_p]$ contains *linear regression coefficients* determining $\mathbf{r}(X)$.

Training data mentioned in the Definition 4.0.13 is given by the historical observations of the time series which are transformed using (4.15) into a set of *input/output (I/O)* observations (X_j, y_j) with $j = 1, \ldots, m$. We use this set to calculate W, so that the estimated curve $\hat{\mathbf{r}}(X)$ best-matches the observed values of Y_t. For this purpose an ordinary least squares technique is commonly applied under the assumption that the data (X_j, y_j) is *independently and identically distributed (IID)* [FHT08].

Having found best possible fit $\hat{\mathbf{r}}(X)$, we calculate the forecast with (4.14) as

$$\hat{y} = X^* W. \tag{4.17}$$

[19] For ease of exposition, we restrict ourself to uniform (i.e., $y_t \in \mathbb{R}$) time series.

As in our example and in general, a parametric model assumes that y_t will always be somewhere close to the fit $\hat{\mathbf{r}}(X)$. This is true as long as the assumption about the underlying form of $\mathbf{r}$ (e.g. linear) is correct.

Parametric regression models all have in common that the model $\mathbf{r}$ is fully determined by a set of parameters before computing $\hat{y}$ for a given X^*. These models numerous applications such in finance, manufacturing systems, health informatics and energy grids among other fields [CSB+15]. They are also standard for load forecasting purposes for which we provide examples further in the text. Next, we describe two parametric regression methodologies: *autoregressive integrated moving average* and *artificial neural network*. They are the most common regression methodologies in the load forecasting literature (Chapter 5) and we will use them to create reference models for our study.

4.1.1 Autoregressive Integrated Moving Average

Autoregressive integrated moving average (ARIMA) methodology is a general form of a linear autoregressive model that was originally developed specifically for time series. The acronym ARIMA captures the main parts of of a time series model that are discussed below.

An ARIMA-model interpolates between autoregressive, integrated and moving average parts and can be written as

$$\hat{y}_t^{(d)} = c + w_1^{\text{AR}} y_{t-1}^{(d)} + \ldots + w_p^{\text{AR}} y_{t-p}^{(d)} + \epsilon_t + w_1^{\text{MA}} \epsilon_{t-1}^{(d)} + \ldots + w_q^{\text{MA}} \epsilon_{t-q}^{(d)} \tag{4.18}$$

while often denoted as ARIMA(p,d,q) with the following hyperparameters:

- p ... order of the autoregressive part
- d ... order of differencing
- q ... order of the moving average part

Once properly set up, an ARIMA-model can deliver reliable forecasts, however it has strong assumptions on the predicted time series and requires substantial amount of historical data for training. Moreover, the selection of the hyperparameters p, d, q for a given time series can be challenging. For this task, researchers would often try out (either manually or automatically) different settings and choose the ones that yield the most accurate forecast on a validation dataset. Each parameter corresponds to one of the model parts described below.

4.1.1.1 Autoregression (AR)

The *autoregressive (AR)* part models the time series observation y_t as a linear combination of past values of the variable. Therefore, AR-model model assumes a linear relationship between an observation y_j and p lagged observations $y_{j-1}, \ldots, y_{j-p}$ named *lags*. An AR-model of order p, describes a time series as

$$\hat{y}_t^{\text{AR}} = c_{\text{AR}} + w_1^{\text{AR}} y_{t-1} + w_2^{\text{AR}} y_{t-2} + \ldots + w_p^{\text{AR}} y_{t-p} + \epsilon_t, \tag{4.19}$$

where c_{AR} is a constant, ϵ_t is an error term that is often assumed to be the white noise while $w_1^{\text{AR}}, \ldots, w_p^{\text{AR}}$ are the model parameters that have to be found during model training.

4.1.1.2 Integration (I)

Linear autoregression requires the time series to be stationary after removing an eventually present trend component. To fulfill this assumption, we can differentiate the time series d times before modeling and forecasting it. Thereby, we regard the original time series is regarded as the integral of the stationary time series. Note that we can differentiate a discrete time series Y_t by subtracting an observation from an observation at the previous time step.

4.1.1.3 Moving Average (MA)

A *moving average (MA)* model represents the time series as a moving average of lagged residual error observations that are assumed to be normally distributed. Consequently, an MA-model of order q can be expressed as

$$\hat{y}_t^{\text{MA}} = c_{\text{MA}} + \epsilon_t + w_1^{\text{MA}} \epsilon_{t-1} + w_2^{\text{MA}} \epsilon_{t-2} + \ldots + w_q^{\text{MA}} \epsilon_{t-q}, \tag{4.20}$$

where c_{MA} is a constant, ϵ_t is a white noise error, and $w_1^{\text{MA}}, \ldots, w_q^{\text{MA}}$ are the model parameters that have to be found during the training. In contrast to the AR-model, a time series is modeled using as a weighted sum of historical errors $\epsilon_{t-1}, \ldots, \epsilon_{t-q}$ rather than its historical values $y_{j-1}, \ldots, y_{j-p}$.

4.1.1.4 Other Variants of ARIMA

There are also other variants of an ARIMA-model. To consider implicit seasonality of a time series, *seasonal ARIMA (SARIMA)* was proposed in [BJRL15]. Another common

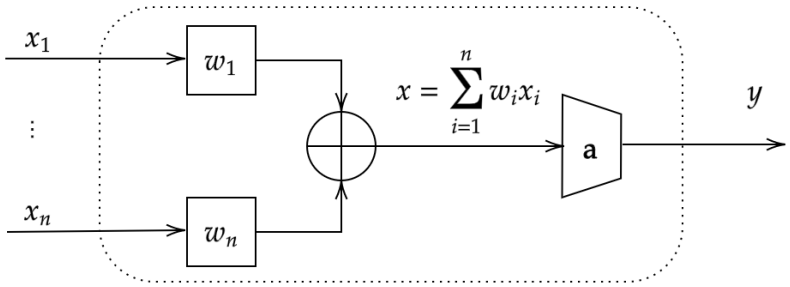

Figure 4.1: Mathematical model of an artificial neuron. Description is provided in the text.

extension is the *ARIMA with exogenous variable (ARIMAX)* that allows to consider exogenous inputs [BJRL15]. A model denoted as ARIMAX(p, d, q, β) model adds the covariate observation x_t to an ARIMA model such that

$$\hat{y}_t^{(d)} = \beta x_t^{(d)} + c + w_1^{\text{AR}} y_{t-1}^{(d)} + \ldots + w_p^{\text{AR}} y_{t-p}^{(d)} + \epsilon_t + w_1^{\text{MA}} \epsilon_{t-1}^{(d)} + \ldots + w_q^{\text{MA}} y_{t-q}^{(d)}. \quad (4.21)$$

Observe that the *covariate coefficient* β is not the direct effect of the change in x_t on y_t. Instead, we have to interpret it considering the past values of y_t. However, the model (4.21) can be reformulated as a transfer function model which simplifies its interpretation [**?**].

4.1.2 Artificial Neural Network

Artificial neural network (ANN) methodology is a group of techniques for parametric regression often used in machine learning applications. With this methodology, we can model almost any nonlinear relation between multiple inputs and outputs. Assuming such relation exists, its form does not have to be determined in advance since it is self-learned from the available historical data using a specialized training algorithm. Due to this fact, ANN became a common approach for solving the regression problem with numerous applications.

A neural network is an interconnection of single elements called *artificial neurons*. Their mathematical model loosely resembles the function of a neuron in a human brain (Figure 4.1). A neuron can have $n \in \mathbb{N}$ inputs with n corresponding weights w_i where $i = 1, \ldots, n$. Weighted inputs are summed and processed by an *activation function* $\mathbf{a}(x)$ that determines the unit output $y \in \mathbb{R}$. While the weights are calculated during the training phase, $\mathbf{a}(x)$ has to be defined a priori. There exist many activation functions (Table 4.1) and, in practice, we intend to use a function that resembles the expected output characteristics (e.g., bounded, smooth, positive range etc.).

Table 4.1: Activation function examples.

Name	$\mathbf{a}(u)$	**Range**
ReLu	$\max(0, x)$	$[0; \infty)$
Log-Sigmoid	$\frac{1}{1+exp(-x)}$	$[0; 1]$
Tanh-Sigmoid	$\frac{2}{1+exp(-2x)} - 1$	$[-1; 1]$

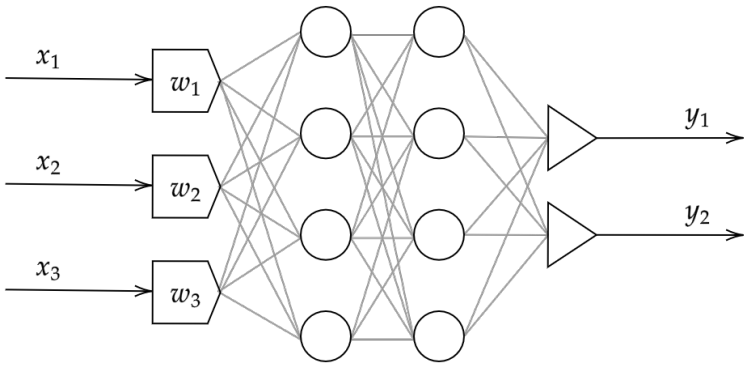

Figure 4.2: An interconnection of artificial neurons constituting a feedforward neural network (multilayer perceptron). In this example, the input layer contains three neurons, two hidden layers contain four neurons each and the output layer contains two neurons. The input data x_1, x_2, x_3 traverse the network from the input layer towards the outputs y_1, y_2.

If several artificial neurons are interconnected into a network, almost any nonlinear relationship can be approximated. The units are organized in layers creating different network architectures. The computation traverses the network from the input layer, to the output layer, passing through some neurons in the *hidden layers* one or several times (Figure 4.2).

We formalize a neural network model as follows. Let $\mathcal{N}$ be a neural network with n_x inputs, n_y outputs, and n_w interconnections between the neurons. The network is fully defined by its architecture and a set of weights $W = [w_1, \ldots, w_{n_w}] \in \mathbb{R}^{n_w}$. We describe it as a regression function

$$Y = \mathbf{r}_{\mathcal{N}}(X, W) \tag{4.22}$$

that maps an input $X \in \mathbb{R}^{n_x}$ to the output $Y \in \mathbb{R}^{n_y}$. In the following, we restrict ourselves to the case where $X \in \mathbb{R}^{n_x}$, $y \in \mathbb{R}$ though the extension to a multivariate output $Y \in \mathbb{R}^{n_y}$ is straightforward as described in [FHT08].

To predict a time series using ANN-methodology, we proceed as follows. First, we select a network architecture and hyperparameters[20], which we consider the most appropriate for a given task. Next, we train the model on the available historical data calculating W that fully determines $\mathbf{r}_{\mathcal{N}}$. At last, trained network yields the forecast

$$\hat{y} = \mathbf{r}_{\mathcal{N}}(X^*), \tag{4.23}$$

given an input X^*.

4.1.2.1 Network Architecture

A *network architecture* includes inputs, network type, number of neurons in the layers, and their hierarchy. Probably, the most common network type is a feedforward network (Figure 4.2). Another common type is a recurrent ANN that allows to feed back the output to create autoregressive models. Extensive overview of existing architectures of both types contrasting the differences can be found in [RK15].

Feedforward neural network is the type where, starting at the input, the data traverses the network through hidden layers to the output without any cycles or loops. The most prominent architecture of this type is the *multilayer perceptron (MLP)*[21]. It includes at least one hidden layer where each neuron has a nonlinear activation function.

As stated by the *universal approximation theory*, a MLP with only one hidden layer and a finite number of neurons can approximate any function $\mathbf{r}$ on a compact subset of $\mathbb{R}^{n_x}$, when given appropriate weights [Cyb89]. However, this does not consider the algorithmic learnability of those weights. The hidden layer may be impractically large and the network may fail to learn and perform well in a practical setting. For this reason, there exist many other architectures.

Recurrent neural networks is a type of network with feedback loops that allow data to traverse the network both ways. In this case, the connections between the neurons form a directed graph and, unlike the feedforward type, a recurrent network has a dynamic internal state. An architecture where past output values are fed back can be particularly useful for time series modeling and prediction.

[20] Hyperparameters are settings that, unlike weights, cannot be calculated directly from the training data and have to be set a priori (e.g., $\mathbf{a}(u)$).

[21] In fact, the terms feedforward neural network and multilayer perceptron are often used interchangeably.

For univariate time series, we can use a recurrent network to create a *nonlinear autoregressive model (NAR)*[22] with p lags defined as

$$\hat{y}_i = \mathbf{r}_{\mathcal{N}}\left(y_{i-1}, \ldots, y_{i-p}\right), \tag{4.24}$$

where prediction $\hat{y}_i$ is a function of the p preceding values $y_{i-1}, \ldots, y_{i-p}$ of the time series y.

For multivariate time series, we can create a *nonlinear autoregressive model with exogenous inputs (NARX)* that considers external inputs:

$$\hat{y}_i = \mathbf{r}_{\mathcal{N}}\left(y_{i-1}, \ldots, y_{i-p}, X_i\right). \tag{4.25}$$

Here, a prediction $\hat{y}_i$ is calculated as a function of its p lags $y_{i-1}, \ldots, y_{i-p}$ and an exogenous input X_i.

Deep neural networks (DNN) are the state of the art in machine learning research. These architectures have two or more hidden layers combined with a complex topology for which there have been remarkable applications in image and speech recognition [GBC16]. There are DNN architectures of recurrent type such as *restricted Boltzmann machine (RBM)* and *long short-term memory (LSTM)* networks and of feedforward type such as *convolutional neural networks (CNN)*. Increasing the number of layers and neurons can improve the accuracy but, also elevates the complexity of the network which increases the training time and amount of historical data that is needed to train the network.

4.1.2.2 Network Training

In the training phase, we estimate $\mathbf{r}$ finding a set of weights W so that $\mathbf{r}_{\mathcal{N}}$ yields the lowest error on a given training data and is expected to *generalize* well – i.e., be accurate predicting unseen data. The weights are initialized randomly and calculated with a *supervised training algorithm*[23] often using a *back-propagation (BP)* of training error [RHW86] combined with different optimization methods.

A training algorithm calculates the weights seeking to minimize EPE (4.11) on the available training data. It is common to use squared error loss (4.13) as a cost function for which finding the weights becomes a *nonlinear least squares fitting problem* formulated as follows. Given a network $\mathcal{N}$ with n_x inputs and a training set with m IID observations

[22] An example of such model is provided in Section 9.2.2.2.
[23] In some contexts, a supervised training algorithm is often called learning algorithm.

(X_j, y_j) where $j = 1, \ldots, m$ and $m \gg n_x$, minimize the mean of squared errors finding optimal weights W^* as:

$$W^* = \arg\min_W \left[\frac{1}{m} \sum_{j=1}^{m} \left(y_j - f(X_i, W) \right)^2 \right]. \tag{4.26}$$

This problem can be solved using an iterative procedure based on *gradient descent optimization*. Starting with a randomly initialized vector W_0, during each step s, the weights W_s are updated in the direction of the negative gradient:

$$W_{s+1} = W_s - \gamma \mathbf{J}^{\mathrm{T}} \mathrm{EPE}(W), \tag{4.27}$$

where *learning rate*[24] γ is another hyperparameter defined in advance. The Jakobian matrix $\mathbf{J}$ contains first order partial derivatives of the inputs and can be calculated by back-propagating EPE to the outputs of individual neurons [RHW86]. Related approaches are also often called *back-propagation algorithm* in context of neural networks [RB93, KW52, KB17].

To speed up the computation, we can use *Levenberg-Marquardt (LM)* algorithm originally introduced in [Mor78] and adopted to train feedforward networks in [HM94]. It applies the approximation of the Hessian matrix

$$\mathbf{H} = \mathbf{J}^{\mathrm{T}} \mathbf{J}, \tag{4.28}$$

so that the weights are updated as

$$W_{s+1} = W_s - \left[\mathbf{J}^{\mathrm{T}} \mathbf{J} + \gamma \mathbf{I} \right]^{-1} \mathbf{J}^{\mathrm{T}} \mathrm{EPE}(W). \tag{4.29}$$

The LM-algorithm interpolates between gradient descent and *Gauss-Newton algorithm* depending on the dynamically chosen learning rate [DFH97a]. While for small learning rate γ, (4.29) resembles Gauss-Newton algorithm, for a large γ, it corresponds to gradient descend (4.27).

In practice, advanced algorithms combining gradient descent and back-propagation approaches such as *stochastic gradient descent* [KW52], *resilient backpropagation* [RB93] and others [KB17] are often used for large neural networks (e.g., DNNs). Alternatively, the LM-algorithm is much faster than gradient descent for moderate-sized networks with up to several hundred weights, which suffices for many applications [HM94].

[24] In some contexts, learning rate is also called damping factor.

4.1.2.3 Network Settings

The choice of a training algorithm and other settings such as activation function, learning rate and the architecture itself significantly affects the training result and the prediction accuracy of the network. The settings must be known before learning the weights from the available training data. They are often chosen and fine-tuned manually relying on problem knowledge, researcher experience and intuition.

Nevertheless, there have been attempts to automate the choice. Automated model setup approaches require training numerous networks and the implementation can be challenging and even unpractical. Given a large space of settings, (grid-) search based methods can become prohibitively time-consuming. Despite the increasing interest, the development of fully automated models based on ANN is in a preliminary stage [HKV19]. As of today, the well-performing network architecture presented in the results is usually found manually through a trial and error process and requires large amounts of historical data and computational resources.

Finding the best settings, together with the ANN training often requires a vast amount of historical data, and it is hard to interpret the weights of the resulting network. Nonetheless, for some applications where a considerable amount of data is available, computation time is not an issue and there is an extensive domain knowledge allowing manual fine-tuning, ANN-model can be very accurate. Therefore, in the recent past, neural networks were used in many propositions for the load forecasting as we discuss further in the text.

4.1.3 Parametric Model Setup

The hyperparameters of a parametric model[25] need to be known before learning the weights from the available training data. Those hyperparameters are often manually estimated and iteratively fine-tuned given an in-depth knowledge of the forecasting problem.

Setting hyperparameters can also be automated based on a *grid search*, *random search* or *Bayesian optimization* procedure which might often become very computationally expensive [KH88, TMA16, **?**]. For such purpose, the available historical data is divided into training and validation sets. Different models with different hyperparameters (grid search) are evaluated on the validation set and the best one is selected.

Having determined the hyperparameters, all historical data is used for training. The trained model is, then, used to obtain a forecast for unseen new data, assuming that the statistical

[25] For instance, the hyperparameters p, d, q of an ARIMA-model need to be set before learning the weights from the available data.

properties of the process generating new data are the same as of the process that provided the historical observations.

For this reason, once trained, parametric models do not adapt to the abrupt changes in data which can be often encountered in a practical situation such as load forecasting of low-voltage end-consumers. For this matter, sliding window or other adaptive learning approaches are the subject of ongoing research [HKV19].

4.2 Nonparametric Regression

Nonparametric regression assumes that similar input are likely to have similar outputs. It is often associated with the most common method family called *locally weighted learning*. There are tasks, where it is hard to assume that there exists a globally (i.e., $\forall X \in \mathbb{R}^q$) valid relation between X and y. Even if such relation exists, it might be very complex and we might not have any a priori knowledge about it to anticipate any predefined form of the regression function. In contrast to the global learning methods (parametric models) where a model is trained to fit all the data, local learning fits the data only in the region around the given input query X^*. For every X^*, a new local model is built[26] avoiding any assumptions about the form of $r(X)$. Instead, the $r(X)$ is approximated *locally*, in the vicinity of X^*, only requiring that $\mathbf{r}(X)$ is smooth in a mathematical sense. There is no pretraining necessary, as the model is determined online for a given X^*. As a result, a nonparametric model predicts the output as a combination of the historical output observations in that region.

Kernel regression is one of the most prominent methods for locally weighted learning. The main idea of this method is to use historical observations $y_1, \ldots, y_m$ according to their relevance for a given input X^*. Such relevance is measured using a predefined *distance measure* between X^* and historical inputs $X_1, \ldots, X_m$. Kernel regression is a flexible technique that can capture even a very complex behavior of $\mathbf{r}(X)$. However, its accuracy deteriorates quickly with the growing input dimension q. This effect named *curse of dimensionality* is one of the main limitations of a nonparametric model.

Consequently, nonparametric models are not as common as parametric models, but for the applications where $q \ll m$, nonparametric methods are valued for their simplicity, intuitiveness and flexibility for predicting complex nonlinear behavior. There are several areas of application[27] with some propositions for the load forecasting which we discuss

[26] This approach is also known as *lazy* or *memory based learning*.

[27] Among other fields, nonparametric models were applied in finance [BCO18, ANHS13, CCL04, Dia09], electricity price forecasting [BSRP08, LSS+02b, LSS+02a, LSE+07], traffic flow forecasting [SC08, SWKO02, ZL15, HC15] and human behavioral sciences [BBK+14].

in detail further in the text (Chapter 5). A recent review on theoretical advances on nonparametric methods can be found in [CSB+15] and references therein.

Subsequently, we describe *kernel density estimator* (Section 4.2.1) that provides theoretical foundation for kernel regression (Section 4.2.2), before focusing on the common nonparametric models in more detail and discussing the curse of dimensionality (Section 4.2.3). Later in the text, we introduce *functional regression* developed to circumvent this substantial limitation of nonparametric models.

4.2.1 Kernel Density Estimator

Kernel density estimator (KDE) provides a way to estimate a PDF from a set of observations. Given a dependency between two random variables $\mathbf{x}, \mathbf{y} \in \mathbb{R}$, with the corresponding marginal PDF $\mathbf{f}_{\mathbf{x}}(x)$ and $\mathbf{f}_{\mathbf{y}}(y)$, we can describe their relationship by the joint PDF $\mathbf{f}_{\mathbf{x},\mathbf{y}}(x, y)$. Hence, knowing the marginal and joint PDFs we can obtain the forecast (4.14) with CPDF $\mathbf{f}_{\mathbf{x}|\mathbf{y}}(x|y)$ using (4.6) and (4.14).

Starting with a one-dimensional example, imagine we have an IID sample of random variable x represented as a vector

$$X_s = [x_1, \cdots, x_m] \in \mathbb{R}^m, \tag{4.30}$$

and we want to estimate its PDF. Such estimate $\hat{\mathbf{f}}(x)$ can be obtained using KDE as [HWMS04]

$$\hat{\mathbf{f}}(x) = \frac{1}{b} \cdot \frac{1}{m} \sum_{j=1}^{m}, \theta(\frac{x - x_j}{b}), \tag{4.31}$$

where parameter b is called *bandwidth* and θ is a kernel defined as follows[28].

Definition 4.2.1. *Kernel* $\theta(z)$ is a function $\theta : \mathbb{R} \to \mathbb{R}$ with following properties:

1. $\theta(z) = \theta(-z)$
2. $\int \theta(z)dz = 1$
3. $\max_{z \in \mathbb{R}} \theta(z) = \theta(0)$
4. piece-wise smooth[29]

We provide some of the commonly used kernels in Table 4.2 and illustrate them in Figure 4.3.

[28] The exact definition of a kernel depends on the application and can be very broad. There exists a large variety of kernels in the statistical literature [GRW79].

[29] In practice, piece-wise smooth means a function that is differentiable except on a discrete set of points.

Table 4.2: Common kernels illustrated in Figure 4.3.

Kernel	Function
Uniform	$\mathbb{1}(\lvert z\rvert \leq 1)$
Epanechnikov	$\frac{3}{2}(1-z^2)\mathbb{1}(\lvert z\rvert \leq 1)$
Gaussian	$\frac{2}{\sqrt{2\pi}}\exp(-\frac{1}{2}z^2)$

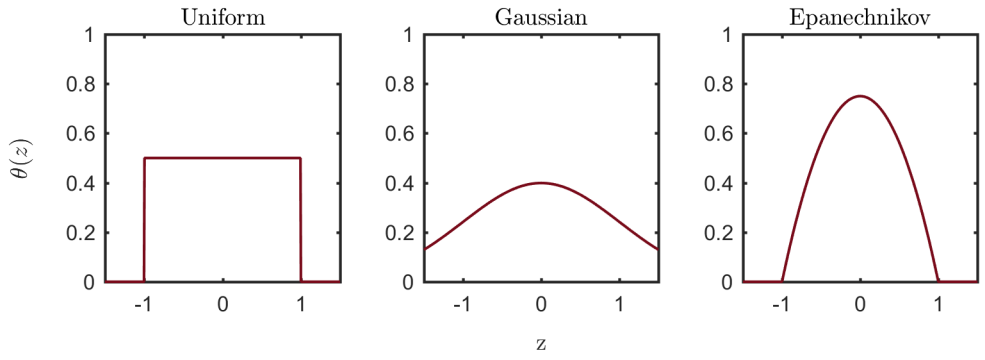

Figure 4.3: Kernels defined in Table 4.2.

While there exist numerous different kernels[30], we can show that almost identical estimates can be obtained using different kernels accordingly adjusting the bandwidth. For any kernel, KDE is asymptotically consistent and the estimated PDF converges towards the actual density [AMS97, HWMS04].

4.2.1.1 Estimator Bias

Having estimated the $\hat{\mathbf{f}}(x)$ using (4.31), we consider some of its properties. The expected value of the estimation error is defined as *bias*

$$\text{bias}[\hat{\mathbf{f}}(x)] = \mathbb{E}\left[\hat{\mathbf{f}}(x) - \mathbf{f}_{\mathbf{x}}(x)\right] = \mathbb{E}\left[\hat{\mathbf{f}}(x)\right] - \mathbf{f}_{\mathbf{x}}(x) \tag{4.32}$$

and represents the difference between the expected value of the estimator and the actual value. For the estimator defined in (4.31) statisticians showed that[31]

$$\text{bias}[\hat{\mathbf{f}}(x)] \propto b^2, \tag{4.33}$$

$$\text{bias}[\hat{\mathbf{f}}(x)] \propto |\mathbf{f}''_{\mathbf{x}}(x)|. \tag{4.34}$$

Increasing b, will have the kernel density estimator consider more of the values from the sample which will result in a greater bias. Moreover, a volatile PDF with a large number

[30] Additional examples can be found in [HWMS04].
[31] For the equations we used the proportionality symbol $\propto$. See [HWMS04] for a detailed derivation.

of sharp peaks reflected in its curvature $|\mathbf{f}''_{\mathbf{x}}(x)|$ will also results in a larger bias. With increasing b our estimator will not be able to reflect the volatility of the actual PDF.

4.2.1.2 Estimator Variance

Another aspect of the estimated PDF is its variance (Definition 4.0.6). It can be expressed as squared sampling deviations:

$$\sigma^2(\hat{\mathbf{f}}(x)) := \mathbb{E}\Big[(\hat{\mathbf{f}}(x) - \mathbb{E}\Big[\hat{\mathbf{f}}(x)\Big])^2\Big] = \mathbb{E}\Big[\hat{\mathbf{f}}(x)^2\Big] - \mathbb{E}\Big[\hat{\mathbf{f}}(x)\Big]^2. \tag{4.35}$$

The variance reflects the dispersion of the estimate and hence the volatility of the approximated PDF $\hat{\mathbf{f}}(x)$. For the estimator defined in (4.31) statisticians showed that [HWMS04]

$$\sigma(\hat{\mathbf{f}}(x)) \propto \frac{1}{mb}. \tag{4.36}$$

In this case, increasing the bandwidth reduces the variance yielding us an $\hat{\mathbf{f}}(x)$ that is less volatile. Hence, b determines the smoothness of the estimated PDF and is, for this fact, sometimes regarded as *smoothing parameter*.

4.2.1.3 Bias-Variance Dilemma

The goodness of the estimator $\hat{\mathbf{f}}(x)$ can be assessed using *mean squared error (MSE)* expressed using expected value as

$$\text{MSE}(\hat{\mathbf{f}}(x)) := \mathbb{E}\Big[(\mathbf{f}_{\mathbf{x}}(x) - \hat{\mathbf{f}}(x))^2\Big]. \tag{4.37}$$

Given the above definitions of bias (4.32) and variance (4.35) MSE, can be rewritten as

$$\text{MSE}(\hat{\mathbf{f}}(x)) = \sigma^2(\hat{\mathbf{f}}(x)) + (\text{bias}[\hat{\mathbf{f}}(x)])^2. \tag{4.38}$$

Considering this equation, we can try to reduce the MSE by decreasing the bandwidth (4.33) while considering a smaller number of values for our estimator. Decreasing the bandwidth, at the same time, increases the variance (4.36) which adds to MSE. This trade-off between variance and bias is known as *bias-variance dilemma*.

4.2.1.4 Optimal Bandwidth

Searching for the optimal bandwidth b^{opt}, we note that MSE also depends on $\mathbf{f}_{\mathbf{x}}(x)$ and $\mathbf{f}''_{\mathbf{x}}(x)$ which are both unknown in the practice since $\mathbf{f}_{\mathbf{x}}(x)$ is the PDF we are estimating.

Thus, it is only be possible to find an approximation $\hat{b}^{\text{opt}}$ using substitutes for $\mathbf{f}_{\mathbf{x}}(x)$ and $\mathbf{f}''_{\mathbf{x}}(x)$.

One way to do so is to apply a so called *rule of thumb*[32] assuming that $\mathbf{f}_{\mathbf{x}}$ has a certain form for which we can derive an analytical expression used as an approximation $\hat{b}^{\text{opt}}$ for the optimal bandwidth. Such approximation of the bandwidth is as good as the estimated PDF resembles the assumed form of the actual PDF $\mathbf{f}_{\mathbf{x}}(x)$.

For example, Bowman et al., approximate the optimal bandwidth assuming a normal distribution of $\mathbf{x}$ obtaining [BA97]:

$$\hat{b}^{\text{opt}} = \hat{\sigma}^2 * (\frac{4}{3m})^{1/5}, \tag{4.39}$$

with

$$\hat{\sigma}^2 = \frac{\text{median}[|X_s - \text{median}(X_s)|]}{0.6745}. \tag{4.40}$$

being a robust estimate of the variance of the assumed normal PDF.

Alternatively, we can approximate the optimal bandwidth calculating the *integrated squared error (ISE)*

$$\text{ISE}(b) = \int_{-\infty}^{\infty} (\hat{\mathbf{f}}(x) - \mathbf{f}_{\mathbf{x}}(x))dx = \int \hat{\mathbf{f}}^2(x)dx - 2\int (\hat{\mathbf{f}} \cdot \mathbf{f}_{\mathbf{x}})(x)dx + \int \mathbf{f}^2_{\mathbf{x}}(x)dx \tag{4.41}$$

and try to find an approximate solution for b^{opt} that minimizes it.

Note that the term $\int \mathbf{f}^2_{\mathbf{x}}dx$ does not depend on b and $\int \hat{\mathbf{f}}dx$ can be calculated for a given data sample. Hence, we focus on the term $\int (\hat{\mathbf{f}} \cdot \mathbf{f}_{\mathbf{x}})(x)dx$ that corresponds to the expected value of $\hat{\mathbf{f}}$ and can be approximated using a *leave-one-out-estimator* defined as

$$\hat{\mathbf{f}}_{-j}(x) = \frac{1}{b(m-1)} \sum_{l=1, l\neq j}^{m} \theta(\frac{x_l - x_j}{b}), \tag{4.42}$$

with an expected value estimate

$$\hat{\mathbb{E}}[\hat{\mathbf{f}}(x)] = \frac{1}{m}\sum_{j=1}^{m} \hat{\mathbf{f}}_{-j}(x_j), \tag{4.43}$$

where $x, x_j, x_l \in \mathbb{R}$ Using the expected value estimate (4.43), we can formulate a so called *leave-one-out cross-validation criterion* [HWMS04]:

$$\text{CV}(b) = \frac{1}{b} \cdot \frac{1}{m^2} \sum_{j=1}^{m}\sum_{l=1}^{m} \theta \circledast \theta(\frac{x_l - x_j}{b}) - \frac{1}{b} \cdot \frac{2}{m(m-1)} \sum_{j=1}^{m}\sum_{l\neq j}^{m} \theta(\frac{x_l - x_j}{b}), \tag{4.44}$$

[32] Rule-of-thumb method is also known as *plug-in method.*

where $\theta \circledast \theta(u)$ is a convolution operation defined as

$$\theta \circledast \theta(u) = \int \theta(u-v)\theta(v)dv \text{ with } u, v \in \mathbb{R}. \tag{4.45}$$

Finding the optimal bandwidth using the leave-one-out estimator (4.44), the solution $\hat{b}^{\text{opt}}$ depends on the data sample X_s and, hence, adapts to the smoothness of the underlying PDF $\mathbf{f}_{\mathbf{x}}(x)$. In contrast, with the rule-of-thumb approximation (4.39) the obtained bandwidth depends on the sample variance of X_s.

4.2.1.5 Multivariate Density Estimation

We need to extend kernel density estimator to a multivariate case in order to use it for regression. Let

$$\mathbf{X}_s = [X_1, \ldots, X_m]^{\mathrm{T}} \in \mathbb{R}^{m \times n} \tag{4.46}$$

be a sample of m observations of a random n-dimensional variable $X = [x_1, \ldots, x_n] \in \mathbb{R}^n$, from which we estimate the joint PDF $\mathbf{f}_{\mathbf{X}}(X) = \mathbf{f}(x_1, \ldots, x_n)$. We denote the i'th dimension of the j'th observation as $x_i^{(j)}$ with $i = 1, ..., n$ and $j = 1, ..., m$.

One dimensional kernel density estimator (4.31) can be generalized to a multivariate case as:

$$\hat{\mathbf{f}}(X) = \frac{1}{\det(\mathbf{B})} \frac{1}{m} \sum_{j=1}^{m} \theta(\mathbf{B}^{-1}(X - X_j)), \tag{4.47}$$

where $\theta : \mathbb{R}^n \to \mathbb{R}$ is a *multiplicative kernel* of the form

$$\theta(Z) = \theta(z_1) \cdot \ldots \cdot \theta(z_n), \text{ for } Z = [z_1, \ldots, z_n] \in \mathbb{R}^n \tag{4.48}$$

and a symmetrical and positive definite (n, n) matrix $\mathbf{B}$ is known as *bandwidth matrix*. If we restrict ourselves to diagonal matrices $\mathbf{B} = \text{diag}(b_1, \ldots, b_n)$, we obtain

$$\hat{\mathbf{f}}(X) = \frac{1}{b_1 \ldots b_n} \cdot \frac{1}{m} \sum_{j=1}^{m} \theta\left(\frac{x_1 - x_1^{(j)}}{b_1}, \ldots, \frac{x_n - x_n^{(j)}}{b_n}\right). \tag{4.49}$$

Herewith, we can estimate a multivariate PDF $\hat{\mathbf{f}}(X)$ as

$$\hat{\mathbf{f}}(X) = \frac{1}{b_1 \ldots b_n} \cdot \frac{1}{m} \sum_{j=1}^{m} \left(\prod_{i=1}^{n} \theta\left(\frac{x_i - x_i^{(j)}}{b_i}\right)\right). \tag{4.50}$$

4.2.2 Kernel Regression

Multivariate KDE provides a foundation for nonparametric methods for solving the regression problem (Definition 4.0.13). For ease of exposition, we focus on the case where we are given a set of observations $X_s = [x_1, \ldots, x_m]$ and $Y_s = [y_1, \ldots, y_m]$ of a one-dimensional input $x \in \mathbb{R}$ and output $y \in \mathbb{R}$ constituting a training set

$$\mathcal{T} := \{(x_j, y_j) \mid 1 \leq j \leq m\} \text{ with } x_j, y_j \in \mathbb{R}. \tag{4.51}$$

Previously introduced multivariate KDE (4.50) allows us to directly obtain a model for the expectation (4.14) which, together with (4.6) can be written as

$$\mathbf{r}(x) = \mathbb{E}\Big[y \mid x\Big] = \int y' \frac{\mathbf{f}(x, y')}{\mathbf{f}_x(x)} dy'. \tag{4.52}$$

Here, $\mathbf{f}(x, y)$ is a two-dimensional joint PDF and $\mathbf{f}_x(x)$ is the marginal PDF calculated as

$$\mathbf{f}_x(x) = \int \mathbf{f}(x, y') dy'. \tag{4.53}$$

We can estimate the joint PDF using (4.50) as

$$\hat{\mathbf{f}}(x, y) = \frac{1}{b_x b_y} \cdot \frac{1}{m} \sum_{j=1}^{m} \theta\left(\frac{x^{(i)} - x_j^{(i)}}{b_x}\right) \theta\left(\frac{y^{(i)} - y_j^{(i)}}{b_y}\right). \tag{4.54}$$

Further, if θ is symmetrical and integrates to zero, we can approximate the marginal PDF as

$$\hat{\mathbf{f}}_x(x) = \frac{1}{b_x b_y} \cdot \frac{1}{m} \sum_{j=1}^{m} \theta\left(\frac{x^{(i)} - x_j^{(i)}}{b_x}\right). \tag{4.55}$$

Introducing equations (4.54) and (4.55) into (4.52) and considering the aforementioned properties of θ, we can estimate the regression function as [Wat64, Nad64]

$$\hat{\mathbf{r}}(x) = \frac{\sum_{j=1}^{m} \theta\left(\frac{x - x_j}{b_x}\right) y_j}{\sum_{j=1}^{m} \theta\left(\frac{x - x_j}{b_x}\right)}. \tag{4.56}$$

Kernel regression approach allows to create flexible nonparametric models such as *Nadaraya-Watson estimator* and *K-nearest neighbors* that we discuss subsequently.

4.2.2.1 Nadaraya-Watson Estimator

The regression function estimate (4.56) is also known as *Nadaraya-Watson estimator (NWE)* and it can be rewritten as a locally weighted average

$$\hat{\mathbf{r}}(x) = \sum_{j=1}^{m} \theta_j(x) y_j, \tag{4.57}$$

where

$$\theta_j(x) = \frac{\theta\left(\frac{x-x_j}{b}\right)}{\sum_{j=1}^{m} \theta\left(\frac{x-x_j}{b}\right)} \tag{4.58}$$

are the weights which for

$$0 \leq \theta_j(x) \leq 1, \forall x \in \mathbb{R} \tag{4.59}$$

ensure that the observations are considered depending on their distance to x. This is the reason why the model (4.56) is also called *locally weighted estimator*.

The weights θ_j not only depend on x but also on the chosen *bandwidth* b. Similar to KDE, the bandwidth determines the smoothness of $\hat{\mathbf{r}}$ and hence of the forecast curve. For $b \to \infty$, every weight becomes $\theta_j = 1/m$ and the forecast is the sample average of Y_s. At the same time, $b \to 0$ leads to $\hat{y} = y_j$ with $j = 1, \ldots, m$ for $x^* = x_j$ and undefined elsewhere.

Therefore, if bandwidth is too small to include any x_j in the vicinity of x the estimate $\hat{y}$ is not defined. For a fixed and given b, this can occur in the regions of sparse data which can be the case if we are forecasting a volatile curve and do not have numerous observations – e.g., due to curse of dimensionality discussed later in the text.

Again, choosing appropriate bandwidth becomes a fundamental problem. Similar to the one-dimensional KDE we can either use a rule of thumb such as in (4.39) or some error based criteria such as (4.44) to estimate the optimal bandwidth.

4.2.2.2 K-Nearest Neighbors

In any case, if the bandwidth is *fixed* and calculated either with a rule of thumb or cross-validation, the weighting function has constant radius of action. The outcome depends largely on the neighborhood of x and on if there are many data-points that will get a considerable weight in its vicinity or not.

An alternative is to use a *variable* bandwidth. The bandwidth $b = b_K$ is set, for a given x, so that for every prediction the model considers only K points assigning them a notable weight in the average. In particular, the bandwidth b in the equation (4.56)

becomes variable depending on the input x and its vicinity. This means, that for a given x, bandwidth $b_K(x)$ is set as:

$$b_K = |x - g_K|, \tag{4.60}$$

where g_K is the K'th nearest neighbor of x.

Setting the bandwidth in such way, together with (4.56), results in a nonparametric regression model that is a locally weighted average of *K-nearest neighbors (KNN)*

$$\hat{\mathbf{r}}_K(x) = \frac{\sum_{j=1}^{m} \theta\left(\frac{x-x_j}{b_K(x)}\right) y_j}{\sum_{j=1}^{m} \theta\left(\frac{x-x_j}{b_K(x)}\right)}. \tag{4.61}$$

Depending on x, it might happen that the neighbors of x are rather far away so that b_K in the equation (4.56) will be set large and the other way around. In this sense, K becomes the smoothing parameter of the estimator, since its increase makes the estimate $\hat{y}$ smoother.

Note that in practice, bandwidth b_K is defined in such a way that the K'th nearest neighbor is still considered. This means that if there are several neighbors with the same distance $d = |x^* - g_K|$, they all will be considered equally in the weighted average (4.61).

4.2.3 Multivariate Nonparametric Model

We can generalize univariate nonparametric models discussed above to a multivariate case, where we observe a q-dimensional input variable X. In particular, Nadaraya-Watson estimator (4.57) can be generalized for such situation introducing a multivariate kernel[33] defined as follows.

Definition 4.2.2. *Multivariate kernel* is a function $\theta_{\mathrm{m}} : \mathbb{R}^q \to \mathbb{R}$ defined as $\theta_{\mathrm{m}}(Z) = \theta(|Z|_2), \forall Z \in \mathbb{R}^q$ with the ℓ^2-norm $|Z|_2 = \sqrt{Z^{\mathrm{T}} \cdot Z}$ and a univariate kernel function $\theta(\cdot)$

With $X \in \mathbb{R}^q$ and $y \in \mathbb{R}$, we define the *multivariate Nadaraya-Watson estimator (MNWE)* as [HWMS04]

$$\hat{\mathbf{r}}_K(X) = \frac{\sum_{j=1}^{m} \theta_{\mathrm{m}}\left(\frac{X-X_j}{b}\right) y_j}{\sum_{j=1}^{m} \theta_{\mathrm{m}}\left(\frac{X-X_j}{b}\right)} = \frac{\sum_{j=1}^{m} \theta\left(\frac{\mathbf{d}_0(X,X_j)}{b}\right) y_j}{\sum_{j=1}^{m} \theta\left(\frac{\mathbf{d}_0(X,X_j)}{b}\right)}, \tag{4.62}$$

where

$$\mathbf{d}_0(X, X_j) = |X - X_j|_2 = \sqrt{\sum_{i=1}^{q} \left(X^{(i)} - X_j^{(i)}\right)^2} \tag{4.63}$$

[33] Multivariate kernels are also named spherical or radial-symmetric [HWMS04].

is referred to as *Euclidean distance*. As in the univariate case, we calculate the prediction using the locally weighted average of the output observations. The bandwidth can be variable in which case such model is referred to as *multivariate K-nearest neighbors (MKNN)*.

There are other nonparametric models such as local polynomial regression, smoothing splines, decision trees and state vector regression. However, the models based on kernel regression, discussed above, are the most common [FHT08]. Therefore, MNWE or MKNN are often simply referred to as the nonparametric model.

4.2.3.1 Model Consistency

Several researchers studied nonparametric models focusing on model consistency and its asymptotic properties [ČS20, HLC+97a]. In fact, the nonparametric models discussed above are consistent [HWMS04]. Statisticians often express theoretic performance of a model in terms of rate of convergence. Proceeding the discussion on nonparametric models, we introduce following definitions.

Definition 4.2.3. The model is *consistent* if and only if it provides an estimate $\mathbf{r}_m$ obtained using m observations for which applies

$$\underset{m\to\infty}{\text{p-lim}}\, \hat{\mathbf{r}}_m = \mathbf{r}, \tag{4.64}$$

i.e., the estimated regression function converges in probability towards the true regression function as number of observations grows.

Definition 4.2.4. Given a consistent estimate $\hat{\mathbf{r}}_m(X)$ obtained on a set of m observations, its *rate of convergence (ROC)* is defined as

$$\text{ROC} = |\hat{\mathbf{r}}_m(X) - \mathbf{r}(X)|, \tag{4.65}$$

that can be described asymptotically for $m \to \infty$.

Bias and variance of a prediction obtained through kernel regression are bounded depending on the neighborhood size that is determined by the bandwidth [HWMS04] as follows:

$$\text{bias}[\hat{\mathbf{r}}(X)] = \mathcal{O}\left(b^2\right), \tag{4.66}$$

$$\text{Var}[\hat{\mathbf{r}}(X)] = \mathcal{O}\left(\frac{1}{mb^q}\right). \tag{4.67}$$

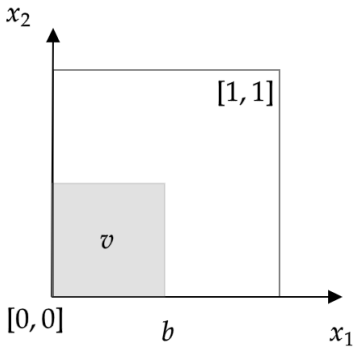

Figure 4.4: Space of a uniformly distributed two-dimensional random variable $X = [x_1, x_2]$. A square with edge length b can be expected to capture v-share of all observations in such space.

Herewith, the model (4.62) is consistent point-wise under the following asymptotic conditions [HWMS04]

$$\lim_{m\to\infty} b = 0 \tag{4.68}$$

$$\lim_{m\to\infty} mb^q = \infty \tag{4.69}$$

For $\hat{\mathbf{r}}(X)$ to converge towards $\mathbf{r}(X)$ in each point X, its neighborhood that we view as local must be as small as possible ($b \to 0$). To this end, we can reduce the bias (4.66) considering only the data in the close vicinity to X. At the same time, we need a large number of observations in that neighborhood ($mb^q \to \infty$) to reduce the variance (4.67).

The result of such trade-off is that a nonparametric regression function estimate, underlying some technical smoothness conditions, has its fastest [Sto82]

$$\text{ROC} = \mathcal{O}\left(\left(\frac{\log m}{m}\right)^{\frac{s}{2s+q}}\right), \tag{4.70}$$

with s representing the smoothness[34] of $\mathbf{r}$. The ROC decreases with the number of inputs q which is a major limitation and a manifestation of a phenomena that we describe next.

4.2.3.2 Curse of Dimensionality

In a multivariate setting $X \in \mathbb{R}^q$, the performance of a nonparametric model is limited by the *curse of dimensionality* phenomena. Given a fixed sample size m, the q-dimensional observation space becomes increasingly sparse with growing dimensionality which undermines the principle of local learning. We illustrate this phenomenon subsequently.

[34] While in traditional multivariate analysis s represents how often r is differentiable, in a more general case, s can come from Lipschitz type of regularity of $\mathbf{r}$. Both definitions are important for theoretical investigations while sufficient degree of smoothness is often given in practice.

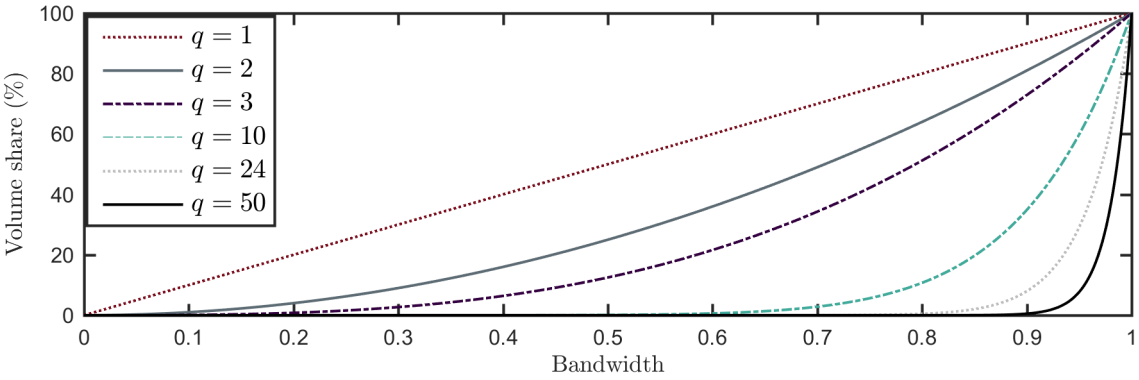

Figure 4.5: Volume share of a hypercube in a q-dimensional space $\mathbb{R}^q$ depending on the edge length (bandwidth).

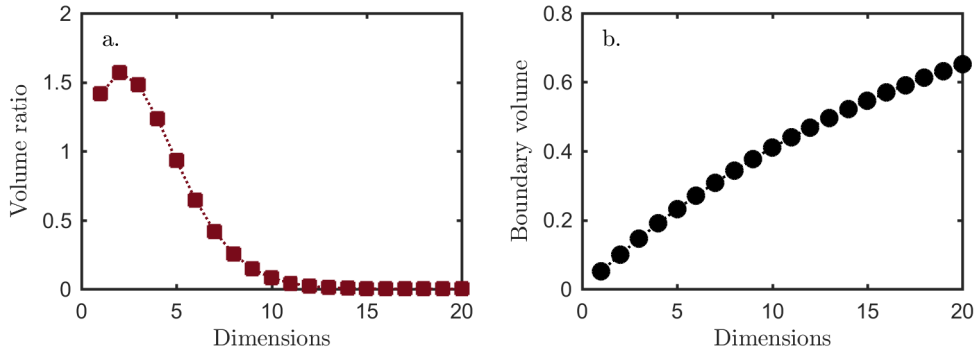

Figure 4.6: Space volume in a q-dimensional space: (a) volume of a hypersphere (unit diameter) within a unit cube relative to the volume of the cube; (b) volume of a boundary region found between two hyperspheres of diameter 0.9 and 1 respectively.

Consider a two-dimensional random variable $X_r = [x_1, x_2]$ that includes two uniformly distributed scalar random variables $x_1, x_2 \in [0, 1]$. For $X^* = [0, 0]$, the neighborhood that we expect to include a share v of all observations is a square with edge length b (bandwidth) and area $v = b^2$ (Figure 4.4). In a q-dimensional space $\mathbb{R}^q$, such neighborhood is a hypercube with volume

$$v_q(b) = V_q^{\text{cube}}(b) = b^q. \tag{4.71}$$

For instance, in a 10-dimensional case, a bandwidth that is half the available range has a volume of only $v_{10}(0.5) = 0.0098$ i.e. it includes less than 1% of all observations. The neighborhood of the same size includes less and less space and data with growing q (Figure 4.5). Put differently, the neighborhood size has to be expanded to include the same amount of observations.

At the same time, data are mostly located at the *boundary* of a high dimensional space. Consider the aforementioned hypercube as such space. We consider the largest hypersphere that we can fit within as the *inner region*. Complementary, we consider the space that does not belong to the inner region as the *outer region*.

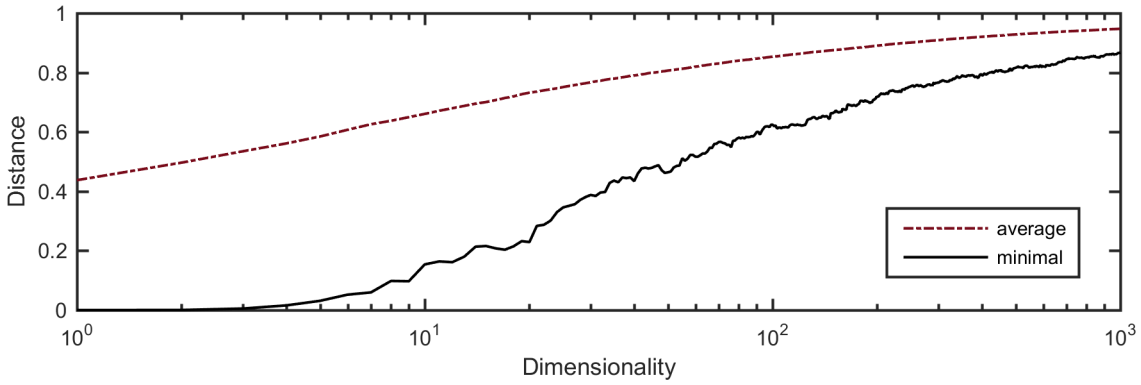

Figure 4.7: Minimal and average distance between the points in a q-dimensional space relative to the maximal distance. For a given dimensionality, distances were calculated for 10000 points that were sampled from a uniform distribution. Distances become indistinguishable as minimal and average distances converge towards the maximal distance.

In a q-dimensional space, a ball with a diameter b has a volume

$$V_q^{\text{sphere}}(b) = (b/2)^q \frac{2\pi^{q/2}}{\Gamma(q/2+1)}, \tag{4.72}$$

where Γ is the *gamma function*[35]. Herewith, the share of inner region

$$\frac{V_q^{\text{sphere}}(1)}{V^{\text{cube}}(1)} = 0.5^q \frac{2\pi^{q/2}}{\Gamma(q/2+1)}. \tag{4.73}$$

vanishes as dimensionality grows towards infinity (Figure 4.6 (a)). On the other hand, the outer region expands at its cost and takes over the entire space already for few dimensions. We observe, same counterintuitive behavior considering two embedded spheres. The relative volume of a thin boundary located between those spheres expressed as

$$V_q^{\text{boundary}} = \frac{V_q^{\text{sphere}}(1) - V_q^{\text{sphere}}(0.9)}{V_q^{\text{sphere}}(1)} \tag{4.74}$$

quickly expands (Figure 4.6 (b)). Therefore, the most of uniformly distributed q-dimensional data is found in the boundary of the space where we no longer can consider it as a local neighborhood of X^*.

The uniformly distributed observations quickly become scarce and equidistant in high dimensional spaces. Given a fixed number of data, the average distance between the points grows unconstrained with dimensionality [ABDM76]. Increasingly, a query has less and less observations in its vicinity which undermines the very idea of local learning.

[35] Gamma function $\Gamma(n)$ is a generalization of the factorial function for real and complex numbers. In particular, $\Gamma(n) = (n-1)!$ when n is a positive integer. Computing the volume of a hypersphere is only one of many applications of the Gamma function in mathematics [Gam].

Moreover, common distance notions (e.g., ℓ^{p}-norms) loose their discriminative capacity. Relative differences between distances vanish as the nearest and farthest points have almost the same distance (Figure 4.7). Under such circumstances, the concepts of *similarity* and *neighborhood* are no longer meaningful since for any query all points appear equally far away.

Consequently, curse of dimensionality impairs the accuracy of a nonparametric model[36]. As the data become scarce, random variation and noise within the data obscure the important features. As the variance of distances becomes negligible we may even get numerical precision problems choosing thresholds, weights, ordering and so on.

4.2.3.3 Data Sparsity in High-Dimensional Space

Data sparsity has a negative effect on the theoretical performance of a multivariate nonparametric model. In practice, data distribution in the space is seldom uniform. However, the *effective number of observations* m'_X used by a local model at X is

$$m'_X = mV_q^{\text{sphere}}(b)\,\mathbf{f}\,(X) \tag{4.75}$$

which is, for some PDF $\mathbf{f}\,(X)$, bounded away from zero. For a nonparametric model to be consistent, the number of observations

$$m'_X \sim mb^q \tag{4.76}$$

must grow sufficiently fast towards the infinity with $b \to 0$ according to (4.72). This becomes increasingly difficult with growing dimensionality which limits the ROC to (4.70).

As a result, theoretical accuracy of a nonparametric model deteriorates with a growing number of inputs (4.70). Due to this limitation, kernel regression is pre-handicapped for multivariate problems. Nevertheless, there has been a notable research effort to overcome the curse of dimensionality by exploring some forms of dimension reduction [HWMS04]. A promising idea to approach data sparsity come from the functional data analysis and regression methodology that we highlight next.

[36] Generally, all models are affected by the curse of dimensionality to some extent. In practice, different models approach high-dimensional tasks using some form of dimensionality reduction for which various methods are available [GT18].

4.3 Functional Regression

In the modern applied science, researchers increasingly the situations in where collected data can be viewed as continuous[37] rather than a set of discrete measurements[38]. While the measurements are still discrete, the advances in measurement and computation methods allow to dramatically increase the sampling rate and apply sophisticated smoothing techniques to a point where obtained curve can be regarded as a continuous function. Though data acquisitions and smoothing techniques are still subject of ongoing research, this motivates a parallel development of a novel data analysis methodology based on the assumption that the underlying data is continuous.

Functional data analysis (FDA)[39] is a young area of statistics and has been investigated in-depth only recently. At the beginning of this century, different researchers provided the first comprehensive FDA-theory focusing on functional parametric [RS02, RS05] and nonparametric [FV06] regression methods. A recently published handbook unites both perspectives and provides an extensive overview of the novel research field [FR11].

4.3.1 Functional Data

To describe functional data, we regard a given continuous curve ϕ as an observation of a *functional random variable (FRV)*

$$\Phi = \{\phi(t); t \in (t_{min}, t_{max})\} \tag{4.77}$$

which takes values in an *infinite-dimensional space of functions* $\mathbb{F}$ endowed with a distance notion defined as follows.

Definition 4.3.1. *Distance notion* $\mathbf{d} : \mathbb{F} \times \mathbb{F} \to \mathbb{R}$ is a semimetric which $\forall \chi_1, \chi_2 \in \mathbb{F}$ has the following properties:

1. $\mathbf{d}(\chi_1, \chi_2) \geq 0$,
2. $\mathbf{d}(\chi_1, \chi_1) = 0$,
3. $\mathbf{d}(\chi_1, \chi_2) = \mathbf{d}(\chi_2, \chi_1)$

[37] Given a very fine resolution, we can view a sampled curve as continuous. Note that this is just an assumption while the data is still stored discreetly.

[38] While, following the topic of this study, we restrict ourselves to curves, the notion of a functional variable is not restricted to curves but can be extended to consider surfaces or any other more complex mathematical objects.

[39] The term functional data analysis was first introduced by Ramsay and Dalzell in 1991 [RD91].

Additionally, a *functional dataset*

$$\mathcal{F} := \{\phi_j \mid 1 \leq j \leq m\} \text{ with } \phi_j \in \mathbb{F} \tag{4.78}$$

includes a collection of continuous observations of Φ.

We can model a functional variable with a regression equation

$$\phi(t) = \mathbf{r}(\chi(t)) + \epsilon(t), \tag{4.79}$$

where the regression operator $\mathbf{r}$ is an element of the space $\mathcal{R}(\mathbb{F}, \mathbb{H})$ of all correspondences between $\mathbb{F}$ and $\mathbb{H}$ that is a measurable and separable Hilbert space[40]. Though if χ, ϕ can be in the same space $\mathbb{F}$, the setting $\mathbb{F} \neq \mathbb{H}$ is the most general for which the consistency of regression models have been studied [FVKV12].

The regression problem is formulated similarly to the multivariate case (Definition 4.0.13). Consider a sample of IID observations $(\chi_1, \phi_1), \ldots, (\chi_m, \phi_m)$ of the random pair (χ, ϕ) valued in $\mathbb{F} \times \mathbb{H}$. Herewith, the problem consists in estimating the operator

$$\mathbf{r}(\chi) = \mathbb{E}\Big[\phi \mid \chi\Big], \tag{4.80}$$

that underlies some smoothness restrictions while, for the *error term* $\epsilon(t)$, we assume that $\mathbb{E}\Big[\epsilon \mid \chi\Big] = 0$.

A model for estimating $\mathbf{r}$ consists of introducing some constraints of the form

$$\phi \in \mathcal{C} \tag{4.81}$$

and is called a *functional parametric model* if $\mathcal{C}$ is indexed by a finite number of elements and *functional nonparametric model* otherwise (Definition 4.0.14). Historically, there was a higher interest in nonparametric models, due to their flexibility and the lack of graphical tools[41] for representing and investigating parametric models [FR11]. Therefore, we proceed describing functional version of the previously introduced nonparametric model based on kernel regression.

[40] Hilbert space is a space of infinite sequences of real numbers that are square summable and is endowed with an inner product that defines the corresponding norm $\|\cdot\| = \langle\cdot,\cdot\rangle$. Further, Hilbert space is called separable if it has an orthonormal and countable basis [Mus14]. Hilbert space is also a generalization of $\mathbb{R}^n$. Euclidean space is a finite dimensional Hilbert space.

[41] Graphical representation of data dependencies in form of scatter and various residual plots are common aids for developing multivariate nonparametric models [HA18]. Such tools are inapplicable in infinite-dimensional setting of functional data analysis.

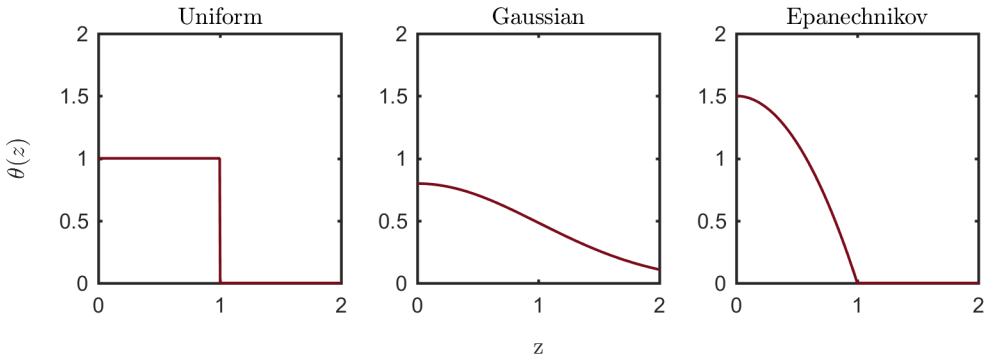

Figure 4.8: Functional kernels examples. The figure shows functional versions of uniform $\mathbb{1}(0 \leq z \leq 1)$, Epanechnikov $\frac{3}{2}(1-z^2)\mathbb{1}(0 \leq z \leq 1)$, and Gaussian $\frac{2}{\sqrt{2\pi}} \exp(-\frac{1}{2}z^2)\mathbb{1}(0 \leq z)$ kernels.

4.3.2 Functional Nadaraya-Watson Estimator

Given a way to measure proximity between functional data-points using $\mathbf{d}$, we adopt local weighting idea for nonparametric regression of the functional data. The operator $\mathbf{r}(\chi)$ can be estimated with *functional Nadaraya-Watson estimator (FNWE)* using a similar kernel regression approach as in the multivariate case [FV06]:

$$\hat{\mathbf{r}}(\chi) = \frac{\sum_{j=1}^{m} \theta_{\mathrm{f}}\left(\frac{\mathbf{d}(\chi,\chi_j)}{b}\right) \phi_j}{\sum_{j=1}^{m} \theta_{\mathrm{f}}\left(\frac{\mathbf{d}(\chi,\chi_j)}{b}\right)}, \tag{4.82}$$

where θ_{f} is a functional kernel defined as follows.

Definition 4.3.2. *Functional kernel* $\theta_{\mathrm{f}}(z)$ is an operator $\theta_{\mathrm{f}} : \mathbb{R} \to \mathbb{R}$ that is Lipschitz[42] for $z \in [0, 1)$ and supported[43] on $[0, 1]$ which $\forall z \in \mathbb{R}$ satisfies:

- $\theta_{\mathrm{f}}(z) \geq 0$
- $\int \theta_{\mathrm{f}}(z)dz = 1.$

In contrast to the Definition 4.2.1 , functional kernel should be asymmetrical since $\mathbf{d}(\chi_1, \chi_2) \geq 0, \forall \chi_1, \chi_2 \in \mathbb{F}$ and *bandwidth* b is positive. We provide some example in Figure 4.8.

Instead of various restrictive assumptions on $\mathbf{r}$ (e.g., linearity), nonparametric approach only assumes the regression operator to be sufficiently smooth. We express this assumption through a Hölder condition, for some positive constant c:

$$\forall s > 0, \forall (\chi_1, \chi_2) \in \mathbb{F} \times \mathbb{F}, |\mathbf{r}(\chi_1) - \mathbf{r}(\chi_2)| \leq c\, \mathbf{d}(\chi_1, \chi_2)^s. \tag{4.83}$$

[42] A Lipschitz function underlies some continuity conditions. In particular, a function is said to be Lipschitz if its first derivative is bounded [Wei].

[43] *Support* of θ_{f} is the domain where function is not equal to zero.

The estimate (4.82) is consistent [FV06]. The studies of its asymptotic properties initially focused on models with a scalar response [FV06]. It was only in the recent past that the nonparametric estimate was shown to be consistent when both χ and ϕ are functions [FVKV12].

Similar to the multivariate method (4.62), the estimate $\hat{\mathbf{r}}(\chi)$ is determined by the data present in the vicinity of χ that we describe as a *ball* with radius b and center χ as

$$\mathcal{B}_{\mathbb{F}}(\chi, b) := \{\chi' \in \mathbb{F} \mid \mathbf{d}(\chi', \chi) \leq b\}. \tag{4.84}$$

The behavior of the estimate depends on the number of data present in $\mathcal{B}_{\mathbb{F}}$ that we describe with the *small ball probability*

$$P_{\mathcal{B}}(\chi, b) = \mathbb{P}\Big[\mathbf{d}(\chi', \chi) \leq b\Big] = \mathbb{P}\Big[\chi' \in \mathcal{B}_{\mathbb{F}}(\chi, b)\Big]. \tag{4.85}$$

representing the *local concentration* of data in the vicinity of χ that satisfies the condition

$$\exists c_1, c_2 \in \mathbb{R}^+, 0 < c_1 P_{\mathcal{B}}(\chi, b) \leq \mathbb{P}\Big[\mathbf{d}(\chi, \chi_r) \leq b\Big] \leq c_2 P_{\mathcal{B}}(\chi, b). \tag{4.86}$$

This condition describes the density of uniformly distributed data in $\mathcal{B}_{\mathbb{F}}(\chi, b)$ (e.g., $P_{\mathcal{B}}(X, b) \sim b^q, \forall X \in \mathbb{R}^q$).

Under the conditions that connect $P_{\mathcal{B}}(\chi, b)$ to the bandwidth:

$$\lim_{m \to \infty} b = 0, \tag{4.87}$$

$$\lim_{m \to \infty} \frac{m P_{\mathcal{B}}(\chi, b)}{\log m} = \infty \tag{4.88}$$

the ROC of the FNWE (4.82) is [FV06]

$$\text{ROC}_{\text{FNWE}} = \mathcal{O}(b^s) + \mathcal{O}\left(\sqrt{\frac{\log m}{m P_{\mathcal{B}}(\chi, b)}}\right). \tag{4.89}$$

Here, the first term corresponds to the estimator bias and depends only on the smoothness of $\mathbf{r}$ expressed through s. The second term corresponds to the variability of the prediction and is determined by the local concentration $P_{\mathcal{B}}$.

Data concentration depends on the data generating process but also on the topology of the observation space according to (4.85). We determine the topology through the distance notion definition. For instance, having chosen $\mathbf{d}$ we call the data generating process of *fractal order* τ if

$$P_{\mathcal{B}}(\chi, b) \sim C b^{\tau}. \tag{4.90}$$

For such process, the functional nonparametric estimator converges at fastest with [FV06]

$$\text{ROC} = \mathcal{O}\left(\left(\frac{\log m}{m}\right)^{\frac{s}{2s+\tau}}\right), \tag{4.91}$$

which is comparable to the ROC_{MNWE} in (4.70).

In fact, functional approach is a generalization of the multivariate nonparametric model. We can show[44] that, for $\mathbb{F} = \mathbb{R}^q$, Euclidean distance notion and a given PDF $\mathbf{f}(X)$, the data concentration is of a form

$$P_{\mathcal{B}}(\chi, b) = V_q^{\text{sphere}} \sim Cb^q. \tag{4.92}$$

Herewith, the estimator (4.82) corresponds to (4.62) and its ROC is given by (4.70). Functional data analysis extends the nonparametric model theory to other processes and cases where $\mathbf{f}(X)$ is not easily described.

Similar to the multivariate approach (4.62), the estimate $\hat{\mathbf{r}}(\chi)$ is determined by the data present in the vicinity of χ. However, which and how many observations we encounter in the neighborhood depends, not only on the ball size controlled by b (4.84), but also on the choice of the semimetric $\mathbf{d}$. Selected distance notion describes the similarity between the curves, determines the topology of $\mathbb{F}$ and, thereby, affects the convergence rate of the functional nonparametric estimator (4.89).

4.3.3 Data Sparsity in Infinite-Dimensional Space

In this section, we discuss the sparsity of infinite-dimensional data and its effect on estimator performance in an infinite-dimensional space. Curse of dimensionality limits the ROC of a multivariate nonparametric estimator (4.70). We have seen that observation space becomes sparse with growing dimensionality of the input vector. In a functional model, the input $\chi \in \mathbb{F}$ is infinite-dimensional. Therefore, we need to discuss how sparsity affects the performance of the functional nonparametric estimator.

If we directly apply the distance based on an ℓ^2-norm to the functional nonparametric estimation problem, we note that many of the continuous data generating processes that we we encounter in practice are of *exponential* type[45] which for some constants $\alpha_1, \alpha_2, \alpha_3, C \in \mathbb{R}^+$ have the associated concentration function of a following form:

$$P_{\mathcal{B}}(\chi, b) \sim C \cdot \exp\left(-\frac{1}{b^{\alpha_1}} \log \frac{1}{b^{\alpha_2}}\right) \text{ as } b \to 0. \tag{4.93}$$

[44] See Proposition 13.14 in [FV06].

[45] For instance, research shows gaussian and diffusion processes are of exponential type [FLV06, LS01].

For such processes the estimator performance is limited by [FV06]

$$\text{ROC}_{\ell^2} = \mathcal{O}\left(\left(\frac{1}{\log m}\right)^{\alpha_3}\right). \tag{4.94}$$

Herewith, the ROC in infinite-dimensional setting is limited by some power of m which is unsatisfactory from statistical point of view and is substantially slower than (4.70). For a given X, there are too few observations in its vicinity in order for (4.82) to converge fast. The reduced concentration measure of the process reduced the ROC. This effect is sometimes regarded as *curse of infinite dimensionality* [FV03, FLV06, FV06, Gee11].

Further, for any $\mathbf{d}$, the bias and variance of the estimator (4.82) are bounded [FMV07]:

$$\text{bias}[\hat{\mathbf{r}}(X)] = \mathcal{O}(b), \tag{4.95}$$

$$\text{Var}[\hat{\mathbf{r}}(X)] = \mathcal{O}\left(\frac{1}{mP_{\mathcal{B}}(\chi, b)}\right). \tag{4.96}$$

Considering bias-variance dilemma, the small ball probability $P_{\mathcal{B}}(\chi, b)$ directly affects the variance in of the estimator (4.96) and influences its ROC (4.89).

However, we can change the topological structure of the observation space by the choice of $\mathbf{d}$ increasing concentration of the data. In fact, there always exists a distance notion according to which data generating process is of fractal type [FV06]. With such distance notion, we can achieve theoretical performance (ROC) that is better than with a multivariate nonparametric model. In practice, we will use this observation developing our model in Section 8.2.

Sparseness of data limiting the performance of the nonparametric model (curse of dimensionality) can be addressed in the infinite space $\mathbb{F}$ by the choice of $\mathbf{d}$.

It appears that, given appropriate choice of a semimetric $\mathbf{d}$, the curse of dimensionality is either partially canceled or has no significant effect on functional data given notable correlation within the curves [FV06]. This holds to a variety of applications such as price [Lie13, AVCMSR13, PMA17] and load forecasting in the transmission system which we discuss next.

5 State-of-the-Art Load Forecasting

Short-term load forecasts (STLF) allow to balance electricity generation and demand on a daily basis and are fundamental for the power system operation and stability. These forecasts are used for control and scheduling allowing grid operators to plan ahead and adjust the production or, if possible, the consumption to avoid congestions and reduce the operating costs. Forecast accuracy presents a large potential for cost savings, and it has been addressed by numerous research works some of which are mentioned in this chapter.

While there is no clear definition for the short-term horizon, in the load forecasting literature, short-term is regarded as a horizon from one to several hours up to one to several days ahead [MTAR15, RK15]. Traditionally, there was a much larger interest in intraday forecasts that are required for power system control [GAWY17]. Additionally, day-ahead forecasts are computed for the entire upcoming day and are required for scheduling the available production and consumption flexibilities.

In this chapter, we discuss the STLF-methods found in the power engineering literature. Historically, balancing and control had to be done globally, nationwide, at the level of power system and transmission grid. Therefore, most of the existing methods were developed for forecasting the loads at the transmission system level (Section 5.1). The same methods were among the first proposals for predicting the loads in the distribution system and building domain (Section 5.2). At the end of this chapter, we summarize the insights from the reviewed literature (Section 5.2) which we will use for developing a novel method for wide-scale day-ahead building load forecasting.

5.1 Transmission System Load Forecasting

The literature on transmission system short-term load forecasting is extensive. In this section, we highlight only some out of the numerous existing works. For a more extensive survey of this field, the reader can consult comprehensive literature reviews [SPS16, KMS+16a, GAWY17, FGSC19, AIJH21]. Transmission system load mostly depends on hour, weekday and weather [AN02]. Consequently, researchers and practitioners predominantly use parametric regression models which presuppose an explicit dependency on the exogenous variables.

5.1.1 Parametric Models

There is a large variety of parametric models for intraday load forecasting in the existing literature. The propositions are often based on statistical techniques from time series analysis that explore correlation, trend and seasonal variation such as it is done by multiple-regression model [MR89], aforementioned ARIMA-model [CHC95, HCC95, **?**], as well as SARIMA model [CPB09]. Another family of methods use machine learning techniques such as ANNs as in [WNJ12, HPS01], *fuzzy logic* [BAB12] and *support vector regression (SVR)*[1] [CCL04]. With ARIMA and ANN being the most common approaches, a recent and more extensive overview of parametric load forecasting models can be found in [SS12, MTAR15, SPS16].

5.1.2 Nonparametric Models

With well known characteristics of the load time series, application of nonparametric regression techniques at the transmission system level is much more rare. Charytoniuk et al., were one of the first to propose nonparametric load model to be used for intraday forecast [CCVO98]. They also extended the autoregressive kernel model to consider the weather forecast. In a simulation, their model was compared with ANN obtaining slightly lower accuracy. Though there have been further propositions to apply KNN [AC13, TLRSR$^+$04], nonparametric methods at the transmission system are mostly used as benchmarks [FM07, TVM02].

5.1.3 Functional Models

Up to the present, there are only few propositions to apply functional regression for the load forecasting. Most notably, recent theoretical advances in functional data analysis reinforced the interest in nonparametric models. At first, research efforts focused on intraday forecasts creating models with functional inputs and scalar response for which the theory was already available [FV06]. More recently, the mathematicians extended the theory and explored the asymptotic properties of the regression models where both input and output are functional data [FVKV12]. These advances allowed to create day-ahead load forecasting models.

While Ferraty and Vieu [FV06] focused on theoretical foundations of a functional nonparametric model, Antoniadis et al., [APS06] proposed a functional nonparametric forecaster for the intraday load of Paris. In the corresponding simulation, the accuracy of the functional forecast was comparable to that of a SARIMA model commonly used for power

[1] The name support vector regression refers to the state vector machine adopted for the regression task.

system load forecasting. Following the nonparametric methodology, Antoniadis et al., measured the distance between the curves with discrete wavelet decomposition and used a fixed bandwidth found through cross-validation method, adopted for time series as described in [Har96]. The authors also observed that, as with a multivariate nonparametric model, the choice of the kernel function had only little influence on the accuracy.

Aneiros and Vieu extended the functional nonparametric approach (4.82) to the *semi-functional partially linear (SFPL)* model that allows to consider linear dependencies on exogenous multivariate inputs [AV06]. Further, they studied the theoretic and asymptotic properties of the SFPL-model assuming independent data. In particular, they showed that the rate of convergence of the nonparametric component is unaffected by the linear part of the model. They generalized the conclusions to the case where the data underlies mixing conditions (i.e. time series) in a subsequent publication [AV08].

Vilar et al., used these results to apply SFPL for predicting the day-ahead load of Spain [VCA12]. They applied 24 separate models with scalar response and considered different variants of the model including fixed bandwidth, variable bandwidth, various kernels and distance notions. The SFPL-model achieved accuracy superior of a naive and comparable to a SARIMA model. Aneiros et al., extended the SFPL-model to having functional response [AVCMSR13, AVR16]. Their approach predicted the curve at once using a single model, that required only few hyperparameters selected through cross-validation. The authors observed a slight but statistically significant improvement over the former proposition and overall accuracy comparable to the ARIMAX model.

Paproditis et al., proposed the *functional similar shape forecaster* — a different way to consider exogenous variables when predicting system load using nonparametric methodology [PS13]. Given the prediction of the exogenous variables (e.g., weekday, weather), the upcoming day was assigned to one of the precalculated reference curves. The forecast is calculated with functional nonparametric regression (4.82) using the reference segment as the query. The authors proved the weak consistency of such model, and demonstrated its accuracy predicting the day-ahead load of Cyprus. The bandwidth of the model was selected using *empirical risk of prediction methodology* previously introduced in [APS09].

Accuracy of the functional forecasters was compared in a series of studies with models that rely on conventional multivariate approaches such as: ARIMA and NWE [SL15], SARIMA [APS06, PS13, VCA12], MLP [PS13] and ARIMAX [AVR16]. Those studies demonstrate that functional models can compete with the more traditional methods for power system load forecasting. Nevertheless, model propositions based on functional regression are very rare to this day.

5.2 Building Load Forecasting

Until recently, most propositions for building load forecasting were focused on adopting transmission system models for lower load aggregation levels. In contrast to the transmission system, forecasting of building power demand has various specifics. Building domain includes a large variety of loads with different time-series characteristics which can be highly volatile, nonstationary and affected by weather and workday calendar to a varying extent, depending on the facility size and purpose. In general, it is easier to forecast large aggregated loads present in high voltage domains [SR18]. Several models that can be very accurate for a transmission system were found failing to reflect the volatility of building electricity consumption where the aggregation is much smaller [JAW+12].

Building load forecasting literature widely discusses predicting intraday energy demand (electrical and thermal) of a specific building or a group of buildings. Contrastingly, in this section, we highlight the approaches for predicting total electricity consumption of a building with hourly or subhourly resolution. A more extensive discussion on building energy demand forecasting can be found in comprehensive reviews [ACGW18, MMA+22].

5.2.1 Parametric Models

As for the transmission system, the absolute majority of load forecasting models proposed for buildings follow a parametric regression approach. In the recent years, several researchers applied traditional statistical techniques such as multiple linear regression and ARIMA. These multiple linear and autoregressive models are commonly set up using in-depth time series analysis of the load curves. At the same time, machine learning techniques (especially ANNs) became the most common methodology with numerous applications for building energy forecasting [RZ19]. However, many propositions focus on forecasting heating and cooling power demand [YRZ05, MST02] or total daily consumption [NF08, BRF16] and are not relevant for day-ahead load forecasting. In this section, we review the parametric regression methodology applications for predicting building electricity demand with hourly and subhourly resolution found in the literature.

5.2.1.1 The Multiple Linear Regression

The *multiple linear regression (MLR)* approaches focus on modeling explicit relation between predicted load and explanatory variables. Such relation needs to be defined manually including the choice of the variables the load is believed to depend upon. For larger buildings, MLR-models can have a reasonable accuracy and are simple to implement [IYIO14, HWVA13]. The major limitation is that those models cannot deal

with nonlinearities that become apparent on smaller buildings [BZN+19]. Moreover, the MLR-approach requires noncolinearity between the input variables which can be addressed through a manual feature selection [FRS+13]. Further, this approach cannot model out-of-sample and hence needs large amounts of data to capture all possible situations.

5.2.1.2 Autoregressive Models

Autoregressive models such as ARIMA, SARIMA and ARIMAX are standard in time series modeling and are among the most common approaches used for building STFL [BZN+19]. These models describe the time series as a linear combination of its past values and require extensive manual setup. To this date, there is no systematic way to set hyperparameters rather than through trial and error and an autoregressive models are usually set up with manual fine-tuning [TB13, NB10]. While, the linearity assumption appears to be too prohibitive for many building load forecasting applications, the statistical autoregressive models were often used as benchmarks [FBP11, PBF11b, PBF11a].

5.2.1.3 Artificial Neural Networks

Several nonlinear parametric modeling methods were proposed for building load forecasting. Among various machine learning approaches discussed in the literature, *artificial neural networks (ANN)* are the most intensively investigated methods [ACGW18, BZN+19, VKS20]. Accuracy and the setup of a particular ANN-model primarily depend on the characteristics of the forecast time-series and the complexity of the modeled relationship [Bis94, LGT98]. Hence, we only consider the studies focusing on predicting building total electricity consumption with hourly and subhourly resolution. Such time series are often more volatile than the widely investigated thermal energy demand [KAS13].

We summarized the forecasting models based on an ANN-approach for predicting either intraday or day-ahead overall load curve of a building in Table 5.1. The ANN-models we found in the literature have several aspects in common (Table 5.1). They were developed for a specific building or building type. The well-performing architecture was set up manually, given explicit knowledge of the problem, researcher experience and intuition combined with a trial and error process. Model inputs were related to historical load, calendar features and, sometimes, daily weather. Further, the ANNs required large amounts of historical data [CBWS16]. The majority of the publications applied a general black box modeling approach with a feedforward neural network set up manually or using some heuristics [RZ19].

Feedforward neural networks are the most common type among load forecasting applications [RZ19]. This approach allows to create a nonlinear model of the relationship

Table 5.1: Publications on neural-network models for predicting intraday or day-ahead total electricity consumption of buildings with hourly and subhourly resolution.

Reference	Network	Load	Dataset (months)	Horizon	Inputs	Setup	Hidden layers	Hidden neurons	Training algorithm
[BFS+15]	MLP	hospital	12	DALF	load, calendar, weather	manual	1	20	BP
[MHD+13]	MLP	aggregation	17	DALF	load, weather	manual	1	15	BP
[CSZ+15a]	RNN	edu. building	12	DALF	load, calendar, weather	manual	1	n/s	LM
[MRCA14]	RNN	lab. building	18	1 h	load, calendar, weather	manual	1	10	LM
[POC+17]	RNN	hypermarket	12	1 h	load, calendar, weather	manual	1	var	var
[MNGK16]	RBM	home	48	DALF	load	manual	1	10	LM
[MAM16a]	LSTM	home	48	60 h	load, calendar	manual	2	10	BP
[AMM17]	CNN	home	48	DALF	load, calendar	manual	2	20	BP
[SLW16]	ESN	office building	48	1 h	load	auto	1	50	n/s
[KDJ+17]	LSTM	69 homes	48	6 h	load, calendar	manual	2	20	BP
[RNK16]	RBM	40 enterprises	36	DALF	load, calendar, weather	manual	4	150	LM

between the forecast load and exogenous inputs that the electricity consumption is believed to depend upon. These inputs are frequently related to historical load, calendar and weather.

Studying various reviews [RZ19, KAS13] and the references therein dedicated to the feedforward neural networks for distribution system load forecasting, we notice that the researchers often assume constant occupancy level for the building and focus on modeling energy demand as a function of weather and calendar features. For instance, Bagnasco et al., did so applying an MLP with one hidden layer containing 20 neurons to forecast electricity consumption of a hospital complex in Italy [BFS+15].

While the occupancy can be assumed to follow a steady weekly pattern for larger buildings, this is not the case for smaller, especially residential, buildings where user behavior is one of the main consumption drivers. When Rodrigues et al., used a similar network in terms of size and inputs and trained it to forecast intraday load of 93 homes, their *multilayer perceptron (MLP)* was half as accurate [RCC14]. This illustrates that using the same network on a different building type can result in significant difference in model performance.

Indeed, there have been recent attempts to consider occupancy explicitly given a set of sensors in a building as by Massana et al., [MPB+15]. The authors showed that using occupancy made the prediction more accurate in their particular case. At the same time, Wang et al., argue that it is impractical to rely on this data for a short-term load forecasting on a wide scale where we might have numerous end-consumers and strict privacy regulations [WCHK18].

Recurrent neural networks are an alternative approach to the feedforward architectures. There are several applications of recurrent neural networks which consider the occupancy

implicitly. In particular, these networks use the inherent structure of the load time series to create nonlinear autoregressive models instead of relying exclusively on exogenous variables. While NAR-models are purely autoregressive, NARX-architectures, additionally, allow to explicitly consider external inputs and are more common [RZ19].

For instance, Mena et al., proposed a NARX-model to predict the intraday load of a laboratory building equipped with an air-conditioning and a photovoltaic module [MRCA14]. Consequently, the model considered workday calendar, weather and air-conditioning actuator signal along with the previous day load curve. The researchers trained the network on one year of data with one minute resolution, while their study focused on finding the best hyperparameters of the network by trial and error.

Likewise, Pirjan et al., empirically compared different learning algorithms, number of hidden neurons and lags for predicting the load of a commercial center in Romany [POC+17]. The authors used eleven months of one minute resolution data for training. A NAR-model was contrasted with NARX using outside ambient temperature as an input on a one month of test data.

Deep neural networks (DNN) are the state of the art in machine learning research, and there have been remarkable applications in image and speech recognition [GBC16]. However, for the building STLF there exist only few propositions because this type of networks requires large amounts of training data which might not be available in the distribution system even despite mass adoption of smart metering [BZN+19].

In fact, many DNN applications were demonstrated on the same research dataset that contains the four years of one-minute resolution load measurements of a single household and its separate rooms. Amarasinghe et al., applied a feedforward methodology with the CNN-architecture. The network included two hidden layers with 20 neurons using historical load, weather, and calendar as inputs. Recurrent-neural-network type was used with *Restricted Boltzmann machine (RBM)* by Mocanu et al., [MNGK16] and *long short-term memory (LSTM)* by Marino et al., [MAM16b]. All propositions used three years of data for training and one year for the test. They all achieved comparable accuracy, only slightly increasing the performance in comparison to a shallow feedforward neural network and exposed some fundamental practical issues when applying DNNs.

As with all neural networks, model accuracy relies on an appropriate choice of hyperparameters [MNGK16]. However, the works mentioned above as well as others, using either a similar architecture (RBM in [RNK16] and LSTM in [KDJ+17]) or another recurrent neural networks named Eco State Network in [SLW16] – they all relied on a manual trial and error process to find an adequate network setup.

Due to the size, training a DNN is a computationally demanding process and trying out numerous networks can become impractical. At the same time, if numerous different loads

are to be forecast, it is senseless to manually set up each network as was also noted by Kong et al., in [KDJ+17] forecasting 69 homes and forced to use some rules of thumb for the setup.

Overfitting presents another limitation. Given a very complex function such as the load of a small building or a single home, increasing the network size only improves the training but not the test error [MAM16b]. Amarasinghe et al., came to the same conclusion for the CNN where many hidden layers produced excellent training error but failed to generalize on an unseen data [AMM17].

5.2.2 Nonparametric Models

Building load forecasters based on nonparametric methodology are scarce in the literature and are mostly used as benchmarks [BM15b]. Nevertheless, there have been some rare propositions that we highlight below.

Brown et al., propose multivariate KNN-model which they tested along-with a feedforward ANN on four different office and educational buildings [BBB†12]. They concluded that a KNN-model can be significantly more accurate than the evaluated ANNs, when training data is scarce. The KNN-model considered weather variables, which the authors showed to have different effect on the energy consumption depending on the building. Paradoxically, the researchers observed better forecasting results on some buildings, when weather information was disregarded. They also compared KNN with other models in a forecasting competition [KH94] and found that their model was often the least accurate comparing to the others.

Arora et al., applied a model based on kernel density estimation onto small buildings in Ireland from a widely used public smart-meter dataset[2] and compared the forecast accuracy on houses and non-domestic buildings over the horizons of up to a week ahead [AT14]. Their approach was developed further by [AAD+17], where authors modeled the relationship between the temperature and the load nonparametrically for five residential buildings in Montreal. The model was based on adaptive conditional density estimation, extracting the temperature-related component such as heating and air conditioning consumption from the total electricity demand.

Chaouch presented a forecasting approach based on functional kernel regression [Cha14]. He combined unsupervised curve clustering [DFV06] with the functional wavelet kernel approach [APS06], to predict the loads from the aforementioned Irish smart-meter dataset.

[2] We used this dataset for the evaluation and discuss it in detail in Section 9.1.1.

5.2.3 Heuristic Models

Currently, the distribution system operators predict building loads on a wide scale using *standard load profiles* [ECo,Ber00]. In praxis, this heuristic approach, proved being simple and effective technique that provides accurate forecasts for larger load aggregations. There are other simple approaches based on *profiling* and *persistence* heuristics that can be as effective as the more sophisticated models discussed above [STH+15].

It suggests itself to use building load measurement data obtained from smart meters for a profiling-based forecasting approach. Baranek et al., proposed a methodology to enhance the existing profiles creating, so called, *individual load profiles (ILP)* for each building in [BPT13]. While KNN is rarely used for regression, it can be used to better classify the low-voltage consumers creating a profiles for each identified class [MRRJ11].

Stephen et al. proposed a bottom-up[3] profiling-based approach for the STLF in the distribution networks or buildings that include numerous end-consumers [STH+15]. Researchers compared their approach to other reference models common such as persistence forecast, feedforward ANN and ARIMA. Forecasting the load of an aggregation of 123 residential consumers, the profiling-based model and ARIMA were the most accurate showing comparable results, while the proposed method was more computationally efficient.

Persistence heuristics are commonly used as benchmark models [DBW15, HGZA18, HFS14]. *Naive model* takes most recent load curve as the forecast for the upcoming day. *Weekday persistence* accounts for workday calendar using the most recent load curve of the same weekday as a forecast. Haben et al. created a more advanced persistence heuristic [HWVG+14]. To compute a baseline profile for each weekday, the authors used a curve similarity notion particularly suited for volatile time series such as load curves measured on small buildings and households.

Other Models

There are further approaches to building load forecasting, though these are less common than ARIMA- and ANN-based methods [ACGW18]. Most notably, parametric *state vector regression (SVR)* technique was often used for building load forecasting. For instance, Dong et al., used SVR predicting the energy demand of four commercial buildings [DCL05]. Similar SVR-approach was used for other buildings equipped with numerous sensors improving the accuracy in comparison to an ANN-forecast [JSCT14].

[3] Given a load aggregation, a bottom-up forecast predicts each individual load separately and, then, aggregates the forecasts to an overall prediction of the aggregated load.

Forecast accuracy can be improved by incorporating several predictions done with different techniques. The models combining multiple forecasts into one start to appear in the literature and are often regarded as *hybrid* or *ensemble models*. For instance, an average of different forecasts can, in some situations, be more accurate than individual predictions [HA18, BM15a, Bur17]. Another approach is to select the best model from a cohort of sub-models on the training data with a dedicated switching function [BM15b, REG15].

Subbayya et al., focused on selection criteria that can be used for hybrid models [SJW13]. Borges et al., also investigated combining different models and tested their approach on a university campus [BPF11]. They concluded that hybrid model only improves the result if no model is consistently better that the others.

Hybrid models usually combine several parametric models and, in some cases, were shown to improve the forecast in comparison to the individual models [MHD+13, BM15a, BM15b]. For instance, a combination of ANN and SVR approaches has been explored obtaining better forecasting results than the individual models [JMC14, AHA+14]. At the same time, hybrid models have in common that they are complex to implement as they require to set up numerous sub-models, while the efficacy still has to be evaluated across the building domain rather than on few single buildings.

5.3 Literature Summary

In this section, we summarize the insights from the reviewed literature (Section 5.2) in terms of data requirements, model setup and comparison. We have observed, that it is still an ongoing challenge to develop a widely applicable forecaster for predicting the load across the building domain – i.e., a forecaster that is applicable on various different buildings. Rather than developing a scalable day-ahead model for the entire domain, most attempts focus on predicting intraday energy demand of a specific building.

In the reviews literature, model setup and input selection, usually required *explicit domain knowledge* i.e., the specific building structure had to be explicitly considered in the model. The well-performing model, presented in the results, for that building was often found through manual trial and error process and required large amounts of historical data and computational resources [CBWS16]. While such model can be very accurate on the given building, the forecast can worsen if the same model is applied for another building [FRS+13]. The comparison between the models is often difficult since those are rarely evaluated on a large variety of buildings and subjected to a statistical analysis accounting for the stochastic variability of the results.

5.3.1 Data Inputs of the Forecasting Models

In the publications reviewed for this work (Section 5.2), model inputs were selected from available data that are expected to influence the load curve. They were frequently chosen manually relying on the problem knowledge (e.g. building with an installed PV module is likely to depend on solar irradiation), researcher experience and intuition [MRCA14, HP16, MHD+13] or an in-depth sensitivity analysis [LSPB+12]. Most often, optimal choice was found by training the same model using different inputs and selecting the variant yielding the lowest validation error. For instance, it is the case in [RCC14, POC+17] and all the DNN applications previously described.

It is widely recognized that local loads are autoregressive and underlie annual and weekly seasonalities [AT14]. Indeed, most applications used inputs related to the calendar (weekday, day-type, month) and historical load (previous day, week, historical mean, etc.) [HWVA13]. For instance, a feedforward neural network can use the most recent load curve as an input to account for the autoregression in the time series demonstrated in [BFS+15, RCC14, MHD+13].

Despite a common preconception, researchers are ambiguous about using weather-related inputs such as *outside ambient temperature* or solar irradiation. Some, researchers did consider weather by modeling electrical heating and photovoltaics at the level of larger buildings [POC+17, CSZ+15b]. In fact, an in-depth sensitivity analysis can highlight an existing weather dependency [LSPB+12]. At the same time, other researchers observed that the models that do not use any weather data can be more accurate for disaggregated loads [HGP15, MHD+13, HBA+14]. In [BFS+15], authors test two ANN-models on the same dataset with and without weather-related data. They observed no consistent advantage for either model. Consequently, some researchers assumed that the instantaneous weather changes do not affect the load significantly and considered only the month and the day arguing that the temperature does not change substantially from day to day [HBA+14, HGZA18].

To select the model inputs, some researchers have proposed either frameworks [BFS+15] or automated approaches [SLW16]. Alternatively, others proposed to quantify the influence of each possible input and select only the variables whose influence exceeds a predefined threshold [AKZ10, CSZ+15a].

5.3.2 Setup of the Forecasting Models

Aside from the choice of inputs, forecaster setup includes the choice of a model itself and its hyperparemeters[4] that have a major impact on the resulting accuracy [MRCA14, POC+17]. As we saw, and considering more general review studies on load forecasting [HWVA13, RZ19], the models were usually set up and fine-tuned manually, given problem knowledge, researcher experience and intuition, using heuristics and, most often, though trial and error.

Alternatively, there are rare attempts to determine the model setup automatically formulating an optimization problem of minimizing the validation set forecast error. There have been attempts to do so using either grid search [HBA+14] or an optimizer based on evolutionary algorithm [AKZ10]. Automated approaches require setting up and training numerous models that can be challenging and even unpractical. Given a large space of hyperparameters (grid-) search based methods can become prohibitively time-consuming. At the same time, an optimizer can still be trapped in local minima [RZ19]. Despite the increasing interest in fully automated machine learning it is in a preliminary stage [HKV19]. We are yet to see how it might apply to the building STLF.

Especially parametric models require large amounts of data for model training and setup. For instance, various DNN models: CNN [AMM17], RBM [MNGK16, RNK16], LSTM [MAM16a, KDJ+17] and ESN [SLW16] – all required years of data to set up and train the network. Other models, including ARIMA and SVR, also often required over a year of training data [TB13, MHD+13].

As of today, the load forecasting models rely on explicit a priori knowledge of the building and manual fine-tuning. The propositions mentioned previously were mostly developed for a specific building using additional inputs available for that building. The well-performing model presented in the results was often found through a trial and error process which required large amounts of historical data and computational resources.

5.3.3 Comparison of the Models

Despite numerous propositions of building load forecasters, a direct comparison of the models from different publications is either not possible or often leads to contradictory conclusions [FRS+13]. Usually, authors compare the proposed model to some other models that they arbitrarily choose as a reference. The models are evaluated on a single or a very small number of buildings with no statistical analysis of the forecast errors. Therefore, the conclusions can rarely be extrapolated to further cases. Moreover, the

[4] For instance, an ANN-model has numerous hyperparameters such as the threshold for the input selection, network size, training algorithm, activation function and other type-specific settings defining the model.

datasets are often not published due to privacy, ethical or other concerns. For instance, Sun et al., reviewed 105 studies concluding that only 17% of the models were tested on public datasets [SHF20]. This makes it difficult to evaluate and improve the models proposed in the literature.

When reviewing the works on building load forecasting, overall publication bias becomes conspicuous. Usually, the proposed model is the most accurate among few reference models whose setup is not validated. Given that the proposed model is often found through trial and error with extensive manual fine-tuning, it is unclear whether the same favorable comparison would hold if the same manual effort were applied to set up the reference models.

For instance, consider [SXL18] where the authors proposed a sophisticated DNN-based model. They demonstrated the 20% improvement against an ARIMA-model which they believe to be the "state-of-the-art method". However, in their setup the ARIMA-model did not consider weekly seasonality in any form. While they did not provide any validation of the reference model, we can see that the ARIMA-setup is inadequate because building load considerably depends on the workday calendar [FBP11, PBF11b, PBF11a]. Additionally, the authors only provided the mean error without any analysis of stochastic variability of the forecasts. Unfortunately, publications with similar deficits are common [KC19, KDJ+17, RZ19].

When comparing the models proposed in the literature, we often have to rely on case-based reasoning with contradictory conclusions. Model accuracy can vary notably depending on the precise setup, because buildings are very diverse and various facilities can have very different consumption patterns. While some authors suggest that at low-voltage level simple nonparametric approaches can be very effective [HGP15], the majority of publications advocate the usage of advanced parametric methods based on statistical or machine learning approaches.

For instance, Penya et al., observed that ARIMA-models could achieve accuracy comparable to the machine learning methods. They compared various statistical techniques (ARIMA, MLR etc.) with machine learning methods (ANN and SVR) on several university campus buildings and observed that including workday calendar addresses the major nonlinearities of a nonresidential load. They also argued that none of the existing parametric models is broadly applicable on numerous buildings of different types and sizes, and that statistical techniques might be more suitable for that application because they do not need so much historical data to train the model [FBP11, PBF11b, PBF11a].

Other authors made similar observations [WVHA15], [MHD+13]. Even MLR was shown to be more accurate than SVR on small buildings, but no statistical analysis was provided [HWVA13, ZLY19]. Zeng et al., [ZLY19] compared various data-driven models predicting

the load of six commercial buildings. The models were manually fine-tuned, yet neither validation results nor enough details were provided to assess the adequacy of each model setup. While authors observed MLR to provide the most accurate forecast, there was no statistical analysis of the results.

At the same time, Massana et al., came to the opposing conclusions demonstrating that the MLP and SVR are more accurate on a comparable set of buildings [MPB+15]. In fact, several authors observed that machine learning approaches were more accurate than statistical methods on various commercial facilities [YBDS17]. Cai et al., noted that DNN models were more accurate than SARIMAX predicting the load of three institutional buildings. Yet again, they drew this conclusion only comparing the average error without any statistical analysis [CPR19]. Mynhoff et al., also made similar observations [MMG18].

Overall, it remains unclear when to use which forecasting methodology and why. Recent reviews feature hundreds of propositions [ACGW18,BZN+19,RZ19,ZDZ+22] for building load forecasting, yet no approach was shown superior on a large variety of buildings. A comprehensive evaluation requires to test a model on an extensive and diverse set of buildings. The comparison must go beyond evaluating the average forecast errors (e.g. RMSE) because those underlie considerable stochastic variation among the buildings. Instead, we need to test the difference for statistical significance as it is common in other fields where the results are stochastic in their nature (e.g., biology). Such evaluation, on a publicly available dataset, will make the quantitative comparisons of novel techniques to the existing approaches more effective and improve the usability of data-driven models for building load forecasting. In our study, we apply these insights when considering forecast accuracy measure (Section 7.3) and evaluating various models later in the text.

6 Smart Grids and Buildings

Fluctuating renewable energy generation is a challenge for the power system operators. In the EU, a substantial part of the commissioned generators are PV-modules installed on buildings and connected to the distribution grid [Eur20]. Their intermittent power supply requires the operators to mitigate possible imbalances locally, using facilities located in the area.

Buildings becoming both consumers and producers of electricity, can interact with the grid contributing to its stability. Automation technologies can reinforce the power network and increase its capacity for allocating renewable electricity generators. Advances in information and communication technology allow the development of smart grids.

Definition 6.0.1. *Smart grid* is an electrical network that can automatically monitor energy flows and adjust itself to the changes in energy supply and demand [EU14].

A smart grid manages the energy flow relying on specialized control equipment and demand response.

Definition 6.0.2. *Demand response* relates to the changes in electrical load from the normal consumption pattern in response to an external signal imposed by the power system operator or another governing entity [Jor19].

Demand response can make a smart gird more robust, increase infrastructure efficiency and provide the users with economic benefits (Section 6.1). For instance, variable electricity price can reduce demand peaks and encourage the participants to store the excessive generation. In fact, modern buildings connected to the distribution system can adjust their net electricity demand without compromising on comfort of the users (Section 6.2). In this chapter, we describe these ideas and their applications (Section 6.3).

6.1 Smart-Grid Applications

Electrical networks for unidirectional electricity distribution are evolving towards smart grids that allow bidirectional power and information exchange between utilities and consumers. The principal components of a smart gird are distributed energy generators,

advanced metering and control infrastructure as well as data exchange with the users. These components allow various demand response strategies and other automated control measures on the local level providing following benefits for the grid [LBH+16]:

- reduction of peak load
- increase in transmission capacity
- reduction of stress on the equipment (e.g., sub-stations)

There exist numerous smart grid research and demonstration projects in Europe and across the globe [EU14, YSP+14, AAH+20, DSKDSS15]. The value of a smart grid approach can vary among distribution system operators depending on the region. In some cases, primary objective is to facilitate the integration of fluctuating renewable power generators [SGQ+19, AHU20]. In other cases, it may be peak load management [KC18, BTD+18]. In most cases, the advantages for the end-consumers are usually economic benefits with minimal negative impact on the service [DSDSMS18].

Apart from the equipment, a smart-grid idea requires novel forecasting approaches considering the increasing penetration of the distributed energy generators and demand response [KMS+16b]. Installations of the behind-the-meter generators change the electricity consumption patterns and alter the corresponding net load profiles. For instance, a PV-installation on the roof leads to a notable deviation from the standard load profile, depending on the current solar irradiation in the area. Moreover, energy storage, commonly used in a smart grid to mitigate renewable energy fluctuations, has a changing behavior that is difficult to model and leads to further deviation from common consumption patterns. Herewith, currently used standard load profiles become inadequate for a smart grid where many applications rely on accurate load forecasts [WCHK18].

Overall, smart grids improve the flexibility in energy management and make the power system more efficient. Local demand can be, at least partially, met with local generation improving the efficiency of electricity transmission and distribution [WCHK18]. For this purpose, smart grids require advanced data analysis methods for load forecasting, anomaly detection, decision making and outage management [ZHB18, GDS19, VPLZ19]. Additionally, the equipment of buildings connected to the smart grid becomes important. The facilities can provide their current status and respond to the requests by the grid operator to adjust their power consumption following economic incentives (e.g. variable electricity price) [EPC18]. As a result, smart buildings participating in demand response can help to manage peak demand and possible congestions in a smart grid.

6.2 Smart-Building Applications

Smart buildings can adjust their net electricity consumption and help integrate renewable energy generators into the distribution system. This concept has been introduced to improve stability and reduce the operating costs of the grid [ADDPAL20]. Such facilities, consider demand response requests by the grid and optimize their energy consumption using modern building automation technologies discussed in this section.

There are different definitions of a smart building, but its functionality must include [GDPB+20]:

- automation of building operation
- facilitation of maintenance
- ability to adapt to the needs of the inhabitants
- enabling the users to directly control of their energy consumption
- ability to adjust the power demand following external signals

From the grid perspective, demand response is one of the most important features of a smart building. Facility managers must balance demand response requests by the grid operators with energy needs of the users, maintaining smooth building operation.

Energy equipment of a typical smart building (Figure 6.1) can include:

- PV-generator
- HVAC-systems
- automated lighting and shading
- thermal and electrical storage
- smart metering devices

In such a building, energy equipment is connected to a *building energy management system (BEMS)*[1] that combines software and hardware controlling the technical systems for:

- providing thermal and visual comfort
- improving energy efficiency
- minimizing operating costs of the building

[1] Due to the lack of universal and standard terms, there exist several definitions of energy management system. In the literature, the energy equipment controller is most commonly called building energy management system (BEMS) or building automation and control system (BACS) [ADDPAL20, GDPB+20].

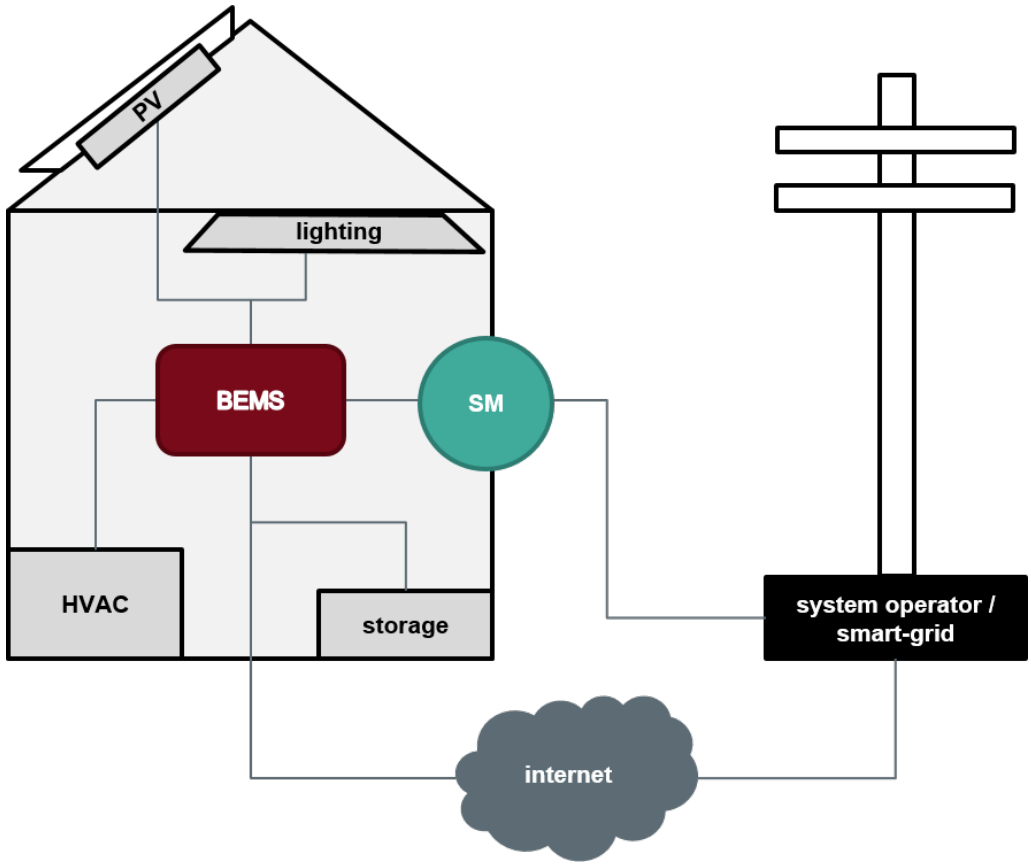

Figure 6.1: Simplified schematic representation of a smart building equipped with various energy equipment: photovoltaic (PV) module, lighting, heating, ventilation and air-conditioning (HVAC), energy storage and other. Building energy management system (BEMS) interconnects the equipment and interfaces with the smart gird. The interface can be either through the two-way communication smart metering (SM) device or over the internet.

- enabling interaction with the smart grid

To reach these objectives, an automated control strategy allows the building to respond to external conditions such as weather and signals from the smart grid operator or the users. At the same time, effective control reduces sizing of the equipment and its operating costs.

Traditional control approaches are inadequate for wide-scale application on smart buildings. *Rule-based methods*, that define the conditions to switch the equipment, have been commonly applied, but lack systematic design methodology [SSM16]. Alternatively, approaches based on the proportional, integral and derivative control are also common for operating the HVAC-systems. Though traditional controllers are easy to implement and inexpensive, they are inapt to consider the external inputs that may come from building or grid operators.

Advanced control strategies address the shortcomings of the traditional controllers. Approaches based on *model predictive control* and *adaptive-predictive control* can consider the uncertainties caused by the external factors using predictions of the exogenous inputs

and dynamic behavior of the building. A sophisticated controller can consider weather, electricity price and load forecasts, users behavior, and other variables to determine optimal control strategy online. Authors of [GDPB+20] reviewed the applications of these strategies. They observed substantial energy and cost savings when compared to the traditional control approaches.

BEMS interfaces with the electric grid providing a demand response capability. In a typical smart building, a PV-module operates as a local electricity generator, while the batteries store excessive energy, optimizing total consumption. Moreover, the system can shift electricity demand in time using high inertia of HVAC processes. Whether over a smart meter or the internet, BEMS can exchange signals with the utility, and adjust its net electricity consumption accordingly [CO17].

Smart buildings interact with the smart grid either directly or through an aggregator. Grid operator can request the buildings in the area to adjust their demand in order to mitigate local power imbalance and limit the stress on the distribution network [LGC+12]. Alternatively, an aggregator can translate global electricity market conditions into specific control signals and incentivize the BEMS to use the available load flexibility accordingly [GKS13]. In this context, variable electricity prices can be effective for balancing supply and demand [LBH+16]. Therefore, integration of the smart grids and buildings is not only a technical problem, but also an economic one that has been addressed in various research projects such as the one described next.

6.3 Smart-City-Demo Aspern Project

Smart-City-Demo Aspern (SCDA) is a research project aiming to carry out a large-scale demonstration of interoperability between smart grid and building domains as well as the interaction with the end-consumers. Supported by the Austrian Energy Fund, the research focuses on the usage of load flexibilities and testing the suitability for everyday usage of the corresponding building technologies [Asp].

The project includes several smart buildings with demand response capabilities, as well as corresponding grid and ICT-infrastructure that were erected in the newly build Aspern district of Vienna. The facilities were constructed using the state-of-the-art building energy equipment such as PV-generators, heat-pumps, solar thermal energy generators, thermal and electrical storage units (Figure 6.2).

Each building participating in the project, has a BEMS optimizing the overall energy consumption using storages. In every facility, the system has to predict the load and the energy generation for which it incorporates the information on user habits, sensor data and weather forecast. Moreover, BEMS enables the interoperability with the smart grid

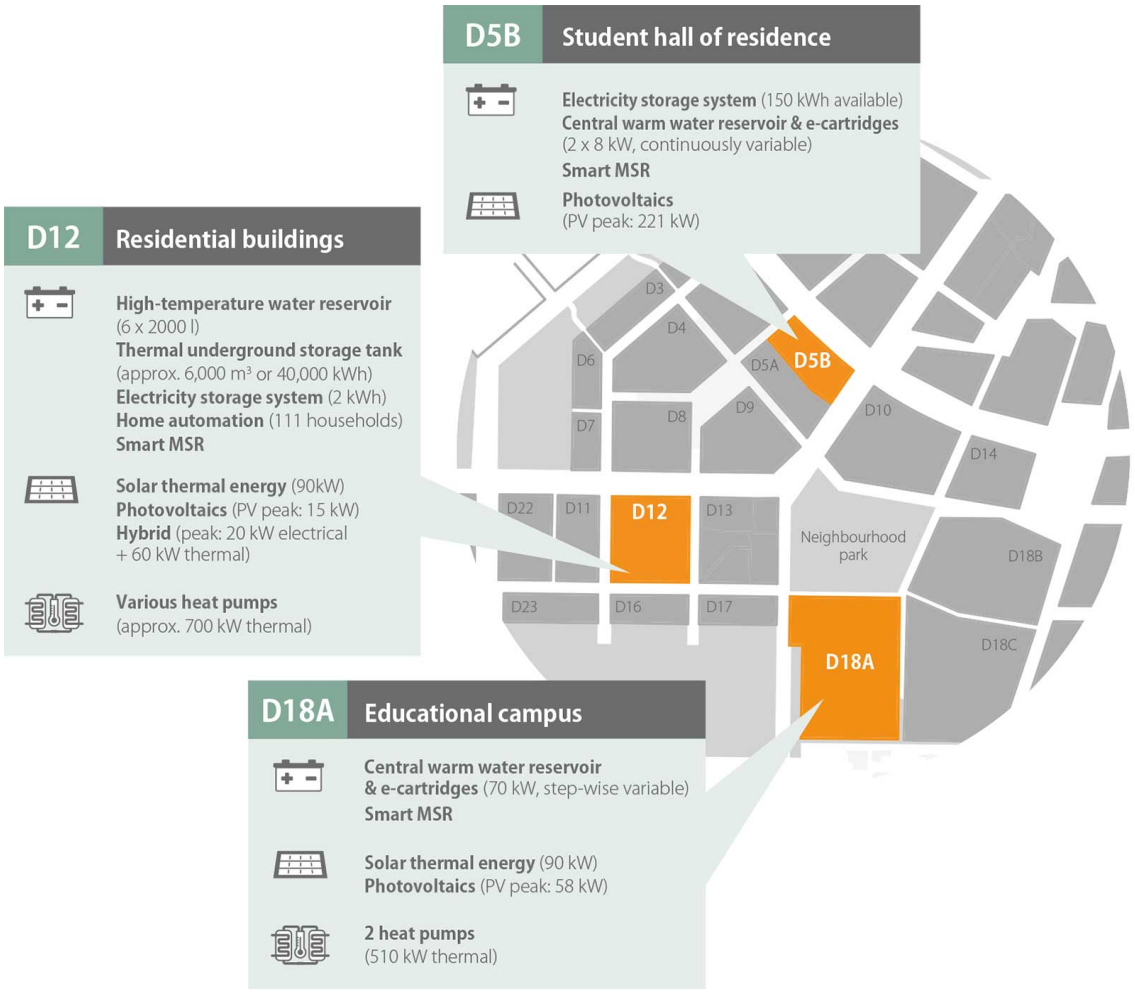

Figure 6.2: Smart buildings and their energy equipment located in the Aspern district of Vienna and that participate in the SCDA project [Asp].

providing demand response capabilities that can either stabilize the local grid or be used in a larger virtual power plant monetizing the available load flexibilities.

In the smart grid domain, the SCDA project uses a district-level network including 500 smart meters, twelve substations and 24 transformers of various types. The research in this domain focuses on potential large-scale roll-out of the smart-grid technology and focuses on: optimal sensor and data-stream configuration, increase of the grid capacity through automation, minimization of investment and operating costs as well as improvement of functionality and safety.

For this purpose, various sensors were installed to monitor the low-voltage grid status. A specialized software was developed to maximize grid utilization capacity and automatically mitigate possible overload by controlling the storages and smart buildings in the area. As a result, the SCDA-project demonstrates an increase in the original grid capacity and provides notable economical benefits for the end-consumers in the building.

Part III

Methodology

In the third part of the thesis, we describe and substantiate the methods that were used to answer the research question. In particular, we discuss how to use the smart-meter data on a wide scale to predict day-ahead electricity consumption of individual buildings and their aggregations. We begin this part by considering the load forecasting problem (Chapter 7). After describing the characteristics of building power demand and its multistep forecasts, we formulate the day-ahead building load forecasting problem and introduce a methodology to evaluate the forecasts in a wide-scale building load forecasting application. Subsequently, we provide a solution to the introduced problem developing a forecaster that is based on a novel *functional neighbor* methodology for predicting day-ahead building load curves (Chapter 8). Finishing the methodological part of the thesis, we describe how we validated the proposed forecaster comparing it to the common reference models from the literature and provide details on the corresponding simulations (Chapter 9).

7 Problem Formulation

This study aims to develop a load forecasting method for a wide-scale application in the building domain. The proposed forecaster has to be applicable on various different buildings disregarding their type, size, or any knowledge of the installed appliances. Moreover, the forecaster can only rely on the data that we expect to have for every building in the distribution system. Given the wide-area installation of smart meters, the forecaster has to predict the load before-the-meter requiring no manual setup or fine-tuning.

Building electricity demand can be represented as a sequence of measurements delivered by a smart meter installed at the connection point to the distribution grid. Assuming that the meter delivers the measurements equidistantly, every Δt minutes, we model the consumption using the notion of a stochastic process (Definition 4.0.11).

$$\mathcal{Y} = \{y_t : t \in \mathbb{R}\}. \tag{7.1}$$

The smart meter delivers a time series $\{y(t)\}_{t\in\mathcal{I}_n}$ where $\mathcal{I}_n = \{1, \ldots, n\}$ is a set of discrete indices corresponding to the measurement time-points. Consequently, we denote a discrete set of n data-points

$$Y_t = \{y(t); t \in (0; (n-1)\Delta t)\}. \tag{7.2}$$

as *load time-series* or simply *load*. Naturally, building load depends on the behavior of the inhabitants but can also be affected by a set of exogenous variables $\mathcal{Z}$ (e.g., weather) which might have an effect on Y_t.

Considering the nature of a wide-scale application on buildings, we formulate the following requirements for the load forecaster:

Requirement 7.0.1. *The forecaster has to be applicable on the largest variety of buildings. Building domain loads are diverse and include consumers of different size and type. The characteristics of the corresponding loads can vary significantly depending on the building.*

Requirement 7.0.2. *The forecaster has to predict the time series that may be neither linear nor stationary. To a varying extent, the load features trends, cycles, and seasonalities,*

depending on the building. Moreover, the statistical properties of the load might abruptly or gradually change in an unpredictable way (i.e., concept change*).*

Requirement 7.0.3. *The forecaster must be able to consider data inputs beyond historical load measurements. Depending on the building, the consumption might depend on exogenous variables (e.g., weather, price, external control signals etc.).*

In this chapter, we describe general characteristics of building loads and provide some examples justifying the requirements listed above (Section 7.1). We formulate the day-ahead forecasting problem (Section 7.2) and define the accuracy measures which we will use to evaluate the models in a wide-scale day-ahead building load forecasting simulation (Section 7.3), before developing the forecaster in the subsequent chapter.

7.1 Building Loads

Generalizing, there are buildings of two main types: residential and nonresidential. Residential buildings can comprise households as predominant end-consumers, while nonresidential buildings include offices, enterprises and community buildings for culture, sports, leisure, education or medical care. Every building can either comprise a single (e.g. household, enterprise) or multiple end-consumers (e.g. apartment/office block). At the same time, a building of a certain size can often have a mixed purpose. For example, a large building in a city can have some floors dedicated for shops and offices while others are strictly residential.

In this section, we discuss the characteristics of building load time-series (Section 7.1.1), highlight the differences among building types (Section 7.1.2) and discuss exogenous variables that can affect the electricity consumption of a building (Section 7.1.3). To exemplify the variety of buildings we use a public smart-meter dataset provided by *Irish Commission for Energy Regulation (ICER)*[1] [Arc16].

7.1.1 Load Measurement Time-Series

Subsequently, we discuss the characteristics of the time series representing the electricity consumption of buildings. Depending on the building, load time-series often feature cycles and various seasonalities. Moreover, building power demand is affected by the

[1] We describe this dataset in detail in Section 9.1.1. In this section, we provide a four-digit ICER dataset smart-meter ID in parenthesis (e.g., Household (1234)) to facilitate the further study for interested readers. This dataset was collected in the Greater Dublin Area, where *households* correspond to the single family *homes*. In this thesis, we use the both terms interchangeably.

nonstationary end-consumer behavior that obstructs the modeling using traditional time series analysis. We study the predictability of different load types considering the auto-correlation of the time series and discuss various exogenous variables that can affect the power demand of a modern building.

7.1.1.1 Annual Cycle

Electricity consumption of a building can follow an annual cycle related to the seasons of the year. Generally, the inhabitants tend to spend more time in the building during the colder months which results in higher average consumption during this period. Annual cycle is particularly notable if a facility is equipped with electrical *heating, ventilation, and air-conditioning (HVAC)* systems. In this case, the dependency on the outside temperature is more pronounced than in a thermally heated building.

In contrast to the transmission system load, the dependency of the daily consumption on the month varies among the buildings. In the electrically heated buildings, we can expect a consumption peak during cold months where the monthly demand can be double as high as in summer (Figure 7.1). Alternatively, buildings with active air-conditioning might have the highest consumption during the warmer months (Figure 7.2).

However, it can be challenging to identify the annual cycle in a building load time-series (Figure 7.3). We can often encounter buildings with no air-conditioning which are heated thermally by a boiler or a district heating network while thermally cooled buildings are also also becoming more common[2]. Moreover, the annual temperature cycle might change depending on the arrival of the cold season which increasingly varies from year to year.

7.1.1.2 Seasonalities

Building electricity consumption features weekly and daily seasonalities. We study the patterns in daily and weekly consumption considering load profiles for each day of the week. For instance, a residential building often has distinguishable morning peaks on each workday (Figure 7.4). The median load curves are similar among each other during the week. During the weekend, the load profile is visibly different and exhibits higher variance while its morning peak is broader.

The distinction between demand patterns during the week and on weekends can be more notable for nonresidential buildings whose consumption is strongly dependent on the business hours. Consider the load of a small enterprise in Figure 7.5. There, the dependency

[2] In the future, district cooling network and passively cooled buildings can often be encountered [GFI18].

is clear and follows the workday calendar. The pattern is more pronounced considering hourly load distribution (Figure 7.6).

Business hours can be different for each enterprise and, thus, cannot be universally determined by the workday calendar of a country where the building is located. To illustrate this, we consider another commercial building whose load we depict in Figure 7.7 and highlight its daily pattern in Figure 7.8. This enterprise is allegedly open every evening except on Tuesday. Such unusual business hours show that, depending on the particular consumer, the load can, to a varying extent depend, on weekday and workday calendar.

Overall, statistical properties such as the mean and variance often depend on the weekday. Therefore, weekly and daily seasonalities are the source of a nonstationarity that has to be addressed by a building load model. The forecaster must be able to predict recurrent daily pattern and seasonal weekly changes in consumption.

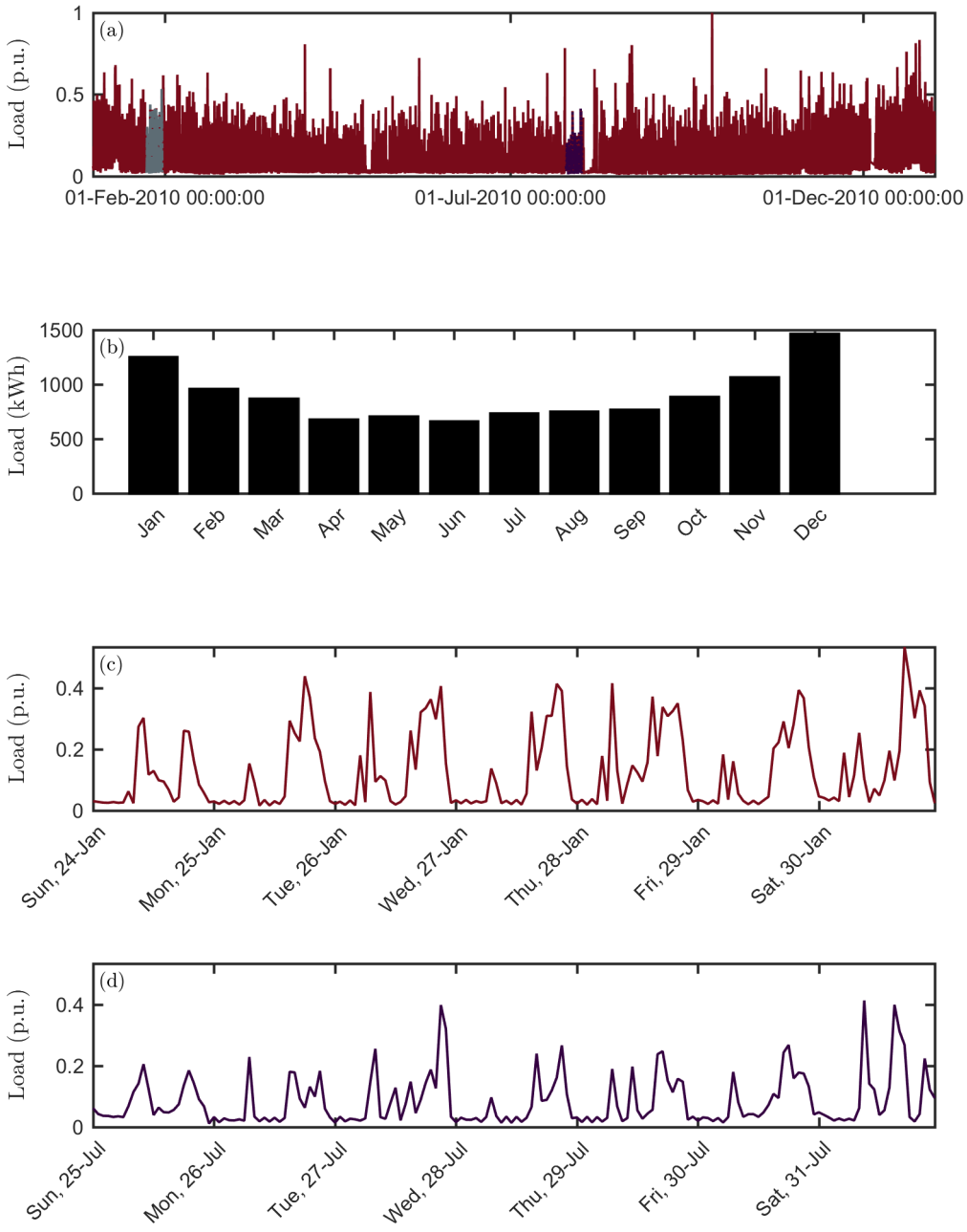

Figure 7.1: Electricity consumption of an electrically heated single family home (Household (1176) from the ICER-dataset [Arc16]). Subplots: (a) load time-series normalized by the maximal value; (b) monthly consumption; (c) load time-series on a selected week in winter; (d) load time-series on a selected week in summer. The power demand in winter is notably higher that in summer. Presumably, the house is heated electrically which increases the load during the colder months.

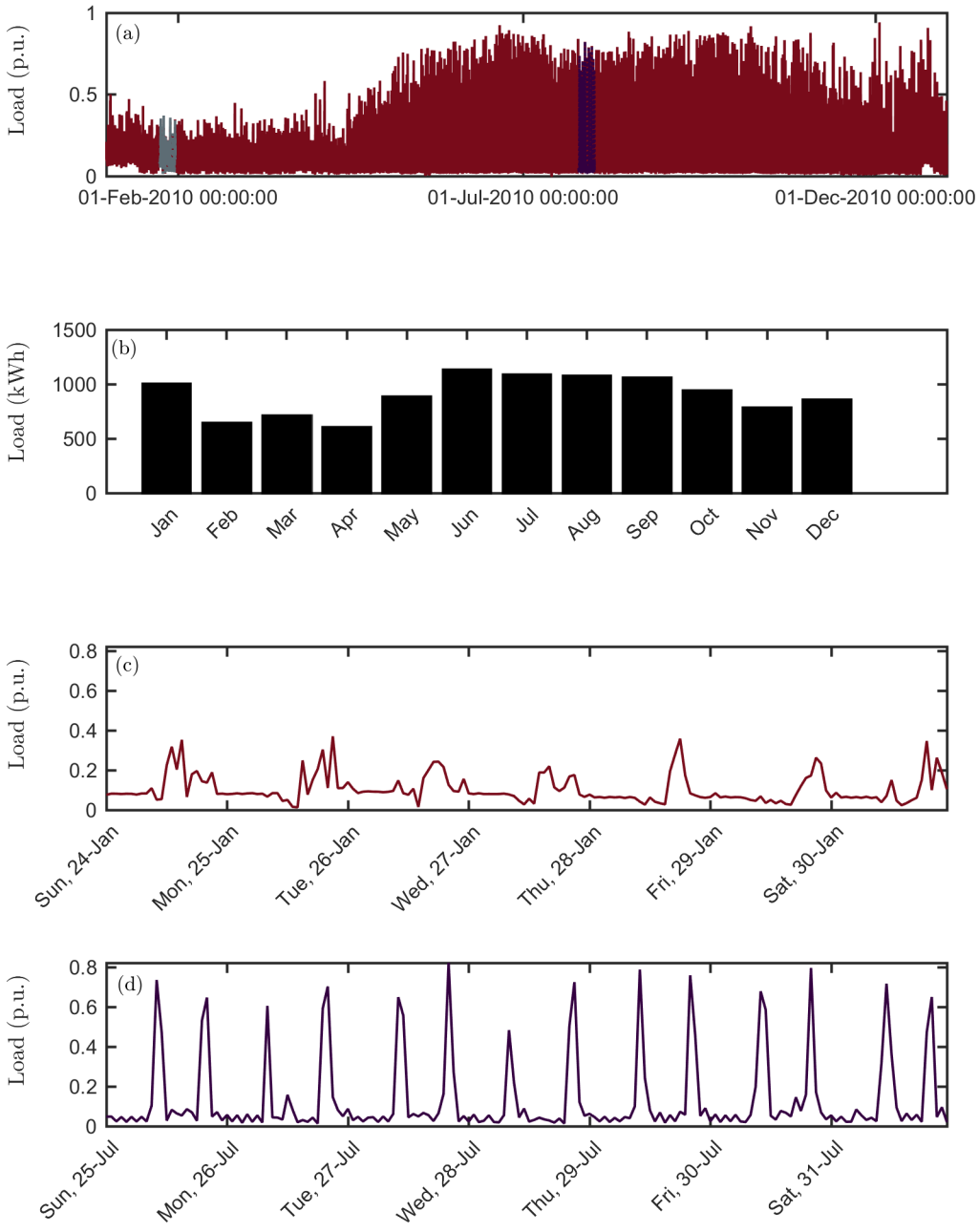

Figure 7.2: Electricity consumption of a single family home with an air-conditioning (Household (1539) from the ICER-dataset [Arc16]). Subplots: (a) load time-series normalized by the maximal value; (b) monthly consumption; (c) load time-series on a selected week in winter; (d) load time-series on a selected week in summer. The power demand in summer is notably higher that in winter. Presumably, the house is cooled actively by an air-conditioning system which increases the load during the warmer months.

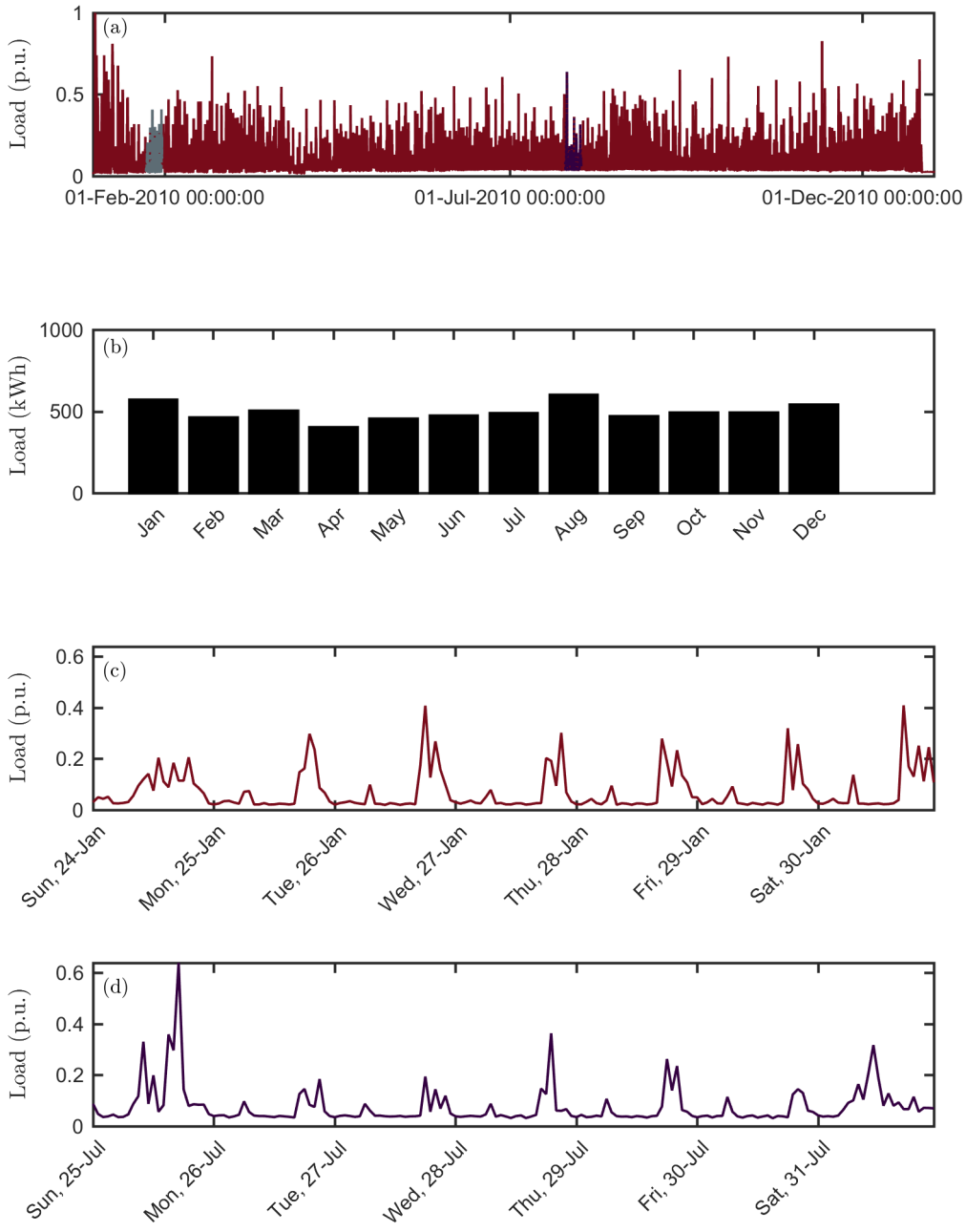

Figure 7.3: Electricity consumption of a single family home (Household (3781) from the ICER-dataset [Arc16]). Subplots: (a) load time-series normalized by the maximal value; (b) monthly consumption; (c) load time-series on a selected week in winter; (d) load time-series on a selected week in summer. There is no clear dependency between the load and the season of the year. The slight demand difference between January and July can be explained by the habits of the users which tend to spend more time indoors during the winter months.

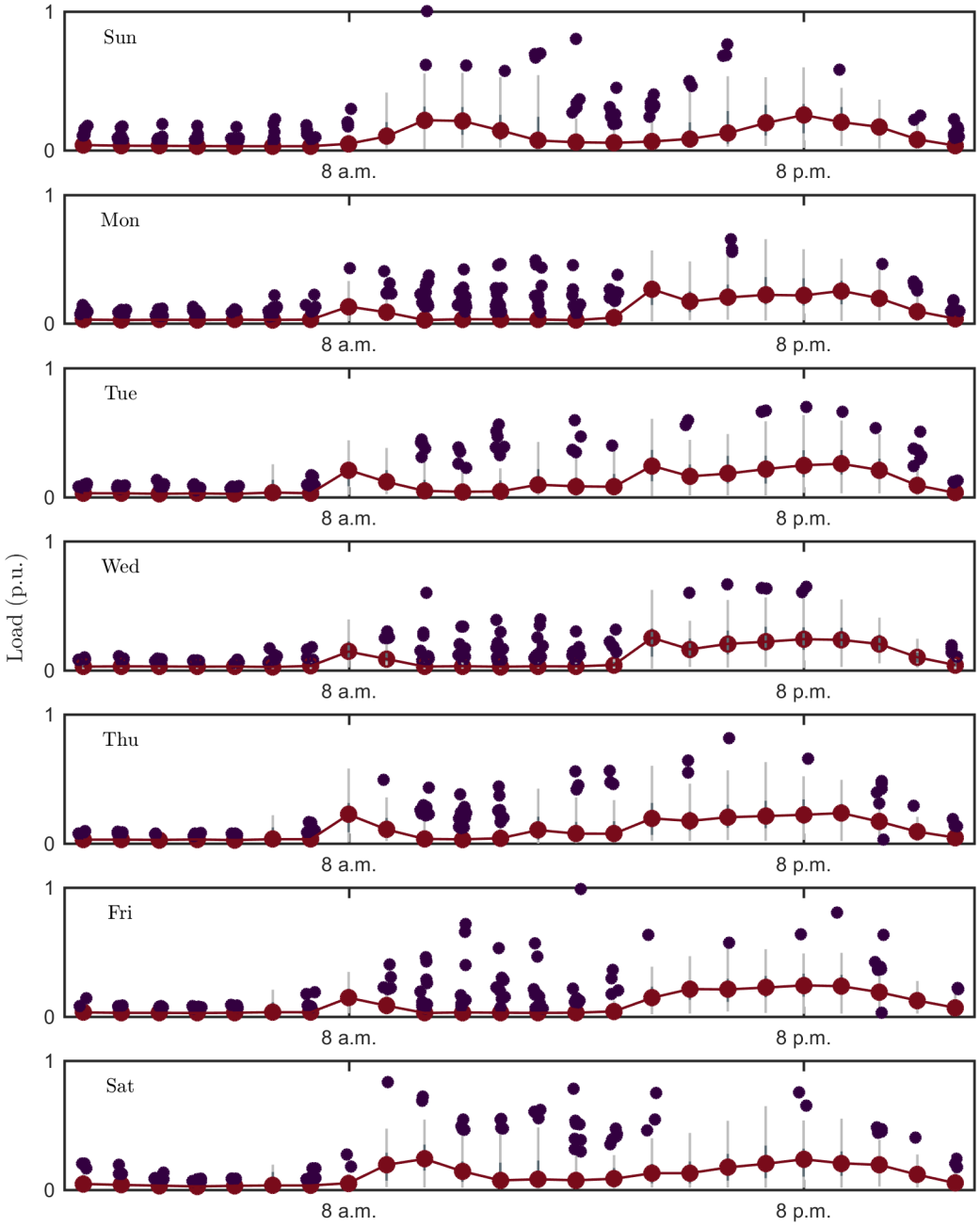

Figure 7.4: Daily load profile and hourly load distribution of a single family home (Household (1176) from the ICER-dataset [Arc16]). For each hour, the distribution of load measurements is represented by a compact box-plot (grey) including outliers (purple). The line interconnects the median values for each hour representing the load profile (red). Each of the seven panels shows the load profile for the corresponding day of the week. From Monday to Friday there is a distinguishable morning peak and the load profiles are similar among each other. During the weekends, the profiles are visibly different, and the load exhibits higher variance while its morning peak is notably broader.

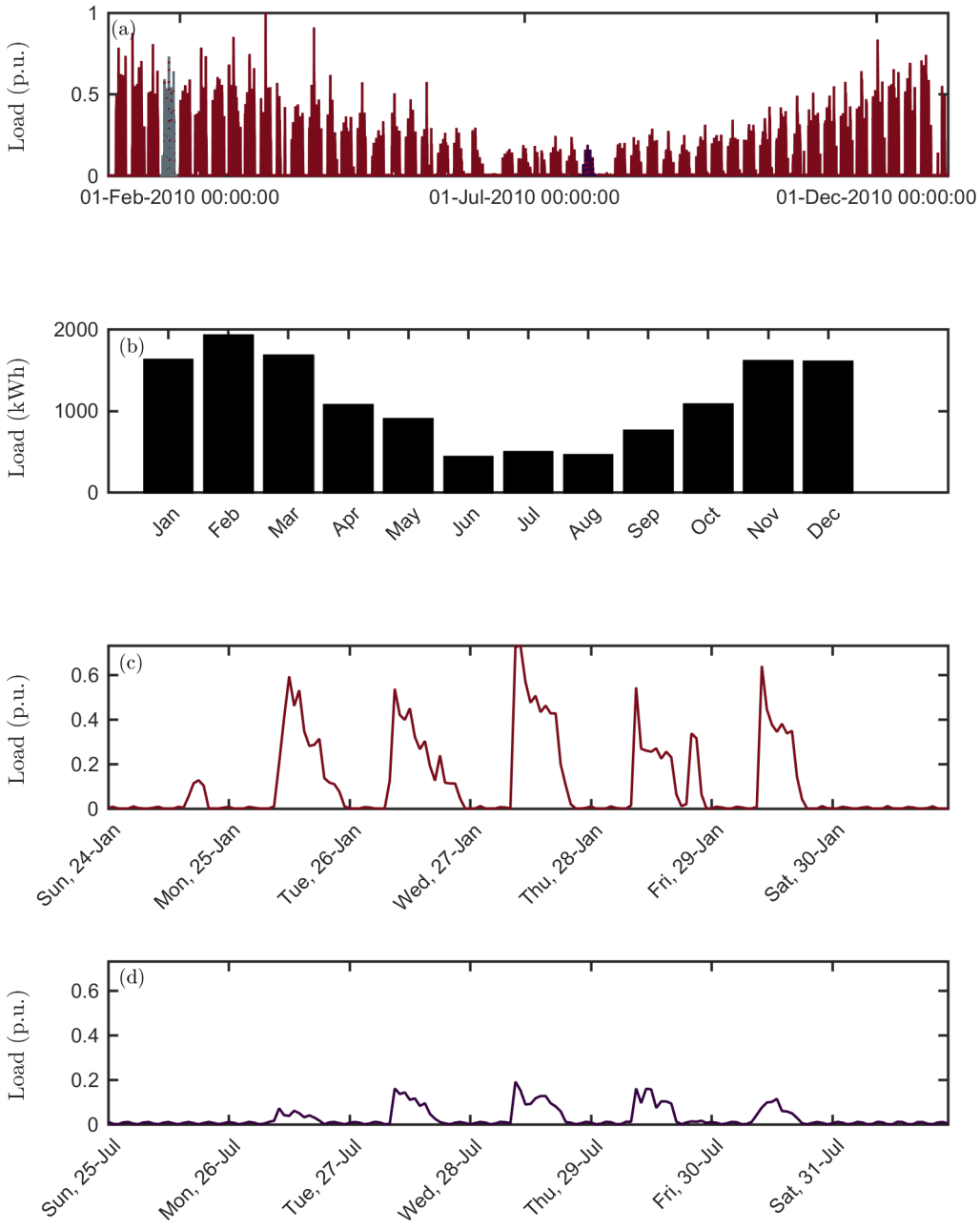

Figure 7.5: Electricity consumption of a commercial building (Enterprise (6520) from the ICER-dataset [Arc16]). Subplots: (a) load time-series normalized by the maximal value; (b) monthly consumption; (c) load time-series on a selected week in winter; (d) load time-series on a selected week in summer. The electricity consumption pattern corresponds to the common business hours following the workday calendar.

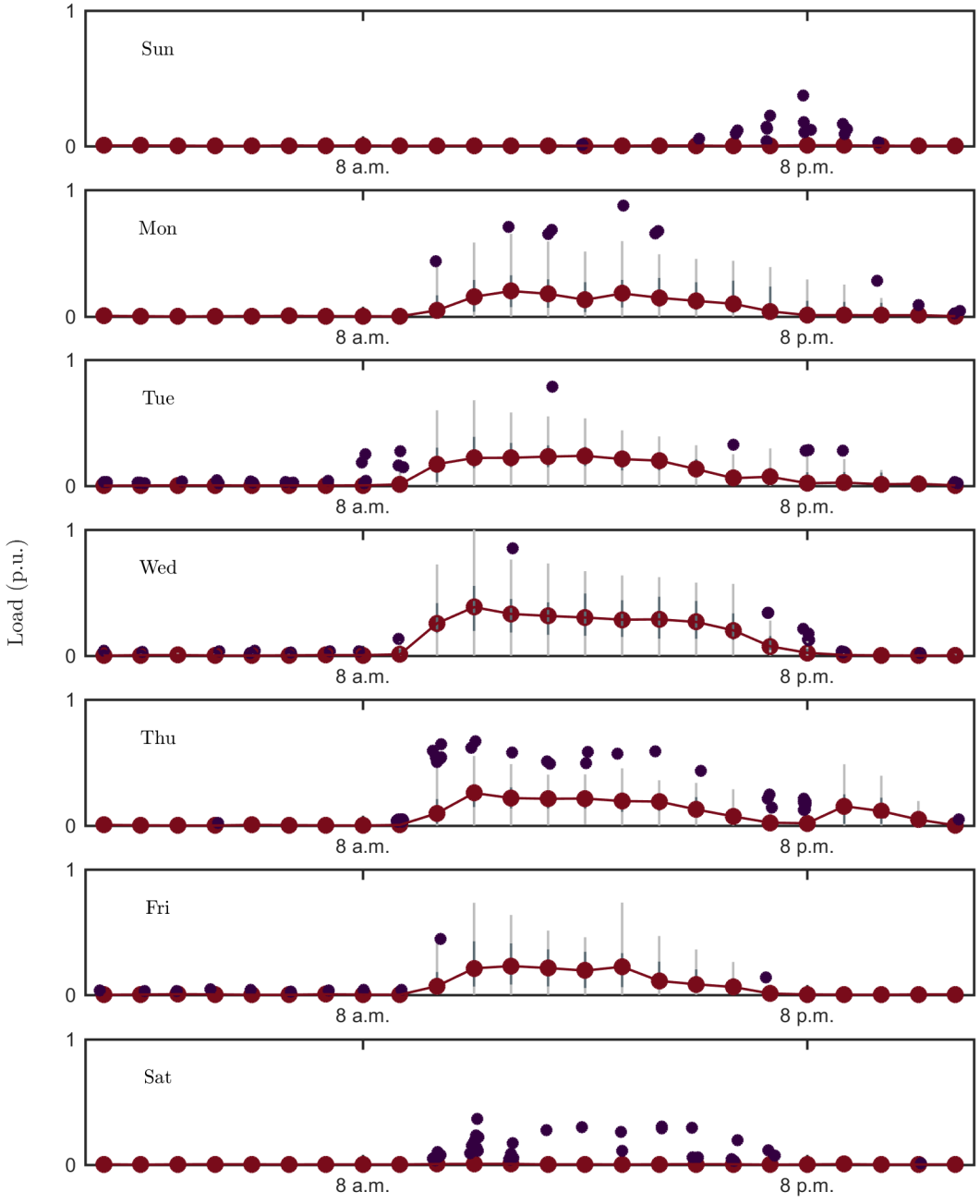

Figure 7.6: Daily load profile and hourly load distribution of a commercial building (Enterprise (6520) from the ICER-dataset [Arc16]). For each hour, the distribution of load measurements is represented by a compact box-plot (grey) including outliers (purple). The line interconnects the median values for each hour representing the load profile (red). Each of the seven panels shows the load profile for the corresponding day of the week. The electricity consumption pattern corresponds to the common business hours following the workday calendar.

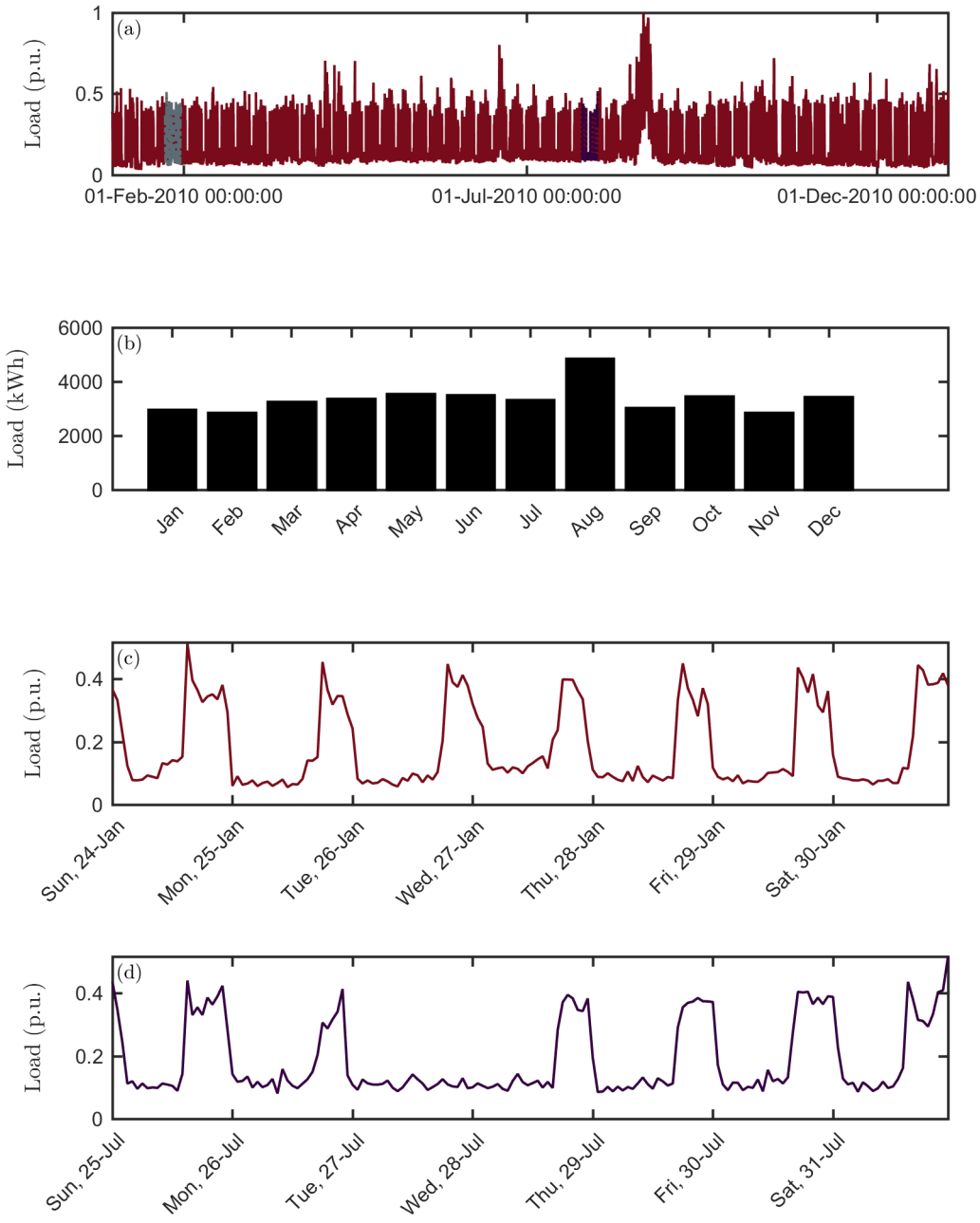

Figure 7.7: Electricity consumption of a commercial building (Enterprise (2916) from the ICER-dataset [Arc16]). Subplots: (a) load time-series normalized by the maximal value; (b) monthly consumption; (c) load time-series on a selected week in winter; (d) load time-series on a selected week in summer. The electricity consumption does not follow the workday calendar. Presumably, this enterprise opens every evening except on Tuesday.

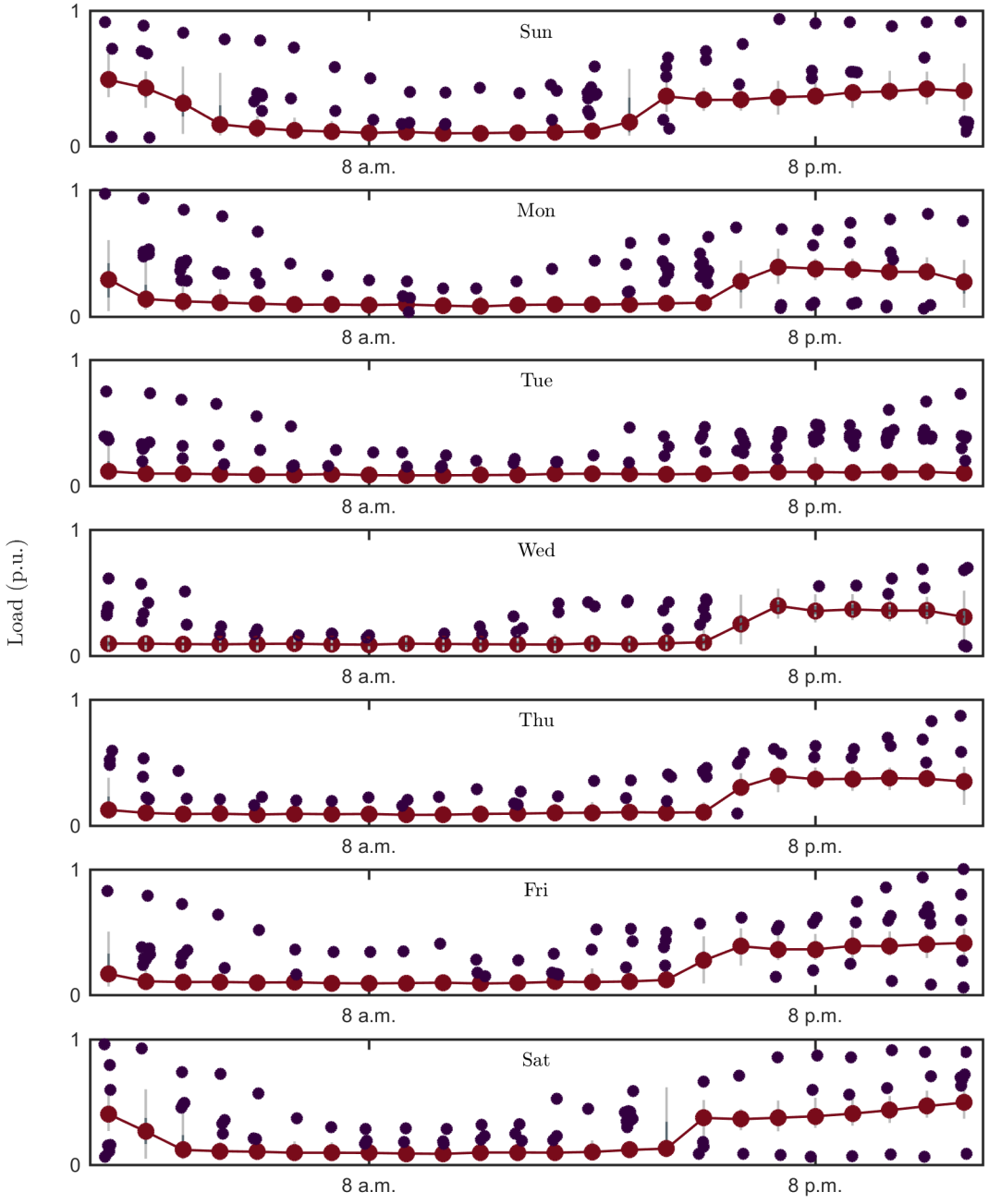

Figure 7.8: Daily load profile and hourly load distribution of a commercial building (Enterprise (2916) from the ICER-dataset [Arc16]). For each hour, the distribution of load measurements is represented by a compact box-plot (grey) including outliers (purple). The line interconnects the median values for each hour representing the load profile (red). Each of the seven panels shows the load profile for the corresponding day of the week. Load profiles indicate that this enterprise has unusual business hours and is closed on Tuesday.

7.1.1.3 Nonstationarity of Load Measurements

Building load measurements are nonstationary and change their statistical properties over time. There is a gradual drift (i.e., *concept drift*) of the average daily consumption following the annual cycle (Section 7.1.1.1). Moreover, mean and variance of the load change depending on weekday and the time-of-day (Section 7.1.1.2). Apart from these changes, we can expect the situations where low-voltage end-consumers abruptly switch their behavioral pattern. In a small building, the few end-consumers can be suddenly absent (e.g., vacation) or the business might temporally close for various reasons (Figure 7.9). Some equipment can operate only during certain periods of time (e.g., storage) or is used only on particular days (e.g., additional heating). New equipment can be installed or removed from the building. Most often, we can only presume why the change in load characteristics happened. Generalizing, a building load can be highly nonstationary, featuring predictable (cycles, seasonalities) and unpredictable changes of the statistical properties. This needs to be considered by an appropriate choice of a model and its setup.

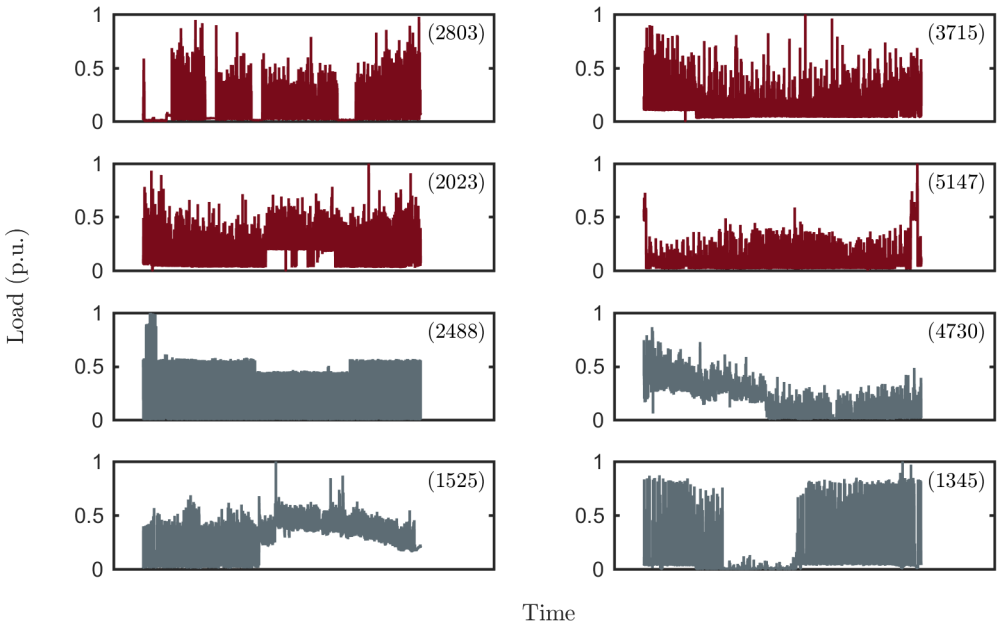

Figure 7.9: Examples of building load nonstationarity. The subplots show electricity consumption of various buildings from the ICER smart-meter dataset [Arc16] that abruptly change their time-series characteristics over the course of a year. For each residential (red) and commercial (grey) building we provide the corresponding smart-meter dataset IDs in parenthesis. We can see examples where the inhabitants of a building are suddenly absent (2803) or the business remains temporary closed (1345). In some examples, we presume that a new piece of equipment is installed or uninstalled, or that an additional electrical HVAC is switched on only on particular days (3715, 2023, 514, 2488). At the same time, the installed equipment (e.g., storage) can operate only during a certain period of the year (2488). Often, we do not know why certain change in the consumption pattern happened (1525, 4730).

7.1.2 Types of Building Loads

There are some general differences between the load of residential and nonresidential buildings. While both types of loads often follow an annual cycle (Section 7.1.1), the electricity consumption of nonresidential buildings shows clearer distinction between working and idle hours where the load is often minimal. In contrast, residential buildings can have similar load profiles across different days of the week disregarding the day-type. Moreover, residential consumption is much more volatile as it highly depends on the end-consumer behavior which makes it harder to predict (Figure 7.1, 7.2 and 7.3).

The predictability of a time series can be studied using the notion of *autocorrelation*. To do so, we compute the correlation (Definition 4.0.10) of a time-series observation y_t with the preceding observation y_{t-p} using the autocorrelation coefficient[3]

$$\gamma_{y,y}(p,t) = \frac{\sum_{t=p+1}^{T} \left(y_t - \mathbb{E}\left[y_t\right]\right)\left(y_{t-p} - \mathbb{E}\left[y_{t-p}\right]\right)}{\sum_{t=1}^{T} \left(y_t - \mathbb{E}\left[y_t\right]\right) \sum_{t=p+1}^{T} \left(y_{t-p} - \mathbb{E}\left[y_{t-p}\right]\right)} \tag{7.3}$$

defined for each time-point t and lag p. Note that the autocorrelation coefficient varies from $\gamma_{y,y}(p) = 1$ indicating ideal positive ($y_t = y_{t-p}$) and $\gamma_{y,y}(p) = -1$ indicating ideal negative relationship ($y_t = -y_{t-p}$). Autocorrelation coefficient represents the strength of a linear relationship and can be represented as an *autocorrelation function (ACF)* on a *correlogram* where $\gamma_{y,y}(p)$ is plotted against the lag p. In a time series with a strong trend, the nearby observations have similar values. This is reflected in larger magnitude of the ACF for smaller lags and its slow decay. Moreover, the ACF reflects the time series seasonalities. In particular, the ACF has notable peaks for the lags corresponding to the multiples of the repeating pattern frequencies.

We compare the correlograms for a residential and a commercial building (Figure 7.10). Observe, that the ACF of a commercial building is notably higher which indicates that the load curve is smoother and more regular. For both buildings, there is a visible increase at the lags corresponding to the multiple of 24 hours and seven days indicating daily and weekly seasonality. The latter is particularly strong for the commercial building.

Generalizing, it is easier to predict the load of a commercial building because it features stronger weekly seasonality and a dependency on the work-day calendar (Figure 7.11). On numerous buildings, we observed that the ACF of commercial loads tends to be higher

[3] The derivation using the probability limit can be found in Chapter 3 of [Ham95]. Note that because of the term $\sum_{t=1}^{T} \left(y_t - \mathbb{E}\left[y_t\right]\right)^2$, autocorrelation coefficient is not defined for constant time series. Further, the lagged time series is not defined for $t < p + 1$. In general, an autocorrelation coefficient depends on time, since nonstationary series change their statistical properties over time such as the expected value or variance. However, this dependency is often disregarded in practice.

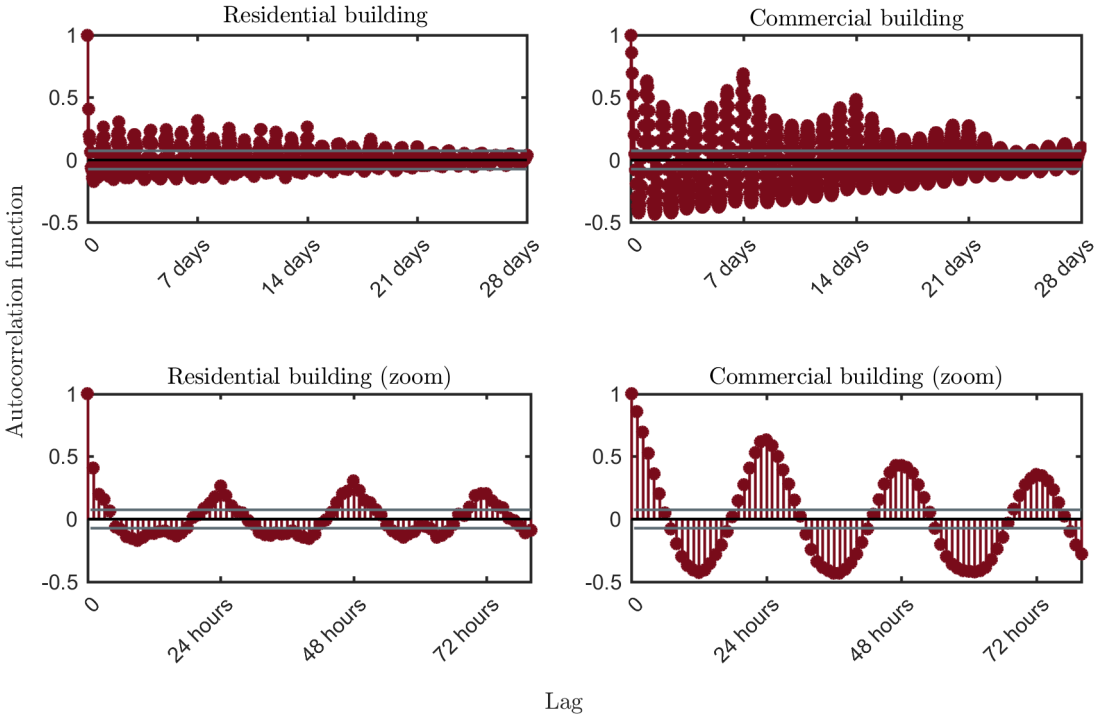

Figure 7.10: Autocorrelation function of a residential (Household (1176)) and a commercial (Enterprise (2916)) building from the ICER smart-meter dataset ICER smart-meter dataset [Arc16]. The panels at the bottom show the enlargement for smaller lags of the corresponding plots at the top. There is a visible increase of the autocorrelation for the lags corresponding to 24 hours, seven days and their multiples. This indicates the presence of a daily and weekly seasonality in the load time-series. The commercial building has notably higher autocorrelation which indicates that its load is more regular (i.e., autocorrelated) and might be easier to predict than the load of the residential building.

compared to the residential loads. The increase of ACF at the lags corresponding to the weekly patterns is also more substantial. Other researchers made similar observations [AT14].

Disregarding the type of a building load, its predictability depends on its size (Figure 7.12). The load curve becomes visibly smoother (Figure 1.1) with the increase of level of aggregation and load size. As a result, the autocorrelation of the load time-series increases (Figure 7.12). In fact, load size is the most important factor for the forecast accuracy, as we will see further in the text.

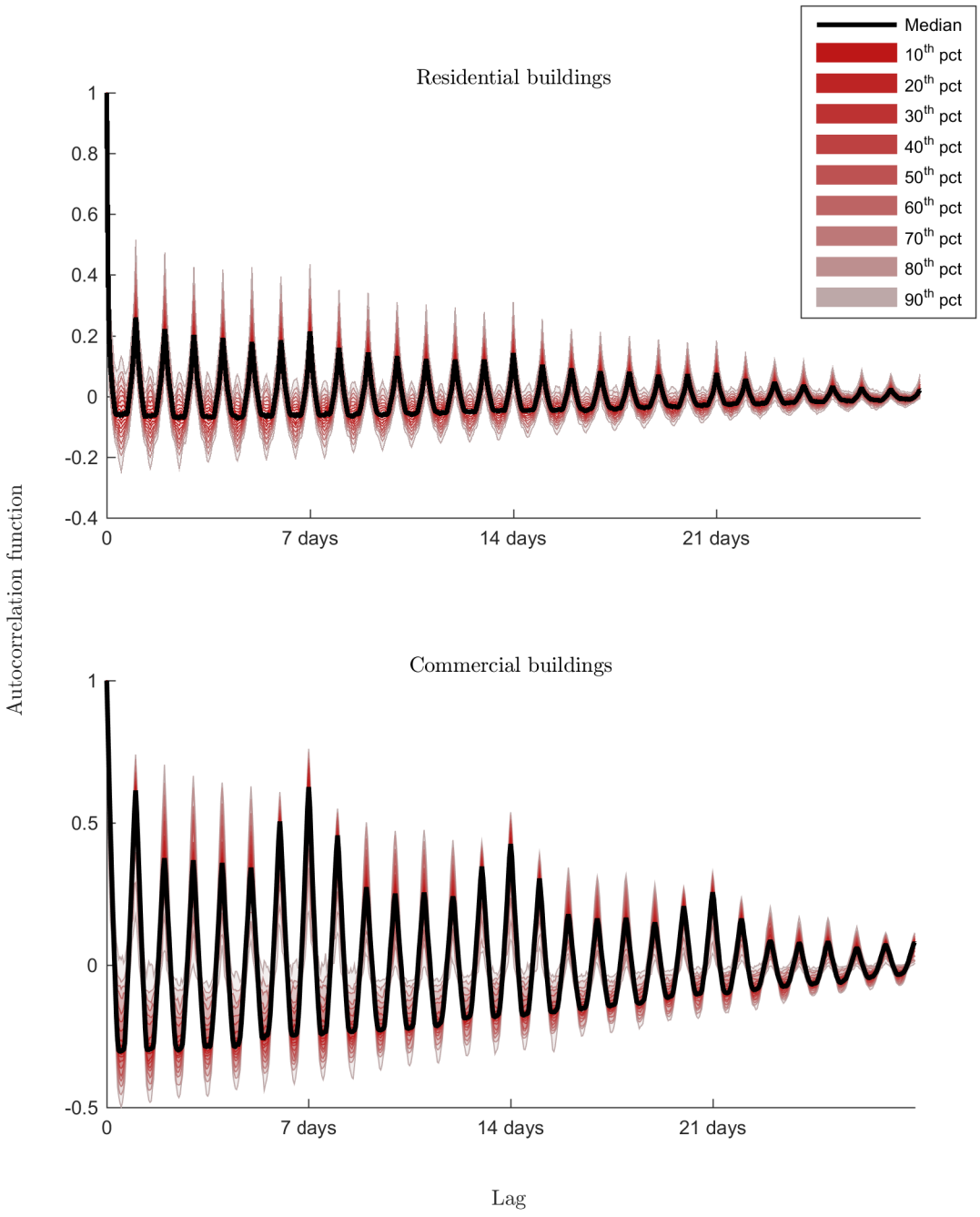

Figure 7.11: Autocorrelation functions of 887 residential (top) and 175 commercial buildings from the ICER smart-meter dataset ICER smart-meter dataset [Arc16]. For each lag, the multitude of the autocorrelation function values is represented with percentiles (pct) and the median. There is a visible increase of the lags that are a multiple of 24 hours. This increase indicates the presence of the daily seasonality in the most loads within the dataset. There is also an increase at the lags that are multiples of seven days that indicates to the weekly seasonality. This increase is more notable for commercial loads that feature strong weekly patterns related to their business hours. Moreover, the autocorrelation of the commercial loads is higher indicating that these loads are, in general, more regular (i.e., autocorrelated) and easier to predict.

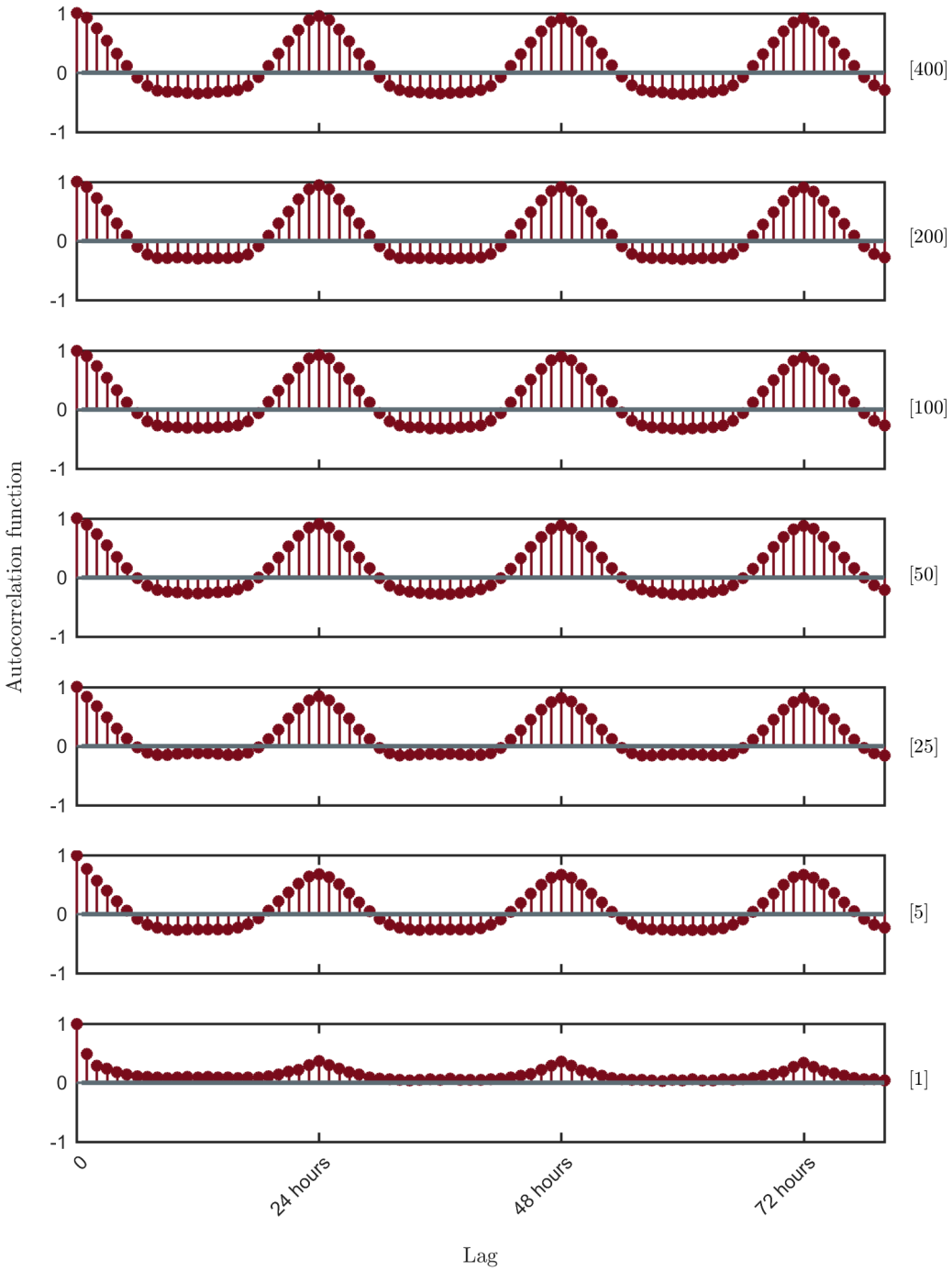

Figure 7.12: Autocorrelation functions of residential aggregations of various sizes. Each building is represented by an aggregation of households from the ICER smart-meter dataset [Arc16]. The magnitude of the autocorrelation function increases with aggregation size indicating the corresponding load curves are smoother (Figure 1.1) and easier to predict.

7.1.3 Exogenous Variables

Electricity consumption of a building is mostly autoregressive but can depend on exogenous variables such as weather[4] or external control signals. The extent of such dependency is case-specific. In general, building power demand depends mainly on human behavior patterns and less on the short-term weather changes while weather becomes more important for larger load aggregations [FBP11, SR18]. However, considering exogenous variables can improve the forecast of a building load in certain cases (Section 5.2). While univariate autoregressive models can be sufficient for the most buildings, for some, a multivariate model considering exogenous variables as external inputs can be more accurate.

For buildings with electrical HVAC-systems, there is an obvious dependency between power consumption and the *outside ambient temperature (OAT)*. The relation between both variables is affected by the magnitude of the temperature [AKS14]. When it is cold outside, there is a negative correlation as more electricity is required for heating. When it is hot, the correlation is positive – the power required for cooling increases with the temperature. Typically, there is also a temperature range where neither heating nor cooling is required and the correlation is non-existent. The degree to which temperature and the load are interrelated depends on the building, its insulation and installed appliances.

Increasingly, buildings become producers of electricity and participate in demand response by interacting with external control signals (Chapter 6). The net consumption of a building with a *photovoltaic (PV)* generator depends on intermittent solar irradiation and can be increasingly volatile. Batteries can mitigate the volatility, yet overall consumption will depend on the amount of insolation. At the same time, a storage can allow the building to adjust its consumption following variable electricity prices or other external signals.

The extent to which exogenous variables affect the load depends on the given building and its energy equipment. The probability to encounter a building with electrical HVAC, PV-generator, or a battery depends on the country. Some countries have high penetration of electrical heating (e.g., France) while in the others the buildings are, predominantly, heated thermally (e.g., Germany) [ÜVCS+15]. Moreover, PV-panels on buildings are more common in certain regions.

Considering exogenous variables as external inputs to a load forecasting model can improve the accuracy in some cases, yet raises several issues when done for numerous different buildings. For instance, weather-related inputs will not be available ex-ante, and require a prediction. An uncertainty of a weather forecast can consume any potential advantage of

[4] While there are various variables describing weather (outside temperature, windspeed, solar irradiation and humidity), outside ambient temperature and solar irradiation are often the most important for individual buildings [BZN+19].

using it as an external input. At the same time, historical weather data required to train the model might not be available for some sites. Despite a common preconception, various researchers observed that the accuracy improvement of a forecasting model using weather can be negligible or even negative when predicting smaller loads (Section 5.3.1). Moreover, when considering external signals, we have to acknowledge that explicit measurements on the appliances that are affected the most by those inputs might not be available.

Therefore, we should minimize the number of the external inputs that are considered by default, when developing a model to be applied on various different buildings. For instance, we can often assume that the instantaneous weather changes do not substantially affect the load of individual buildings[5]. Instead, we can consider the month and day arguing that the temperature does not change considerably from hour to hour [HBA+14, HGZA18]. Ideally, we need a forecasting model that is autoregressive but can optionally consider exogenous variables that might be important in some special cases.

7.2 Wide-Scale Day-Ahead Load Forecasting

Smart grids require wide-scale day-ahead forecasts for effective operation. Currently, loads in the distribution system are predicted using standard load profiles. However, the consumption patterns change following increasing adoption of smart buildings with decentralized renewable energy generators and storages. This development requires novel approaches for wide-scale building load forecasting (Section 7.2.1). Many of the proposed methods were developed for the one-step ahead intraday forecasts. However, for a day-ahead forecast, we need to predict several consecutive values at once (Section 7.2.2). We formulate corresponding forecasting problem mathematically at the end of this section (Section 7.2.3).

7.2.1 Local Load Forecasting in Smart Grids

The notion of a smart grid includes smart buildings that are equipped with renewable energy generators and can adjust their consumption according to external inputs. Local load forecasts at the level of single buildings allow the smart grid to apply demand response locally (Chapter 6). When local demand can be met with local generation, the energy transmission becomes more efficient. As this concept gets commonly applied in modern distribution systems, the operators will require a method for wide-scale local load forecasting. In this context, we introduce the following definitions.

[5] In contrast, transmission-system-level loads are characterized by their seasonality, regularity, and sensitivity to meteorological conditions (Section 5.1).

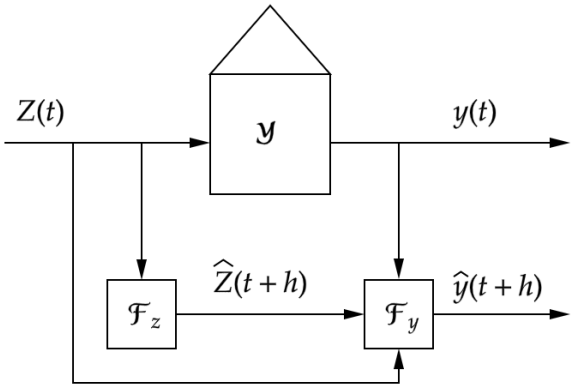

Figure 7.13: Before-the-meter building load forecasting in a wide-scale application (Definition 7.2.1). In this application, we might not have any explicit knowledge about the building $\mathcal{Y}$ or the data from its internal sensors. Forecast $\hat{y}(t+h)$ has to be computed relying on net electricity demand measurements $y(t)$ and the prediction $\hat{Z}(t+h)$ of the exogenous variable $Z(t)$. Further discussion is provided in the text.

Definition 7.2.1. *Wide-scale local load forecast* is a before-the-meter prediction of the net electricity demand on a large and diverse set of local loads connected to the distribution system that is obtained without any explicit domain knowledge.

Definition 7.2.2. *Local load* is a load connected to the distribution system that includes only few end-consumers located in the same geographical area (e.g., buildings).

Definition 7.2.3. *Explicit domain knowledge* in the context of load forecasting denotes explicit knowledge about the predicted load such as its size, type, or installed equipment.

In the forecasting literature, researchers often use explicit domain knowledge to set up and manually fine-tune their building load model (Chapter 5). If we intend to forecast numerous local loads, we might not have any information about each building and its appliances installed behind-the-meter. In our case, we can only use the net demand measurements and commonly available exogenous variables as the inputs for our model[6].

For a wide-scale forecast, each building load has to be predicted *before-the-meter* i.e., without any data from within the building (Figure 7.15). We can represent the building as a stochastic process $\mathcal{Y}$. Given multivariate input[7] $Z(t)$, the building responds with a net electricity consumption $y(t)$ that we predict with horizon h. To do so, the load forecaster $\mathcal{F}_y$ uses historical observations of $y(t), Z(t)$ and the input prediction $\hat{Z}(t+h)$ computed by a separate model $\mathcal{F}_z$ (e.g. weather forecast).

[6] Following an area-wide introduction, smart meters will provide load measurements of each low-voltage consumer. External inputs like calendar information or low resolution weather data are also available on a wide scale.

[7] Inputs may include weather and external control signals as discussed in Section 7.1.3.

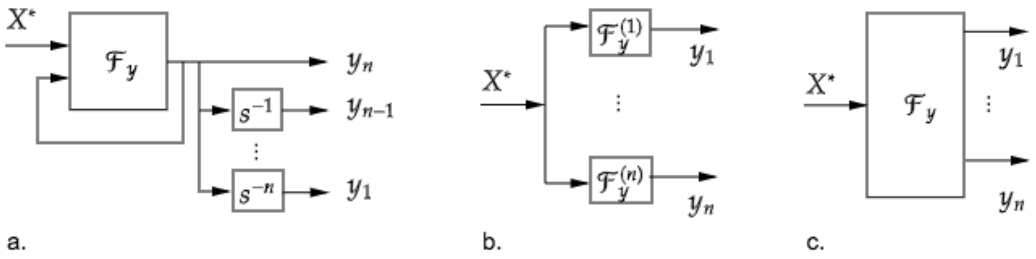

Figure 7.14: Strategies for multistep predictions: (a) recursive, (b) direct, (c) multi-out. Description is provided in the text.

Table 7.1: Summary of multistep strategies and corresponding model architectures predicting n consecutive time steps simultaneously. The models are denoted using following acronyms: single (S), multiple (M), input (I), output(O). Forecast horizon of each individual model is denoted in parentheses.

Input	**Recursive strategy**	**Direct strategy**	**Multi-out strategy**
univariate	SISO(n)	$n \times$ SISO(1)	SIMO(1)
multivariate	MISO(n)	$n \times$ MISO(1)	MIMO(1)

A forecasting model is said to be *process-based*[8] if it is obtained using explicit knowledge about the underlying stochastic process. Alternatively, *data-driven*[9] models can be formulated using only measurements of inputs and outputs. In absence of any explicit information about the load, for a wide-scale load forecasting application, we need a data-driven model that does not require any manual setup or parametrization. Moreover, focusing on the day-ahead prediction of the entire load curve, we have to extend the forecaster to calculate several time steps at once, as we discuss next.

7.2.2 Multistep Prediction

Many of the existing load forecasting methods originate from the research where only *one-step ahead forecast* is required or considered (Chapter 5). Such predictions mostly correspond to the intraday load forecasting. When focusing on the day-ahead forecasting we need to consider *multistep predictions* requiring us to predict n consecutive values at once. To do so, there are several strategies to adjust a one-step ahead forecasting model for a multistep prediction (Figure 7.14)[10].

[8] Process-based models are also called *physical* and are common for building simulations. Under the name *model-based approaches*, they can also be found in other research fields such as econometrics.

[9] Data-driven models are also called *statistical* in some contexts.

[10] See [Tai14, BTBAS12] and references therein for an in-depth discussion on different strategies and comparison of the strategies in terms of performance on different datasets.

Recursive strategy implies forecasting one-step ahead and using the forecast value $\hat{y}_{t+1}$ as the observation with which the forecast is done for $t + 2$ and so on. The fundamental drawback of such strategy is the sensitivity to the forecast error in each step – the error accumulates while advancing the multistep forecast.

Direct strategy requires to set up and train n individual models $\mathcal{F}_y^{(1)}, \ldots, \mathcal{F}_y^{(n)}$ – one for each time step we are predicting. By doing so, we avoid the propagation of the error present with the recursive strategy. However, direct strategy disregards any dependency between the predicted time steps. Moreover, it is computationally expensive, since we have to train n different models. This can become an issue for models with computationally intense training phase such as many parametric models.

Multi-out strategy considers the dependencies between the predicted time steps. It avoids the conditional independence assumption made by the direct strategy, but requires a complex model with a vector output $Y \in \mathbb{R}^n$.

Each of the strategies can be realized through one of the model architectures listed in Table 7.1 and presented in Figure 7.14. Given a univariate time series, a recursive multistep forecaster with a set of internal delay elements results in a SIMO-model. Its prediction can be accurate if we have a good model of the underlying process. The direct strategy requires to set up and train n SISO-models, or, if several inputs are to be considered, n MISO-models. Such approach is the most common for day-ahead load forecasting (Chapter 5). To follow the multi-out strategy, we need to develop and train one MIMO-model with n outputs corresponding to the points of the forecast curve. For example instead of training n ANN-models, we can train a one, more complex network with n outputs.

7.2.3 Day-Ahead Building Load Forecasting Problem

We can now mathematically formulate the day-ahead building load forecasting problem. The before-the meter prediction of building net electricity demand is illustrated in Figure 7.15.

For ease of exposition, and without restricting the generality, we assume that the forecast has to be done shortly before midnight for the entire upcoming day[11]. The smart meter $\mathcal{M}$ provides us with load measurements that we divide into daily load curves

$$Y_j(t) = (y(t); t \in [(j-1)\Delta_s t, j\Delta_s t] \text{ with } j = 1, \ldots, m)\,, \tag{7.4}$$

[11] Such approach is commonly used in the literature on day-ahead load forecasting [CPR19, AVR16, APS06, PS13]. In praxis, the exact time at which we calculate the day-ahead forecast depends on the particular application.

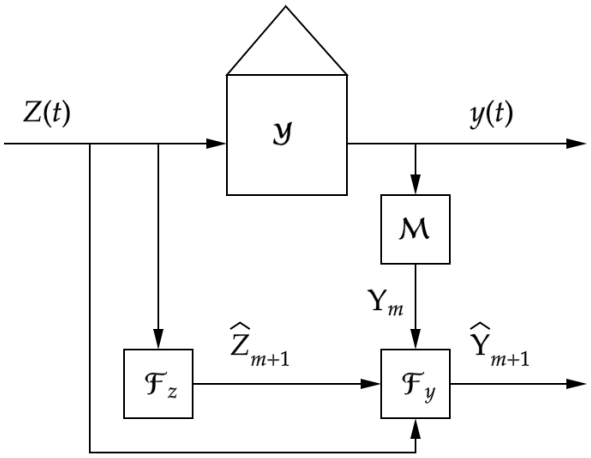

Figure 7.15: Before-the-meter forecast of a building load. At the time t and for a given independent variable z_t (e.g., weather), a building represented by the stochastic process $\mathcal{Y}$ responds with a load y_t. Delays values of z_t and y_t are fed respectively into the forecasters V for external variables and F for the yielding the forecasts $\hat{z}_t$ and $\hat{y}_t$.

where $\Delta_s t = 24$ hours and m is the number of observed days. Herewith, each curve Y_j corresponds to the load measured on the day j, whereby Y_m is the observation of the most recent 24 hours.

Each load curve can be represented by a set of time-discrete measurements

$$Y_j = [Y_j(t_1), \ldots, Y_j(t_n)], \tag{7.5}$$

where $Y_j(t_i)$ with $i = 1, \ldots, n$ denotes the measurement of the daily load curve Y_j at the time t_i where n corresponds to the smart-meter resolution.

Given m daily load curve observations $Y_1, \ldots, Y_m$, corresponding input observations $Z_1, \ldots, Z_m$, and the input forecast $\hat{Z}_{m+1}$ for the upcoming day, we need to find a prediction $\hat{Y}_{m+1}$ of the next-day load curve:

$$Y_{m+1} = [Y_{m+1}(t_1), \ldots, Y_{m+1}(t_n)], \tag{7.6}$$

that will minimize the forecast error according to the measures we discuss in the next section.

7.3 Forecast Evaluation Methodology

In this section, we discuss the forecast evaluation methodology for assessing and comparing predictive models in the context of wide-scale building load forecasting. We begin by describing various traditional error notions to quantify the accuracy of a daily load curve forecast. Additionally, we introduce an error notion that is more adequate for assessing

low-voltage load forecasts than the traditional notions (Section 7.3.1). Afterwards, we discuss how to compare the forecasts of different loads, considering the error scale and stochastic variation (Section 7.3.2). We conclude this section summarizing the insights and formulating the methodology for evaluating the building load forecasts in the context of a wide-scale application (Section 7.3.3).

7.3.1 Daily Error Notion

Assessing forecast accuracy of a volatile and intermittent time series is not a trivial task. In our case, building load can often be negligible or even zero over a considerable period of time[12]. We might also encounter negative load in a case where energy production of a building is larger than its consumption[13]. To evaluate a day-ahead building load forecast, we need a *primary error notion* that can cope with the specifics of volatile low-voltage load time-series and allows us to access the forecast accuracy on a daily basis. Below, we consider several error notions that can be used to evaluate forecast accuracy of a given daily load curve.

7.3.1.1 Point-Wise Error Notions

Traditionally, error notions used to evaluate the forecast accuracy quantify the point-wise deviation between the actual and the forecast time series. Such notions, commonly found in the literature, are based on the concept of a *residual* (error).

Definition 7.3.1. *Residual*

$$\epsilon_i = Y(t_i) - \hat{Y}(t_i) \tag{7.7}$$

is the deviation of the forecast value $\hat{Y}(t_i)$ from the actual value $Y(t_i)$ at the time point t_i.

Studying the *residual time series*, $\epsilon = [\epsilon_1, \ldots, \epsilon_n]$, we can draw preliminary conclusions about the goodness of a forecast and its possible bias. Moreover, we can define a *daily error notion* that quantifies the forecast accuracy on a given day. Subsequently, we present the most common traditional error notions for doing so.

Definition 7.3.2. *Mean average percentage error (MAPE)* is defined as:

$$\text{MAPE} = 100\% \cdot \frac{1}{n} \sum_{i=1}^{n} \left| \frac{\epsilon_i}{y_i} \right|. \tag{7.8}$$

[12] For example, a small enterprise can be closed for a vacation.

[13] For example, a single family home with a large PV-installation might sometimes have negative net electricity demand.

This error notion allows a scale-free comparison between the forecasts and is one of the most widely used error notions found in the load forecasting literature (Chapter 5).

However, for several reasons, MAPE is inadequate to measure forecast accuracy on low-voltage loads. In a situation where y_i can be very small or even zero[14], MAPE can grow to infinity which can substantially distort the result. Moreover, this error notion is not symmetrical since it has a bias favoring the underestimates of an actual value[15]. Therefore, MAPE that is commonly used in the forecasting field can be a poor accuracy measure for low-voltage load forecasts [Arm85, HGZA18]. Alternatively, there are various symmetrical error notions that are also common.

Definition 7.3.3. *Mean average error (MAE)* is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\epsilon_i|. \tag{7.9}$$

The MAE measures the average magnitude of absolute differences between actual observations and their predictions with all residuals ϵ_i having an equal weight.

Definition 7.3.4. *Root mean square error (RMSE)* is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\epsilon_i^2}. \tag{7.10}$$

The RMSE measures the average magnitude of squared residuals.

Both, MAE and RMSE are absolute and negatively oriented[16] notions measured in units of the forecast variable. Both error notions are based on the ℓ^{p}-norm and are linked in terms of the bounds derived with the Cauchy-Schwarz inequality:

$$\text{MAE} \leq \text{RMSE} \leq \sqrt{n} * \text{MAE}. \tag{7.11}$$

However, there is also an important difference between the two notions. In case of MAE, all residuals ϵ_i have the same weight when computing the average. In case of RMSE, the errors ϵ_i are squared before averaging, which increases the emphasis on larger residuals. On smaller loads where such residuals are often, there can be a notable deviation between both notions (Figure 7.16). On larger loads, both errors are strongly correlated.

[14] Consider previously discussed weekly patterns of individual buildings.

[15] Consider an example where with $y = 150$ and $\hat{y} = 100$ we obtain MAPE = 33% while for $y = 100$ and $\hat{y} = 150$ we obtain MAPE = 50%.

[16] Negatively oriented means that a more accurate forecast corresponds to smaller MAE or RMSE.

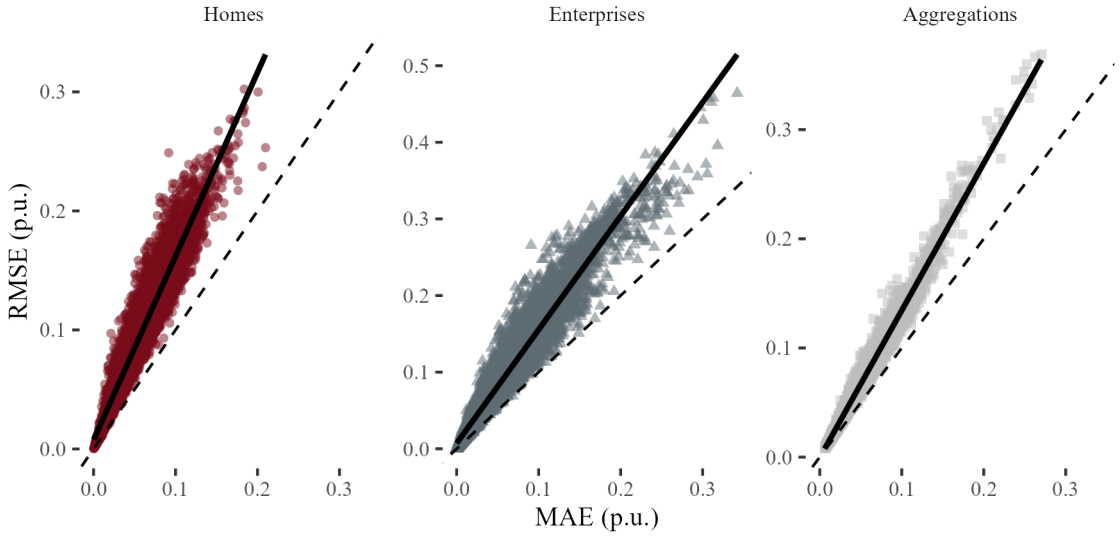

Figure 7.16: Comparison of the MAE and RMSE notions. We applied a naive model (Section 9.2.1.2) predicting the 300 loads in the validation dataset (Section 9.1.1.3) day-by-day for 100 consecutive days. For each load, we evaluated the daily forecast accuracy computing the MAE (7.9) and RMSE (7.10). In the figure, each panel shows the daily forecast errors obtained on individual loads of the corresponding type. Additionally, we denoted the linear regression line (solid) and the line representing the ideal correlation (dashed). We observed that both error notions are strongly correlated. However, the RMSE emphasizes larger residuals which leads to a notable deviation from the MAE on smaller loads (homes, enterprises) where such residuals occur more often and larger forecast errors are to be expected.

In building load forecasting literature, RMSE appears to be more prominent. While MAE tends to be less sensitive to the outliers [Arm01], RMSE is usually preferred, especially when large residuals are particularly undesirable. This is often the case for the low-voltage load forecasting where missing a large load spike can theoretically result in a damage of the equipment [HSG16]. Moreover, RMSE plays an important role within the theory of statistical learning. In many fields where optimization is applied, the cost function is often defined as the mean of squared errors and many theoretical results were derived under such definition [GZ05, CD14, Vap13, FHT08].

7.3.1.2 Permutation-Adjusted Error Notion

Point-wise error notions discussed above can lead to erroneous conclusions when evaluating forecast accuracy on volatile and noisy time series such as low-voltage power demand. Consider a situation where a model had accurately forecast a load peak in terms of size and amplitude, but the prediction was slightly displaced in time, relative to the actual peak. If we were to use a point-wise error notion to evaluate such a forecast, it would penalize the deviation twice for:

1. missing the actual peak

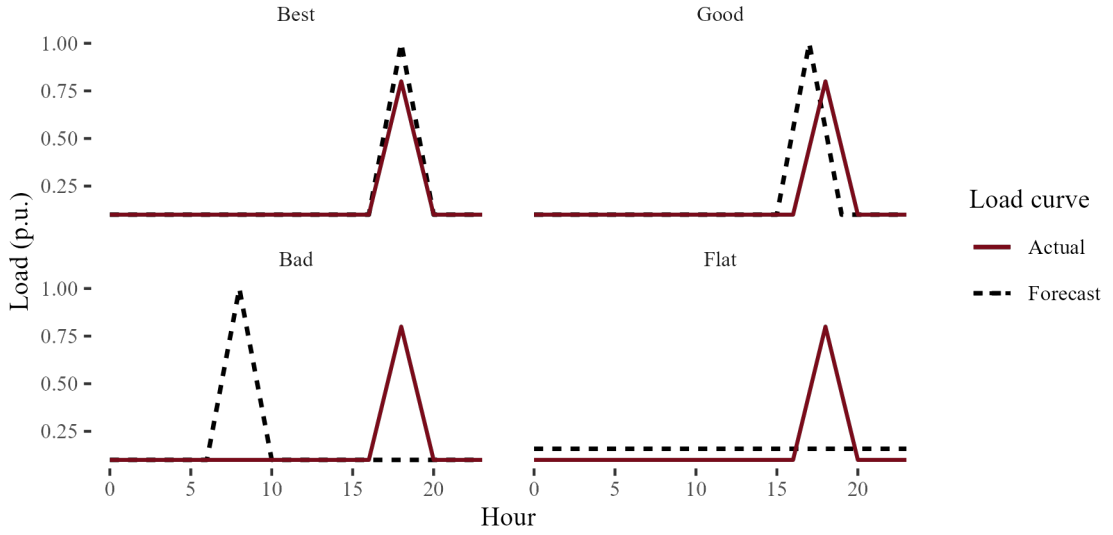

Figure 7.17: Example motivating the usage of a permutation-adjusted error notion. In the figure, each panel shows a different forecast (black) of the same illustrative load curve (red). Four exemplary forecasts – $\hat{Y}_1$ (best), $\hat{Y}_2$ (bad), $\hat{Y}_3$ (good), $\hat{Y}_4$ (flat) – were evaluated using different error notions with the results summarized in Table 7.2. Further discussion is provided in the text.

Table 7.2: Forecast accuracy in the illustrative example (Figure 7.17). Four different forecasts – $\hat{Y}_1$ (best), $\hat{Y}_2$ (bad), $\hat{Y}_3$ (good), $\hat{Y}_4$ (flat) – were evaluated with RMSE (7.10) and PRMSE (7.12) allowing permutations of various range u.

Error notion	Permutation range	$\hat{Y}_1$	$\hat{Y}_2$	$\hat{Y}_3$	$\hat{Y}_4$
RMSE	0	0.24	1.4	0.83	0.81
PRMSE	1	0.24	1.4	0.61	0.81
PRMSE	2	0.24	1.4	0.61	0.81
PRMSE	3	0.24	1.4	0.24	0.81

2. forecasting the peak at the wrong point in time.

Under these circumstances, it can be difficult for a plausible prediction to outperform even a flat forecast that is of little informative value. We demonstrate this on an example adopted from [HWVG+14].

Figure 7.17 depicts an illustrative load curve Y with four different forecasts $\hat{Y}_1, \hat{Y}_2, \hat{Y}_3, \hat{Y}_4$. In this example, $\hat{Y}_1$ and $\hat{Y}_2$ are the best and the worst forecasts respectively. Additionally, for the most use cases, we regard $\hat{Y}_3$ as superior to a simple flat forecast $\hat{Y}_4$. Note that RMSE is based on the ℓ^2-norm (Euclidean distance) quantifying the point-wise distance between the vectors (7.10). Therefore, the flat forecast $\hat{Y}_4$ has lower RMSE than $\hat{Y}_3$ (Table 7.2).

While time series can be represented as vectors, in context of forecast accuracy we are interested in similarity of a forecast to the actual time series, rather than in a point-wise

distance. As an alternative to a point-wise distance, Haben et al., propose a concept of the *permuted ℓ^{p}-semimetric*[17] which quantifies such similarity rather than a point-wise distance [HWVG+14]. We use this semimetric to define the following error notion.

Definition 7.3.5. *Permuted root mean squared error* (PRMSE) is defined as:

$$\mathrm{PRMSE}(Y,\hat{Y}) := \sqrt{\frac{1}{n}\left(\min_{\pi\in\mathcal{P}(u,n)}\sum_{i=1}^{n}|Y(t_i)-\pi(\hat{Y})(t_i)|^2\right)}, \tag{7.12}$$

where $\mathcal{P}(u,n)$ is the set of all u-local permutations of n-points. A u-local permutation rearranges the time series by moving each point forwards or backwards by up to u time-units[18].

Allowing permutations, we get a more adequate evaluation of the curve similarity. Over the course of this study we allow *permutations up to one hour* (independent of granularity). For our illustrative example, the differences between RMSE and PRMSE can be seen in Table 7.2. In practice, the difference between a permuted and a traditional error notion, such as MAE and RMSE, only becomes notable on volatile loads such as single family homes and small enterprises (Figure 7.18).

With the exception of MAPE, we observed strong correlation among all error notions. Hence, as also noted by other researchers, it can be superfluous to present the results in terms of several daily error notions [HGZA18]. Though both MAPE and RMSE are ubiquitous in the literature, both can be inapt for evaluating forecast accuracy on small loads. Therefore, unless stated differently, we will use PRMSE (7.12) as the primary error notion for quantifying daily forecast error.

7.3.2 Forecast Comparison

We need to extend the forecast evaluation methodology beyond the daily error notion, in order to compare different predictive models applied on various buildings. To compare the forecasts computed for different loads, we need an error notion that is scale-independent. Moreover, forecast errors underlie a considerable stochastic variation which requires to consider their distribution and use appropriate descriptive and inferential statistics in order to draw any general conclusions about the relative accuracy of different models. In this section, we discuss how to compare the forecasting models that are to be applied to the entire building domain rather than to an individual building load.

[17] We will discuss the permuted ℓ^{p}-semimetric in detail in Section 8.2.2.3.
[18] Note that with $u = 0$, the equation (7.12) corresponds to RMSE.

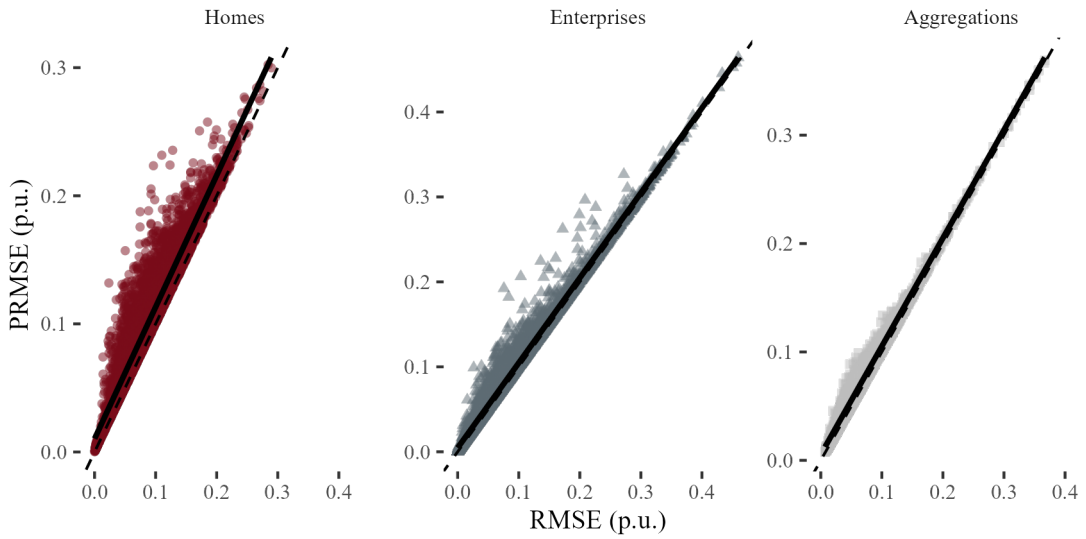

Figure 7.18: Comparison of the RMSE and PRMSE notions. We applied a naive model (Section 9.2.1.2) predicting the 300 loads in the validation dataset (Section 9.1.1.3) day-by-day for 100 consecutive days. For each load, we evaluated the daily forecast accuracy computing the RMSE (7.10) and PRMSE (7.12). In the figure, each panel shows the daily forecast errors obtained on individual loads of the corresponding type. Additionally, we denoted the linear regression line (solid) and the line representing the ideal correlation (dashed). We observed that both error notions are notably correlated. However, the difference between permuted (PRMSE) and traditional (RMSE) error notion becomes notable on volatile loads such as homes and small enterprises.

7.3.2.1 Scale-Independent Forecast Error

We need a scale-independent accuracy measure in order to evaluate and compare the forecasts across the building domain. The absolute prediction error depends on the magnitude of the time series and one of the main advantages of the ubiquitous MAPE-notion is its scale-independence. In our context of low-voltage load forecasts, we quantify the error E using the PRMSE (7.12) which we also make scale-independent by expressing the forecast error in terms of the *error coefficient of variation (ECV)*

$$\mathrm{ECV} = \frac{E}{\bar{y}}. \tag{7.13}$$

Herewith, we scale the daily error E by the average load $\bar{y}$ and allow a scale-free comparison of the forecast accuracy. Since $\bar{y} > 0$, combining PRMSE with ECV provides a notion that is less intermittent than MAPE. Though not as ubiquitous as MAPE, expressing the error in terms of ECV is more adequate for comparing the forecasts of low-voltage loads and was sometimes suggested in the literature [HGZA18].

Disregarding the notion, daily errors underlie a considerable stochastic variation as we will see further in the text. This can reduce the sensitivity of an error notion, obstructing parametrization and comparison between the models. For such case, we can define *relative*

error notions. Hyndman et al., advocate the usage of such notions, suggesting to divide the primary error E by the error of a benchmark model E_b [HK06]. For our study, we define the *improvement* as

$$\mathrm{R} = (1 - \frac{E}{E_\mathrm{b}}) \cdot 100\%, \tag{7.14}$$

which quantifies the daily error reduction in % relative to a benchmark.

Researchers often use the naive model as the benchmark. However, in some situations, this model can achieve very small errors (Figure 7.19) which obstructs an objective overall evaluation[19]. Instead, for the evaluation (Chapter 10) we use standard load profiles forecast as such benchmark since this approach has more consistent accuracy than the naive model. For the model parametrization (Chapter 8), the basic setup can be chosen as the benchmark.

When comparing several forecasts, the difference in terms of a daily error or improvement, might not appear statistically significant at first. We encountered such situations when assessing the effect of a single model-parameter on forecast accuracy as we do further in the text. For such case, we can count the days where the model had the same or smaller error than the benchmark [HGZA18]. Such situations occur when there is a large variation among daily accuracy observations as we discuss subsequently.

7.3.2.2 Statistical Variation of Errors

Load curves vary substantially depending on the day and building. Consequently, forecast errors underlie a notable stochastic variation depending on the predicted day and load. Hence, we must consider statistical variation and error distribution when evaluating forecast accuracy.

For a given load, we cannot expect daily errors to follow the commonly assumed normal distribution. Forecast errors cannot be negative, hence, their distribution is not symmetrical. Further, there might be days of a sudden concept change (e.g., inhabitants leave on holiday) where the forecast accuracy will drop. If anything, the approximate daily error distribution is log-normal rather than normal (Figure 7.19).

Under this approximation, we can expect a considerable number of positive outliers and right-skew of the daily error distribution. For such non-symmetrical and skewed distribution, summarizing the errors using the mean and variance can be misleading, despite the common convention. Instead, daily errors should be reported in terms of a median and the corresponding *interquartile range (IQR)*.

[19] Naive model (Section 9.2.1.2) can sometimes have very small errors. For instance, when the electricity consumption a building is very low for a prolonged period of time.

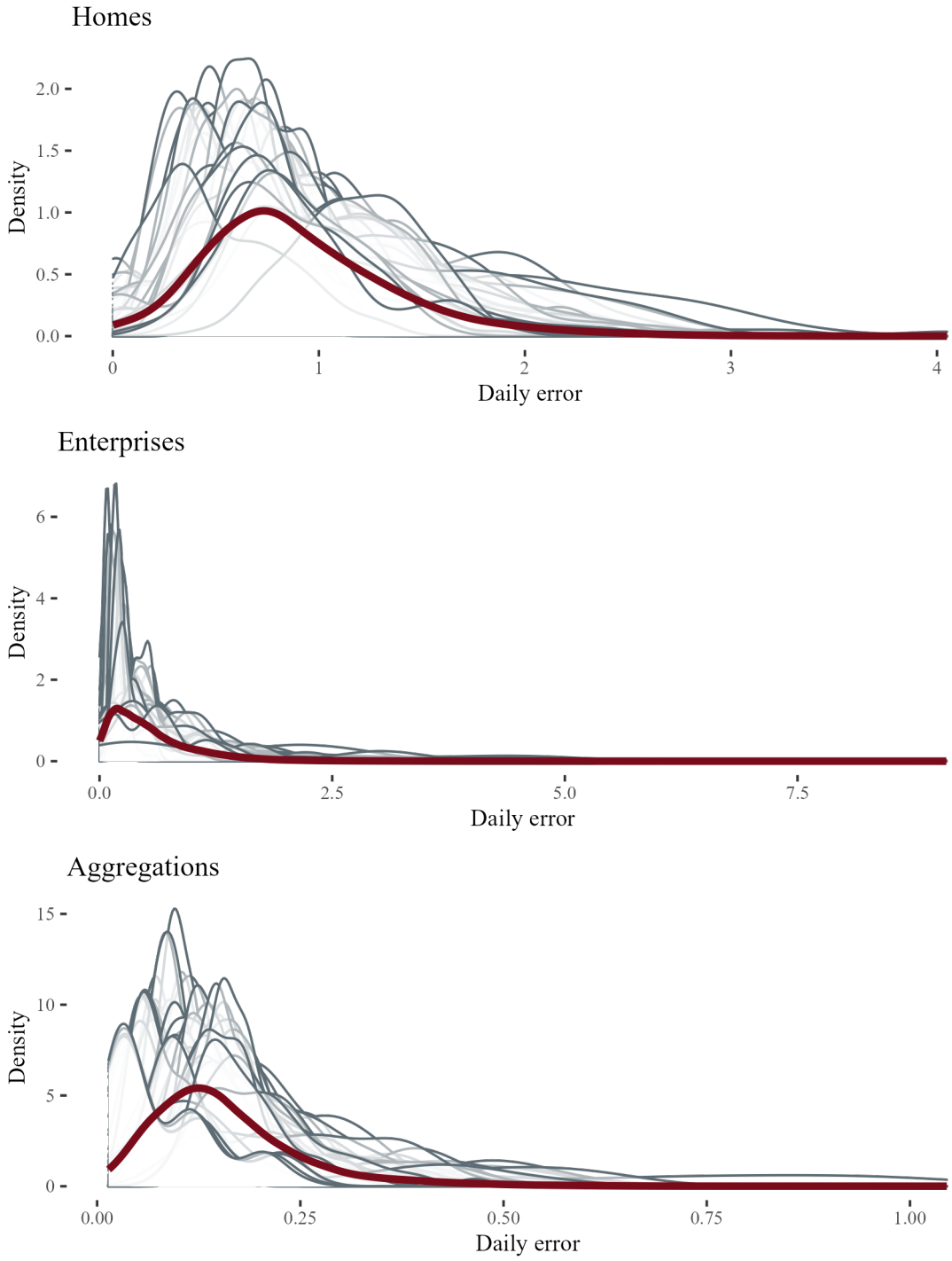

Figure 7.19: Daily error distribution shape of a naive model. We applied the naive model (Section 9.2.1.2) predicting the 300 loads in the validation dataset (Section 9.1.1.3) day-by-day for 100 consecutive days. For every daily load forecast, we computed the PRMSE (7.12) expressed in terms of coefficient of variation (7.13). Each panel shows the daily error distributions in form of the probability density function (grey) of each load and the average distribution (red) in the corresponding load group. We observed that daily forecast errors are often approximately log-normally distributed. Further, the model often produced very small forecast errors since the naive approach can deliver an almost perfect prediction when the building inhabitants are absent for two or more days.

In contrast, the distribution of improvement R can be approximated with a normal distribution (Figure 7.20). The improvement can be both positive or negative. Moreover, the skew is reduced since on the special days where a model performs poorly, the benchmark is also likely to be deficient. Hence, we can summarize the improvement on a set of days or loads using the mean and some notion of variability such as *standard error (SE)* or *confidence intervals (CI)*.

To quantify the overall forecast accuracy on a given load, we define the following *secondary error notion*.

Definition 7.3.6. *Expected daily error (EDE)*

$$\mathrm{EDE} = \mathbb{E}\Big[E_j\Big] = \frac{\sum_j^m E_j}{m} \tag{7.15}$$

is the average of m daily errors E_j obtained by a forecasting model predicting a given load.

Consider the distribution of EDE computed for a group of households, enterprises and aggregations (Figure 7.21). The distribution is approximately normal for households, yet is notably asymmetrical and skewed for the other two load types. Therefore, median is a more appropriate statistic to summarize an EDE-distribution. To quantify the overall forecast error obtained on a set of loads, we define an additional secondary error notion.

Definition 7.3.7. *Total error (TE)*

$$\mathrm{TE} = \mathrm{median}\Big[\mathrm{EDE}\Big] \tag{7.16}$$

is the median of expected daily errors obtained on a set of individual loads.

In the load forecasting literature, the results are often reported in terms of some summary statistics (e.g., EDE, TE) and, if at all, a measure of variability for the given dataset (Chapter 5). The models are evaluated on a single load or a group of loads and compared in terms of the overall accuracy obtained on the given dataset.

To make general statements about the accuracy of a model, we have to evaluate it on various building loads of different type and size. In fact, there is a strong connection between the load size and the forecast accuracy [HWVA13, SR14, SR18]. Sevlian et al., propose an empirical scaling law that allows to estimate the forecast error that a model will obtain on a load of a given size [SR14]. Applying this law, we define a secondary error notion that allows to estimate model accuracy on the entire spectrum of buildings with all possible sizes.

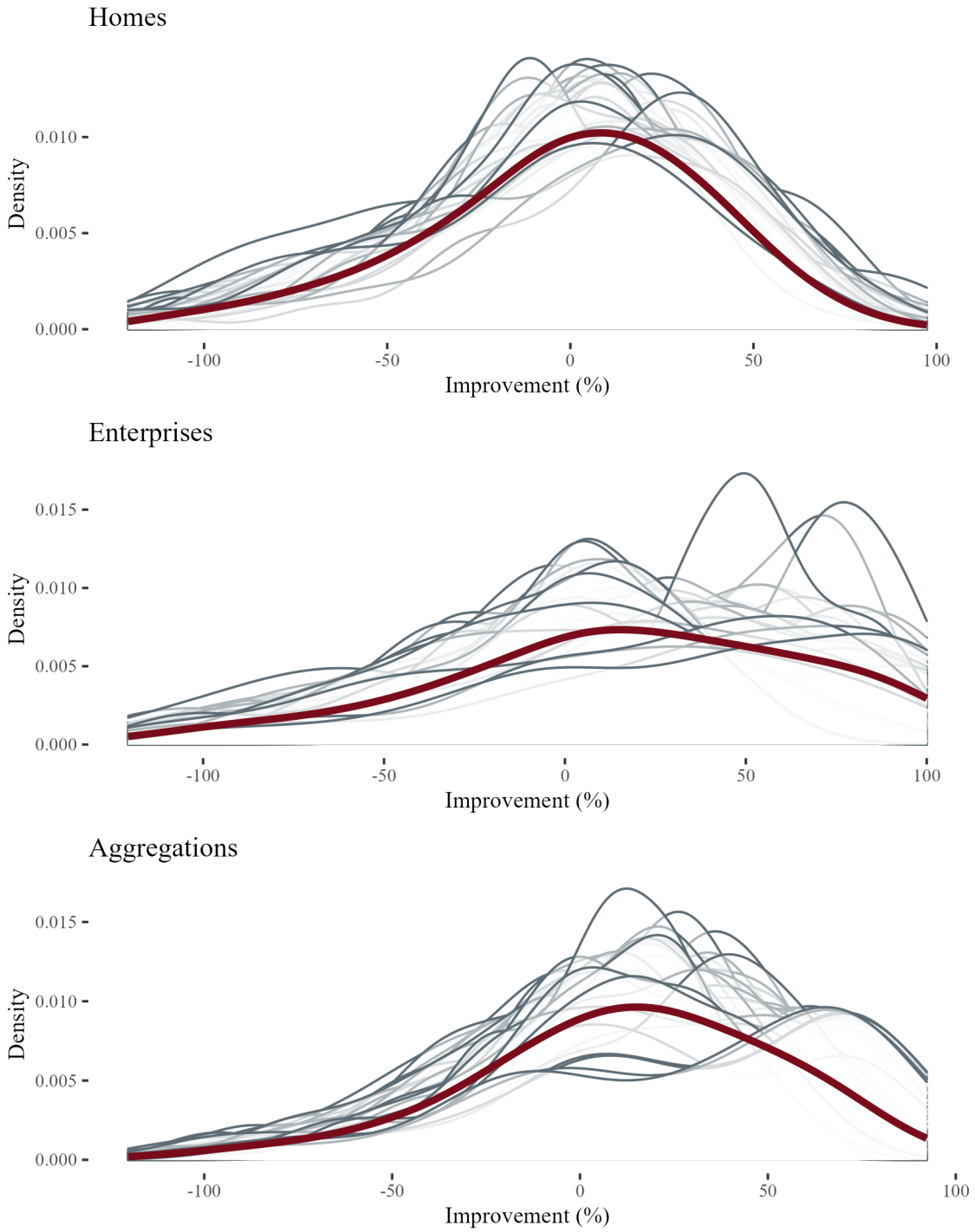

Figure 7.20: Improvement distribution. We applied a naive model (Section 9.2.1.2) and the standard load profiles (Section 9.2.1.1) predicting the 300 loads in the validation dataset (Section 9.1.1.3) day-by-day for 100 consecutive days. For every daily load forecast, we computed the improvement (7.14) relative to the SLP-forecast with the PRMSE (7.12) expressed in terms of coefficient of variation (7.13). Each panel shows the improvement in form of the probability density function (grey) of each load and the average distribution (red) in the corresponding load group. We observed that daily relative forecast errors (e.g., improvement) are often approximately normally distributed.

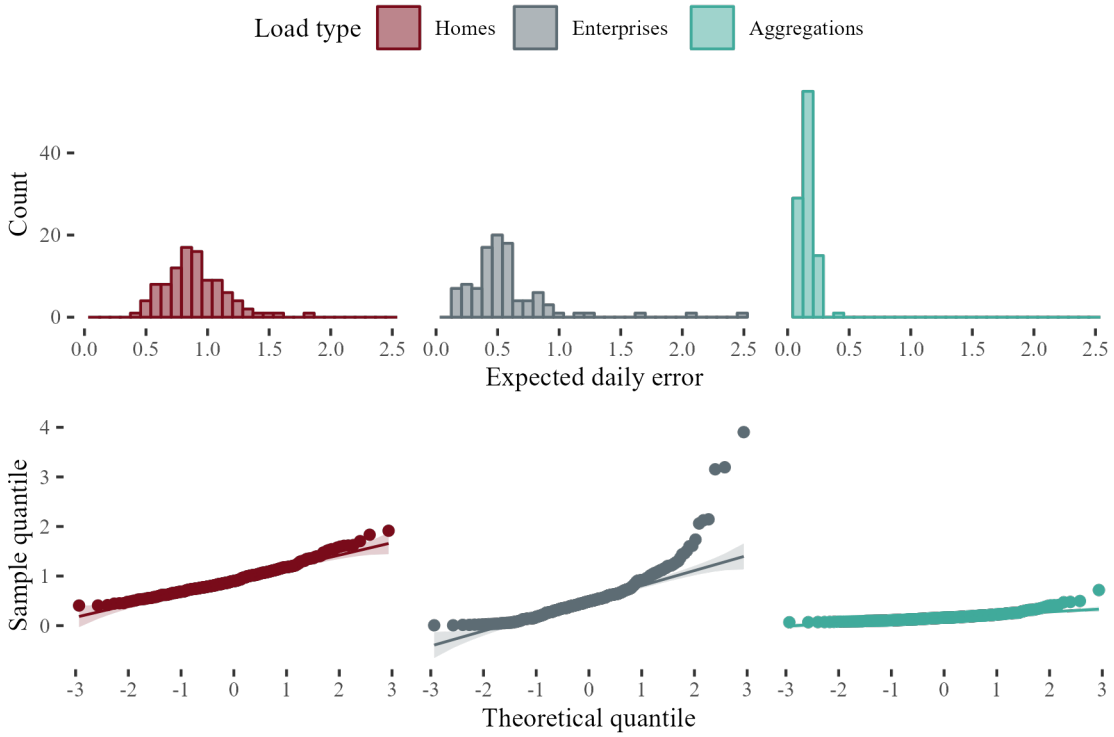

Figure 7.21: Distribution of expected daily errors in different load groups. We applied a naive model (Section 9.2.1.2) predicting the 300 loads in the validation dataset (Section 9.1.1.3) day-by-day for 100 consecutive days. For each load, we computed the expected daily error (7.15). In the figure, each panel represents the distribution in the corresponding load group. The top panels show the distribution while the lower panels show the corresponding Q-Q-plots. We observed that the distribution is approximately normal in case of households, yet is notably asymmetrical and skewed for the enterprises and aggregations. Further discussion is provided in the text.

Definition 7.3.8. *Expected model error (EME)*

$$\mathrm{EME}(S) = \mathbb{E}\big[\mathrm{ECV} \mid \mathrm{S}\big] = \sqrt{\frac{\alpha}{S^p} + \beta} \tag{7.17}$$

is the error that can be expected from a forecasting model applied on the load of a given size S according to the empirical scaling law [SR14] with predetermined parameters p, α, β.

The EME is scale-independent and is related to the load size that we express in terms of annual consumption (MWh). The relation is determined by the parameters p, α and β that we compute for the evaluated forecasting model. Given a sample of forecast errors collected on a set of loads of different sizes, we estimate the values of p, α, β applying a nonlinear weighted regression technique. Herewith, for a given model and load size, we can provide an out-of-sample estimation of the forecast accuracy on the loads that are not part of the evaluated dataset.

As noted previously, larger loads are easier to forecast (Section 7.1) and we can expect the scale-independent forecast error to decrease with growing building size (Figure 7.22).

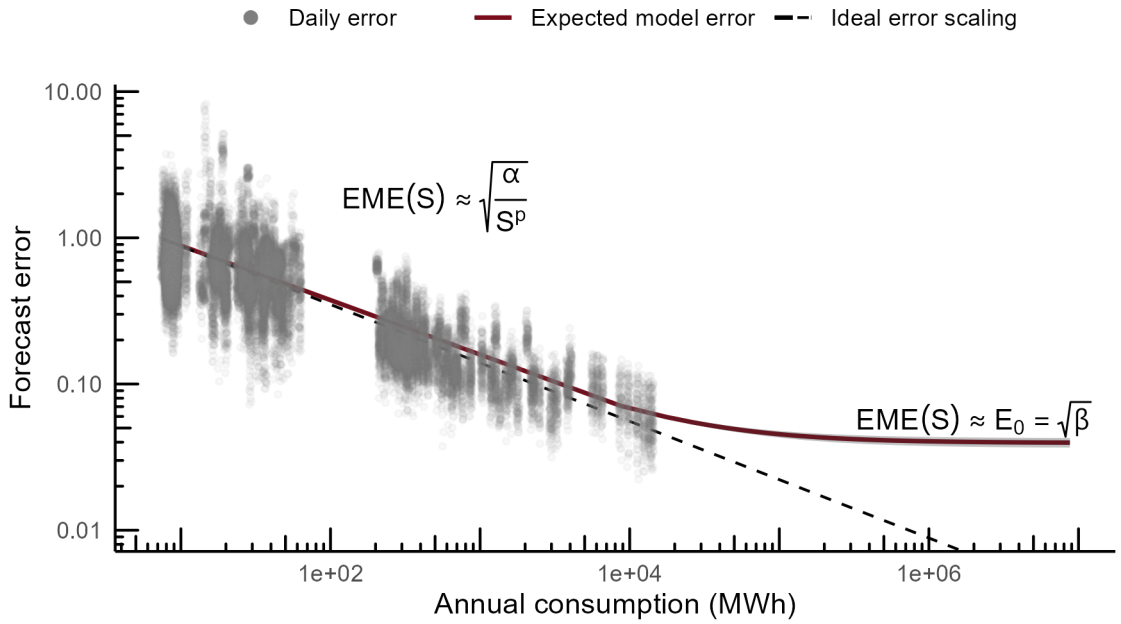

Figure 7.22: *Expected model error (EME)* of a forecast. The figure shows the 30000 daily errors (grey dots) obtained by the standard load profile forecast (Algorithm 3) predicting the loads in the validation dataset (Section 9.1.1.3). Having obtained a set of daily errors according to PRMSE (7.12) expressed in terms of coefficient of variation (7.13), we computed the EME (7.17) (red line) according to the empirical scaling law (7.17) using nonlinear weighted regression and compared it to the ideal scaling (black line). Further discussion is provided in the text.

Indeed, when describing how EME scales with the load size S, we can identify two regimes. For smaller loads, we have the *reduction regime*

$$\sqrt{\frac{\alpha}{S^p}} \gg \sqrt{\beta}, \tag{7.18}$$

where the EME decreases steeply with increasing load size and closely follows the *ideal error scaling*

$$\mathrm{EME}(S) \approx \sqrt{\frac{\alpha}{S^p}}, \tag{7.19}$$

where parameters p and α determine how fast the error reduces with the size – i.e., slope of the curve. However, as the load size increases past a certain point, we enter the *saturation regime*

$$\sqrt{\beta} \gg \sqrt{\frac{\alpha}{S^p}}, \tag{7.20}$$

where EME converges towards the *irreducible error*

$$\mathrm{EME}(S) \approx E_0 = \sqrt{\beta} \tag{7.21}$$

beyond which the model accuracy will not improve. The size S_{crit} where the regime changes, i.e.,

$$\sqrt{\frac{\alpha}{S^p}} = \sqrt{\beta}, \tag{7.22}$$

is called *critical load size* defined as

$$S_{\text{crit}} = \left(\frac{\alpha}{\beta}\right)^{1/p}. \tag{7.23}$$

The secondary error notions (EDE, TE, EME) introduced above allow us to compare forecasting models in terms of the expected error on a particular load or a group of loads and estimate the accuracy on the loads which are not part of our dataset. The usual comparisons of EDE and TE obtained on the loads of a particular type extend the analysis preventing Simpson paradox and complement the computation of the EME. The EME estimates the error that we can expect when forecasting a load of a given size. With these secondary error notions, we can evaluate a forecasting model across the building domain estimating its accuracy on the loads of different types and all possible sizes.

7.3.2.3 Statistical Tests

Regardless of the chosen notion, forecast errors underlie considerable stochastic variation. For a given building, daily errors vary substantially depending on the day. Moreover, overall error will also vary depending on the building. Hence, to compare different models applied on a single or several buildings, we need to assess if the difference in forecast accuracy is statistically significant.

Consequently, our evaluation methodology must include the *hypothesis testing* – a method of making inferences from empirical evidence [Lav21]. We use samples of observations to substantiate statements about the underlying population distributions and make the inferences using *statistical tests*. Prior to conducting a test, we formulate two mutually exclusive hypotheses about the population distribution. The *null hypothesis (H0)* is presumed true until the test indicates otherwise. The *alternative hypothesis (H1)* can refute the H0 and corresponds to the investigated question. By comparing the observed samples to what we would expect if a hypothesis was true, we can draw substantiated conclusions about the population distributions.

A statistical test can only provide evidence for the H1 with a certain *significance level* which represents the probability of mistakenly rejecting the null hypothesis if it was true. For instance, a commonly used significance level of 5% means that there is a 1 in 20 chance of doing so. Alternatively, we can also report the p-value representing the probability of

obtaining the evidence[20] at least as extreme as ours if H0 was true. Usually, the results are viewed as significant for $p < 0.05$ [Lav21].

There exist different statistical tests for various situations. The choice must depend on the test assumptions and experiment setup. Imagine, we want to compare the accuracy of two models predicting a set of load curves[21]. We collect two samples of daily error observations and want to know if any of the two models can be *expected* to be more accurate than the other. To compare the models, we can use different tests discussed below.

One way to make a statistically substantiated comparison, is to combine both samples into one by computing the relative error R (7.14) for each predicted load curve. With a sample of R, we can verify if the *expected* improvement is significantly larger than zero using the following test.

Test 1: One-sided Independent t-Test Assuming that the observations are approximately normally[22] distributed, this parametric test verifies the following hypotheses:

H0 Population mean is not larger than zero.

H1 Population mean is larger than zero.

With this test, we can provide statistical evidence that a predictive model is significantly more accurate than a benchmark. To do so, we collect a sample of improvement observations, conduct one-sided independent t-test and report the mean improvement, 95%-confidence intervals together with the p-value.

Alternatively, we can test if the observed average difference in model accuracy is significant. Note that the observations in the samples are paired since each model forecasts the same load curve. Herewith, if an independent test does not provide evidence for statistical significance, we can apply a paired test that is more sensitive.

Test 2: Paired t-Test This parametric test requires a sample of paired differences. Assuming the difference observations are normally distributed and there are no extreme outliers, we verify the following hypotheses:

H0 Population means are equal.

H1 Population means are different.

[20] The evidence is commonly expressed in terms of a test statistic.

[21] In our case, these curves can be daily load curves of a single building or a set of buildings.

[22] In its original form (Student's t-test), this test also assumes homoscedasticity. Alternatively, its modified form (Welch's t-test) can be used if homoscedasticity is not given.

With this test, we can make statistically founded pair-wise comparisons of predictive models. To do so, we collect the observations of improvement relative to a common benchmark (7.14), conduct the test for each pair of the models and report the differences in the mean improvement, 95%-confidence intervals, together with the p-value.

In this case, the usage of a mean as a summary statistic is adequate since the improvement observations are, disregarding eventual outliers, approximately normally distributed (Figure 7.20). At the same time, absolute forecast errors (e.g., EDE) are often not normally distributed (Figure 7.19). To evaluate the differences in absolute forecast errors, we can use the following nonparametric test that has much weaker assumptions on the sample distribution.

Test 3: Paired Wilcoxon Signed Rank Test This test is a nonparametric equivalent of the paired t-test. Unlike the t-test, paired difference distribution does not need to be normal. Assuming that the distribution is symmetrical, we verify the following hypotheses:

H0 Medians of the population distributions are equal.

H1 Medians of the population distributions are different.

Given that the differences in forecast errors are often symmetrically distributed, we will use this test to evaluate if the observed difference in median absolute forecast errors is significant. For each model comparison, we report the median errors together with the corresponding p-values.

Overall, statistical tests allow us to evaluate the accuracy despite the uncertainty in the observed forecast errors. Parametric tests have more power comparing to the nonparametric tests but have strong assumptions about the sample distribution. We will use the tests described above to compare forecasting models between each other as a part of our evaluation methodology summarized subsequently.

7.3.3 Evaluating Models Across the Building Domain

In this chapter, we discussed the specialties of day-ahead building power demand forecasting. We studied the diversity of the loads, which we encounter when applying a predictive model in the building domain on a wide scale (Section 7.1). In the context of day-ahead forecasts, we can expect the daily electricity consumption pattern to vary substantially depending on the predicted day and load. Consequently, the forecast errors of a model underlie considerable stochastic variation. In order to guide the solution of the wide-scale day-ahead local load forecasting problem (Section 7.2), we introduce a methodology for

evaluating predictive models on a diverse set of low-voltage loads that can be found across the building domain (Section 7.3).

On volatile low-voltage loads, traditional point-wise accuracy measures such as MAPE and RMSE, that are ubiquitous in the literature, can be deficient (Section 7.3.1.1). Further, a single forecast error notion can be insufficient to evaluate the accuracy of a building load forecasting model since building domain includes a large diversity of loads of different type and size. For an adequate evaluation across the building domain, we need to apply descriptive statistics considering the stochastic variation of the forecast errors (Section 7.3.2.2). Additionally, forecast accuracy of any predictive model depends on the size of the predicted load. Therefore, we need to estimate the forecast error that we can expect on a building of a given size, when considering a model for a wide-scale application across the entire building domain. Moreover, we have to rely on inferential statistics such as hypothesis testing to assess the significance of the results and avoid case-based reasoning when comparing different models for the wide-scale day-ahead building load forecasting application (Section 7.3.2.3).

For our study, we propose a methodology for evaluating predictive models across the entire building domain. Focusing on volatile low-voltage loads, our methodology is based on the scale-independent and unitless *permuted root mean squared error (PRMSE)* which we use as the *primary error notion* for quantifying the daily forecast errors. We remove the scale by expressing the error in terms of coefficient of variation which allows us to compare the models that were applied to different loads. Additionally, our evaluation methodology includes various *secondary error notions* (EDE, TE, EME) that quantify the overall forecast accuracy obtained on a given load or a sample of loads and are based on descriptive statistics adequate for the error samples collected within building domain load forecasting. Given a sample of forecast errors obtained predicting the loads of various size, we can apply the empirical scaling law estimating the error which we can expect from the model when predicting a load of a given size. Therefore, the proposed evaluation methodology allows to compare forecasting models in terms of the expected error on a particular load or a group of loads and estimate the accuracy on loads that are not part of the given dataset.

With our methodology, we evaluate a predictive model proceeding as follows:

1. We apply the model computing day-ahead load forecasts on a diverse smart-meter dataset[23] that includes buildings of different type and size.
2. For each forecast, we compute the daily error (7.12) expressed in terms of the coefficient of variation (7.13).

[23] An example of such dataset can be found in Section 9.1.1.

3. For each load, we compute the EDE (7.15) obtaining a sample consisting of EDE-observations.
4. Preventing Simpson's paradox, we split the EDE-sample into load groups conditioning on the consumer type and size. For each group, we quantify the overall accuracy computing TE and IQR (7.16).
5. If possible[24], we use the daily error sample to estimate the parameters α, β, p for the EME computation (7.17). With these parameters, we compute the critical load size S_{crit} (7.23) and the irreducible error E_0 (7.21).

Having followed these steps, we can compare different forecasting models using appropriate statistical tests (Section 7.3.2.3). In particular, we can compare:

- EDE-distribution and its summary statistics (TE, IQR) in each load group
- improvement relative to a common benchmark forecast in each load group
- critical load size S_{crit}
- irreducible error E_0.

The proposed forecast evaluation methodology allows to access and compare forecasting models across the building domain estimating their accuracy on the loads of different types and all possible sizes. In the subsequent chapters, we consider various predictive models which we will evaluate applying the proposed methodology.

[24] To estimate the parameters determining the error scaling (EME), smart-meter dataset must include the loads of various size with larger loads exceeding 100 MWh of annual consumption.

8 The Forecaster

In this chapter, we present the model that can be applied for a wide-scale day-ahead building load forecasting. In this application, explicit knowledge about each building and its physics is not available (Section 7.2.1). Hence, we focus on developing a data-driven regression-based model. Subsequently, we solve the forecasting problem (Section 7.2.3) using statistical learning theory[1] and considering time-series nature of the data.

To outline the solution, we assume that $x, y \in \mathbb{R}$, though same argumentation holds for multidimensional spaces $X \in \mathbb{R}^q, Y \in \mathbb{R}^n$. Further, we consider each point of the time series Y_t as being generated by the *regression model*

$$y = \mathbf{r}(x) + \epsilon, \tag{8.1}$$

where *regression function* $\mathbf{r}(x)$ describes the deterministic relationship between x and y. A random *additive error* ϵ captures all other influences independent of x.

Given a set of functions $\mathbf{r}_\alpha(x), \alpha \in \Lambda$ indexed by a parameter or a set of parameters α, we intend to select $\mathbf{r}_\alpha$ using a sample of m input observations X_s and output observations Y_s combined into a training set

$$\mathcal{T} := \{(x_j, y_j) \mid 1 \leq j \leq m\} \text{ with } x_j, y_j \in \mathbb{R} \tag{8.2}$$

minimizing the *expected prediction error (EPE)*

$$\text{EPE}(\mathbf{r}) = \mathbb{E}\Big[L(y, \mathbf{r}(x))\Big], \tag{8.3}$$

where $L(y, \mathbf{r}(x))$ is a predefined loss function for which applies

$$L(y, \mathbf{r}(x)) \geq 0, \forall x, y \in \mathbb{R}. \tag{8.4}$$

For finding $\mathbf{r}$, the following assumptions are common for the regression methods developed within the theory of statistical learning [FHT08, Vap10]:

[1] For more details, see comprehensive works by Vapnik [Vap10] and by Friedman et al., [FHT08].

Assumption 1. *Unbiased error i.e.,*

$$\mathbb{E}\Big[\epsilon\Big] = 0. \tag{8.5}$$

This assumption simplifies mathematical derivations and, once we have a prediction, we can validate this assumption by examining the forecast error. An existing systematic error $\mathbb{E}\Big[\epsilon\Big] > 0$ named *bias* should be considered by the model[2].

Assumption 2. *The data in $\mathcal{T}$ are sampled independently and identically distributed (IID) from an unknown but constant distribution expressed through a joint PDF $\mathbf{f}(x, y)$.*

Data-points are assumed to be uncorrelated and obtained from the same distribution. This assumption is fundamental for large parts of the theory and is standard in machine learning applications[3]. It implies that the data-points carry the same information about $\mathcal{Y}$, independently from each other. For a random set of points, there must be a common generative mechanism that we intend to learn about[4]. Assuming $\mathcal{T}$ is IID simplifies the underlying derivations and is necessary for many statistical inference methods[5].

Assumption 3. *Output values y_j are observed following an unknown but constant conditional distribution $\mathbf{f}(y|x = x_j)$.*

This assumption states the existence of a conditional PDF $\mathbf{f}(y|x)$ that we intend to learn. In other words, for all $j = 1, \ldots, m$, recorded data y_j depends on j only through x_j which makes our model useful for prediction.

With the above assumptions, we expand EPE (4.2) to

$$\text{EPE}(\mathbf{r}) = \mathbb{E}\Big[\mathbb{E}\Big[L(y, \mathbf{r}(x)) \mid x\Big]_{y|x}\Big]_x = \int_{-\infty}^{\infty} \mathbb{E}\Big[L(y, \mathbf{r}(x)) \mid x = x'\Big]_{y|x} dx', \tag{8.6}$$

[2] It is also common to assume that the error has a zero-mean Gaussian distribution. This assumption is useful but not necessary for the consistency of the derived models as it was shown by [Hub64].

[3] See [FHT08, JWHT13, Vap10, VS78] and the references therein.

[4] If every data-point is generated by a different PDF, having a sample $X_s = x_1, \ldots, x_m$ is equivalent to estimating an m-dimensional PDF using just one point in a training set – an infeasible task.

[5] While IID assumption is very strong, theoretical results such as proofs of model consistency or error bounds calculation – all rely on some kind of assumptions about $\mathbf{f}(x, y)$. Assuming IID is sufficient, though, not always a required condition. In general, it is not required for some of the, so called, discriminative models such as ANN, random forests and regression trees. Even if this assumption does not hold, a method can still provide a sensible estimation of $\mathbf{f}(y|x)$. Further, this assumption can be substituted by other, weaker assumptions, such as *exchangeability*, *conditional independence*, *ergodic data generating process* or *sufficiently fast mixing* [KM].

where (8.4) assures that the loss expectation is non-negative. Hence, a regression function $\mathbf{r}_0$ that minimizes EPE($\mathbf{r}$) can be found as[6]

$$\mathbf{r}_0(x) = \arg\min_{\mathbf{r}_a} \mathbb{E}\Big[L(y, \mathbf{r}_a(x')) \mid x' = x\Big]_{y|x}. \tag{8.7}$$

We can show that the theoretical *best fit* solution is [FHT08]

$$\mathbf{r}_0(x) = \mathbb{E}\Big[y \mid x' = x\Big]_y \tag{8.8}$$

using the common *squared error loss* function

$$L(y, \mathbf{r}(x)) = (y - \mathbf{r}(x))^2. \tag{8.9}$$

In praxis, there are different regression methods that can find an *approximately* best solution (Chapter 4). For our application, we are predicting nonstationary time series and are confronted with the following situation:

- The data in $\mathcal{T}$ is not independent, but correlated in time.
- The data in $\mathcal{T}$ is not identical since the underlying PDF changes over time.
- Selecting a random subset of $\mathcal{T}$ disturbs the autocorrelated structure of the data.
- Once a good solution close to $\mathbf{r}_0(x)$ is found, it remains valid only as long as the expectation $\mathbb{E}\Big[y \mid x\Big]_y$ remains approximately constant.

Under these circumstances, the assumptions above might not hold. Therefore, using traditional parametric regression methods might not be the best approach for our application and we have to develop a novel, specialized method for the data-driven load forecasting. Considering the differences between parametric and nonparametric models (Chapter 4), we focus on developing a forecaster based on a nonparametric regression technique due to the following reasons.

Reason 1. *Nonparametric regression does not require the existence of a globally valid relationship between inputs and outputs.*

Building electricity consumption can notably depend on human behavior. Inhabitant activities are difficult to model by a constant function $\mathbf{f}(y|x)$ that is valid for all inputs (i.e., *globally*). However, assuming that such relationship exists is fundamental for the

[6] This can also be formulated in terms of *risk* and *empirical risk minimization (ERM)* problem for a discrete set $\mathcal{T}$ which is more common for the machine learning field of study [Vap91].

parametric regression approach[7] (Section 4.1). Alternatively, nonparametric techniques model the input-output relationship only in the vicinity of a given x (i.e., *locally*) and do not require any a priori knowledge about the underlying stochastic process (Section 4.2).

Reason 2. *Nonparametric regression typically requires less data than parametric regression.*

Training data can be scarce if a building is new or just underwent a retrofit that might have considerably changed its energy consumption. A parametric model approximates the regression function globally, while the number of parameters required to approximate such function grows with its complexity[8]. Each parameter requires several training points for its computation. In contrast, a nonparametric model approximates the regression function locally while its form is given by the training data itself (Section 4.2). As a consequence, load forecasting models based on nonparametric regression usually require less training data (Chapter 5).

Reason 3. *Nonparametric models require less computation.*

Once trained, a parametric model remains valid for a short time due to the nonstationarity of our data (Section 7.1.1.3). The model has to be retrained regularly, which further increases computational burden. Alternatively, nonparametric regression is particularly adequate for an online training, where the model is retrained after each step and where new data is continuously added to the training set. Moreover, automatically setting hyperparameters of a regression model often implies selecting among numerous variants of the model (Section 4.1.3). Training various models becomes computationally expensive when multiple hyperparameters have to be set.

These reasons qualify nonparametric regression models for the wide-scale building load forecasting. Many practical problems feature nonstationary variable data with measurement errors and missing values. The literature shows that the parametric models can be effective for smooth load time-series encountered with larger buildings or in higher domains of a power system. At the same time, even simple heuristic models have been shown to outperform sophisticated parametric models on small disaggregated loads (Chapter 5).

In the rest of this chapter, we develop a forecaster based on the nonparametric regression approach. We begin the exposition, presenting the general setup of a nonparametric model

[7] For instance, commonly used ARIMA-model assumes linear relationship between the lags. A model based on an ANN is more general about its assumptions allowing to model a nonlinear dependency. However, the function form Λ is still given by the hyperparameters (number of neurons, layers, etc.), though there is no systematic methodology for the ANN-design.

[8] For instance, an ANN-based load forecasting model can have thousands of parameters that have to be computed during the training phase (Table 9.1).

that considers the seasonalities of the load and allows automated parametrization (Section 8.1). Subsequently, we introduce the functional neighbor methodology and propose a load forecasting algorithm for predicting building power demand on a wide scale (Section 8.2). While the corresponding model is purely autoregressive, we conclude this chapter by describing an extension that allows to explicitly consider exogenous variables that can affect a given building load (Section 8.3).

8.1 Nonparametric Load Forecasting

In this section, we describe the general setup of a nonparametric model for wide-scale day-ahead building load forecasting. In particular, we describe how to consider the inherent seasonalities and trends of the load curves. Moreover, we discuss how to select model hyperparameters adopting common cross-validation ideas for nonstationary time series. The specialized model we propose for solving the day-ahead building load forecasting problem will be described further in the text.

The idea of a nonparametric forecasting is simple. We find historical observations that are relevant for the given situation and predict the output as a combination of historical outputs. This approach does not require any model training or assumptions about the regression function. Nonparametric regression only assumes that similar inputs are likely to produce similar outputs.

For instance, a nonparametric model based on kernel regression predicts the output as an average of the relevant historical output observations y_j i.e.,

$$\hat{y} = \hat{\mathbf{r}}(x) = \mu(y_j). \tag{8.10}$$

The relevant I/O-observations (x_j, y_j) are the ones where x_j belongs to the neighborhood of the current input x. Note that the model $\hat{\mathbf{r}}(x)$ is determined on the go for a given x. It approximates the best-fit solution $\mathbf{r}_0(x)$ locally, in the vicinity of x. The approximation converges towards the conditional expectation $\mathbf{r}_0(x) = \mathbb{E}[y|x' = x]_y$ according to (8.8) if we are given a large amount of training data [FHT08].

We implement the nonparametric model using the K-nearest neighbors algorithm (Algorithm 1). Following the procedure, the distance $d = |x^* - x_j|$ quantifies the relevance of historical observations. Variable bandwidth K determines the size of the relevant region (*neighborhood*) around the input query x^*. Note that a fixed bandwidth could result in undefined predictions when the training data is limited – for some x, the fixed-sized neighborhood might have no data within it. The KNN algorithm is robust against such situations and allows to capture even very complex behavior of the regression function $r(x)$.

Algorithm 1: K-nearest neighbors algorithm

Inputs: input query $x^* \in \mathbb{R}$
Outputs: $\hat{y} \in \mathbb{R}$
Data: training set $\mathcal{T} := \{(x_j, y_j) \mid 1 \leq j \leq m\}$
Parameters: number of nearest neighbors K

1 Sort $\mathcal{T}$ with respect to distance $d = |x^* - x_j|$.
2 Determine K closest historical inputs x_j of the query x^*.
3 Average historical outputs y_j obtaining the prediction $\hat{y} = \mu(y_1, \ldots, y_K)$.

The intuition for nonparametric prediction of a univariate time series is as follows. Assume, the data generating process shows observable patterns of behaviors which are repeated over time. If we can find historical patterns similar to the current behavior of the series, then the previously observed subsequent behavior can be predicative for the immediate future.

The nonstationarity of the load has to be reflected by the nonparametric model. We can consider the inherent nonstationarities to a large extent by accounting for the seasonalities and trends of the load (Section 8.1.1). Nonstationarity of the data also complicates the usage of standard methods for model validation (Section 8.1.2) and selection (Section 8.1.3). Subsequently, we discuss these aspects of using the nonparametric modeling approach for the load forecasting before presenting our model afterwards.

8.1.1 Seasonality and Annual Cycle

Building electricity consumption underlies weekly and daily seasonality at all levels of load aggregation (Section 7.1.1.2). In other words, the load time-series notably depends on weekday and intraday time point (e.g., hour). In this section, we describe how to consider these dependencies with a nonparametric model.

8.1.1.1 Weekly Patterns

We saw that building power demand depends on the weekday to a varying extent (Section 7.1.1.2). A parametric model could account for this dependency *explicitly* by using weekday as an input variable. In contrast, a univariate nonparametric model can consider this dependency *implicitly*, as we explain through the following example.

Figure 8.1 shows the power demand of an enterprise and its forecast obtained by the KNN-model (Algorithm 1). At the top panel, we see that a notable prediction error occurs on Tuesday, where the enterprise remains closed. Evidently, the model considers historical observations that are not relevant for the day we are forecasting.

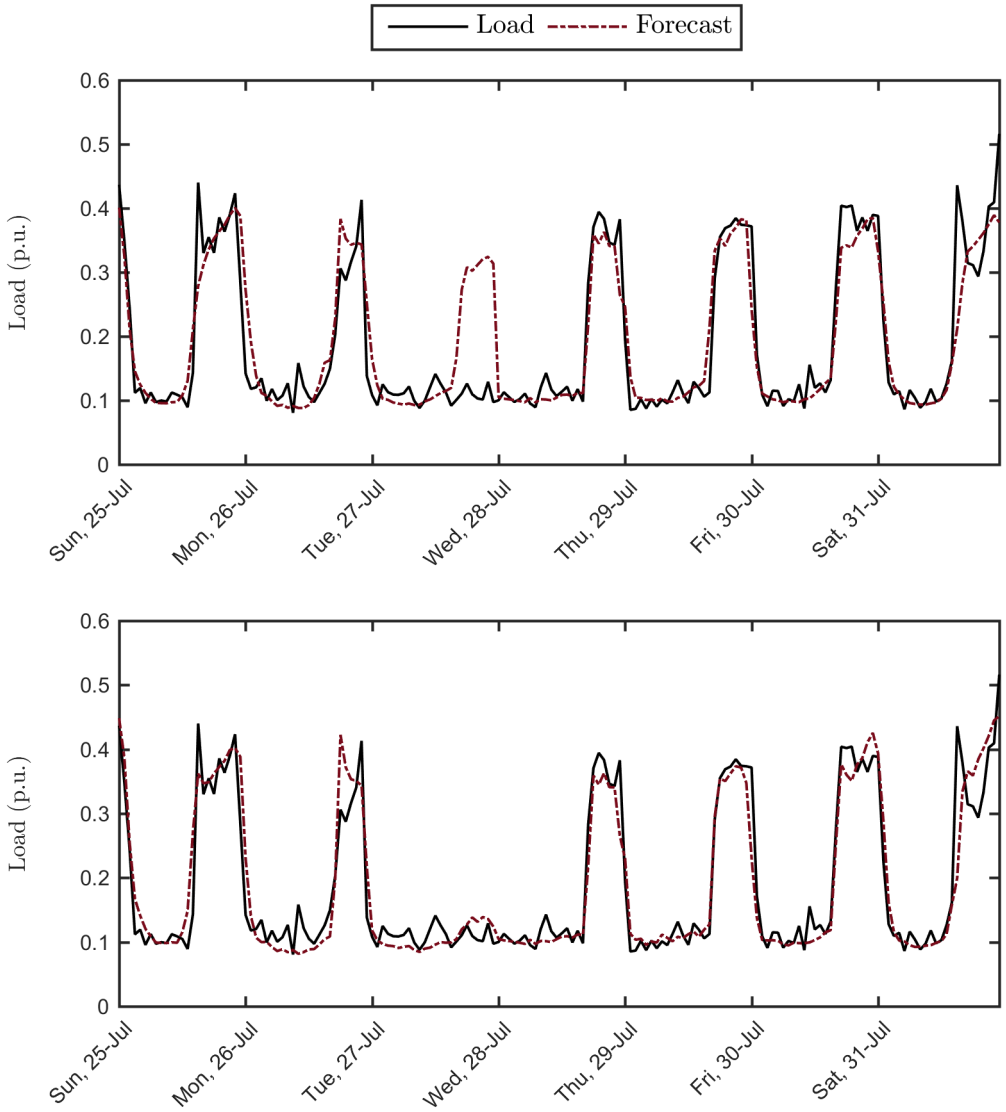

Figure 8.1: Forecast example illustrating weekly seasonality modeling of the load. Nonparametric forecast (Algorithm 1) that considers all historical observations results in a notable forecast error on Tuesday where the enterprise whose load we are predicting is supposedly closed (upper panel). Using only the historical days of the corresponding weekday allows the model to consider weekly seasonality and avoid such error (lower panel). Further discussion is provided in the text.

We can address weekly seasonality implicitly discarding any *interday* dependencies and filtering the observations using one of the following approaches:

- *filter by weekday (FbW)*,
- *filter by day-type (FbD)*.

Herewith, we obtain several groups of observations[9] and create a separate model for each group. In our example, FbW results in a better forecast as we see in the lower panel of the Figure 8.1.

Creating a separate model for each weekday considers any eventual weekly seasonality. However, this approach has several drawbacks. First, the model can only notice any abrupt change in the load characteristics after seven days. Imagine, if the discussed enterprise closes for a holiday or, in case of a residential building, the inhabitants leave on a vacation – the model filtering the historical observations by weekday will deliver erroneous forecast for the entire week. Moreover, weekday filtering requires a large amount of historical data. In order to consider 100 data-points in a model, we would require almost two years of measurements. During this period, the load can change its characteristics several times (Section 7.1.1.3). Additionally, there might be not much difference between business days (Figure 7.4). Especially in residential buildings, it may be hard to identify a pattern during the week, despite a notable difference between working and non-working days.

For these reasons, we can apply FbD instead. Rather than grouping by weekday, we create only three different groups: business days, Saturdays and holidays (incl. Sundays). Such approach is also followed by the standard load profiles which are commonly used for predicting the electricity consumption in distribution grids [ECo, Zuo00].

Nonetheless, we acknowledge that for some loads, weekly seasonality may persist within the business days (e.g. enterprise is closed every Tuesday). Therefore, we apply day-type filtering first. Then, we consider further filtering of business days by weekday if we expect this to reduce the forecast error even more[10]. The way to estimate this error is discussed further in the text.

8.1.1.2 Daily Patterns

To account for the daily seasonality, the nonparametric load model has to consider daily consumption patterns of a building. Traditionally, daily seasonality is modeled explicitly introducing a time-related variable (e.g., hour) as a model input. Another possibility is to consider the daily seasonality implicitly as described below.

[9] As a result, we have one group of observations for each weekday or day-type.
[10] Filtering by day-type and, then, by weekday will result in overall weekday filtering.

We model the daily seasonality by applying the multi-out multistep strategy with the nonparametric forecasting approach (Algorithm 2). This approach considers the intraday dependencies between the load measurements and allows to model the daily consumption patterns. Alternatively, having a separate model for each time-point (i.e., direct multistep strategy) also considers the daily seasonality to some extent, but discards any intraday dependencies.

In fact, the residuals of a KNN forecast using the direct strategy (Algorithm 1) are highly autocorrelated which indicates that there is still a substantial information in the time series that was not extracted by the model (Figure 8.2). In contrast, the residuals of a multi-out KNN forecast (Algorithm 2) are notably less autocorrelated which allows us to expect a more accurate forecast when we consider the daily seasonality implicitly by using the multi-out multistep strategy.

Algorithm 2: Multivariate K-nearest neighbors algorithm

Inputs: input query $X^* \in \mathbb{R}^n$
Outputs: $\hat{Y} \in \mathbb{R}^n$
Data: training set $\mathcal{T} := \{(X_j, Y_j) \mid 1 \leq j \leq m\}$
Parameters: number of nearest neighbors K

1 Sort $\mathcal{T}$ with respect to ℓ^2 norm $\ell^2(X^*, X_j)$.
2 Determine K closest historical inputs X_j of the query X^*.
3 Average historical outputs Y_j obtaining the prediction $\hat{Y} = \mu(Y_1, \ldots, Y_K)$.

8.1.1.3 Annual Cycle

We account for the annual cycle of the load in the nonparametric model *implicitly* by limiting the training data to 17 most recent observations. Consider the autocorrelation function of the loads in the ICER smart-meter dataset (Figure 8.3). For the majority of loads, we observed that there is a substantial correlation between the measurements obtained within ten weeks. At the same time, the autocorrelation disappears after six months (27-30 weeks) indicating that it might be counter-productive to include older data into the training set. Neglecting the short-term weather changes, we can consider the effect of the annual cycle on electricity consumption by limiting the history length to 17 weeks. This history length corresponds to the length of one season that is assumed for the computation of a standard load profile [ECo,Zuo00] and other profiling heuristics [BPT13].

Alternatively, the annual cycle can be considered *explicitly* by introducing the related time-variable (month, day) as a model input. This approach is often used in parametric

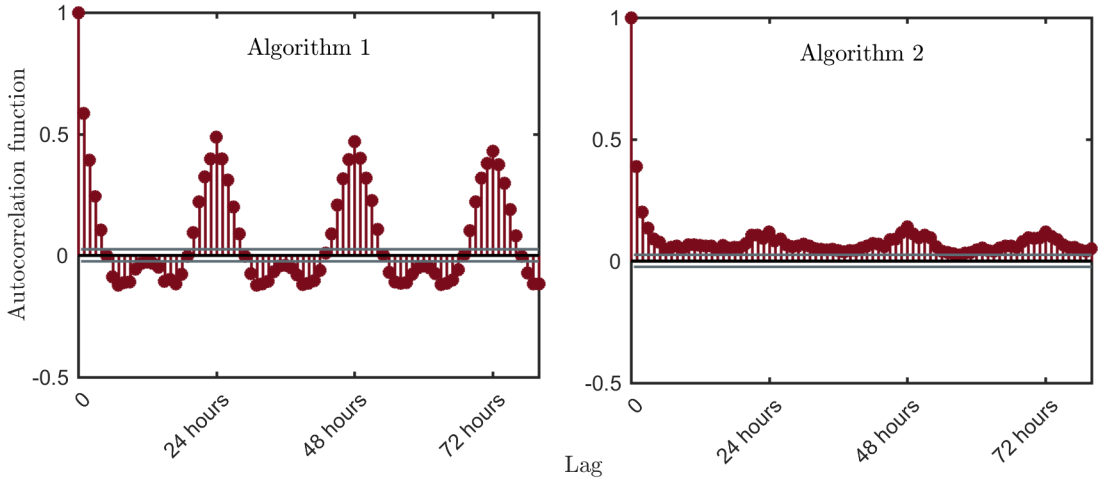

Figure 8.2: Residuals autocorrelation with different multistep strategies considering daily seasonality. The forecast of a residential building load was computed using the nearest neighbors model applying the direct (Algorithm 1) and the multi-out (Algorithm 2) multistep strategy. The residuals of the direct KNN forecast (left panel) are notably more autocorrelated than the residuals of the multi-out KNN forecast (right panel). Autocorrelated residuals indicate that the multi-out strategy allowed the KNN-model to extract more information from the time series and is better for considering daily seasonality of the load.

load models (Section 5.2.1) and requires to have, at least, one year of data for training[11]. However, we observed that the load autocorrelation is negligible for the corresponding lags (52 weeks) and there might be no use to consider the observations that are older than several months.

8.1.2 Validation Methods

Model validation relates to the estimation of the forecast error E which we can expect when applying the model on the unseen data. The error has to be estimated using the available historical data which constitutes our *training set* $\mathcal{T}$. Given $\mathcal{T}$, we are interested in the error that our model is expected to obtain on a, previously unseen, *test set* $\mathcal{U}$[12]. In this section, we consider various validation methods for estimating the forecast error (Table 8.1).

Out-of-sample (OOS) validation is a model validation method common in time series analysis. Its idea is to withhold the most recent part of the historical data as a *validation set* and evaluate the model on it in order to estimate the future model accuracy. Further, this method assumes that the performance on the validation set will be similar to the

[11] We would require historical observations of the same month as that of the day which we are forecasting.
[12] In statistical learning theory such error is called *generalization or test error*.

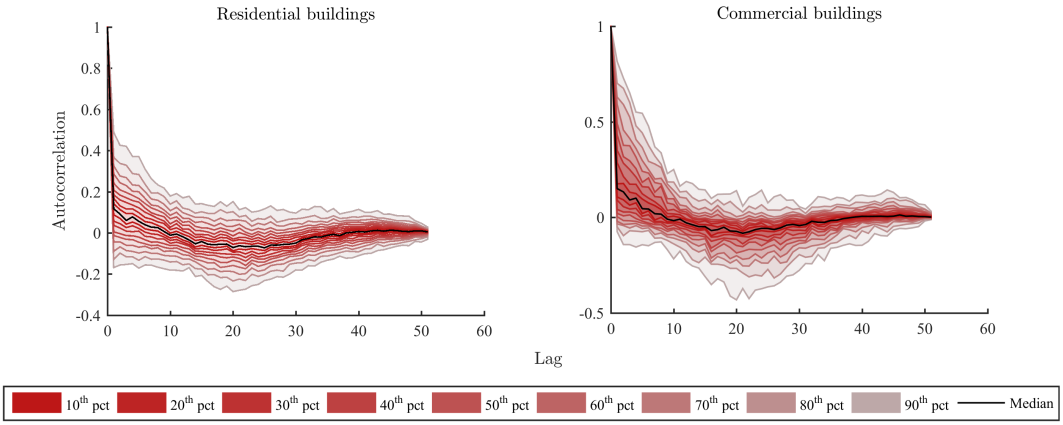

Figure 8.3: Autocorrelation function of the load measurements in the ICER smart-meter dataset. To exclude the influence of the weekly and daily seasonality, we only computed the autocorrelation of the time series consisting of the load measurements on a particular hour and weekday (Saturday, 8pm). At each lag, the multitude of the autocorrelation function values of the ICER-dataset is represented with percentiles (pct) and the median. For the majority of residential (left) and commercial (right) buildings, we see a substantial autocorrelation of the load to its recent observations (<20 weeks old) that quickly decays and becomes negligible for older observations (>30 weeks old). Therefore, older measurements might contain less information that can be used by the load model.

performance on the unseen data. Given enough historical data, out-of-sample validation can yield an accurate error estimate and is often used to validate time series models [BD10].

For our application, we do not have abundant data for training and validation. One year of load measurements contains 52 daily load curves. Given that we only use few weeks of data, our training and validation sets can be very small[13]. This is diminutive comparing to orders of magnitude that amount which are used for validating machine learning models. In our situation, we might need to estimate the anticipated model accuracy *in-sample* using the data in the most effective way.

Cross-validation (CV) is a standard machine learning method for efficient sample re-use [Sto74, AC10]. However, it relies on the Assumption 2 which does not hold for time series. Nevertheless, cross-validation can be still used, at least for trend-stationary time series, if the validated model delivers uncorrelated forecast errors [BHK18].

[13] As previously discussed, we use 17 weeks of data to consider annual cycle and apply FbW or FbD which leaves us with just 17 points in the training set in case of a Saturday or Sunday.

To apply k-fold cross-validation on a given training set $\mathcal{T}$ we proceed as follows. At first, we randomly split the training set $\mathcal{T}$ into k disjoint subsets $\mathcal{D}_i$ named *folds*[14] so that

$$\mathcal{T} = \bigcup_{i=1}^{k} \mathcal{D}_i. \tag{8.11}$$

Next, we use each fold $\mathcal{D}_i$ as a test set while training the model $\hat{\mathbf{r}}_{-i}$ on $\mathcal{T} \setminus \mathcal{D}_i$. With $|\mathcal{D}_i|$ test samples, we compute the mean test error

$$E_{\mathcal{D}_i}(\hat{\mathbf{r}}_{-i}) = \frac{1}{|\mathcal{D}_i|} \sum_{j \in \mathcal{D}_i} L(Y_j, \hat{\mathbf{r}}_{-i}(x_j)). \tag{8.12}$$

The overall error estimated by the CV is the average of errors obtained on all folds

$$\hat{E}^{\text{CV}}(\hat{\mathbf{r}}) = \frac{1}{k} \sum_{i=1}^{k} E_{\mathcal{D}_i}(\hat{\mathbf{r}}_{-i}). \tag{8.13}$$

Though the k-fold cross-validation was originally developed for $x, y \in \mathbb{R}$ [Sto74], we can expand it for a multidimensional case. With a training set $\mathcal{T}$ containing m observations of $X \in \mathbb{R}^q$ and $Y \in \mathbb{R}^n$ we apply the regression equation (4.10) obtaining

$$\begin{bmatrix} y_{11} & \cdots & y_{1n} \\ \vdots & \ddots & \vdots \\ y_{m1} & \cdots & y_{mn} \end{bmatrix} = \begin{bmatrix} \mathbf{r}(x_{11} & \cdots & x_{1q}) \\ \vdots & \ddots & \vdots \\ \mathbf{r}(x_{m1} & \cdots & x_{mq}) \end{bmatrix} + \begin{bmatrix} E_1(\epsilon_{11}, \cdots, \epsilon_{1n}) \\ \vdots \\ E_m(\epsilon_{m1}, \cdots, \epsilon_{mn}) \end{bmatrix} \tag{8.14}$$

where each row corresponds to an observation of $(X_j, Y_j) \in \mathcal{T}$. Following the cross-validation idea, we can interchange the rows as long as they are uncorrelated [BHK18].

A nonparametric model assumes that the most important information about Y_j is contained in the most recent observation $Y_{j-1} = [y_{j1}, \cdots, y_{jn}]$ which can be included into the input $X_j = [x_{j1}, \cdots, y_{jq}]$. The residuals $\epsilon_{j1}, \cdots, \epsilon_{jn}$ within a day j can be correlated among them, but the daily errors $E_1, \cdots, E_m$ have to be uncorrelated between each other allowing to mix the rows of the above matrix applying the cross-validation approach.

We apply the validation methods discussed above and summarized in Table 8.1 to estimate the forecast error of the nonparametric model (Algorithm 2). In particular, we use $\mathcal{T}$ consisting of 17 most recent historical days to estimate the error for the upcoming days

[14] For $k = m$ such variant is called *leave-one-out cross-validation (LOOCV)* that we introduced previously. A computationally efficient way to compute LOOCV-error is using PRESS-statistic as discussed in [BBTLB13].

Table 8.1: Validation methods overview.

Validation type	Validation method	Estimator name	Reference
out-of-sample	out-of-sample validation	OOS	[BD10, BHK18]
in-sample	5-fold cross-validation	5-CV	[Sto74, AC10]
in-sample	10-fold cross-validation	10-CV	[Sto74, AC10]
in-sample	leave-one-out cross-validation	LOOCV	[Sto74, AC10]

obtaining a series of estimated errors $\hat{E}$ and actual daily prediction errors E presented in Figure 8.4. Note that rather than estimating the E, all validation methods estimate the expected daily error (7.15)

$$\text{EDE} = \mathbb{E}\Big[E\Big]. \tag{8.15}$$

Observing that the validation methods, if anything, only estimate the EDE, we define the *relative estimation bias (REB)*

$$\text{REB} = \frac{\hat{E} - \text{EDE}}{\text{EDE}} \cdot 100\%. \tag{8.16}$$

With this measure, we studied the extent to which a validation method can provide an unbiased estimate of the forecast error that can be expected from a nonparametric model predicting the load day-ahead.

We observed that an unbiased estimate was rarely achieved by any of the methods (Figure 8.5). For some weekdays, the average REB was up to 20%. The in-sample validation method had a similar bias and provided no significant ($p < 0.05$) advantage compared to the out-of-sample validation. In fact, out-of-sample validation was often the most accurate estimating the EDE. At the same time, out-of-sample validation on a small dataset was sensible to the outliers (special days)[15]. As a result, we saw the largest confidence interval for this validation method. In contrast, leave-one-out cross-validation and 10-fold cross-validation had the smallest variation since the training set did not change substantially from day to day.

[15] Same observation is confirmed by the previously discussed Figure 8.4.

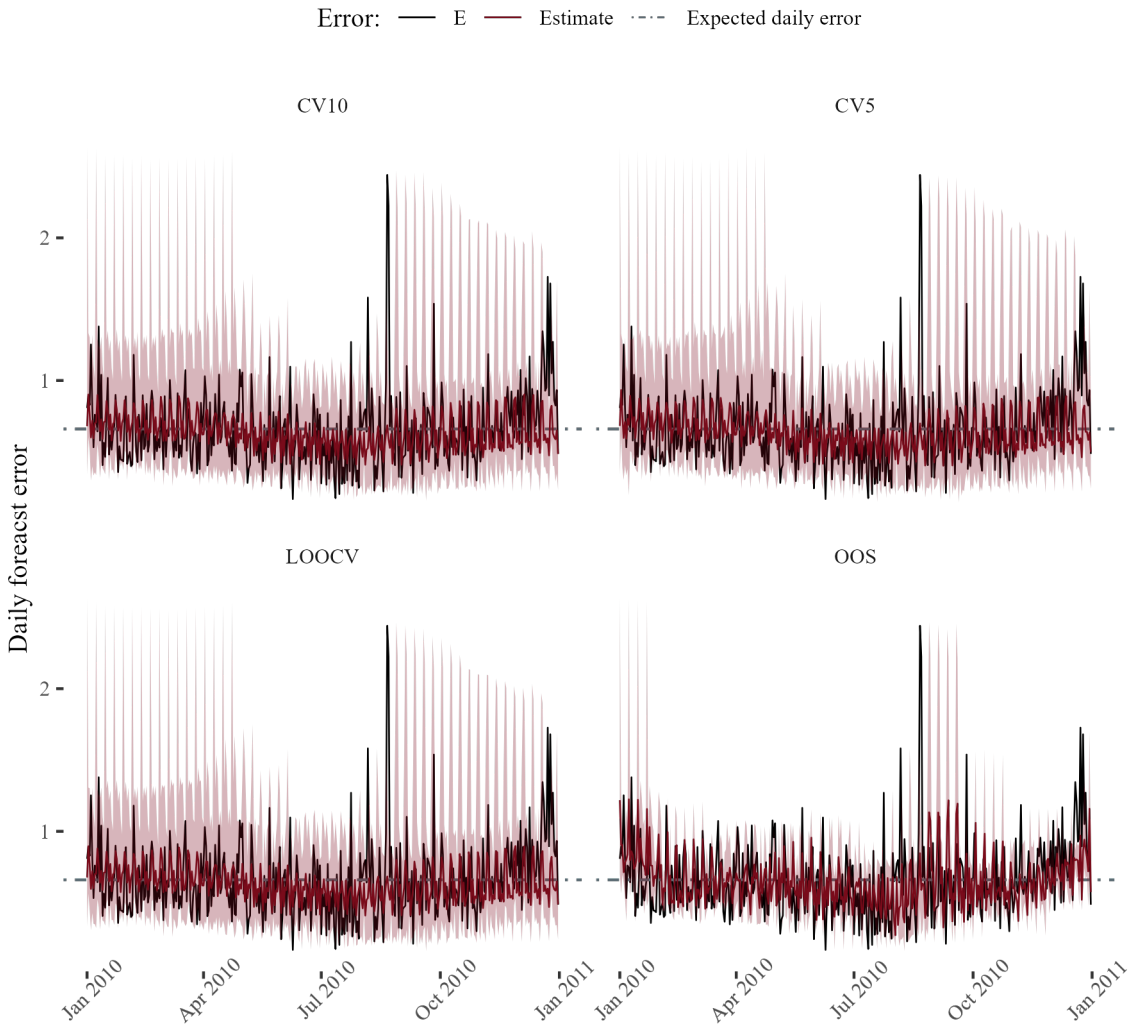

Figure 8.4: Forecast error estimation using various validation methods. In the figure, each panel shows the actual daily prediction error E, average daily error (i.e., EDE (7.15)) and estimated daily error $\hat{E}$ (red line) with its spread (red shadow) obtained by the corresponding validation method (Table 8.1). To collect the data, we applied the MKNN-model predicting the day-ahead load of a single family home (ID 1176) from the ICER smart-meter dataset [Arc16] for one year (Algorithm 2 with $K = 1$, FbW, and 17 weeks of training data). Further, we applied different validation methods (Table 8.1) to estimate the forecast error and compare the estimate to E. We observed that all estimators, failed to estimate E and instead estimated the EDE. Further, all estimators reliably estimated the spread, while its ripple could be explained by the filtering approach of the model (FbW). Further discussion is provided in the text.

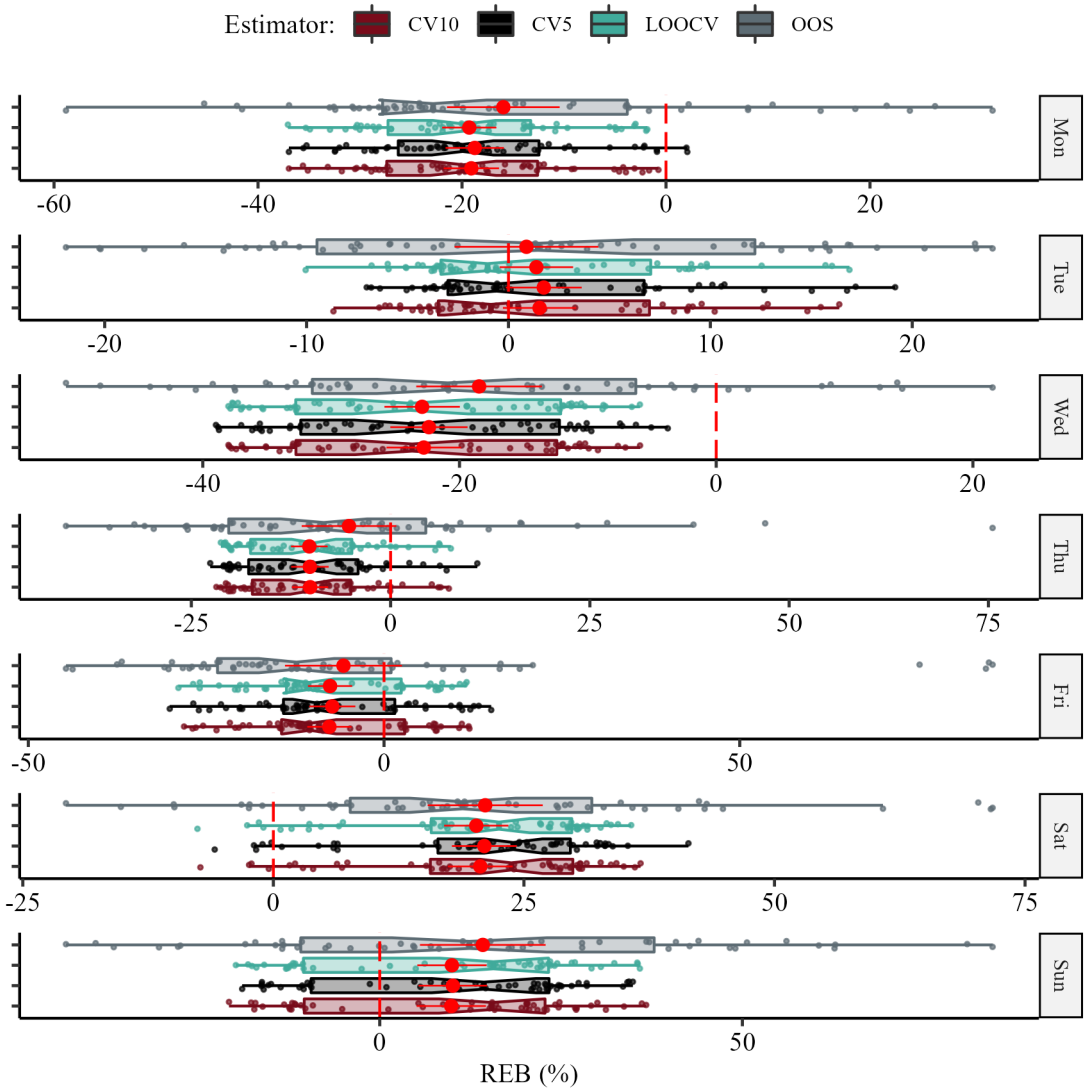

Figure 8.5: Relative estimation bias (REB) with various validation methods. In the figure, each panes shows the REB (8.16) observations of different validation methods that were collected on the corresponding weekday. In each panel, their distribution is summarized with a box-plot where the notch denotes the 95%-confidence interval of the median and the red dot represents the average shown together with its 95%-confidence interval (red horizontal bar). To collect the data, we applied the MKNN-model predicting the day-ahead load of a single family home (ID 1176) from the ICER smart-meter dataset [Arc16] for one year (Algorithm 2 with $K = 1$, FbW, and 17 weeks of training data). Further, we applied different validation methods (Table 8.1) to estimate the forecast error and computed the REB for each forecast day. We observed that all validation methods rarely achieved an unbiased estimate. For some weekdays, the average REB was up to 20%. The in-sample validation methods had similar average bias and provided no significant ($p < 0.05$) advantage compared to the OOS-validation method. In fact, the OOS-validation was often the most accurate estimating the EDE. Further discussion is provided in the text.

8.1.3 Model Selector

The error estimation bias becomes important when using the validation methods for designing a *model selector*. In a model selection task, we are given a set of models $\mathcal{M} := \{M_1, \cdots, M_d\}$ where the subscript enumerates the candidates and need to select the model M_α that minimizes the estimated error $\hat{E}$ that is determined by the previously discussed error estimator (Table 8.1). To investigate how the REB affects the model selection, we defined the *model selector success rate (MSSR)* as

$$\text{MSSR} = \frac{\text{NBS}}{\text{NHO}} \cdot 100\%, \tag{8.17}$$

which is as the *number of best selections (NBS)* relative to the total *number of historical observations (NHO)* that were used for the model selection. Here, the NBS is the number of occasions, where the model selector was able to determine the best model $M_\alpha = M_{\text{best}}$. The MSSR represents the relative share of observations where the model selector chose the best model from a set of candidates and reflects the quality of the model selector.

Nonstationarity of the low-voltage electricity consumption data is a challenge for the traditional validation methods and model selectors based upon them. In particular, we observed that the model selectors based on the in-sample validation was usually worse than using the out-of-sample validation (Figure 8.6). This indicates that for a nonparametric model applied for the day-ahead local load forecasting, there is a notable remaining correlation among the rows of (8.14). Hence, when selecting parameters of a nonparametric model, we should rely on the out-of sample validation that is commonly used in time series analysis, despite limited historical data [BD10, BHK18].

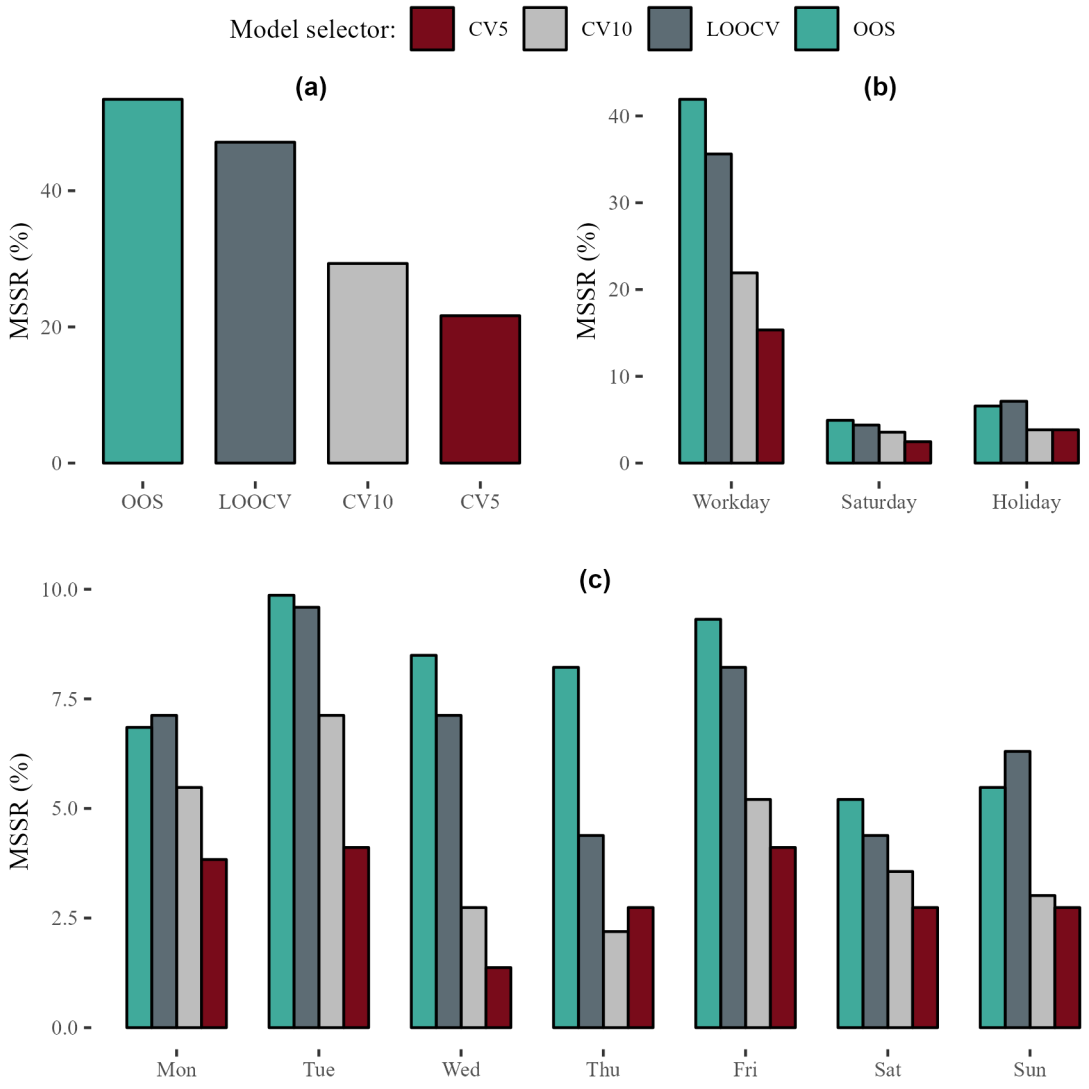

Figure 8.6: Model selector success rate (MSSR) using different validation methods. The figure shows the quality of model selection observed in a load forecasting experiment. In this experiment, we applied the MKNN-model predicting the day-ahead load of a single family home (ID 1176) from the ICER smart-meter dataset [Arc16] for one year (Algorithm 2, FbW and 17 weeks of training data). For this model, we used different model selectors with corresponding validation methods (Table 8.1) selecting the bandwidth K before each daily forecast. After the experiment, we computed the MSSR (8.17) of each model selector and represented it on a bar-plot for all forecast days (a), conditioned on day-type (b) and weekday (c). We observed that, on most days, the best model was found using OOS and (less often) LOOCV estimators. Further discussion is provided in the text.

8.2 Functional Neighbor Model

We propose a model for a wide-scale day-ahead local load forecasting based on the nonparametric approach (Figure 8.7). The proposed *functional neighbor (FN)* model is a generalization of the previously discussed multivariate KNN-regression (Algorithm 2). First, we use historical observations to create an object space

$$\mathcal{H} := \{\mathcal{D}_j \mid 1 \leq j \leq h\}, \tag{8.18}$$

where each object

$$\mathcal{D}_j := \{X_j, Y_j, \mathcal{Z}_j\} \tag{8.19}$$

corresponds to an observed day j and contains historical input X_j, output Y_j and a general set of features $\mathcal{Z}_j$ that includes further information about the day (e.g., weekday, weather etc.) as well as other exogenous variables[16]. With this data, we set up the model finding its hyperparameters using previously discussed model validation methods (Section 8.1.2). Next, given a query X^*, we find its K-nearest neighbors

$$\mathcal{G} := \{\mathcal{G}_j \mid 1 \leq j \leq K\} \tag{8.20}$$

using a predefined distance notion (Section 8.2.2). At last, we combine historical outputs of $\mathcal{G}$ to the forecast $\hat{Y}$ (Section 8.2.3).

The main drawback of a nonparametric approach is that it is notably affected by the curse of dimensionality[17]. To mitigate this effect, we combine nonparametric modeling approach with functional methodology discussed next.

8.2.1 Functional Methodology

We apply functional methodology[18] to develop the load model. This methodology is an extension of the traditional multivariate analysis which we considered so far (Section 8.1). Functional methodology has several advantages for modeling the situations where the data generating process is driven by human behavior, is nonstationary but exhibits repeated patterns [BRXK02, RS02, LNVR07]. Moreover, functional methodology addresses the curse

[16] In this section, we consider a univariate autoregressive model with $X, Y \in \mathbb{R}^n$ and no exogenous inputs. Further in the text, we provide an extension that will allow our model to consider external inputs (Section 8.3).

[17] We discussed this effect in Section 4.2.3.2.

[18] We provided a short introduction to functional data analysis in Section 4.3.

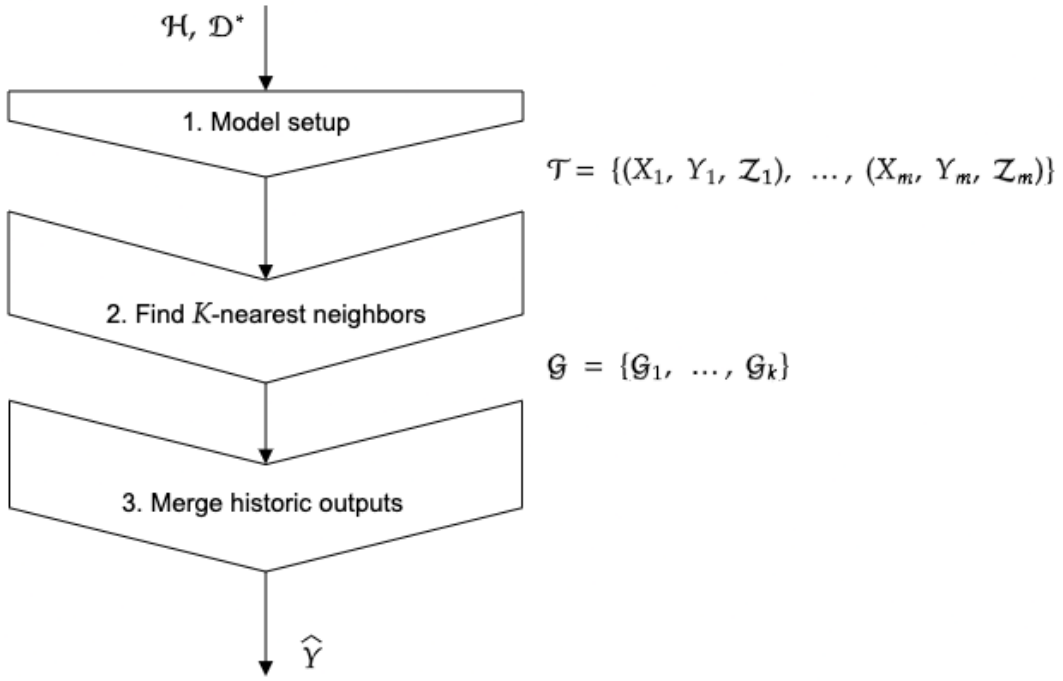

Figure 8.7: Functional neighbor forecaster. Description is provided in the text.

of dimensionality – the major limitation of a nonparametric model (Section 4.2.3.2). Consequently, functional load forecasts can be at least as accurate as conventional multivariate models (Section 5.1.3).

The term *functional* stresses that there is an underlying continuous function from an infinite-dimensional space of functions that produced the observed data. The main idea of the functional approach is to consider observations as single entities rather than a sequence of individual data-points. The developed theory is based on the functional data analysis and exploits additional structure of the continuous data. The functional view is only conceptual. The data is still measured on a finite discrete grid and processed digitally. However, using appropriate smoothing techniques allows us to view the measurements as a continuous curve, develop novel models and study their properties[19].

[19] In this section, we use Latin letters X, Y to denote time-discrete I/O observations and Greek letters χ, ϕ to denote the corresponding continuous curves.

8.2.1.1 Day-Ahead Load Forecasting in Functional Setting

We approach the day-ahead building load forecasting problem (Section 7.2.3) using functional regression. The load curve $y(t)$ is a continuous random variable generated by a continuous stochastic process

$$\mathcal{Y} = \{y(t) \mid t \in [0, \infty)\}. \tag{8.21}$$

The observed realization of $\mathcal{Y}$ is a time series of discrete and sequenced load measurements

$$y_t := \{y(t) \mid t \in T\} \tag{8.22}$$

with index set

$$T_0 := \{0, \cdots, (n_T - 1)\Delta t\} \tag{8.23}$$

consisting of the measurement time points with Δt corresponding to the measurement resolution.

In the context of building load forecasting (Section 7.1), we can assume that $\mathcal{Y}$ is a seasonal process with the *smallest season length* $\Delta_s t$. Considering daily seasonality of the demand, we define the *functional time series* $\{\phi_j\}_{j=1}^m$ where

$$\phi_j(t) = \{y(t) \mid t \in [(j-1)\Delta_s t, j\Delta_s t]\} \tag{8.24}$$

is a daily load curve. This functional series can be seen as a realization of a discrete functional-valued stochastic process

$$\Phi = \{\phi_j(t) \mid j \in \Big\{1, \cdots, m\Big\}\}. \tag{8.25}$$

For the given functional time series $\{\chi_j\}_{j=1}^m$, $\{\phi_j\}_{j=1}^m$, we assume a functional regression model

$$\phi_j = \mathbf{r}(\chi_j) + \epsilon, \tag{8.26}$$

where, analog to the multivariate approach, the output consists of deterministic relation captured by the regression function $\mathbf{r}(\chi)$ and uncaptured influences summarized in an error term ϵ. In this model, functional input variable

$$\chi \in (\mathbb{F}, \mathbf{d}) \tag{8.27}$$

lives in an infinite-dimensional space of functions $\mathbb{F}$ for which we will define a semimetric $\mathbf{d}$ named *distance notion* (Definition 4.3.1). Functional output variable

$$\phi \in (\mathbb{H}, \|\cdot\|) \tag{8.28}$$

lives in a measurable and separable Hilbert space $\mathbb{H}$ with a corresponding norm $\|\cdot\|$. Further, $\mathbb{E}\Big[\epsilon \mid \chi\Big] = 0$ and $\mathbf{r}(\chi)$ underly the Hölder condition (4.83) that is usually fulfilled in practice [FR11]. Even if χ, ϕ could be in the same space, we need to define a separate space $\mathbb{F} \neq \mathbb{H}$ for χ. Otherwise we can only rely on semimetrics that have a corresponding norm in $\mathbb{H}$ which appears limiting [FV06].

Each of the two structures endows the corresponding object space with a *topology*. In $\mathbb{F}$, we describe it defining a *ball* with a radius b and a center χ as

$$\mathcal{B}_{\mathbb{F}}(\chi, b) := \Big\{\chi' \in \mathbb{F} \mid \mathbf{d}(\chi', \chi) \leq b\Big\} \tag{8.29}$$

and the *small ball probability*

$$P_{\mathcal{B}}(\chi, b) = \mathbb{P}\Big[\mathbf{d}(\chi', \chi) \leq b\Big] = \mathbb{P}\Big[\chi' \in \mathcal{B}_{\mathbb{F}}(\chi, b)\Big]. \tag{8.30}$$

For $\mathbb{F} = \mathbb{R}^q$ and $\mathbf{d}_0$, the topology corresponds to the one of a Euclidean vector space. In $\mathbb{H}$, the ball definition is straightforward using the corresponding norm and includes the case where $\mathbb{H} = \mathbb{R}^n$.

8.2.1.2 Smoothing

Theoretical studies within the functional data analysis operate with continuous functions of time [FV06, FR11, FVKV12]. Yet in practice, we observe the curves $\chi(t), \phi(t)$ at the discrete times within the index set $T := \Big\{t_1, \cdots, t_n\Big\}$. Therefore, we might need to convert discrete observations X, Y into continuous functions of time in order to use a functional model. To do so, there is a wide range of smoothing techniques at our disposal.

A pre-processing smoothing step is required if:

- Time series is *unbalanced* – measured at non-equidistant time points[20]
- Time series is *noisy* – underly a substantial measurement error.
- Time series needs to be *re-sampled* – i.e., sampled on a different time-grid.

[20] See [YMW05] for detailed discussion on the cases where data is sparse and unbalanced.

- Forecaster relies on information not directly available in the original data (e.g., derivatives).

Smoothing technique $\mathbf{s}(X)$ converts discrete measurements to a functional observations (*smooth*) of a form

$$\chi(t) = \mathbf{s}(X) = \sum_{h=1}^{l} c_h \xi_h(t) \tag{8.31}$$

computing the coefficients (*expansion*)

$$C = [c_1, \cdots, c_l]^{\mathrm{T}} \tag{8.32}$$

with respect to the chosen basis

$$\mathcal{S} := \Big\{ \xi_h(t) \mid 1 \leq h \leq l \Big\}, \tag{8.33}$$

that consists of l orthonormal functions $\xi_h(t)$ pre-specified in advance. We need to choose $\mathcal{S}$ considering the problem at hand for which common basis expansions are Fourier, polynomial, splines and wavelets[21].

Smoothing eventually results in a dimensionality reduction. It discards the original measurements and, instead, stores only the coefficients c_h for further processing. The number l of these coefficients and corresponding basis functions $\xi_h(t)$ should be chosen such that $\chi(t)$ resembles the original data eliminating the obvious noise and outliers. While performance of some functional methods can depend on l, most methods are rather insensitive to it [HK12].

A general procedure to find the expansion C representing the curve $\chi(t)$ with respect to the chosen basis $\mathcal{S}$ is as follows. Given a vector of measurement times T and values

$$X = [\chi(t_1), \cdots, \chi(t_n)], \tag{8.34}$$

we compute the *basis matrix*

$$\mathbf{A} := (a_{ih}) \in \mathbb{R}^{n \times l} \text{ with } a_{ih} = \xi_h(t_i) \tag{8.35}$$

such that every measured curve X is represented as

$$X = \mathbf{A}C. \tag{8.36}$$

[21] For instance, if we are to use function derivatives and the data is known to be (near-) periodic, Fourier basis might be an appropriate choice. Alternatively, a basis can also be a data-based orthonormal system e.g., principal component analysis. Consider [RS05] for an extensive discussion on basis expansions in context of functional data analysis and further references therein.

We find C solving the least-squares problem

$$C = \underset{C' \in \mathbb{R}^l}{\arg\min} \left[\sum_{i=1}^{n} \left(\chi(t_i) - \sum_{h=1}^{l} c'_h \xi_h(t_i) \right)^2 \right] \tag{8.37}$$

and calculate C as

$$C = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{-1}\mathbf{A}^{\mathrm{T}}X^{\mathrm{T}}. \tag{8.38}$$

While often used in signal processing, smoothing with various basis expansions is widely available within common statistical software [RHG09].

Practical applications of functional data analysis often rely on smoothing with a basis expansion described above. This procedure converts the measured data into continuous curves $\chi(t), \phi(t)$. Analyzing continuous data allows to derive theoretical results and novel forecasting methods [FV06,FR11,FVKV12]. For instance, it allows to create novel models such as the one we describe next.

8.2.1.3 Functional Nonparametric Model

The functional methodology is more general than the traditional multivariate analysis (Section 4.2.3) since it includes the finite-dimensional techniques for $\mathbb{R}^n$ which is also a Hilbert space. With a basis expansion, functional data analysis also includes the case where $\mathbb{F}$ and $\mathbb{H}$ are infinite-dimensional functional spaces[22] (Section 4.3). Subsequently, we apply the functional methodology to predict $\phi_{j+1}(t)$ with a functional version of locally weighted learning.

Functional nonparametric approach estimates the regression function $\mathbf{r}$ in (8.26). For a given sample of curves

$$\mathcal{T}_{\mathbf{f}} := \Big\{ (\chi_j, \phi_j) \mid 1 \leq j \leq m \Big\} \text{ with } \chi_j, \phi_j \in (\mathbb{F} \times \mathbb{R}^n) \tag{8.39}$$

drawn from a pair of functional valued random variables (χ', ϕ'), we apply *functional nonparametric model* to compute the estimate

$$\hat{\mathbf{r}}(\chi) = \frac{\sum_{j=1}^{m} \theta_{\mathrm{f}} \left(\frac{\mathbf{d}(\chi, \chi_j)}{b} \right) \phi_j}{\sum_{j=1}^{m} \theta_{\mathrm{f}} \left(\frac{\mathbf{d}(\chi, \chi_j)}{b} \right)} \tag{8.40}$$

of the regression function $\mathbf{r}$ using asymmetrical kernel θ_{f} (Definition 4.3.2).

[22] For example, Lebesgue space, sequence space et cetera. [Mus14].

Table 8.2: Differences between multivariate and functional methodologies on the example of a nonparametric model. Further description is provided in the text.

	Multivariate	**Functional**
Data	vectors $X \in \mathbb{R}^q, Y \in \mathbb{R}^n$	continuous curves $\chi(t) \in \mathbb{F}, \phi(t) \in \mathbb{H}$
Distance notion	Euclidean distance $\mathbf{d}_0$	semimetric $\mathbf{d}$
Kernel	Definition 4.2.2	Definition 4.3.2
Topology	$B(X_c, b) := \{X \in \mathbb{R}^q : \mathbf{d}_0(X_c, X) \leq b\}$	$\mathcal{B}_{\mathbb{F}}(\chi_c, b) := \{\chi \in \mathbb{F} : \mathbf{d}(\chi_c, \chi) \leq b\}$
Training set	$\mathcal{T} := \{(X_j, Y_j) \mid 1 \leq j \leq m\}$	$\mathcal{T}_{\mathbf{f}} := \{(\chi_j, \phi_j) \mid 1 \leq j \leq m\}$
Model	$\hat{\mathbf{r}}(X) = \frac{\sum_{j=1}^{m} \theta\left(,\frac{\mathbf{d}_0(X,X_j)}{b(X)}\right) Y_j}{\sum_{j=1}^{m} \theta\left(\frac{\mathbf{d}_0(X,X_j)}{b(X)}\right)}$	$\hat{\mathbf{r}}(\chi) = \frac{\sum_{j=1}^{m} \theta_{\mathrm{f}}\left(\frac{\mathbf{d}(\chi,\chi_j)}{b}\right) \phi_j}{\sum_{j=1}^{m} \theta_{\mathrm{f}}\left(\frac{\mathbf{d}(\chi,\chi_j)}{b}\right)}$
Asymptotics	[HWMS04]	[FV06, FVKV12]

The estimate $\hat{\mathbf{r}}(\chi)$ is the average of the observed outputs ϕ_j for which the respective inputs χ_j are in the *neighborhood*

$$\mathcal{G}_{\chi} := \left\{(\chi_j, \phi_j) \mid (\chi_j, \phi_j) \in \mathcal{T}_{\mathbf{f}}, \chi_j \in \mathcal{B}_{\mathbb{F}}(\chi, b)\right\} \tag{8.41}$$

of the functional input χ. The neighborhood size is controlled by the *bandwidth* b, as with a multivariate nonparametric model (Section 4.2). At the same time, neighborhood shape is dependent on the topological structure of the space $\mathbb{F}$, which is determined by the distance notion $\mathbf{d}$. The choice of the semimetric $\mathbf{d}$ affects the asymptotic behavior of $\hat{\mathbf{r}}$ through the topology described by $\mathcal{B}_{\mathbb{F}}$ and $P_{\mathcal{B}}(\chi)(b)$ [FV06].

For a wide range of semimetrics, the regression function estimate (8.40) is consistent [FVKV12]. Distance notion becomes a tuning parameter of the model controlling the rate at which $\hat{\mathbf{r}}$ converges to the true regression function $\mathbf{r}$. Using the model (8.40), we can now introduce a day-ahead load forecasting algorithm (Section 8.2.1.5). But before, we contrast the differences between multivariate and functional nonparametric models.

8.2.1.4 Differences Between Multivariate and Functional Approaches

What are the fundamental differences between multivariate and functional forecasting methodologies? Indeed, the computations are done with finite dimensional vectors in both cases. To clarify the distinction, we summarize the differences between the nonparametric models in Table 8.2.

The multivariate models are studied using calculus on finite dimensional vectors $X \in \mathbb{R}^q$ that belong to the Euclidean space. Consequently, multivariate approach is limited to this space and ignores the important information about the smoothness (in mathematical sense) and a continuous behavior of the curve. It also suffers from issues associated with highly correlated data within the vectors (Section 4.2.3).

In contrast, functional approach considers objects in an abstract space $\mathbb{F}$ endowed with a semimetric $\mathbf{d}$ and is more general reaching beyond Euclidean calculus. We view X as a noisy representation of a continuous function $\chi(t)$ from an infinite-dimensional space $\mathbb{F}$ and use smoothing together with functional data analysis to derive mathematical concepts and tools (Section 4.3).

Indeed, we have to compute continuous curves given their discrete representations when applying a functional model. To do so, various smoothing techniques are available [RHG09]. Moreover, $\hat{\phi}(t)$ has to be discretized for subsequent processing. In return, a functional model can handle data that is unbalanced, has missing values and is corrupted by noise [Fer11]. Having a functional prediction $\hat{\phi}(t)$, we can re-sample it on a desired time-grid.

Functional nonparametric model (8.40) is an extension of the multivariate nonparametric model (4.62) that uses a generalized distance notion. The most tangible difference is that the shape of the neighborhood is dependent on the topological structure introduced onto functional space $\mathbb{F}$ through the semimetric $\mathbf{d}$. The usage of an abstract distance notion $\mathbf{d}$ instead of the Euclidean distance $\mathbf{d}_0$ is the main practical difference when developing the forecasting algorithm we discuss next.

8.2.1.5 Functional Neighbors Forecasting Algorithm

Functional neighbors (FN) algorithm computes the forecast $\hat{Y}$ using the discussed functional nonparametric regression approach. We introduce an algorithm that applies the functional model (8.40) to the day-ahead local load forecasting problem (Section 7.2.3). For this problem, we are given a training set $\mathcal{T}$ of m time-discrete load observations. An observation of a day j consists of the observed load curve $Y_j \in \mathbb{R}^n$ (output) and its historical predecessor $X_j \in \mathbb{R}^n$ (input). Moreover, input query $X^* \in \mathbb{R}^n$ corresponds to the most recently measured load curve[23]. With these data, we compute the forecast $\hat{Y}$ of the next-day load curve $Y_{m+1} \in \mathbb{R}^n$. Subsequently, we describe each step of the proposed algorithm (Algorithm 3).

Step 1 We apply the smoothing splines to obtain a continuous function from discrete measurements as described in [RS05]. This method is flexible and particularly apt for representing nonperiodic signals with strong local features of the curves (e.g., daily load

[23] The extension to consider further input variables will be provided in Section 8.3.

Algorithm 3: Functional neighbors (FN)

Inputs: most recent curve $X^* \in \mathbb{R}^n$
Outputs: forecast curve $\hat{Y} \in \mathbb{R}^n$
Data: training set $\mathcal{T} := \{(X_j, Y_j) \mid 1 \leq j \leq m\}$
Parameters: distance notion $\mathbf{d}$, number of nearest neighbors K

1 smooth time-discrete measurements: $\rightarrow \mathcal{T}_\mathbf{f}, \chi^*$
2 sort $\mathcal{T}_\mathbf{f}$ by the distance $\mathbf{d}(\chi^*, \cdot)$ from the query χ^*
3 find K-nearest neighbors of the query: $\rightarrow \mathcal{G}_{\chi^*}$
4 merge historical outputs to a consensus representation of $\phi_j \in \mathcal{G}_{\chi^*}$: $\rightarrow \hat{\phi}$
5 re-sample $\hat{\phi}$: $\rightarrow \hat{Y}$

peaks). To compute the continuous function that corresponds to a vector of measurements X and Y (Figure 8.8), we expand it using the B-spline basis obtaining

$$\chi \approx \sum_{i=1}^{n} c_i^{(x)} \xi_i = \mathbf{s}(X), \tag{8.42}$$

$$\phi \approx \sum_{i=1}^{n} c_i^{(y)} \xi_i = \mathbf{s}(Y), \tag{8.43}$$

which can be calculated very efficiently [DBDBM$^+$78]. This step provides us with a functional training set $\mathcal{T}_\mathbf{f}$ and a query χ^*, that are continuous functions and can be subject to the functional data analysis techniques.

Step 2 We compute the distances between χ^* and all input observations in $\mathcal{T}_\mathbf{f}$ to sort them in ascending order accordingly. The distances are computed by a predefined distance notion $\mathbf{d}$ that we discuss in detail in Section 8.2.2.

Step 3 To find the K-nearest neighbors of χ^*, we determine corresponding neighborhood $\mathcal{G}_{\chi^*}$ by selecting the observations whose distance from χ^* is within the variable bandwidth

$$b_K = \mathbf{d}(\chi^*, \chi_K), \tag{8.44}$$

that corresponds to the distance between χ^* and its K'th nearest neighbor χ_K.

Step 4 We merge the outputs of historical observations located in $\mathcal{G}_{\chi^*}$ to a forecast $\hat{\phi}$. For this step, we use a merger function discussed in Section 8.2.3.

Step 5 At last, we output a time-discrete prediction $\hat{Y} \in \mathbb{R}^n$ resampling $\hat{\phi}$ on the desired time-grid.

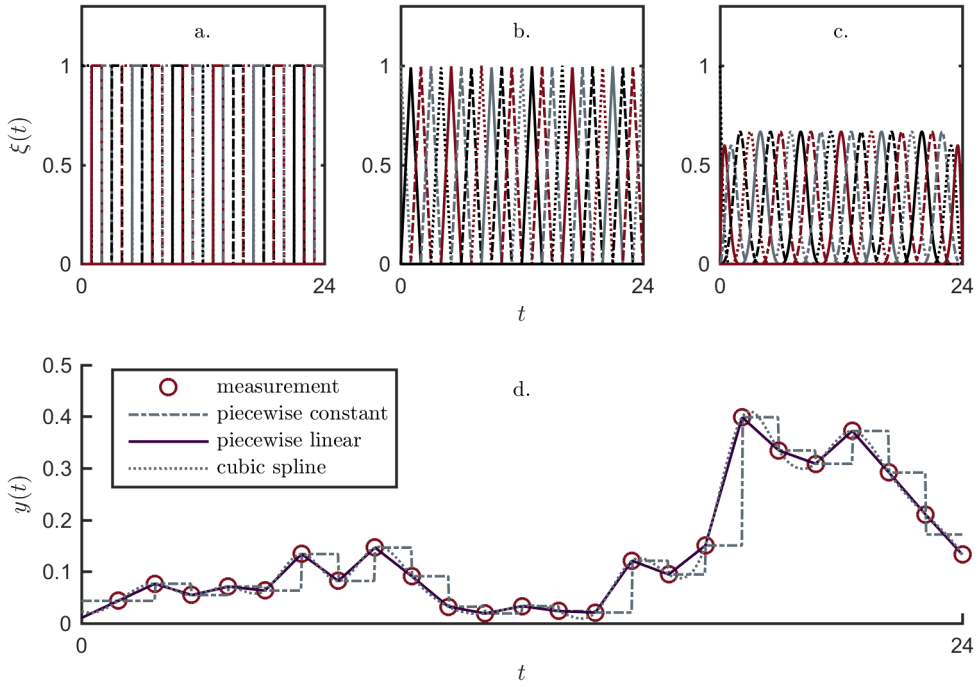

Figure 8.8: Computation of a smooth. Figure demonstrates the computation of a continuous function $y(t)$ from a series of load measurements using B-splines basis functions $\xi_i(t)$ of different orders: (a) step-wise splines – zero-order basis functions; (b) linear splines – first order basis functions; (c) cubic splines – third order basis functions; (d) comparison of a daily load curve to the smoothed curves which were calculated using splines of different order. Here, piecewise constant smooth corresponds to the sample-and-hold technique using a step-wise function (zero-order splines). Piecewise linear smooth corresponds to a simple interpolation joining the points of adjacent observations (first-order splines). Cubic smooth is calculated with splines of the third-order that are among the most commonly used [RS05].

In this study, we assume to have ideal load measurements[24] and use B-splines of zero-order for which the basis functions are given as

$$\xi_i(t) = \begin{cases} 1 & t_{i-1} \leq t < t_i \\ 0 & \text{otherwise} \end{cases} \text{ for } i = 1 \in [1, \ldots, n]$$

With this step-function basis, the coefficients c_i correspond to the measured values $Y(i)$. Herewith, the continuous curves χ^*, χ_j, ϕ_j are uniquely represented by the respective vectors X^*, X_j, Y_j that contain the load measurements. We proceed operating with these unprocessed values, and notate model inputs and outputs, from now on, with X and Y respectively. The disadvantage using zero-order splines is that we have to abstain from using the derivatives in our computations.

[24] Ideal load measurements imply small measurement errors and no missing data.

More generally, we can use B-splines of higher order if any of the algorithm steps rely on computing the derivatives of the smoothed curves[25] or if the measurements are far from being ideal. Using smoothing splines in our model facilitates its practical applications where the data is often unbalanced and has substantial share of measurement errors or missing values. The literature provides a detailed exposition on how a smooth can be computed under these conditions [DBDBM$^+$78, RS05]. While with step-wise splines there is no additional computation required, the smooth of a higher order can be routinely computed with a common statistical software [RHG09].

Additionally, variable bandwidth makes the model more robust when working with non-stationary data. The fixed bandwidth alternative results in a fixed-sized neighborhood that can become increasingly empty with growing dimensionality of the data (Section 4.2.3.2). In the worst case, an empty neighborhood might result in an undefined prediction. Moreover, it is hard to find the optimal neighborhood size if it is determined by a continuous variable b. In contrast, variable bandwidth b_K, determined by the number of neighbors K, allows the model to adapt to the data distribution, its heteroscedasticity as well as the changes in smoothness or curvature of the regression function [FM94].

8.2.2 Finding Nearest Neighbors

In order to find the nearest neighbors of X^* we have to calculate the distance between X^* and historical inputs in $\mathcal{T}$ according to a predefined distance notion $\mathbf{d}$. The choice of the semimetric can affect the asymptotic behavior of the model (8.40) in a notable way (Section 4.3). Therefore, we have to investigate how such a choice will affect the quality of the forecast provided by the Algorithm 3.

The role of $\mathbf{d}$ is related to the previously described curse of dimensionality (Section 4.2). From a practical point of view, we can experience the dimensionality issue by the growing sparsity of the data in the vicinity of X^* with increasing vector dimension q. Evidently, this problem must be addressed in the infinite-dimensional context of functional data. In fact, research shows that it is possible to construct a semimetric[26] so that the functional nonparametric model has a similar or superior rate of convergence to the multivariate model (4.61) if the data within the curves are correlated [FV06]. For a wide range of semimetrics, curse of dimensionality can be effectively addressed.

[25] For example, we can use a distance notion that considers the derivatives of χ as proposed in [FV06].
[26] The usage of semimetrics instead of metric appears to be particularly important [FV06].

8.2.2.1 Data Sparsity in the Load Curves Space

Performance of the functional neighbor model depends on the data concentration that characterizes the stochastic process generating the data (Section 4.3.3). Assuming the process is fixed, topology and with it the concentration properties of the observation space can be altered changing $\mathbf{d}$.

Multivariate models are often set up with uncorrelated explanatory variables (Chapter 5). For the MKNN in particular, curse of dimensionality is a major limitation that has to be compensated by an un-proportionally increase of training data amount (4.76). At the same time, correlated inputs allow a dimensionality reduction[27]. As we saw, having less dimensions allows Euclidean distance to be more discriminative and improves the performance of the model (4.70).

Alternatively, in a functional setting we can make the data denser by a wise choice of the distance notion that explores the correlation within the curves. Researchers have observed that the space of highly correlated curves can be made denser depending on the choice of $\mathbf{d}$ to a point, where growing data sparsity connected to the curse of dimensionality is partially or fully canceled [FV06].

Consider the Euclidean space $(\mathbb{R}^q, \mathbf{d}_0)$ of q-dimensional vectors that represent the daily load curves which we can commonly encounter in our problem. We study the concentration, counting the curves as a relative number of observations

$$N(q) = 100\% \cdot \frac{1}{m} \sum_{j=1}^{m} \mathbf{1}_{\left\{ \frac{\mathbf{d}(\bar{X}, X_j)}{\max_j \mathbf{d}(\bar{X}, X_j)} < b \right\}}, \tag{8.45}$$

that indicates which percentage of the observed curves is encountered in the ball $\mathcal{B}_{\mathbb{F}}\left(\bar{X}, b\right)$ that is centered around the average curve $\bar{X}$. Further, we define the autocorrelation coefficient

$$a = \gamma_{X,X}(1) = \frac{\sum_{t=2}^{T} \left(y_t - \mathbb{E}\Big[y_t\Big]\right)\left(y_{t-1} - \mathbb{E}\Big[y_{t-1}\Big]\right)}{\sum_{t=1}^{T} \left(y_t - \mathbb{E}\Big[y_t\Big]\right) \sum_{t=2}^{T} \left(y_{t-1} - \mathbb{E}\Big[y_{t-1}\Big]\right)}, \tag{8.46}$$

that expresses the extent to which the load time-series y_t is correlated with itself.

In the Euclidean space, vectors with uncorrelated data rapidly become more sparse as we increase their dimensionality (Figure 8.9). We observe this effect for the synthetic example of a vector with uniformly distributed random variables ($a = 0$) which demonstrates the curse of dimensionality (Section 4.2). In practice, the load curve of a large load

[27] Most relevant inputs can either be selected manually or using an algorithm that finds the optimal linear or nonlinear combination of original inputs.

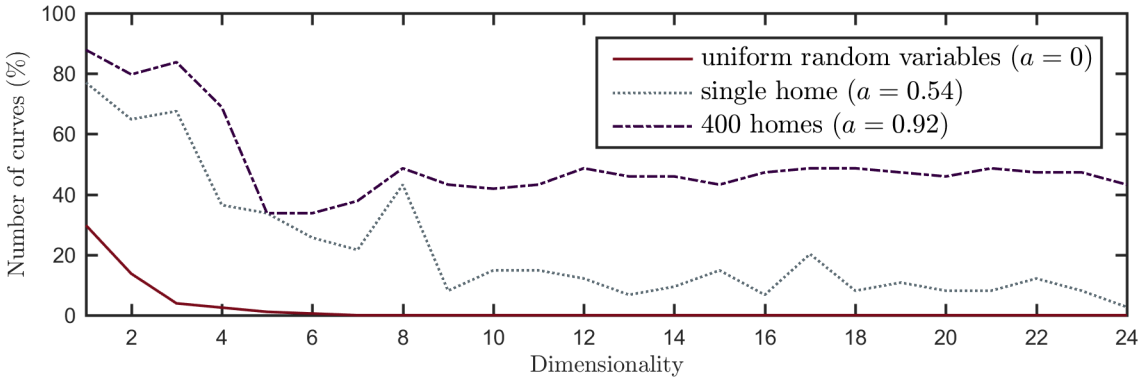

Figure 8.9: Load curve sparsity in a multidimensional Euclidean space. In this space, each point is a q-dimensional vector that can represent a time series. For this example, we considered three different sets of time series represented by the vector in the Euclidean space: uniformly randomly-generated time series, daily load curves of a single home, load curves of a larger residential aggregation (400 homes). In each set, the time series feature a different degree of autocorrelation expressed by the first autocorrelation function coefficient a. For each set, we computed the average curve and counted the number of observations in a ball with radius $b = 0.3$ centered at the vector corresponding to the average curve. Resampling the curves with various resolution, we show how the sparsity of the observations varied with dimensionality of the space. Observe, that for uncorrelated data, the sparsity rapidly decreases with the dimensionality. At the same time, the density of the highly correlated data (400 homes aggregation) remained stable despite the dimensionality increase.

aggregation is highly autocorrelated (e.g., 400 homes with $a = 0.92$ in Figure 8.9). In such case, data density remains stable despite the dimensionality increase. Evidently, curse of dimensionality becomes partially canceled. This explains why, despite dimensionality issue, nonparametric models still yield acceptable results in various applications discussed previously (Chapter 5).

Nevertheless, the sparsity is still notable for the load curves that are less correlated such as those of a single home ($a = 0.54$). In this case, the performance of a nonparametric model in terms of maximal ROC (4.70) deteriorates quickly with growing time-series resolution (dimensionality). In practice, we can expect such model to yield less accurate prediction for disaggregated loads when increasing smart-meter resolution.

More generally, the sparsity of data depends on the measure of closeness between the observations (8.45). Intuitively, by an appropriate choice of such measure, we can increase the data density and improve the rate of convergence of the model. Therefore, we proceed discussing the choice of a distance notion for the functional neighbor model.

8.2.2.2 Distance Notion Design

The main idea of nonparametric learning is that similar inputs are likely to produce similar outputs. Therefore, we are foremost interested in similarity between the curves rather than a point-wise distance of the corresponding vectors. Consequently, we want to choose a

distance notion that reveals more pertinent information of the curve. A suitable choice of $\mathbf{d}$ can change the concentration property of $\mathbb{F}$ and improve the rate of convergence addressing the curse of dimensionality.

Note that in $\mathbb{R}^q$ there is an equivalence between all norms including the Euclidean distance [FV06]. In a strict mathematical sense, the distance notion choice is not important for the multivariate model, apart from some practical constraints (e.g., computation time). In contrast, within an infinite-dimensional space $\mathbb{F}$, the equivalence between the norms fails and even the usage of a norm as a distance notion appears too restrictive [FV06]. The usage of a semimetric and its choice becomes fundamental through the equations (4.86), (4.87), and (4.88) which describe the fact that an estimation of a point $\chi \in \mathbb{F}$ needs a sufficiently large number of data around it (i.e., concentration).

We introduce the following corollary to justify the usage of distance notions other than the Euclidean distance $\mathbf{d}_0$ (4.63) in an infinite-dimensional space.

Corollary 8.2.1. Consider an infinite-dimensional space of curves $\mathbb{F}$ with an infinite-dimensional basis $\mathcal{E} := \{\xi_i, i \in \mathbb{N}^+\}$ so that

$$\forall \chi \in \mathbb{F}, \chi(t) \approx \sum_{i=1}^{q'} X(i)\xi_i = \sum_{i=1}^{q'} \langle \chi(t), \xi_i \rangle \xi_i, \tag{8.47}$$

where the *inner product*

$$\langle \chi_1, \chi_2 \rangle = \sqrt{\int_T \chi_1(t)\chi_2(t)dt} \tag{8.48}$$

projects the curve χ onto $\mathcal{E}$. The rate of convergence of the nonparametric model (8.40) applied in such space is limited by

$$\text{ROC} = (\frac{\log m}{m})^{\frac{s}{2s+q'}} \tag{8.49}$$

if the distance notion is defined as a semimetric of a form

$$\mathbf{d}(\chi_1, \chi_2) = \sqrt{\sum_{i=1}^{q'} \langle \chi_1 - \chi_2, \xi_i \rangle^2}. \tag{8.50}$$

Proof. The proof follows directly from Lemma 13.6 and Proposition 13.2 in [FV06]. □

Being itself a semimetric, such a *projection-type of distance* (8.50) returns the ℓ^2-norm of the projection of χ onto the q'-dimensional basis $\mathcal{E}$. Therefore, we can view the $\mathbf{d}_0$ as a distance notion projecting continuous curves onto a *standard basis* of q-dimensions with $\text{ROC} = \text{ROC}_{\text{MNWE}}$ (4.70). If we can find a basis for our data that aptly uses only $q' < q$

dimensions to approximate the observations in $\mathbb{F}$, we obtain functional nonparametric model for whose maximal rate of convergence $\mathrm{ROC}_{\mathrm{FNWE}}$ holds the following:

$$\mathrm{ROC}_{\mathrm{FNWE}} = (\frac{\log m}{m})^{\frac{s}{2s+q'}} > \mathrm{ROC}_{\mathrm{MNWE}}. \tag{8.51}$$

Therefore, finding a suitable basis to represent the observations and using the corresponding semimetrics is an opportunity for improving our model and gives potential for further developments of functional nonparametric models in general.

There are numerous widely known methods to compute a basis $\mathcal{E}$ for a set of curves such as *principle component analysis (PCA)*, polynomial basis, B-splines, wavelets, Fourier transform etc. Given an expansion in $\mathcal{E}$, we can always use a distance notion that truncates the projection using fewer dimensions. However, this would also change the topology of $\mathbb{F}$ such that the estimated regression operator might become less smooth, and the condition (4.83) can only be satisfied with a smaller s. Therefore, there is a trade-off (8.49) between reducing effective dimensionality q' and smoothness s.

It is hard to find a suitable semimetric analytically for a problem at hand. Currently, research suggests that the choice must be guided by practical considerations for the given task [FV06, Gee11, FR11]. Some known aids to select the semimetric can be: parametrizing some family of semimetrics (e.g., PCA) by an integer [FV04, FGV02] or using a functional single index model (i.e., adaptive semimetric) [FPV03, AFKV08, FGSV13].

For our load forecasting problem:

- We want to choose a distance notion that reveals the most pertinent information of the load curves – occurrence of peaks and their size.
- We are foremost interested in similarity between the curve shapes rather than a point-wise distance.
- The time-series characteristics, in particular the degree of smoothness, can be very different depending on load aggregation level.

For the functional neighbor model described in this chapter, we propose to use the distance notion described below.

8.2.2.3 Permuted ℓ^2_{u} Distance Notion

Following the same rationale as in the context of forecast evaluation (Section 7.3), we define the ℓ^2_{u}*-distance* with the same *permuted* ℓ^2*-semimetric* that we described previously

for evaluating the forecasts [HWVG+14]. Given that X_k and X_j are vectors representing the daily curves, the distance

$$\mathbf{d}_u(X_k, X_j) := \min_{\pi \in \mathcal{P}(X_j,u,q)} \sqrt{\sum_{i=1}^{q} \left|\pi_{X_j}(i) - X_k(i)\right|^2} \tag{8.52}$$

is calculated allowing small time-permutations $\pi_{X_j}(i)$ of each point $i = 1, \ldots, q$ of X_j, while comparing the time series and penalizing the amplitude differences using the ℓ^2-norm. Here, $\mathcal{P}(u, q)$ is the set of all u-local permutations on q-points of the vector X_j. Such permutations are defined as follows.

Definition 8.2.1. A vector $\pi \in \mathcal{P}(X, u, q)$ is a *u-local permutation* of a vector X if it was obtained rearranging the coordinates $X(1), \ldots, X(q)$ by moving each one forwards or backwards[28] by up to u-time units.

Out of all u-local permutations, a minimal cost local permutation π_X solves the corresponding combinatorial optimization problem defined as follows.

Definition 8.2.2. *Minimal cost local permutation (MCLP)* problem consists in finding a permutation[29] $\pi \in \mathcal{P}(X, u, q)$ that minimizes the cost function of a form

$$\mathrm{Cost}(\pi) := \sum_{i=1}^{q} \mathrm{C}\left(i, \pi(i)\right),$$

where $\mathrm{C}(i_1, i_2)$ is the cost of mapping the point at i_1 to the point $\pi(i)$.

We can solve the MCLP-problem using the graph-based method in $\mathcal{O}(q)$ time [CGS13a]. The MCLP-problem is a special case of the fundamental *assignment problem* that is commonly solved using Hungarian algorithm [JV86] in polynomial time $\mathcal{O}(q^3)$ [HWVG+14]. The graph-based method reduces the assignment problem to computing the shortest path in a directed acyclic graph.

In general, the ℓ^2_{u}-distance is a semimetric since it does not fulfill the triangle inequality. Nevertheless, it complies with the requirements on a distance notion (Definition 4.3.1). Further, this distance notion is parametrized by u which allows for automated search of the best distance notion for a given load time-series as we discuss further in the text. Note that for $u = 0$, the equation (8.52) corresponds to the Euclidean distance (4.63) that is commonly used with multivariate nonparametric models [AC13].

[28] Every point can be moved only once so that it can be found at the maximum distance u from its original location.

[29] Note that the MCLP-problem allows for several solutions.

The ℓ^2_{u}-distance is of projection type. Allowing to permute single coordinates of each vector reduces the dimensionality of the observation space. Increasing the range of allowed permutations makes the space smaller and denser (Figure 8.10). At the same time, the corresponding density function becomes less smooth. There is the aforementioned trade-off between the effective dimensionality q' and the smoothness s determining the rate of convergence of the model (8.49).

In practice, increasing permutation range of the distance notion applied with the functional neighbor model results in a growing spread of the daily forecast errors (Figure 8.11). The forecaster becomes less consistent in its accuracy. With increasing u, we see more outliers, which leads to a lower overall improvement relative to the Euclidean distance. This is particularly notable for homes and aggregations.

Empirically, we have found that limiting allowed permutations to one hour (i.e., $u = 1$ given hourly resolution), results in the largest accuracy improvement (Figure 8.12). On the vast majority of days, the $\mathbf{d}_{\mathrm{u}=1}$-notion results in a more accurate forecast than with the $\mathbf{d}_0$-notion. It does so, on the larger percentage of forecast days than the distance notions with larger permutation range.

Nevertheless, we cannot generalize assuming that $\mathbf{d}_{\mathrm{u}=1}$ is a better distance notion for the functional neighbors forecaster[30]. If we directly compare the EDE obtained with $\mathbf{d}_0$ and $\mathbf{d}_{\mathrm{u}=1}$ distance notions, we see that only about half of the loads were predicted more accurately with $\mathbf{d}_{\mathrm{u}=1}$-notion (Figure 8.13). This indicates that for some loads and days, $\mathbf{d}_{\mathrm{u}=1}$-notion was inadequate to quantify the similarity. Hence, if we choose the distance notion depending on the load and weekday, we can notably improve the forecast (Figure 8.13). Therefore, the best distance notion choice for the forecaster depends on the load and day-type we predict.

[30] In our case, we consider the time series with the hourly resolution. For other resolutions, different values of u might be advisable.

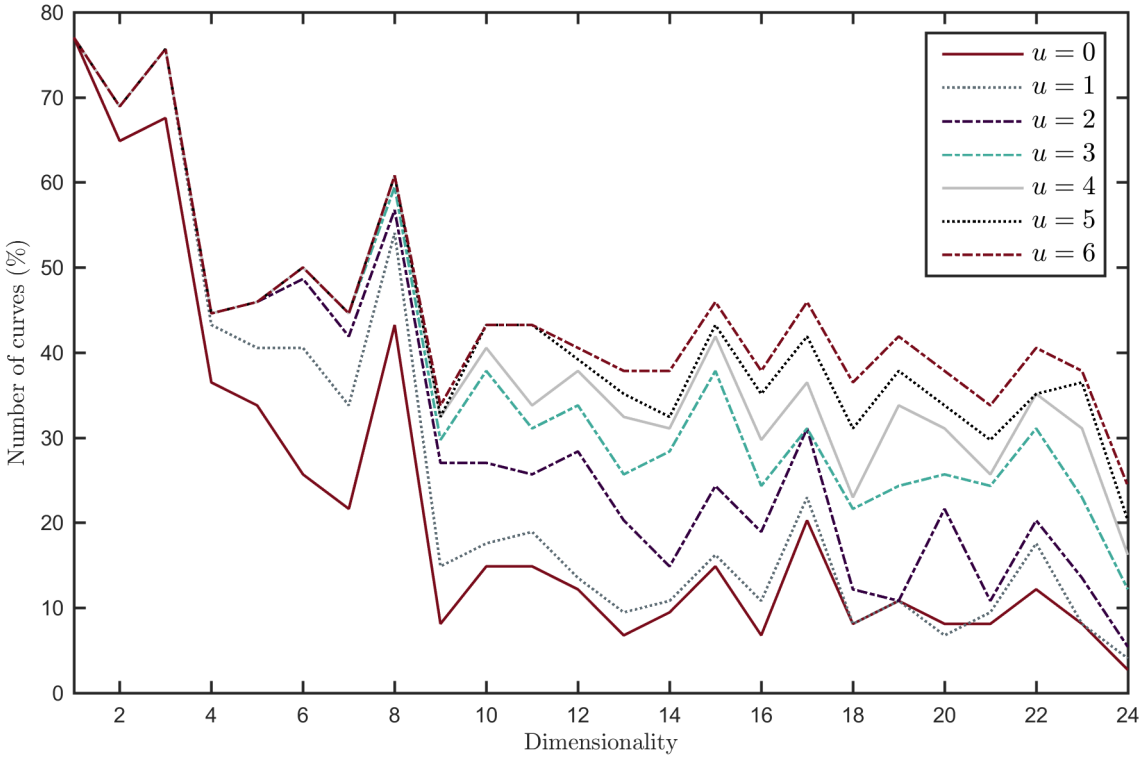

Figure 8.10: Change in data-sparsity in $(\mathbb{F}, \mathbf{d}_{\mathrm{u}})$. The space of daily load curves $(\mathbb{F}, \mathbf{d}_{\mathrm{u}})$ includes 153 observations collected for a single family home (ICER dataset discussed in Section 9.1) and is endowed with a distance notion $\mathbf{d}_{\mathrm{u}}$ based on the permuted ℓ^2-semimetric (8.52). To quantify data-sparsity, we computed the average curve $\bar{X}$ and counted the curves located in the ball $\mathcal{B}_{\mathbb{F}}\left(\bar{X}, b\right)$ centered at $\bar{X}$. On the figure, we denote the % of the curves whose distance from $\bar{X}$ was less than $b = 0.3$. Resampling the curves with various resolutions, we show how the observation sparsity in $\mathbb{F}$ varies with dimensionality of the vectors that would represent the time series in a Euclidean space. Additionally, the sparseness depends on the dimensionality of the basis $\mathcal{E}$ which changes depending on the permutation range u of the $\mathbf{d}_{\mathrm{u}}$-distance notion. Observe, that the sparsity decreases with permutation range at every dimensionality. Therefore, by increasing the permutation range, we can make reduce the sparsity of the observation space.

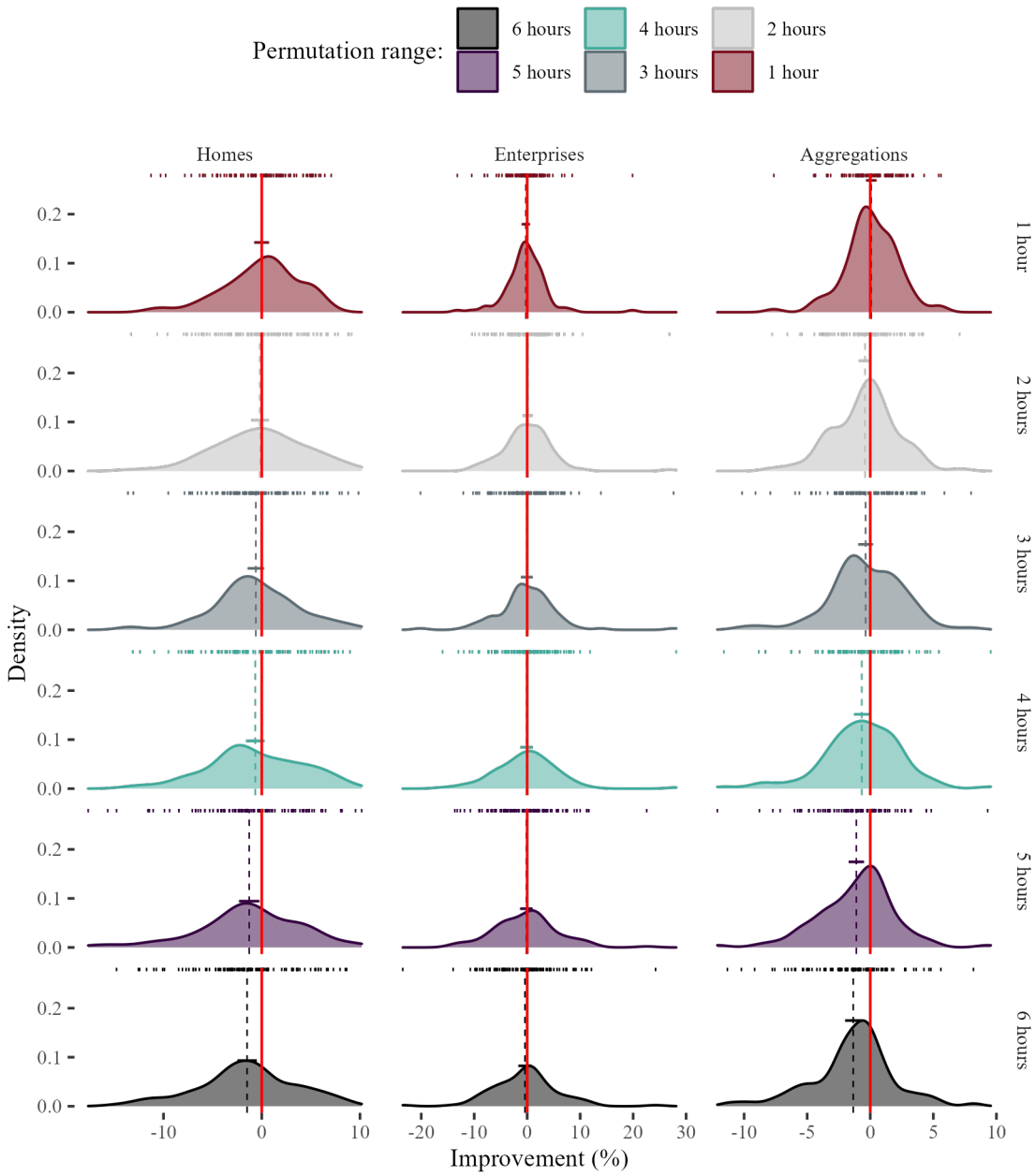

Figure 8.11: Forecast improvement with the ℓ^2_{u}-distance. In a validation experiment (Section 9.3.1.1), we applied the functional neighbor forecaster (Algorithm 3) using the $\mathbf{d}_{\mathrm{u}}$-distance notion (8.52) to predict the 300 loads of different groups obtaining a sample of 30000 daily forecast errors. For the distance notion, we used different permutation ranges (1 hour to 6 hours, given hourly resolution of the time series). Additionally, we applied the functional neighbor forecaster with ℓ^2-distance to predict the same loads and used these results as a benchmark. Relative to this benchmark, we computed the forecast improvement (7.14) for each predicted daily load curve. In the figure, every panel presents the sampling distribution of the mean improvement for each load (rugs at the top), expected improvement in the load group (dotted vertical line) with the 95%-confidence interval (horizontal bar), and the zero-improvement mark (red vertical line). Notably, increasing the permutation range resulted in a growing spread of the improvement observations. Numerous outliers lowered the overall improvement which was particularly notable for homes and aggregations.

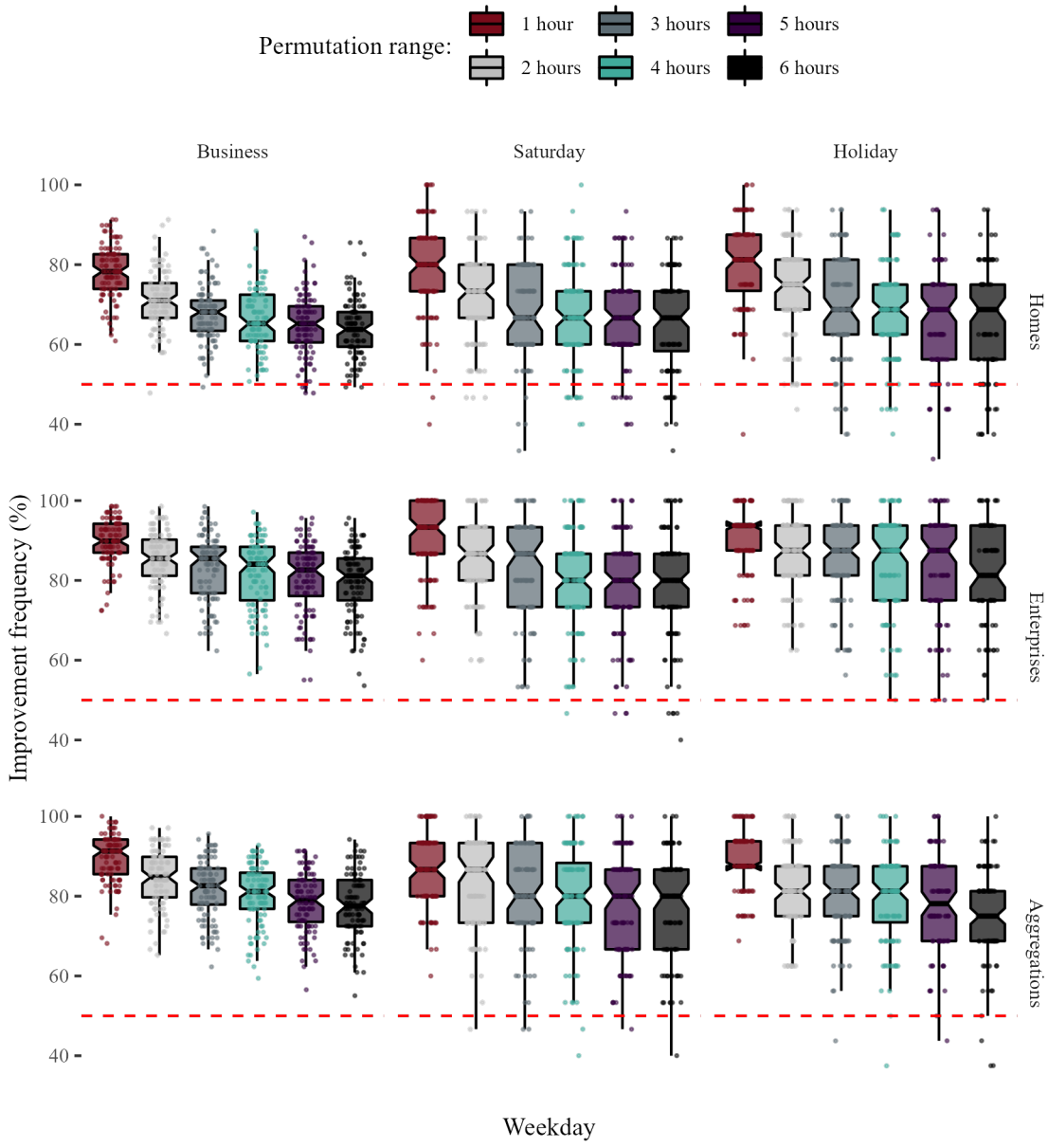

Figure 8.12: Selecting the permutation range for the $\mathbf{d}_{\mathrm{u}}$-distance notion. In a validation experiment (Section 9.3.1.1), we applied the functional neighbor model (Algorithm 3) using the $\mathbf{d}_{\mathrm{u}}$-distance notion (8.52) with various permutation ranges (1 hour to 6 hours, given hourly resolution of the time series) to predict the 300 loads of different groups obtaining a sample of 30000 daily forecast errors for each model variant. Additionally, we predicted the same loads using the same model with the ℓ^2-distance considering the results as a benchmark. Relative to this benchmark, we computed the forecast improvement (7.14) for each predicted daily load curve. Conditioned on load group and day-type, each panel shows the percentage of predicted daily load curves where the usage of ℓ^2_{u}-distance resulted in a forecast at least as accurate as when using the ℓ^2-distance. In each panel, the distribution is summarized with a box-plot where the notch denotes the 95%-confidence interval of the median and red dotted line denotes the 50% improvement frequency mark. On average, $\mathbf{d}_{\mathrm{u}=1}$-distance notion improved the forecast with the ℓ^2-distance on more days than any other distance notion with larger permutation range.

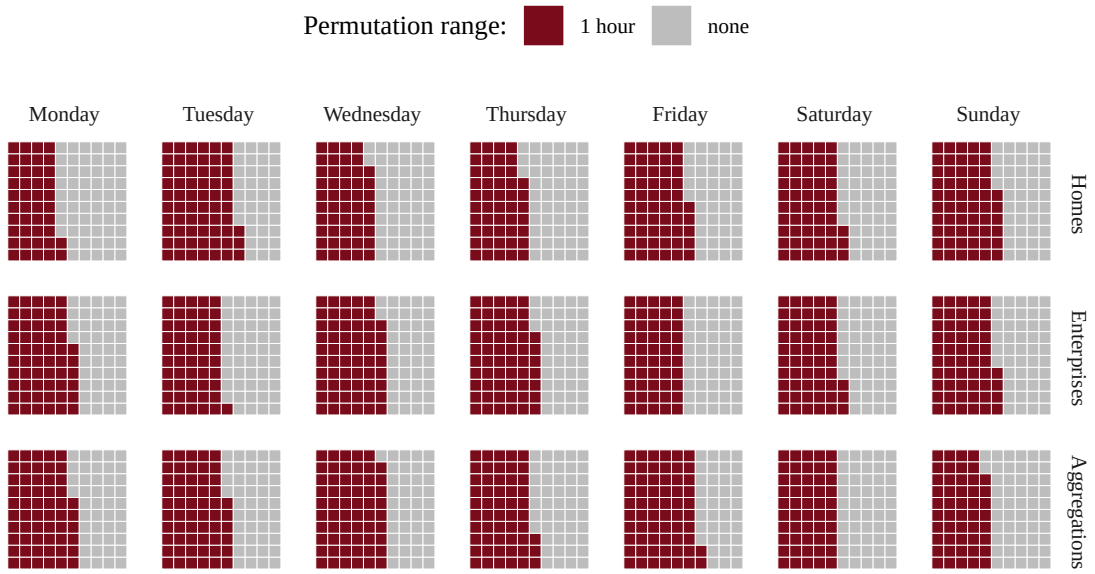

Figure 8.13: Distance notion comparison by load. In a validation experiment (Section 9.3.1.1), we applied the functional neighbor model (Algorithm 3) using the Euclidean (4.63) distance (no permutations) and the $\mathbf{d}_{\mathrm{u}=1}$-distance notion (1 hour permutation range according to (8.52)) to predict the 300 loads of different groups. Conditioning on load type (panel row) and weekday (panel column), we represent each individual load by a square filled depending on the model that provided the smallest expected daily error (7.15) on the days of the corresponding weekday. Notably, there was a similar number of loads where each of the notions delivered the most accurate forecast.

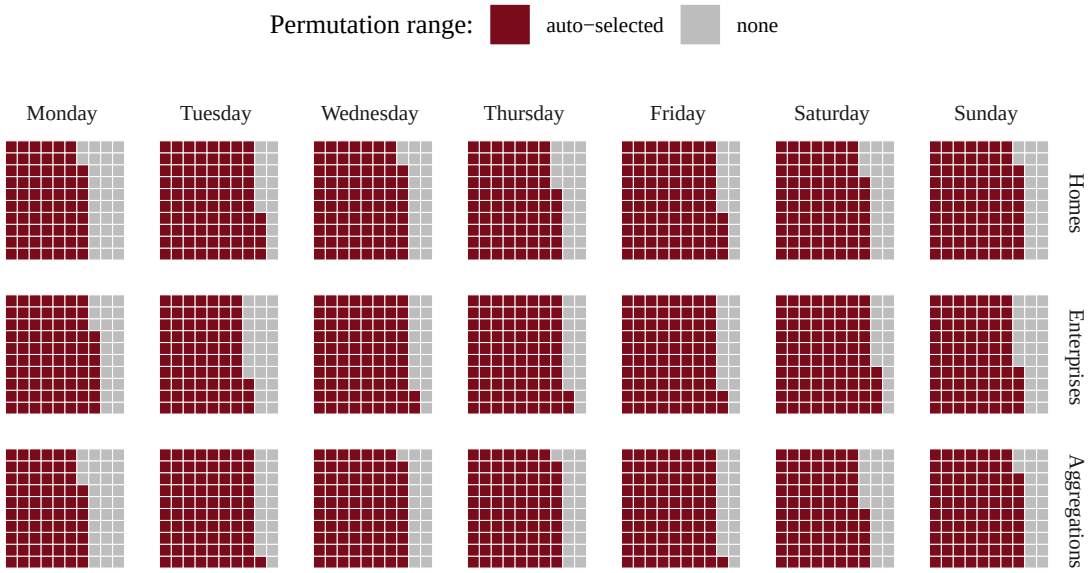

Figure 8.14: Comparison of the distance notion with auto-selected permutation range. In a validation experiment (Section 9.3.1.1), we applied the functional neighbor model (Algorithm 3) using the ℓ^2-distance (no permutations) and ℓ^2_{u}-distance to predict the 300 loads of different groups. The permutation range ($u \in \{0, 1\}$) for the ℓ^2_{u}-distance was selected automatically for the given load using leave-one-out cross-validation prior to the forecast. Conditioning on load type (panel row) and weekday (panel column), we represent each individual load by a square filled depending on the distance notion that provided the smallest expected daily error (7.15) on the days of the corresponding weekday. Notably, the ℓ^2_{u}-distance notion provided the most accurate forecast for the vast majority of loads.

8.2.3 Merging Historical Outputs

Local learning provides a forecast merging historical observations to a consensus representation. Such a representation can be seen as the center among the observations with respect to the chosen distance notion. Generalizing, the concept of a center, we introduce the following definition.

Definition 8.2.3. *Center* of a set of curves $\mathcal{F}$ is a curve

$$Y_c = \arg\min_{v} \sum_{Y_j \in \mathcal{F}} \theta_j \left[\mathbf{d}(v, Y_j,)\right]^2, \tag{8.53}$$

that minimizes the weighted *sum of squared distances (SSD)*

$$\mathrm{SSD}(y) = \sum_{j=1}^{K} \theta_j \Big(\mathbf{d}(Y_j, y) \Big)^2 \tag{8.54}$$

to other curves in $\mathcal{F}$.

For instance, nonparametric models commonly find the center of the vectors in $\mathcal{G}_{X^*}$ by computing their (weighted) average [HLC+97b]. Using the Euclidean distance notion $\mathbf{d}_0$ (4.63), a weighted average of the curves in $\mathcal{F}$ corresponds to the center of $\mathcal{F}$ according to the Definition 8.2.3. In the context of functional neighbor forecasting methodology, we view the average computation as an example of a merger defined as follows.

Definition 8.2.4. *Merger* is a function $\mathbf{m}_c : \mathbb{H} \times \cdots \times \mathbb{H} \to \mathbb{R}^n$ that finds the center of a functional dataset $\mathcal{F}$ with respect to a distance notion $\mathbf{d}$.

Applying a distance notion, beyond the Euclidean distance, allows to compute the forecast $\hat{Y} = Y_c$ (Step 4 in Algorithm 3) with various mergers discussed in this section[31]. We begin considering the standard average-based mergers (Section 8.2.3.1) that find the center with respect to $\mathbf{d}_0$ (ℓ^2-distance). With these mergers, we compute the forecast as a weighted average of historical observations assigning the weights according to a predefined kernel function. Subsequently, we discuss the permutation merger (Section 8.2.3.2) that finds the center with respect to $\mathbf{d}_{\mathrm{u}}$ but does not weight the observations. At last, we propose the weighted permutation merger (Section 8.2.3.3) and compare it to the aforementioned approaches when used with the functional neighbor model predicting the load curves day-ahead.

[31] For what follows, we are given a set of K vectors $Y_j \in \mathbb{R}^n, j \in \{1, \ldots, K\}$ representing load curves and, following the nonparametric approach (Algorithm 3), we compute the forecast as $\hat{Y} = Y_c$ as the center among relevant historical outputs Y_j.

8.2.3.1 Average-Based Merger

Average-based mergers are standard for the conventional nonparametric models used in load forecasting [AC13]. The observations in $\mathcal{G}_{X^*}$ are merged computing the average that corresponds to the center of $\mathcal{G}_{X^*}$ with respect to the Euclidean distance (4.63). Additionally, the observations can be assigned individual weights θ_j such that

$$\sum_{j=1}^{K} \theta_j = 1 \text{ with } \theta_j \geq 0 \text{ for } j \in \{1, \ldots, K\}.$$

The weighted average of the curves in $\mathcal{G}_{X^*}$ minimizes the weighted SSD as we demonstrate by the following theorem.

Theorem 8.2.1. *Weighted average*

$$\bar{Y}_\theta = \sum_{j=1}^{K} \theta_j Y_j \tag{8.55}$$

is the center (Definition 8.2.3) with respect to the ℓ^2-distance ($\mathbf{d}_0$ distance notion).

Proof. We begin by expressing ℓ^2-distance in terms of inner-product and use its properties:

$$\begin{aligned}
\hat{Y} &= \arg\min_{v \in \mathbb{R}^n} \sum_{j=1}^{K} \theta_j \Big(\mathbf{d}_0(Y_j, v) \Big)^2 = \arg\min_{v \in \mathbb{R}^n} \sum_{j=1}^{K} \theta_j \langle Y_j - v, Y_j - v \rangle \\
&= \arg\min_{v \in \mathbb{R}^n} \sum_{j=1}^{K} \theta_j \Big[\langle Y_j, Y_j \rangle - 2\langle Y_j, v \rangle + \langle v, v \rangle \Big] \\
&= \arg\min_{v \in \mathbb{R}^n} \Big[\sum_{j=1}^{K} \theta_j \langle Y_j, Y_j \rangle - 2 \sum_{j=1}^{K} \theta_j \langle Y_j, v \rangle + \sum_{j=1}^{K} \theta_j \langle v, v \rangle \Big].
\end{aligned}$$

Note that $\sum_{j=1}^{K} \theta_j \langle Y_j, Y_j \rangle$ is a constant term and does not affect the solution. Thus, $\hat{Y}$ can be computed as:

$$\begin{aligned}
\hat{Y} &= \arg\min_{v \in \mathbb{R}^n} \Big[\sum_{j=1}^{K} \theta_j \langle v, v \rangle - 2 \sum_{j=1}^{K} \theta_j \langle Y_j, v \rangle \Big] \\
&= \arg\min_{v \in \mathbb{R}^n} \Big[\sum_{j=1}^{K} \theta_j \langle v, v \rangle - 2 \langle \sum_{j=1}^{K} \theta_j Y_j, v \rangle \Big] \\
&= \arg\min_{v \in \mathbb{R}^n} \Big[\langle v, v \rangle - 2 \langle \bar{Y}_\theta, v \rangle \Big].
\end{aligned}$$

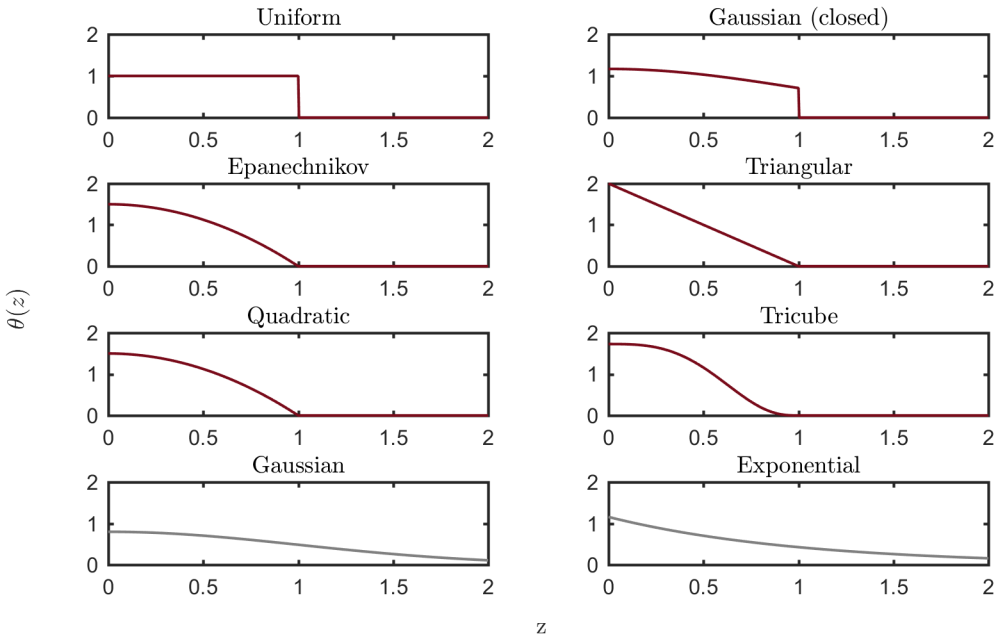

Figure 8.15: Kernel functions defined in Table 8.3.

We denote the weighted average (8.55) as $\bar{Y}_\theta$ and add a constant term $\langle \bar{Y}_\theta, \bar{Y}_\theta \rangle$ into the minimized expression. Herewith,

$$\begin{aligned}\hat{Y} &= \arg\min_{v \in \mathbb{R}^n} \left[\langle v, v \rangle - 2\langle \bar{Y}_\theta, v \rangle + \langle \bar{Y}_\theta, \bar{Y}_\theta \rangle \right] \\ &= \arg\min_{v \in \mathbb{R}^n} \langle v - \bar{Y}_\theta, v - \bar{Y}_\theta \rangle \\ &= \arg\min_{v \in \mathbb{R}^n} \mathbf{d}_0(v, \bar{Y}_\theta),\end{aligned}$$

for which $v = \bar{Y}_\theta$ is the solution. □

Calculating the average curve is trivial and uniform or weighted averaging are the standard mergers for nonparametric models [AC13]. Weighting the observations depending on their distance can significantly improve the nonparametric forecast as we will see further in the text. While the weights are usually calculated using Gaussian kernel, any kernel function (Definition 4.3.2) can be used for this purpose.

Some of the most common kernels that we have encountered in statistical literature [AMS97, HWMS04] are summarized in Table 8.3 and illustrated in Figure 8.15. We can distinguish between two types of kernels:

- *Closed kernels* only consider observations within the bandwidth to which they assign weights $\theta > 0$ while discarding ($\theta = 0$) the observations elsewhere.

Table 8.3: Asymmetrical kernel functions.

Kernel	Function
Uniform	$\mathbb{1}(0 \leq z \leq 1)$
Gaussian (closed)	$1.37\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}z^2)\mathbb{1}(0 \leq z \leq 1)$
Epanechnikov	$\frac{3}{2}(1-z^2)\mathbb{1}(0 \leq z \leq 1)$
Triangular	$2(1-\lvert z\rvert)\mathbb{1}(0 \leq z \leq 1)$
Biweight	$\frac{15}{8}(1-z^2)^2\mathbb{1}(0 \leq z \leq 1)$
Triweight	$\frac{35}{16}(1-\lvert z\rvert^3)^3\mathbb{1}(0 \leq z \leq 1)$
Gaussian	$\frac{2}{\sqrt{2\pi}}\exp(-\frac{1}{2}z^2)\mathbb{1}(0 \leq z)$
Exponential	$3.46\exp(-z)\mathbb{1}(0 \leq z)$

- *Open kernels* do not discard the observations outside of bandwidth and instead assign very small weights to them.

We observed that the nonparametric models using closed kernels were significantly more accurate than when using the open kernels (Figure 8.16). In theory, when considering asymptotic properties of the models (i.e., assuming infinite observations), all kernels are almost equivalent [HWMS04]. In practice, indeed, models with closed kernels had comparable accuracy among them. At the same time, there was a notable difference between open and closed kernel types. For some loads, the recency of observations used for the forecast notably affects the accuracy. Consequently, the difference between the kernel types was particularly notable on the loads with a pronounced annual cycle (enterprises, aggregations) that often feature a strong concept drift (Section 7.1.1.1). A model with an open kernel assigns small weights to the observations outside of $\mathcal{G}_{X^*}$ instead of discarding them. As a result, nonparametric models with closed kernel, considering only the most relevant observations located in $\mathcal{G}_{X^*}$ were significantly more accurate.

Consider the improvement of the functional neighbor model through using triangular and Gaussian kernels relative to the uniform average (Figure 8.17). We observed that the Gaussian kernel, not only provided no improvement against the uniform average, but reduced the accuracy (on average down to 30%). This is somewhat surprising since (open) Gaussian kernel is the most common kernel for the nonparametric models (Chapter 5). Though often applied without any justification in related works, we observed that the Gaussian kernel might not be the best choice for our nonparametric model. Alternatively, triangular kernel appears to be a better choice. It corresponds to weighting observations proportionally to their distance to the query and results in a significantly ($p < 0.05$) more accurate forecast.

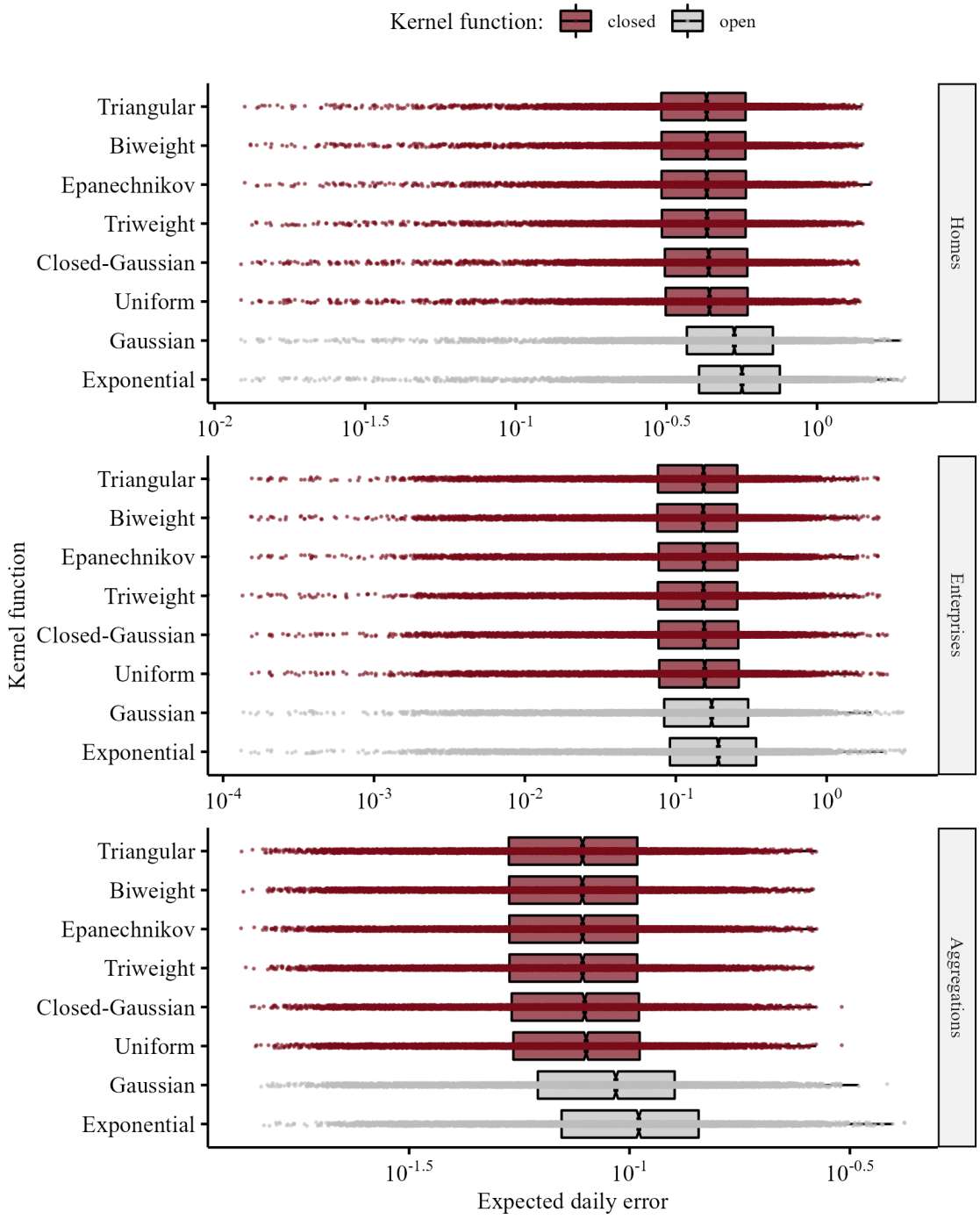

Figure 8.16: Kernel function comparison. Multivariate nonparametric model (Algorithm 2) using average-based merger with various kernel functions to determine the weights of historical observations predicted 300 loads in a validation experiment (Section 9.3.1.1). Each panel shows the expected daily error (7.15) distribution in the corresponding load group. The distribution of expected daily errors for each load (dots) is summarized by a box-plot where the notch denotes the 95%-confidence interval of the median.

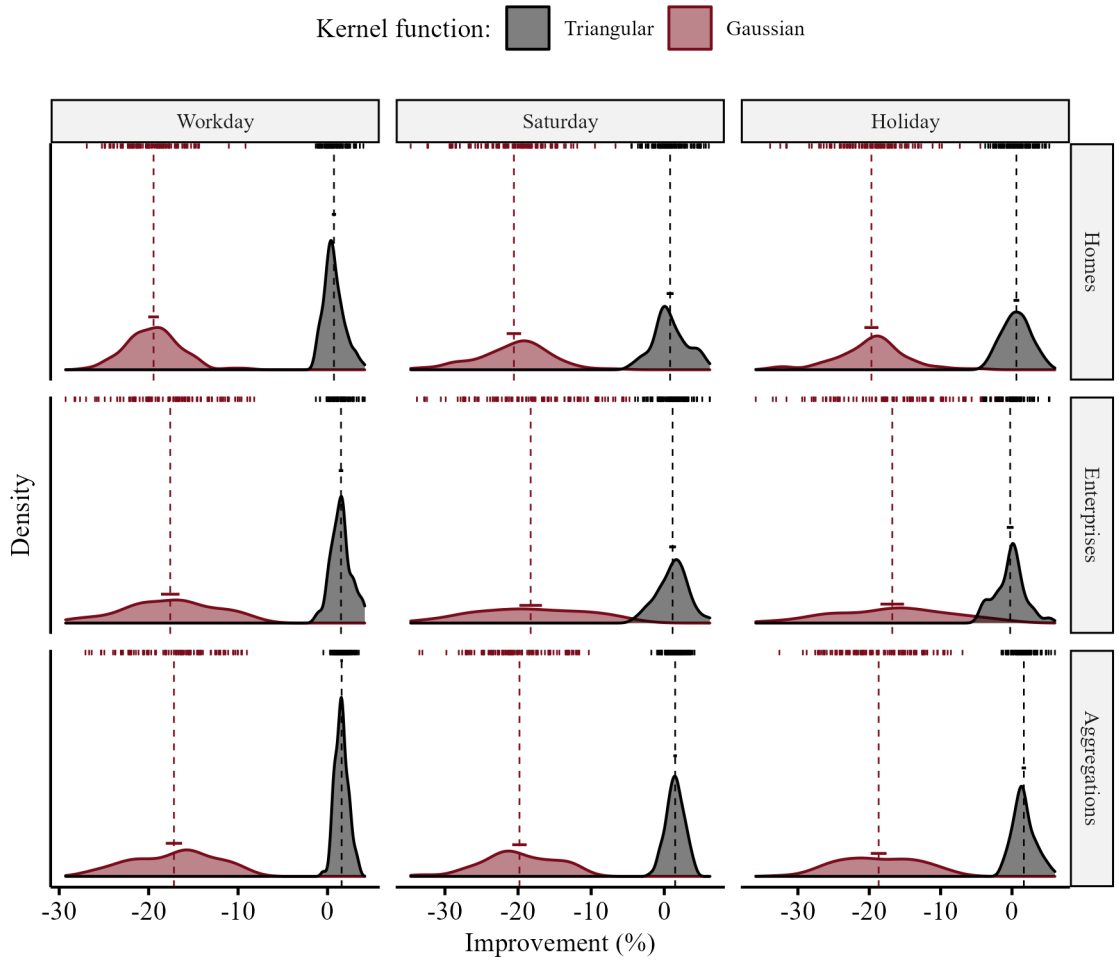

Figure 8.17: Forecast improvement through kernel weighting. In a validation experiment (Section 9.3.1.1), we applied the functional neighbor model (Algorithm 3) using an average-based merger with Gaussian and triangular kernels weighting historical observations to predict 300 loads of different groups and obtaining a sample of 30000 daily forecast errors. Additionally, we predicted the same loads with the uniform-average-based merger using the results as a benchmark. Relative to the benchmark, we computed the forecast improvement (7.14) for each predicted daily load curve. In the figure, every panel presents the sampling distribution of the mean improvement for each load (rugs), expected improvement in the load group (dotted vertical line) with the 95%-confidence interval (horizontal bar) obtained by the functional neighbor forecaster using the denoted kernel function in the corresponding day-type (panel column) and load group (panel row). Notably, Gaussian kernel provided no improvement against uniform average, but reduced the accuracy (on average down to 30%). At the same time, triangular kernel often resulted in a significantly ($p < 0.05$) more accurate forecast than when using the uniform kernel.

8.2.3.2 Permutation Merger

Permutation merger (PM) is a merger that can improve the forecast accuracy by calculating the center $\hat{Y}$ with respect to the ℓ^2_{u}-distance. We saw that ℓ^2_{u}-distance used to quantifying the similarity between the curves instead of the ℓ^2-distance can lead to a more accurate forecast. To further improve the forecast of the functional neighbor model, we consider finding the center of $\mathcal{G}_Y$ by using the ℓ^2_{u}-distance finding the center of historical output observations.

Permutation merger provides a consensus representation of historical outputs in $\mathcal{G}_Y$ allowing for small permutations within the load curves. The center load curve $\hat{Y}$ can be expressed as:

$$\hat{Y} = \underset{v\in\mathbb{R}^n}{\arg\min} \sum_{j=1}^{K} \Big[\mathbf{d}_{\mathrm{u}}(Y_j, v)\Big]^2. \tag{8.56}$$

To find $\hat{Y}$ we must solve a K-MCLP-problem defined as follows [CGS13a, CGS13b].

Definition 8.2.5. *K-dimensional minimal cost local permutation (K-MCLP)* problem consists of finding a set of u-local permutations[32] $\pi_1, \ldots, \pi_K \in \mathcal{P}(X, u, q)$ of the curves $Y_1, \ldots, Y_K$ which minimize the cost function of a form

$$\mathrm{Cost}(\pi) := \sum_{i=1}^{q} \mathrm{C}\Big(i, \pi_1(i), \ldots, \pi_K(i)\Big),$$

where $\mathrm{C}\Big(i, \pi_1(i), \ldots, \pi_K(i)\Big)$ is the cost of mapping the point at i to the points $\pi_1(i), \ldots, \pi_K(i)$.

Note that there might exist several solutions for a K-MCLP-problem. This problem can be effectively solved in $\mathcal{O}(nuK4^{Ku})$ time using a graph-based approach as described in [CGS13a, CGS13b].

We contrast the differences between the permutation merger and a uniform average. Imagine, an artificial example where we are merging two curves Y_1, Y_2 – both with a pronounced peak of a similar size, yet at a slightly different time (Figure 8.18). Merging with uniform average, results in a curve that features both peaks but with reduced amplitude. Permutation merger returns a curve that has one peak in between the peaks of the Y_1 and Y_2. Assuming that Y_1, Y_2 represent load curves with a pronounced peak, we need merger to provide a consensus representation of the curves that also features a single peak.

In practice, a load curve of a typical household often features a morning and an evening peak that is slightly shifted from day to day (Figure 8.19). Again, we combine two curves $Y_1, Y_2 \in \mathcal{G}_Y$ comparing the uniform average with the permutation merger allowing to

[32] Several solutions are possible.

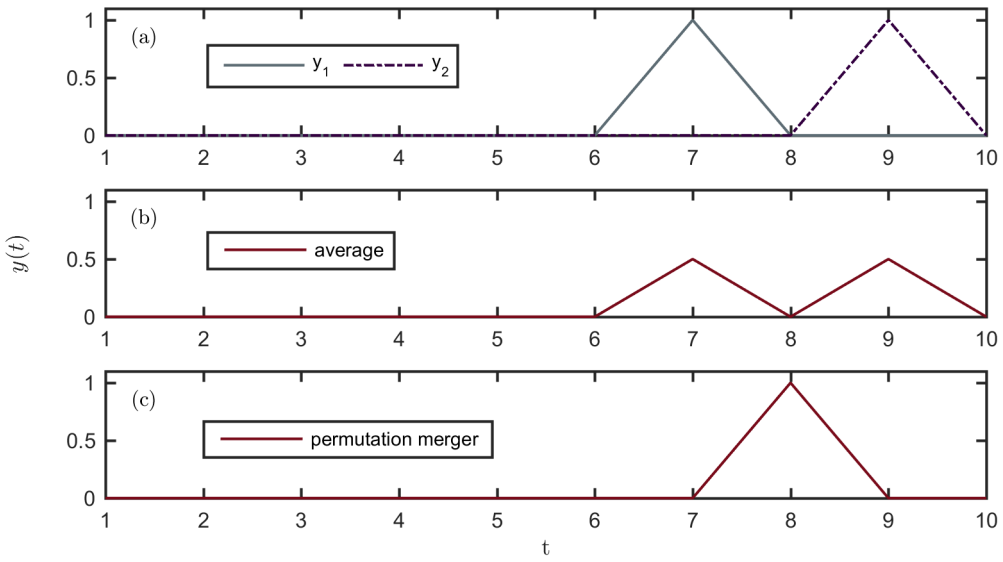

Figure 8.18: Contrasting uniform average and a permutation merger on an artificial example. In this example, we applied uniform average and permutation merger to compute a consensus representation of $\{Y_1, Y_2\}$. The curves Y_1, Y_2 have a distinctive peak of the same magnitude but slightly shifted in time (a). Uniform average provided a curve that features the peaks of both curves with reduced amplitude (b). In contrast, permutation merger provides the curve with one peak between the original peaks (c).

permute each point by one hour (i.e., $u = 1$). Both Y_1 and Y_2 have a visible morning and evening peak – each occurring at slightly different time. The forecast using the uniform average features both peaks with a reduced amplitude. On the other hand, the forecast using the permutation merger finds a better consensus representation. Same as the original curves, the merge has one peak, which is located in-between the original peaks of Y_1 and Y_2.

Allowing small permutations significantly improves the functional-neighbor forecast comparing to the uniform-average-based merger (Figure 8.20). Nevertheless, it is hard to imagine that permutations of more than one hour are adequate for the wide-scale day-ahead building load forecasting application. We observed that the forecast accuracy of the functional neighbor model using the permutation merger with $u = 1$ was the most accurate variant on the majority of forecast loads (Figure 8.21). Allowing permutations beyond one hour (i.e., $u > 1$ for our example) might be too permissive and perjudicate the forecast accuracy. For instance, $u = 2$ allows to match the points that are four hours away[33]. Four hours can already make a difference between an evening and an afternoon activity. Additionally, it is computationally expensive to increase permutation range [CGS13a, CGS13b].

[33] As mentioned previously, we consider hourly resolution of the time series in this study.

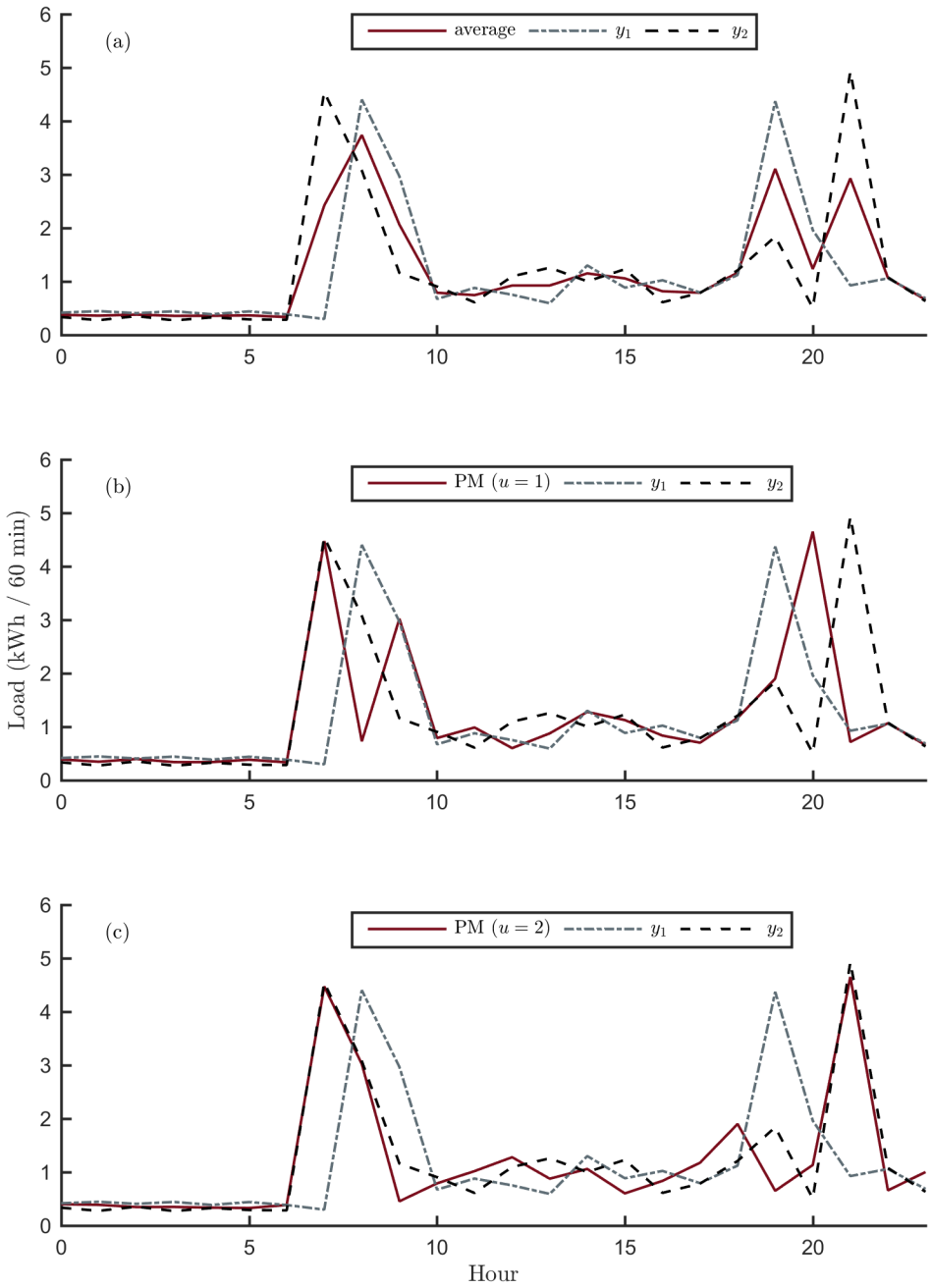

Figure 8.19: Demonstration of different mergers finding the consensus representation of two daily load curves with an hourly resolution: (a) uniform average; (b) permutation merger with one hour range; (c) permutation merger with a two hour range. Detailed description is provided in the text.

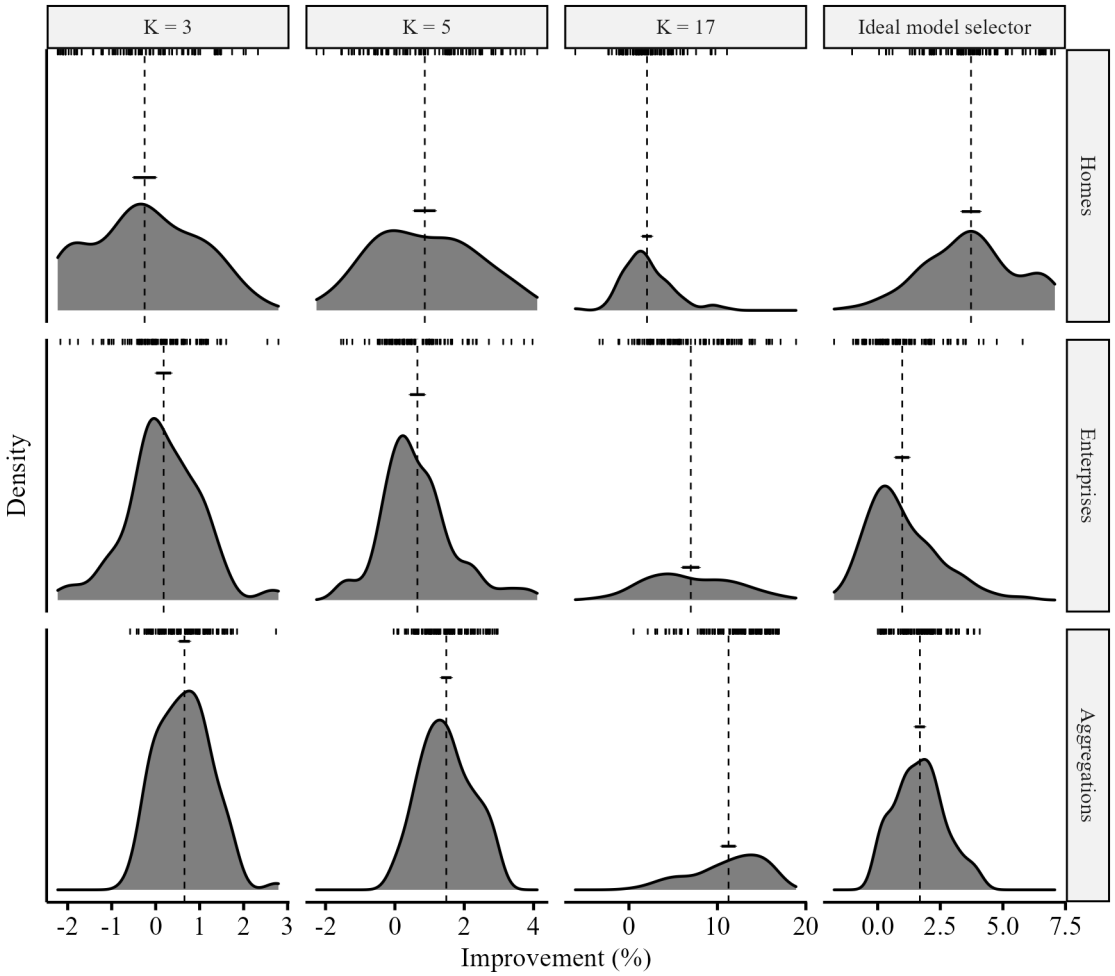

Figure 8.20: Forecast improvement with the permutation merger. In a validation experiment (Section 9.3.1.1), we applied the functional neighbor model (Algorithm 3) with a one-hour permutation merger ($u = 1$) to predict the 300 loads of different groups obtaining a sample of 30000 daily forecast errors. We used various bandwidths K determining the number of curves to be merged. Additionally, we applied the functional neighbor forecaster with uniform-average-based merger to predict the same loads and used these results as a benchmark. Relative to the benchmark, we computed the forecast improvement (7.14) for each predicted daily load curve. In the figure, every panel presents the sampling distribution of the mean improvement for each load (rugs at the top), expected improvement in the load group (dotted vertical line) with the 95%-confidence interval (horizontal bar) obtained by the functional neighbor forecaster with the specified bandwidth K (panel column) on the loads of the corresponding group (panel row). We observed that the permutation merger significantly ($p < 0.05$) improved the functional neighbor forecast. The average improvement depended on the chosen bandwidth. The improvement becomes more notable with larger K that requires more load curves to be merged. Further, we provided the results obtained by the model with an ideal model selector choosing the best possible bandwidth (Section 9.3.1.1). These results show that we can expect a significant forecast improvement when using the permutation merger instead of the uniform average for the functional neighbor model.

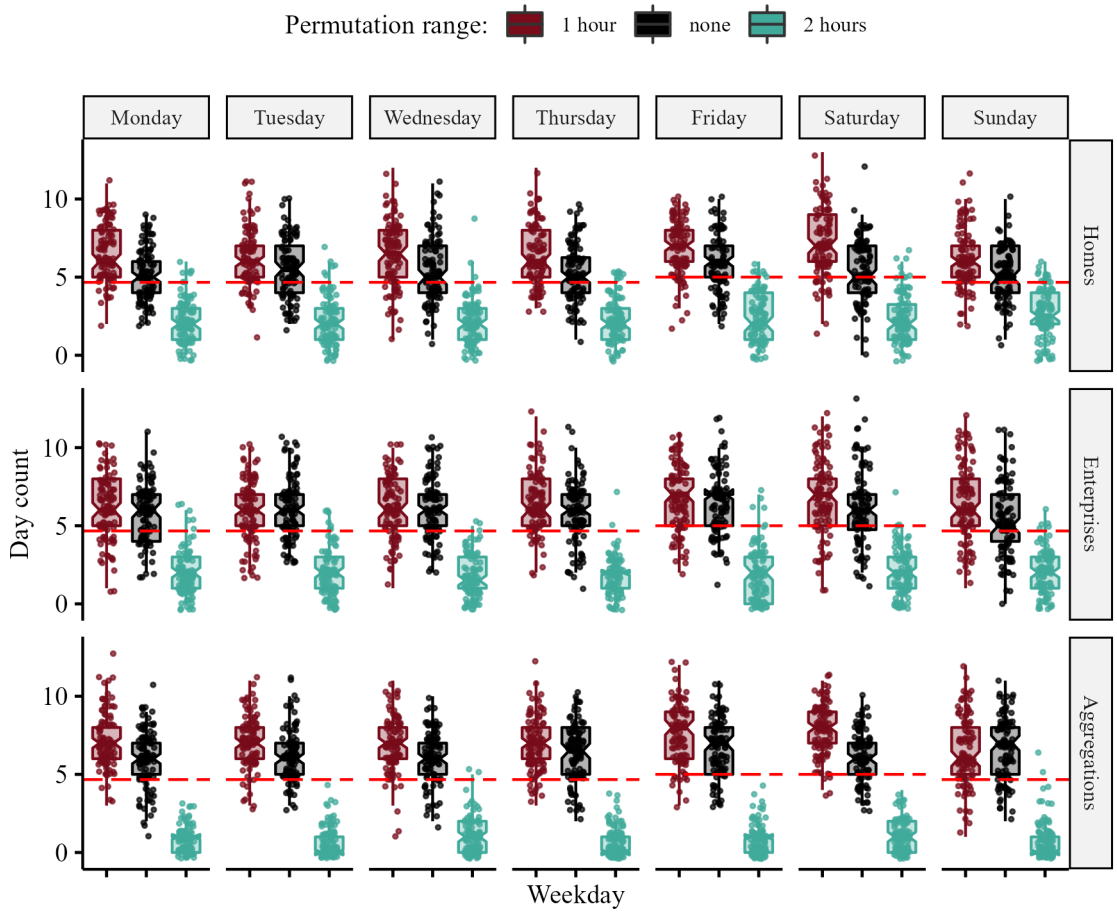

Figure 8.21: Selecting permutation range for the merger. In a validation experiment (Section 9.3.1.1), we applied the functional three-neighbor model (Algorithm 3 with $K = 3$) using the permutation merger with various ranges to predict the 300 loads of different groups obtaining a sample of 30000 daily forecast errors. Conditioning on weekday and load group, for each load, we counted the days where a model variant provided the smallest daily forecast error among other permutation merger variants predicting the same load. Each panel presents these day counts (dots). For the corresponding load group (panel row) and weekday (panel column), the distributions of individual load day counts are summarized by box-plots where the notch represents the 95%-confidence interval of the median and the dotted horizontal line represents the corresponding average count for the panel. Notably, one-hour permutation merger provided a significantly ($p < 0.05$) more accurate forecast in the majority of cases.

8.2.3.3 Weighted Permutation Merger

We observed how weighting the observations based on their distance to the query X^* can improve the accuracy (Section 8.2.3.1). Further, we saw that allowing permutations when merging the curves also improves the forecast comparing to the simple average (Section 8.2.3.2). In this section, we combine these insights introducing *weighted permutation merger* that modifies the original permutation-merger method allowing to weight the observations using a kernel function.

We begin extending the original theorem [CGS13b] through the introduction of the weights θ_j.

Theorem 8.2.2. A vector $\hat{Y} \in \mathbb{R}^n$ with

$$\hat{Y}(i) := \underset{y \in \mathbb{R}}{\arg\min} \sum_{j=1}^{K} \theta_j \Big[y - \pi_j(i) \Big]^2 \tag{8.57}$$

is the center (Definition 8.2.3) of $\mathcal{G} = \{ Y_j \mid j \in \{1, \ldots, K\} \}$ with respect to the ℓ^2_{u} distance if permutations $\pi_1, \ldots, \pi_K$ solve the K-MCLP-problem (Definition 8.2.5) with

$$C(i,\ i_1, \ldots i_K) := \min_{v \in \mathbb{R}} \sum_{j=1}^{K} \theta_j \Big[v - \pi_j(i_j) \Big]^2. \tag{8.58}$$

Proof. The proof is similar to the original line of argument that authors of the *permutation merger* provide in [CGS13b]. We extend it introducing the weights θ_j and prove theorem using Lemmas 8.2.1, 8.2.2, 8.2.3 presented further in the text.

Suppose there exists a curve $Y' \in \mathbb{R}^n$ such that

$$\mathrm{SSD}(Y') \leq \mathrm{SSD}(\hat{Y}) \tag{8.59}$$

with another set of permutations $\pi'_1, \ldots, \pi'_K$ solving the K-MCLP-problem which can have multiple solutions.

Using the ℓ^2_{u}-distance definition (8.52) we can write:

$$\mathrm{SSD}(Y') = \sum_{j=1}^{K} \theta_j \Big(\mathbf{d}_{\mathrm{u}}(Y_j, Y') \Big)^2 = \sum_{j=1}^{K} \theta_j \min_{v \in \mathbb{R}} \sum_{i=1}^{n} \Big[v - \pi'_j(i) \Big]^2. \tag{8.60}$$

Given that $v = Y'(i)$ minimizes the SSD on the right hand side of (8.59)) and introducing (8.60) on the left hand side, we obtain

$$\sum_{j=1}^{K} \theta_j \sum_{i=1}^{n} \Big[Y'(i) - \pi'_j(i) \Big]^2 < \sum_{j=1}^{K} \theta_j \Big(\mathbf{d}_{\mathrm{u}}(Y_j, \hat{Y}) \Big)^2. \tag{8.61}$$

With Lemma 8.2.1 (presented subsequently)

$$\mathrm{Cost}\left(\pi'_1, \ldots, \pi'_K\right) \leq \sum_{i=1}^{n} \sum_{j=1}^{K} \theta_j \Big[Y'(i) - \pi'_j(i) \Big]^2. \tag{8.62}$$

Linearity of the inner sum with θ_j allows us to rewrite (8.61) to

$$\mathrm{Cost}\left(\pi'_1, \ldots, \pi'_K\right) < \sum_{j=1}^{K} \theta_j \Big(\mathbf{d}_{\mathrm{u}}(Y_j, \hat{Y}) \Big)^2.$$

By Lemma 8.2.2 (presented subsequently)

$$\sum_{j=1}^{K} \theta_j \Big(\mathbf{d}_{\mathrm{u}}(Y_j, \hat{Y}) \Big)^2 \leq \mathrm{Cost}\left(\pi_1, \ldots, \pi_K\right),$$

we get

$$\mathrm{Cost}\left(\pi'_1, \ldots, \pi'_K\right) < \mathrm{Cost}\left(\pi_1, \ldots, \pi_K\right),$$

which contradicts the fact that $\pi_1, \ldots, \pi_K$ solves the K-MCLP-problem and, therefore, the $\mathrm{Cost}\left(\pi_1, \ldots, \pi_K\right)$ must be the minimal cost.

□

We complete the proof of Theorem 8.2.2 deriving the aforementioned lemmas. Same as previously, for $j \in \{1, \ldots, K\}$, Y_j are the curves to be merged with corresponding weights θ_j and π_j are the minimal cost permutations that solve K-MCLP-problem with $C(i, i_1, \ldots, i_K)$ defined in (8.58).

Lemma 8.2.1.

$$\mathrm{Cost}\left(\pi_1, \ldots, \pi_K\right) \leq \sum_{i=1}^{n} \sum_{j=1}^{K} \theta_j \Big[\hat{Y}(i) - \pi_j(i) \Big]^2.$$

Proof. Using the cost definition (8.58) we expand the left hand side to

$$\mathrm{Cost}\left(\pi_1, \ldots, \pi_K\right) = \sum_{i=1}^{n} C(i,\ i_1, \ldots i_K) = \sum_{i=1}^{n} \min_{v \in \mathbb{R}} \sum_{j=1}^{K} \theta_j \Big[v - \pi_j(i) \Big]^2.$$

Therefore,

$$\sum_{i=1}^{n} \min_{v\in\mathbb{R}} \sum_{j=1}^{K} \theta_j \Big[v - \pi_j(i_j)\Big]^2 \leq \sum_{i=1}^{n}\sum_{j=1}^{K} \theta_j \Big[\hat{Y}(i) - \pi_j(i)\Big]^2,$$

which is true for all $\hat{Y} \in \mathbb{R}^n$. □

Lemma 8.2.2.

$$\sum_{j=1}^{K} \theta_j \Big(\mathbf{d}_{\mathrm{u}}(Y_j, \hat{Y})\Big)^2 \leq C(\pi_1, \ldots, \pi_K)$$

Proof. We begin by expanding the left hand side using the ℓ^2_{u}-distance definition (8.52) and considering the fact that π_j is the solution to the MCLP-problem:

$$\begin{aligned}\sum_{j=1}^{K} \theta_j \Big(\mathbf{d}_{\mathrm{u}}(Y_j, \hat{Y})\Big)^2 &= \sum_{j=1}^{K} \min_{\pi'_j \in \mathcal{P}(Y_j,u,n)} \sqrt{\sum_{i=1}^{n} \Big|\pi'_j(i) - \hat{Y}(i)\Big|^2} \\ &= \sum_{j=1}^{K}\sum_{i=1}^{n} \theta_j \Big|\pi_j - \hat{Y}(i)\Big|^2.\end{aligned}$$

On the right hand side of the Lemma 8.2.2, we apply Lemma 8.2.3 presented subsequently:

$$\mathrm{Cost}\left(\pi_1, \ldots, \pi_K\right) = \sum_{i=1}^{n}\sum_{j=1}^{K} \theta_j \Big[\hat{Y}(i) - \pi_{Y_j}(i)\Big]^2 = \sum_{j=1}^{K}\sum_{i=1}^{n} \theta_j \Big[\hat{Y}(i) - \pi_{Y_j}(i)\Big]^2.$$

Hence,

$$\sum_{j=1}^{K}\sum_{i=1}^{n} \theta_j \Big|\pi_j - \hat{Y}(i)\Big|^2 \leq \sum_{j=1}^{K}\sum_{i=1}^{n} \theta_j \Big[\hat{Y}(i) - \pi_{Y_j}(i)\Big]^2,$$

which holds for any $\pi_j, \hat{Y} \in \mathbb{R}^n$. □

Lemma 8.2.3.

$$\mathrm{Cost}\left(\pi_1, \ldots, \pi_K\right) = \sum_{i=1}^{n}\sum_{j=1}^{K} \theta_j \Big[\hat{Y}(i) - \pi_{Y_j}(i)\Big]^2.$$

Proof. By definition

$$\mathrm{Cost}\left(\pi_1, \ldots, \pi_K\right) = \sum_{i=1}^{n} C(i, \pi_1(i), \ldots, \pi_K(i)),$$

for which with (8.58) we get

$$\mathrm{Cost}\left(\pi_1, \ldots, \pi_K\right) = \sum_{i=1}^{n} \min_{v\in\mathbb{R}} \sum_{j=1}^{K} \theta_j \Big[\hat{Y}(i) - \pi_{Y_j}(i)\Big]^2.$$

This equality is true since, according to (8.57), since the vector $\hat{Y}$ does minimize the expression on right hand side. □

Weighted permutation merger allows us to weight the observations depending on $\mathbf{d}_{\mathrm{u}}(X, X^*)$ as we did previously with a kernel function (Section 8.2.3.1). Consider illustrative example (Figure 8.22) where we intend to merge three curves Y_1, Y_2, Y_3 to which we assign corresponding weights $\theta_1, \theta_2, \theta_3$. Imagine, Y_3 represents the least relevant observation which is reflected by $\theta_3 \lll \theta_1$ and $\theta_3 \lll \theta_2$. Despite that, uniform average and permutation merger will consider Y_3 to a full extent. In contrast, weighted average will suppress the irrelevant observation but the resulting forecast will suffer under the same problem as uniform average in the previously discussed example (Figure 8.18): both peaks of Y_1 and Y_2 will appear with reduced amplitude. Weighted permutation merge combines the weighting and permutation resulting in the merge that features only one peak.

We expect the weighting to improve the quality of the permutation merger and, thereby, the accuracy of the forecast. Indeed, on the validation dataset, we observed a significant improvement in comparison to the original permutation merger (Figure 8.23). As with other mergers discussed in this chapter, the improvement of the merger became more notable when merging more curves (i.e., a larger K). Further, the largest improvement was observed for enterprises and aggregations where annual cycle is more prominent. Using the weighting helps to consider the annual cycle to which larger loads are often subject to. Overall, it appears that weighting does improve the accuracy of the permutation merger.

To conclude the discussion on the mergers, we make an overall comparison (Figure 8.24). We observed that the permutation-based mergers and, in particular, weighted permutation merger were the most accurate on the majority of the forecast days. As a result, weighted permutation merger was the best merger on the majority of loads (Figure 8.25). Using it instead of the kernel function, we can expect a notable improvement of the forecast provided by our functional neighbor model.

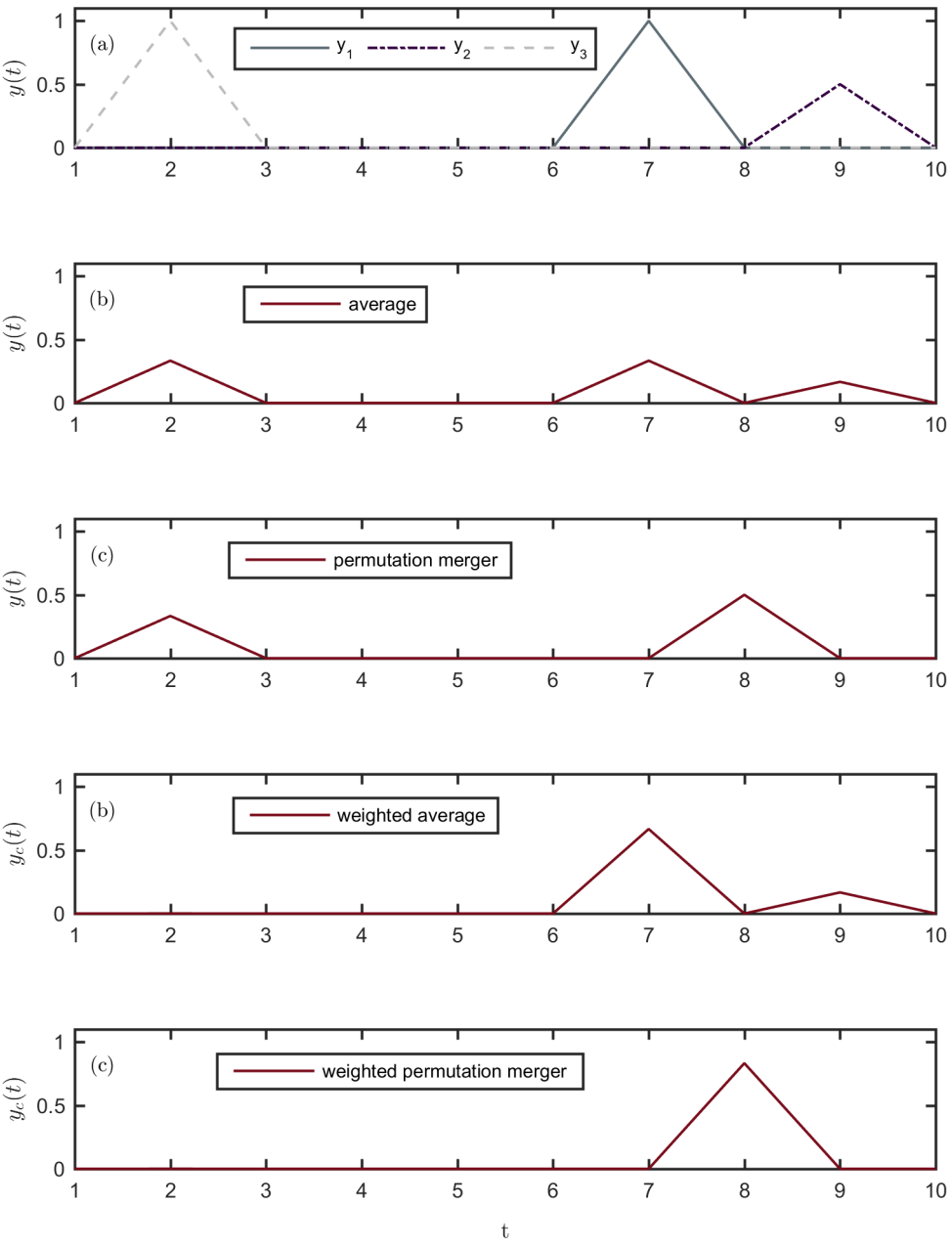

Figure 8.22: Demonstration of different mergers computing consensus representation of the illustrative curves Y_1, Y_2, Y_3. Detailed description is provided in the text.

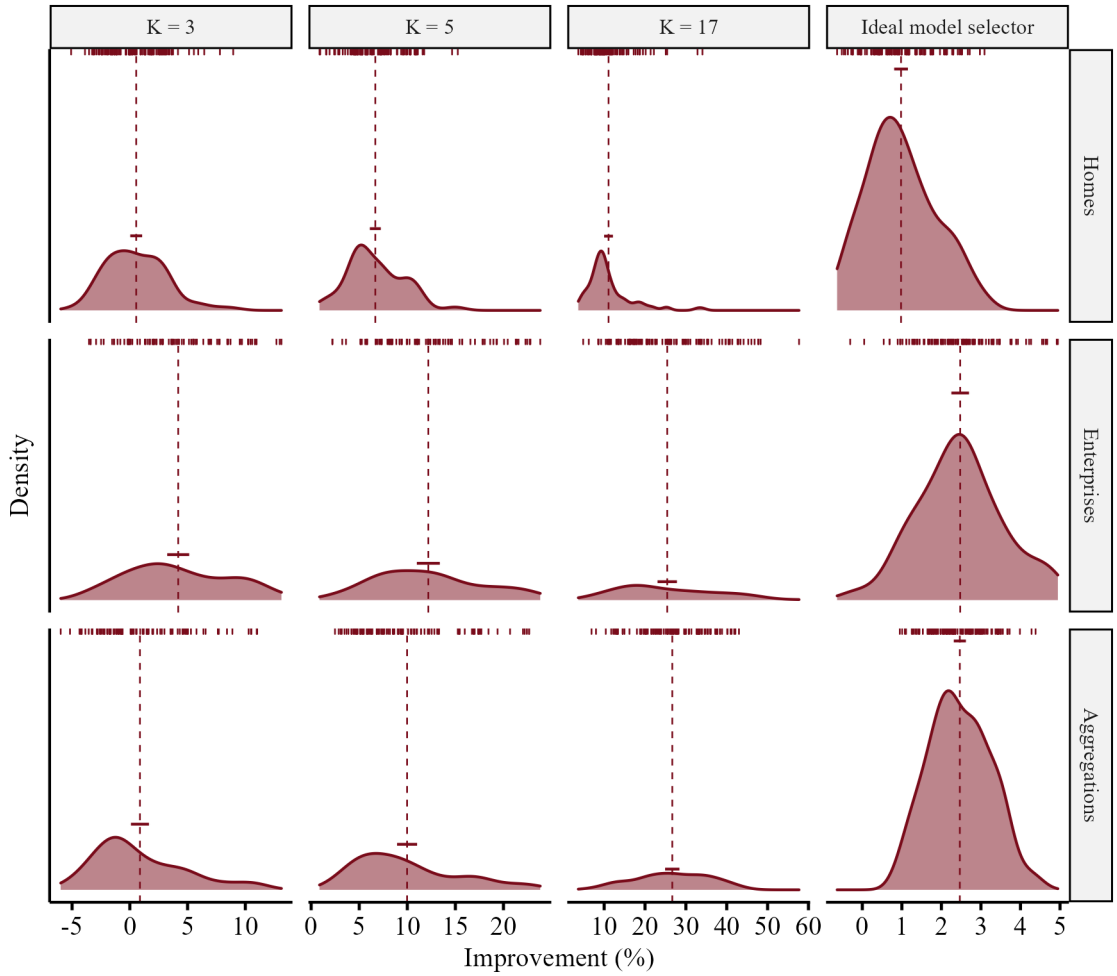

Figure 8.23: Permutation merger improvement through weighting. In a validation experiment (Section 9.3.1.1), we applied the functional neighbor forecaster (Algorithm 3) with a one-hour weighted permutation merger ($u = 1$) to predict the 300 loads of different groups obtaining a sample of 30000 daily forecast errors. We used various bandwidths K determining the number of curves to be merged. Additionally, we applied the functional neighbor forecaster with the (uniform) permutation merger to predict the same loads and used these results as a benchmark. Relative to the benchmark, we computed the forecast improvement (7.14) for each predicted daily load curve. In the figure, every panel presents the sampling distribution of the mean improvement for each load (rugs at the top) and the expected improvement in the load group (dotted vertical line) with the 95%-confidence interval (horizontal bar) obtained by the functional neighbor forecaster with the specified bandwidth K (panel column) on the loads of the corresponding group (panel row). We observed that weighting the observations significantly ($p < 0.05$) improved the functional neighbor forecast. The average improvement depended on the chosen bandwidth. The improvement becomes more notable with larger K that requires more load curves to be merged. Further, we provided the results obtained by the model with an ideal model selector choosing the best possible bandwidth (Section 9.3.1.1). These results show that we can expect a significant forecast improvement when using the weighted permutation merger instead of the original permutation merger.

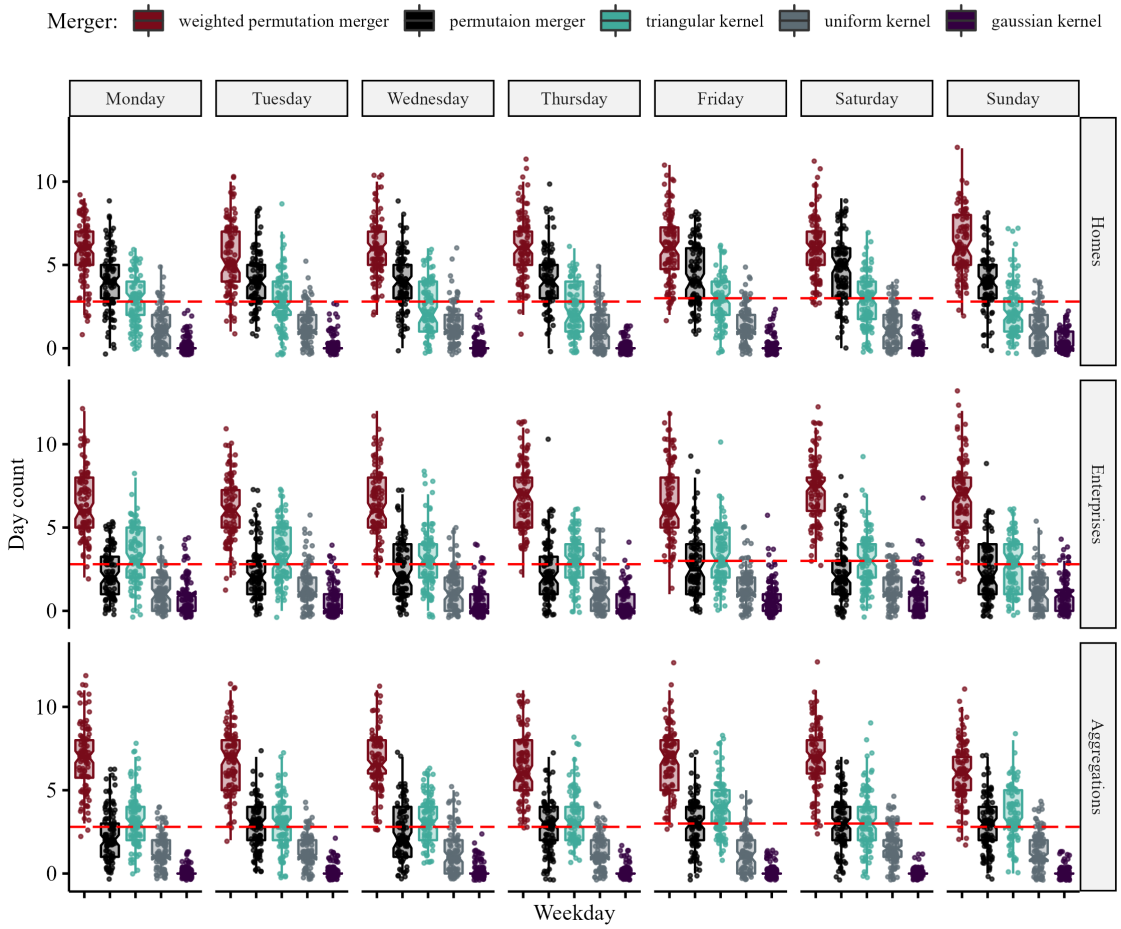

Figure 8.24: Comparison of different mergers. In a validation experiment (Section 9.3.1.1), we applied the functional neighbor forecaster (Algorithm 3) using various average-based mergers (uniform, Gaussian and triangular kernel functions), permutation merger ($u = 1$) and weighted permutation merger ($u = 1$) to predict 300 loads of different groups obtaining a sample of 30000 daily forecast errors. Conditioning on weekday and load group, for each load, we counted the days where a model variant provided the smallest daily forecast error among other merger variants predicting the same load. Each panel presents these day counts (dots). For the corresponding load group (panel row) and weekday (panel column) the distributions of individual load day counts are summarized by box-plots where the notch represents the 95%-confidence interval of the median and the dotted horizontal line represents the corresponding average count for the panel. Notably, the weighted permutation merger provided a significantly ($p < 0.05$) more accurate forecast than other mergers in the majority of cases.

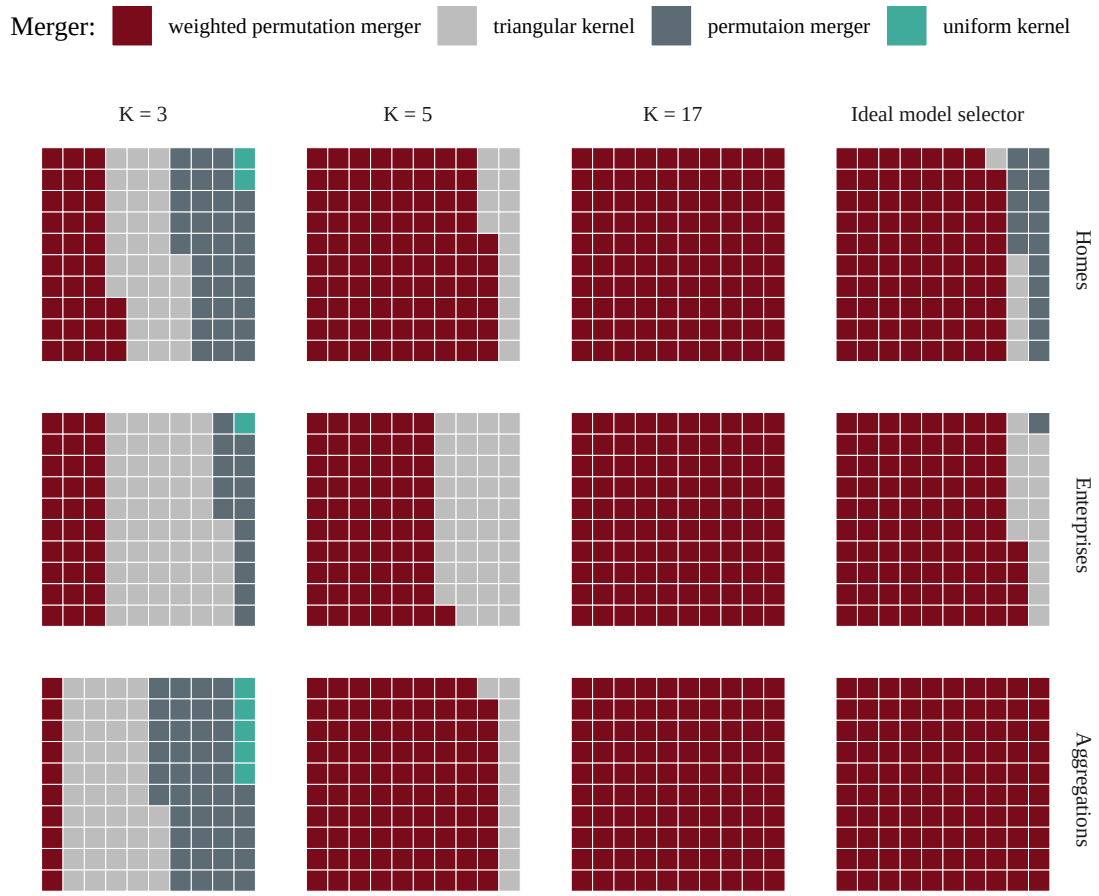

Figure 8.25: Comparison of different mergers by load. In a validation experiment (Section 9.3.1.1), we applied the functional neighbor forecaster (Algorithm 3) using the selected average-based mergers (uniform, triangular kernel functions), permutation merger ($u = 1$) and weighted permutation merger ($u = 1$) to predict 300 loads of different groups obtaining a sample of 30000 daily forecast errors. Moreover, we used various bandwidths K determining the number of curves to be merged. Conditioning on load type (panel row) and bandwidth (panel column), we represent each individual load by a square filled depending on the multistep strategy that provided the smallest expected daily error (7.15) on the days of the corresponding weekday and bandwidth. Notably, the model using the weighted permutation merger provided a more accurate forecast on the vast majority of loads. The dominance becomes more notable with larger K that requires more load curves to be merged. Further, we provided the results obtained by the model with an ideal model selector choosing the best possible bandwidth (Section 9.3.1.1). These results show that we can expect a significant forecast improvement when using the weighted permutation merger.

8.3 Functional Neighbor Extension

In this section, we extend the functional neighbor methodology to consider external inputs in the load forecasting model. As we noted previously, building power demand mostly depends on behavioral patterns of the inhabitants (Section 7.1). Nevertheless, power demand forecast, for some buildings can be improved by incorporating exogenous variables into the model (Section 7.1.3). The previously introduced functional neighbor forecaster (Algorithm 3) considers the behavioral patterns implicitly but disregards any exogenous variables. However, our forecasting methodology allows to extend the forecaster to consider external inputs that might affect the electricity consumption of a building.

The functional neighbor forecaster proposed above is based on the univariate autoregressive functional nonparametric load model and can ignore some of the available information about the predicted day. The basic assumption of the model is that all relevant information about the predicted load curve Y_j is contained in the load curve of the preceding day X_j. This assumption ignores any additional information about the predicted day. In general, however, the load can depend in an unknown way on several qualitative (e.g., weekday) and quantitative (e.g., solar irradiation) variables that characterize the predicted day. This information is not included in X_j and, in some cases, the model can be improved if it considers such information as a set of external inputs $\mathcal{Z}_j$.

8.3.1 Existing Approaches

There are two general approaches for modeling the dependency on the external inputs. Process-based models assume an explicit dependency, and the input-output relationship form needs to be assumed a priori (e.g., linear, quadratic, etc.). As a result, such models can extrapolate with non-observed covariates. The most common example is a linear model that extrapolates assuming linear dependency. However, the output often depends on external variables in an unknown way which makes it hard to make an adequate assumption about the dependency form. Alternatively, data-driven models, such as the ones considered in this study (Chapter 5), rely on interpolation of historical data. Making a prediction, these models interpolate the outputs for the previously observed covariates. They rely on weaker assumptions[34] but cannot extrapolate for inputs that were not observed previously (e.g. temperature on an extraordinarily hot day).

At the moment, we are aware of only two functional nonparametric load forecasting approaches that allow to consider external inputs. Aneiros and Vieu extended the functional nonparametric approach (4.82) to a *semi-functional partially linear (SFPL)* model that

[34] For instance, an ANN-model only has to assume that such dependency exists without specifying its form.

allows to consider linear dependencies on exogenous multivariate inputs [AV06]. In a series of publications, the authors generalized functional nonparametric approach incorporating linear component into the regression function. First, the model was proposed for scalar inputs [AV06] and then extended to functional inputs in the subsequent publications [AVCMSR13, AVR16].

The SFPL model can extrapolate for non-observed covariates but assumes linear dependency on the input and can only consider exogenous variables with a linear effect on the load. For instance, to model the nonlinear effect of the temperature on the consumption, the authors use a transformation to linearize this dependency [AVR16]. In practice, the dependency of the load on various external inputs (e.g., weather and external demand response incentives) can often be nonlinear. Further, the same behavior of the input can lead to a different response. In particular, the reaction of the daily load can depend not only on temperature but also on some qualitative variables (e.g., weekday). Moreover, the SFPL-model relies on accurate next-day forecasts of the inputs. While available for the temperature, this might not be the case for solar irradiation or other exogenous variables that are more volatile.

Alternatively, a functional *similar shape forecaster* presents a different way to consider external inputs when predicting the load with a nonparametric approach [PS13]. Using the prediction of the exogenous variables (e.g., weekday, weather), the upcoming day is assigned to one of the precalculated reference curves. The forecast is computed with functional nonparametric regression (4.82) using the assigned reference curve as a query. In theory, the implementation of this method is straightforward. However in practice, its accuracy depends on the availability of the required time-series data while this method is highly susceptible to the curse of dimensionality — with every additional input we need exponentially more historical data (Section 4.2.3.2).

8.3.2 Functional Neighbor Extension Model

Functional neighbor forecasting methodology (Section 8.2) allows to extend the univariate forecaster enabling it to consider external inputs (*features*). Recall that the relevance of the historical days constituting the object space

$$\mathcal{H} := \{\mathcal{D}_j \mid 1 \leq j \leq h\} \text{ with } \mathcal{D}_j := \{X_j, Y_j, \mathcal{Z}_j\}, \tag{8.63}$$

for the predicted day $\mathcal{D}_{j+1}$, is evaluated using an abstract distance notion that quantifies the similarity between the most recently observed load curve X^* and historical predecessor load curves $X_1, \ldots, X_h$. Beyond the load curve, each day j can include several qualitative and quantitative variables that characterize it (e.g., weather measurements, day-type etc.).

Therefore, we define the *extended query*

$$\mathcal{D}^* := \{X^*, \hat{\mathcal{Z}}^*\}, \tag{8.64}$$

where X^* is the most recently observed load curve (as in Section 8.2) and $\hat{\mathcal{Z}}^*$ is a set containing the predictions of v exogenous variables obtained by an independent forecaster $\mathcal{F}_z$ for the upcoming day (Figure 7.13). historical days in $\mathcal{H}$ can be viewed as points in a $v+1$ dimensional space where the coordinates correspond to the $v+1$ distances between the corresponding variables.

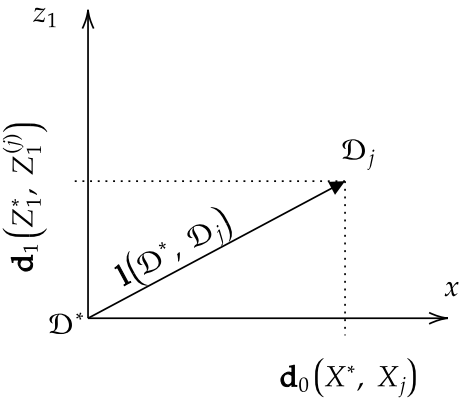

Figure 8.26: Two-dimensional demonstration of relevance computation based on triangulation. The relevance $\mathbf{l}(\mathcal{D}^*, \mathcal{D}_j)$ of the historical day $\mathcal{D}_j = \{X_j, Z_1^{(j)}\}$ to the extended query $\mathcal{D}^* = \{X^*, Z_1^*\}$ can be computed as the square root of the sum of the squares of distances between individual features of the day applying the Pythagorean theorem (8.65). Further discussion is provided in the text.

Consequently, the *relevance* of a historical day $\mathcal{D}_j$, for a given $\mathcal{D}^*$, can be calculated as a distance using triangulation (Figure 8.26):

$$\mathbf{l}(\mathcal{D}^*, \mathcal{D}_j) := \sqrt{w_0 \mathbf{d}_0(X^*, X_j)^2 + w_1 \mathbf{d}_1(Z_1^*, Z_1^{(j)})^2 + \ldots + w_v \mathbf{d}_v(Z_v^*, Z_v^{(j)})^2}. \tag{8.65}$$

The *feature weights* $w_0, \ldots, w_v$ allow to account for the fact that different features can have different influence on the load curve.

The weights can be defined manually or determined by a separate feature selector module. For instance, the importance of an individual feature can be related to the correlation with the electricity consumption. In the future, we will develop an automated feature selector module, facilitating a wide-scale application of the proposed model extension.

For this study, we manually select the most relevant features of the simulated smart building (Section 9.1.2). Note that a manual input selection does not impede a practical application of the FNX-model, since a larger energy equipment (e.g., a PV-generator) has to be registered by the grid operator.

8.3.3 Functional Neighbor Extension Algorithm

Functional neighbor extension (FNX) model described above allows to create a load forecasting algorithm that considers exogenous variables for predicting the day-ahead power demand of a building (Algorithm 4). The algorithm consists of the following steps.

Algorithm 4: Functional neighbor extension (FNX)

Inputs: extended query $\mathcal{D}^*$
Outputs: forecast curve $\hat{Y} \in \mathbb{R}^n$
Data: historical daily observations $\mathcal{H} := \{\mathcal{D}_j \mid 1 \leq j \leq h\}$
Parameters: number of nearest neighbors K, distance notions $\mathbf{d}_0, \ldots, \mathbf{d}_v$, feature weights $w_0, \ldots, w_v$

1 smooth time-discrete measurements of the quantitative inputs: $\rightarrow \mathcal{H}_\mathbf{f}, \chi^*, \hat{\mathcal{Z}}^*_\mathbf{f}$
2 compute the distances between the features using corresponding distance notions: $\rightarrow \mathbf{d}_0(\chi^*, \chi_j), \mathbf{d}_1(\hat{\zeta}^*, \zeta_1^{(j)}), \ldots, \mathbf{d}_v(\hat{\zeta}^*, \zeta_v^{(j)})$
3 compute the relevance of the historical days to $\mathcal{D}^*$ with (8.65)
4 sort $\mathcal{H}_\mathbf{f}$ by the relevance to the query $\mathcal{D}^*$
5 find K-nearest neighbors of $\mathcal{D}^*$: $\rightarrow \mathcal{G}_{\mathcal{D}^*}$
6 merge historical outputs to a consensus representation of $\phi_j \in \mathcal{G}_{\mathcal{D}^*}$: $\rightarrow \hat{\phi}$
7 re-sample $\hat{\phi}$: $\rightarrow \hat{Y}$

Step 1 Where necessary for the distance computation, we apply smoothing splines [RS05], to obtain continuous functions χ, ζ from time-discrete measurements X, Z representing the corresponding quantitative features[35].

Step 2 For each quantitative feature of a historical day j, we compute the distance to the prediction of the corresponding feature contained in $\mathcal{D}^*$. Each distance is computed with the corresponding distance notion[36] that was defined specifically for that feature.

[35] This step is similar to the Step 1 in Algorithm 3.
[36] Distance notions were discussed in Section 8.2.2.2.

Step 3 We compute the relevance of each historical day with respect to the given extended query $\mathcal{D}^*$ using (8.65).

Step 4 We sort historical days by the relevance in a descending order.

Step 5 To find the K-nearest neighbors of $\mathcal{D}^*$, we determine corresponding neighborhood $\mathcal{G}_{\mathcal{D}^*}$ by selecting the observations whose relevance from $\mathcal{D}^*$ is within the variable bandwidth

$$b_K = \mathrm{l}(\mathcal{D}^*, \mathcal{D}_K), \tag{8.66}$$

which corresponds to the distance between $\mathcal{D}^*$ and its K'th nearest neighbor $\mathcal{D}_K$.

Step 6 We merge the historical output observations ϕ_j in $\mathcal{G}_{\mathcal{D}^*}$ to a forecast $\hat{\phi}$[37].

Step 7 At last, we compute a time-discrete prediction $\hat{Y} \in \mathbb{R}^n$ resampling $\hat{\phi}$ on the desired sampling grid[38].

In order to validate the FNX-model, we apply it on the existing smart building within the Smart-City-Demo Aspern project (Section 9.1.2). In particular, we simulated the load forecasting of a student dorm facility whose net electricity demand substantially depends on the solar irradiation due to a large PV-installation on the roof. The results are presented further in the text (Section 10.3).

[37] This step is similar to Step 4 in Algorithm 3. Merger functions were discussed in Section 8.2.3.
[38] This step is similar to Step 5 in Algorithm 3.

9 Evaluation

In this chapter, we describe the evaluation of the functional neighbor forecasting methodology introduced previously. To validate the functional neighbor forecaster (Algorithm 3), we conducted a wide-scale day-ahead building load forecasting simulation (Section 9.1). In particular, we simulated a day-by-day forecast of numerous loads of different size and type using a public smart-meter dataset. Additionally, we applied our model on the buildings in Aspern validating the extension (Algorithm 4) of the functional neighbor forecaster that can be used on smart buildings where the consumption notably depends on external inputs. Moreover, we compared the accuracy of our forecaster with several reference models that can be often found in the load forecasting literature (Section 9.2). The simulations are summarized at the end of the chapter (Section 9.3).

9.1 Simulations

In this section, we describe the simulations that were used to validate the functional neighbor forecasting methodology. In particular, we evaluated the functional neighbor forecaster (Algorithm 3) in a wide-scale day-ahead building load forecasting simulation using an extensive smart-meter dataset (Section 9.1.1). Additionally, we validated the extension for considering external variables (Algorithm 4) on a smart building from the Smart-City-Demo Aspern Project (Section 9.1.2). Concluding this section, we provide the details on the performed computations and tools that we used for our study (Section 9.1.3).

9.1.1 Wide-Scale Building Load Forecasting Simulation

We predicted numerous loads of different size and type simulating the day-ahead building load forecasting on a wide scale. Simulated loads included individual buildings and their aggregations from the public smart-meter dataset provided by *Irish Commission for Energy Regulation (ICER)* [Arc16]. The data was collected for a variable electricity tariff trial in the Great Dublin Area (Ireland) and includes load time-series measured by 6445 smart meters installed on various households and enterprises. For the vast majority of buildings, the load was measured from 15th July 2009 to 31st December 2010 with a 30-minute resolution.

After the preparation, we split the data into a validation and evaluation datasets as described below. We used the validation dataset including the measurements of 300 loads over twelve months for the problem formulation (Chapter 7), for the design of the functional neighbor forecaster (Chapter 8) and the pre-selection reference models (Chapter 9.3). The evaluation dataset consisting of 1851 loads and extending over five months was used only for the final evaluation whose results we describe in the next part of the thesis.

9.1.1.1 Data Preparation

The original ICER-dataset required data preparation described in this section. Most importantly, we selected only the loads from the control group (1126 loads) that were not subject to the variable tariffs. Further, we discarded any load time-series that had more than 1% of measurements equal to zero or were otherwise obviously corrupted.

The time series in this dataset use sample and hold forward sampling. A value at 00:00 indicates the amount of energy in kWh consumed between 00:00 and 00:30. Further, the original load curves include daylight saving time. The load curve on the winter time fall back day (25th of October 2009 and 31st of October 2010) contains 50 points. The load curve on the summer time spring forward day (28th of March 2010) has only 46 points. To simplify the computation, for each load we discard the extra hour on the winter fall back days, and interpolate for the missing hour on the spring forward day. Moreover, we re-sampled each load time-series equidistantly with a *60-minute resolution* and normalized it by its maximum value to facilitate the comparison between the loads and demonstration. After down-sampling and mitigating the daylight saving time unbalances, each daily load curve contained exactly 24 points.

Using the remaining 1062 loads, we created 789 aggregations to evaluate the forecasts also on larger loads representing larger buildings. In particular, we aggregated the households and enterprises to residential, commercial, and mixed aggregations of different size. Mixed aggregations contain 80% of households and 20% of enterprises which corresponds to the ratio commonly found at the transformers in the distribution grid. For each aggregation size, the loads were selected randomly from the correspondent group without replacement.

As a result, the *extended smart-meter dataset* that we used in this study contained:

- 887 single family homes (households)
- 175 small and middle enterprises (enterprises)
- 360 residential aggregations
- 67 commercial aggregations
- 362 mixed aggregations

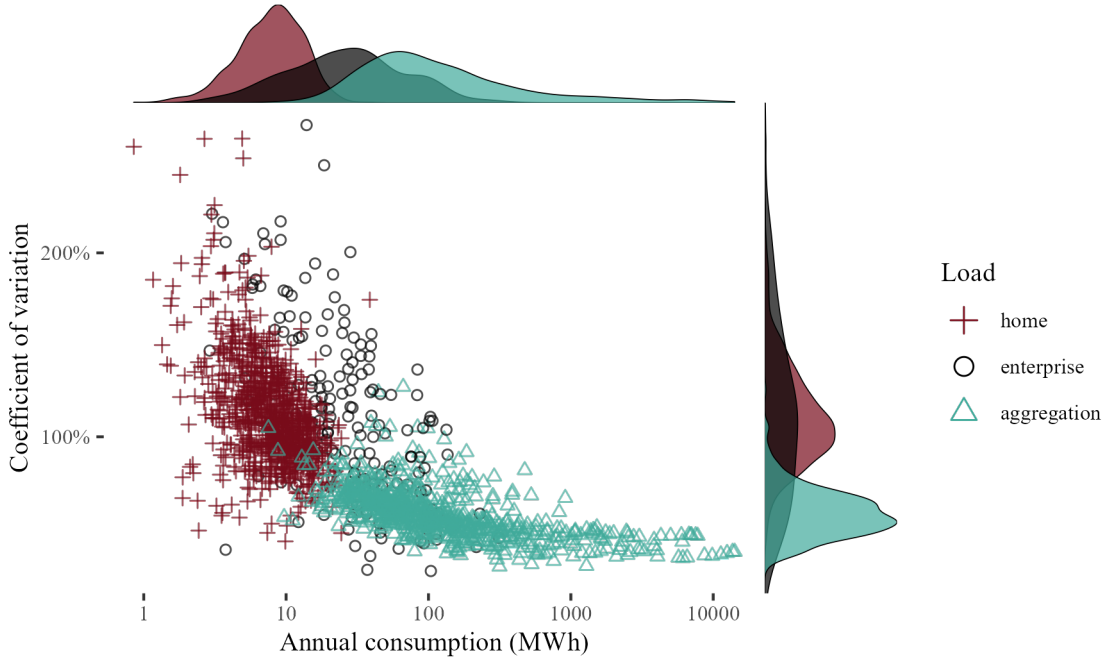

Figure 9.1: Loads of the extended ICER smart-meter dataset that were used for the wide-scale building load day-ahead forecasting simulation. Homes, enterprises and aggregations are denoted according to their size (annual consumption) and variability (coefficient of variation). The distribution of the size and variability within each load group is denoted alongside the main plot with the same color.

For each of these 1851 loads, we had complete data from 15$^{\text{th}}$ of July 2009 to 31$^{\text{st}}$ of December 2010 with no missing values at the 60-minute resolution.

The dataset is represented in Figure 9.1 where we denote each household, enterprise and aggregation in terms of its size (annual consumption) and variability expressed through the *coefficient of variation (CV)*

$$\mathrm{CV}(y) = \frac{\sigma(y)}{\mu(y)}. \tag{9.1}$$

Observing the distributions of the loads, we noted that the households and enterprises had, on average, different annual consumption but similar CV. In each group, the loads were log-normally distributed in terms of size (logarithmic x axis). Enterprises tended to have higher consumption and were more diverse, having a wider range of CV. Aggregations had the largest range in terms of size but the smallest in terms of variability (both average and spread). Therefore, they were the most consistent group in terms of variability.

We split the extended dataset into validation and evaluation datasets as described subsequently.

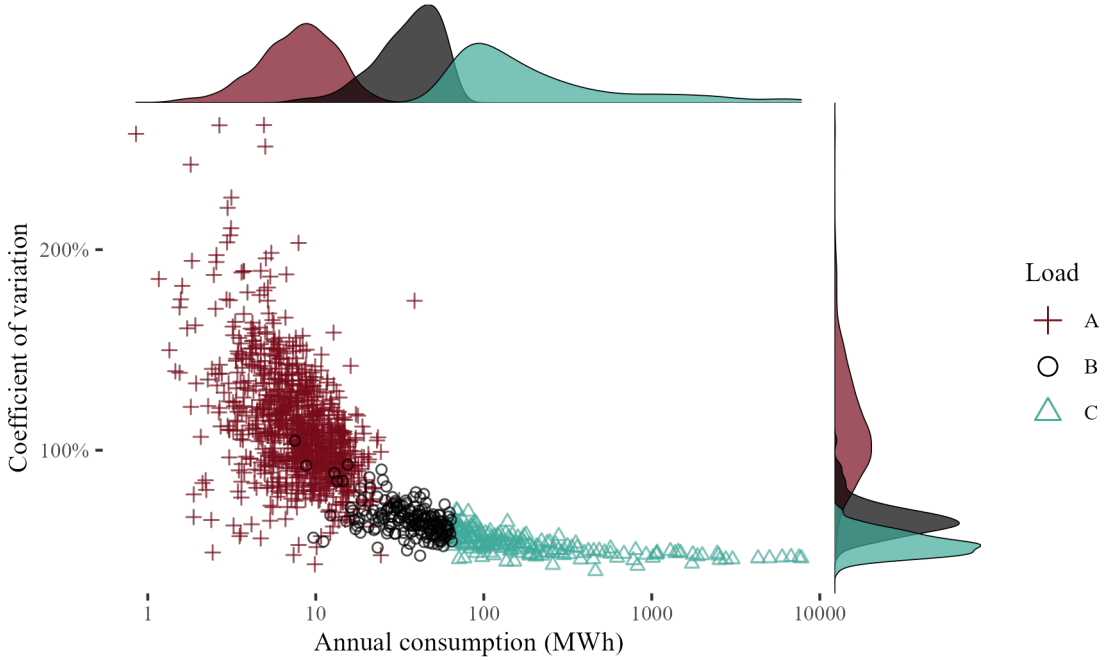

Figure 9.2: Residential load groups included in the extended ICER smart-meter dataset that were used for the wide-scale building load day-ahead forecasting simulation. Single family homes (A), residential aggregations (B) and large residential aggregations (C) are denoted according to their size (annual consumption) and variability (coefficient of variation). The distribution of the size and variability within each load group is denoted alongside the main plot with the same color.

9.1.1.2 Evaluation Dataset

The evaluation dataset consisted of 1851 load time-series from 1st of August 2010 to 31st of December 2010 (five months). We defined the following separate residential and commercial load groups summarized in Table 9.3.

(**A**) *single family homes* – as encountered in the ICER-dataset

(**B**) *residential aggregations* – first two quartiles of all residential aggregations

(**C**) *large residential aggregations* – last two quartiles of all residential aggregations

(**D**) *single enterprises* – as encountered in the ICER-dataset

(**E**) *commercial aggregations* – first two quartiles of the commercial aggregations

(**F**) *large commercial aggregations* – last two quartiles of commercial aggregations

The distribution of the loads in terms of size and variability in the residential groups A, B, and C is presented in Figure 9.2. Notably, there is an overlap between the groups in terms of annual consumption. At the same time, the variability drops with aggregations size. While for single households it can reach over 250%, for larger aggregations it drops under 50%.

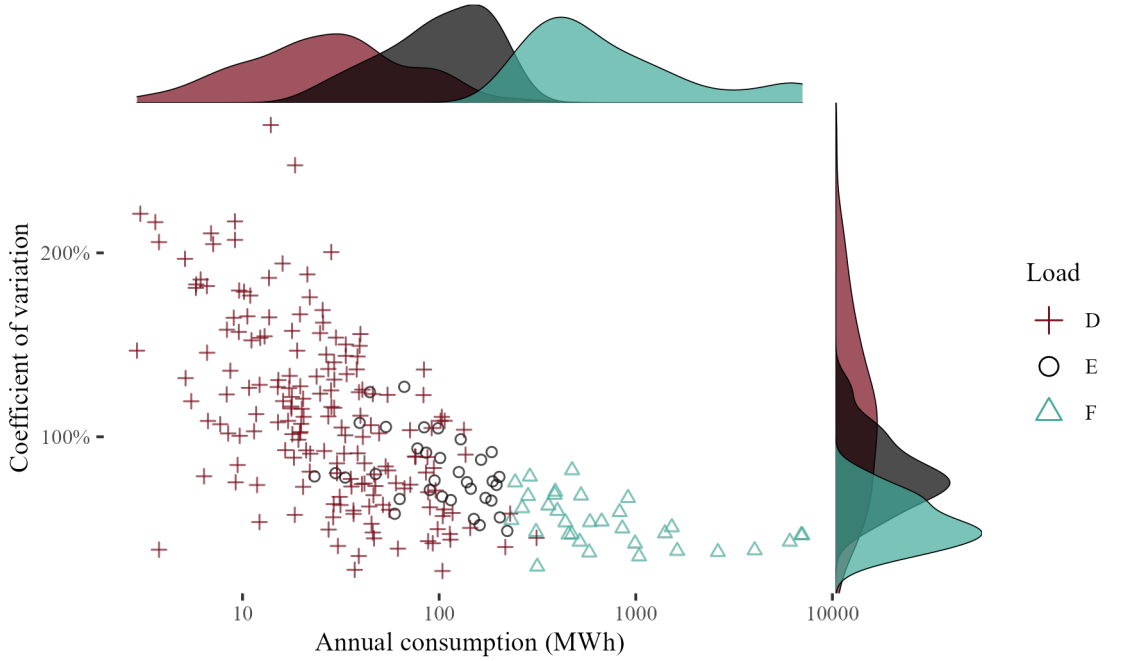

Figure 9.3: Commercial load groups included in the extended ICER smart-meter dataset that were used for the wide-scale building load day-ahead forecasting simulation. Single enterprises (D), commercial aggregations (E) and large commercial aggregations (F) are denoted according to their size (annual consumption) and variability (coefficient of variation). The distribution of the size and variability within each load group is denoted alongside the main plot with the corresponding color.

The distribution of the loads in the commercial groups D, E, and F in terms of size and variability is presented in Figure 9.3. There are much fewer commercial loads in our dataset. However, the distribution in terms of annual consumption and variability is similar to the residential groups. Again, there is a small overlap between the groups in terms of annual consumption. The variability was substantial for enterprises and smaller commercial aggregations but dropped with the increasing load size. Notably, the variability of average-sized loads (around 100 MWh) was more substantial than for residential consumers of similar size.

9.1.1.3 Validation Dataset

For the validation dataset, we selected 300 loads and used the measurements from 1st of August 2009 to 31st of July 2010 (twelve months). In particular, we defined three load groups of different type:

- single family homes (residential)
- enterprises (commercial)
- mixed aggregations (mixed)

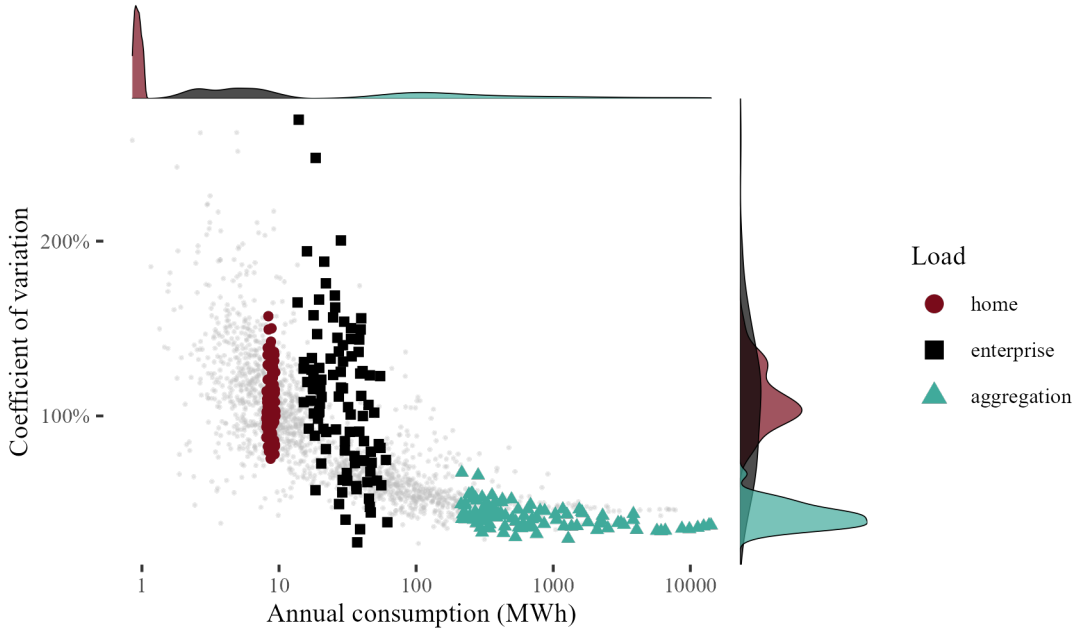

Figure 9.4: Loads from the extended ICER smart-meter dataset that were included in the validation dataset. For each of the three load groups (single family homes, enterprises, mixed aggregations), we selected 100 loads that were closes to the average annual consumption among the corresponding group in the extended ICER smart-meter dataset.

For each group, we selected 100 loads of each type that had annual consumption closest to the average among the loads of same type in the extended ICER smart-meter dataset.

The selection is presented in Figure 9.4 and summarized in Table 9.4. We see that the three groups do not overlap in terms of size. Again, the enterprises are the most diverse group in terms of variability. Aggregations is the most consistent group with the smallest variability average and spread.

9.1.2 Smart-Building Load Forecasting Simulation

In this section, we describe the smart-building load forecasting simulation that we used to validate the FNX-model – the extension of the functional neighbor model that can consider external inputs (Section 8.3). We simulated a smart building from the Smart-City-Demo Aspern project (Section 6.3) which we describe below.

The smart building is a student home constructed to the highest sustainability standards. It accommodates over 300 students on 7000 m^2 and features modern energy equipment (Figure 9.5). The building has a *photovoltaic (PV)* electricity generator (221 kWp) on the roof, electrical battery (150 kWh) and a building energy management system connecting all the energy equipment. The facility is heated thermally by the district heating network [Asp].

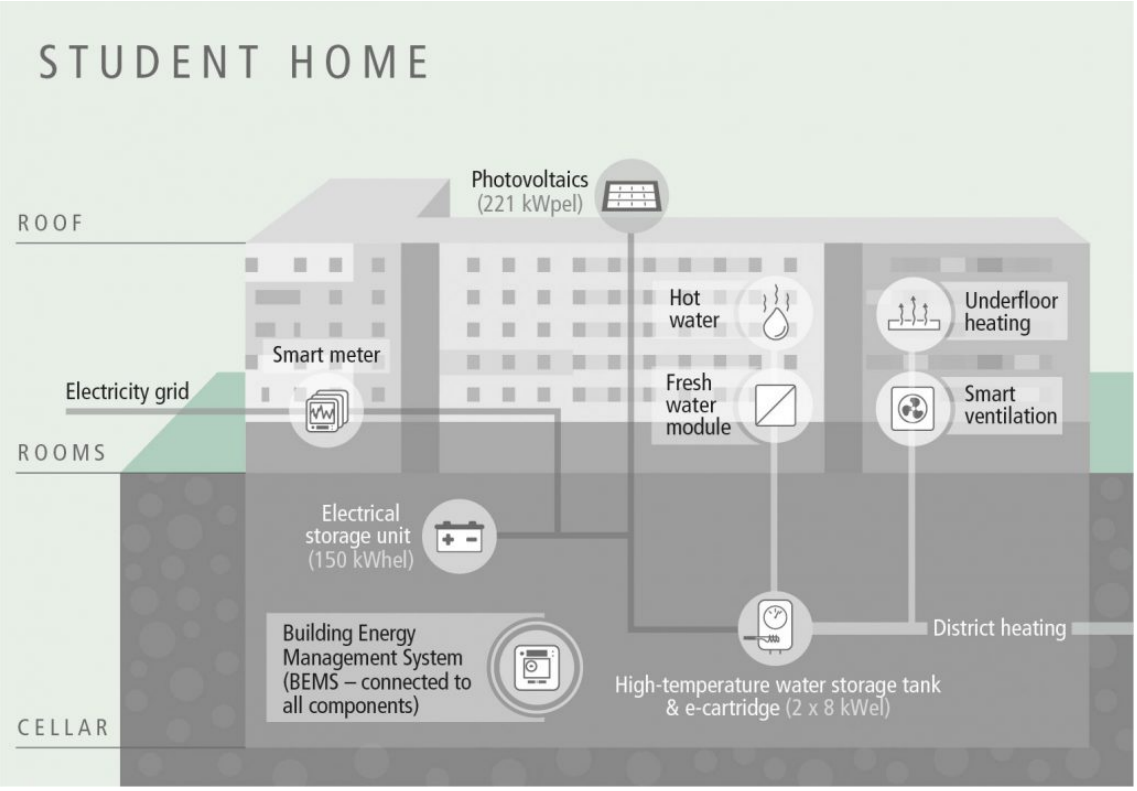

Figure 9.5: Smart building from the Smart-City-Demo Aspern project [Asp]. The student home accommodates over 300 students on 7000 m^2 and features various energy equipment denoted on the figure.

For this smart building, we obtained one year of load measurements[1] that we re-sampled equidistantly with a 60-minute resolution. The average load curves computed for each season show that the highest load occurs in winter, despite the fact that the building is heated thermally (Figure 9.6). In every season, we observed pronounced load peaks during early afternoon and late evening, on each day of the week. The evening peak occurred after 20:00 and was, probably, due to the habits of the students as well as possible peak shifting operation of the battery. Notably, the smallest load occurred in the first half of the day where the PV-generator can produce power. Moreover, in summer, the inhabitants tended to spend more time outside and leave on vacation. At the same time, there is an abundant solar irradiation to fully supply the building with power during several hours and charge the battery. As a result, the net consumption in the first half of the day was often negligible and the evening peak was reduced.

We used this building to demonstrate the FNX-model considering weather-related exogenous variables. For this building, the standard profile is a poor representation of the load

[1] The data contains the net electricity consumption measured from 1st of July 2016 to 30th of June 2017. Of the available twelve months of data, we used three months (1st of January 2017 – 31st of March 2017) as a validation dataset while preserving the last three months (1st of April 2017 – 31st of June 2017) for the evaluation.

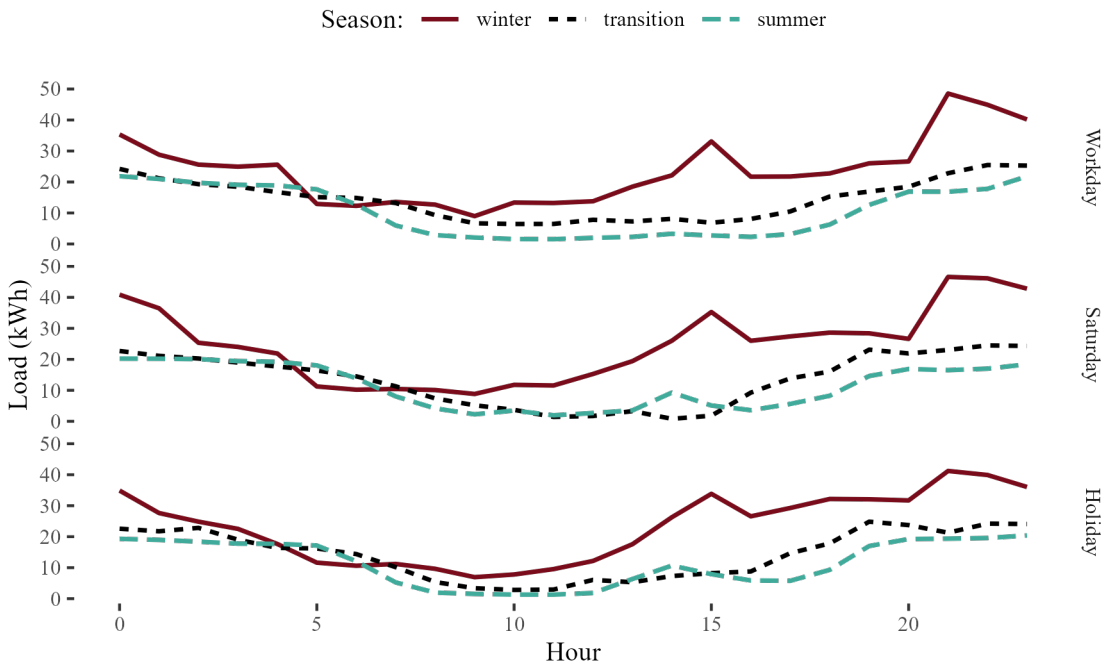

Figure 9.6: Average daily load curves of the smart building (Figure 9.5). In winter, there is a distinct consumption peak in the afternoon and evening. In summer and during the warmer months, the consumption is notably lower due to the installed photovoltaic generator.

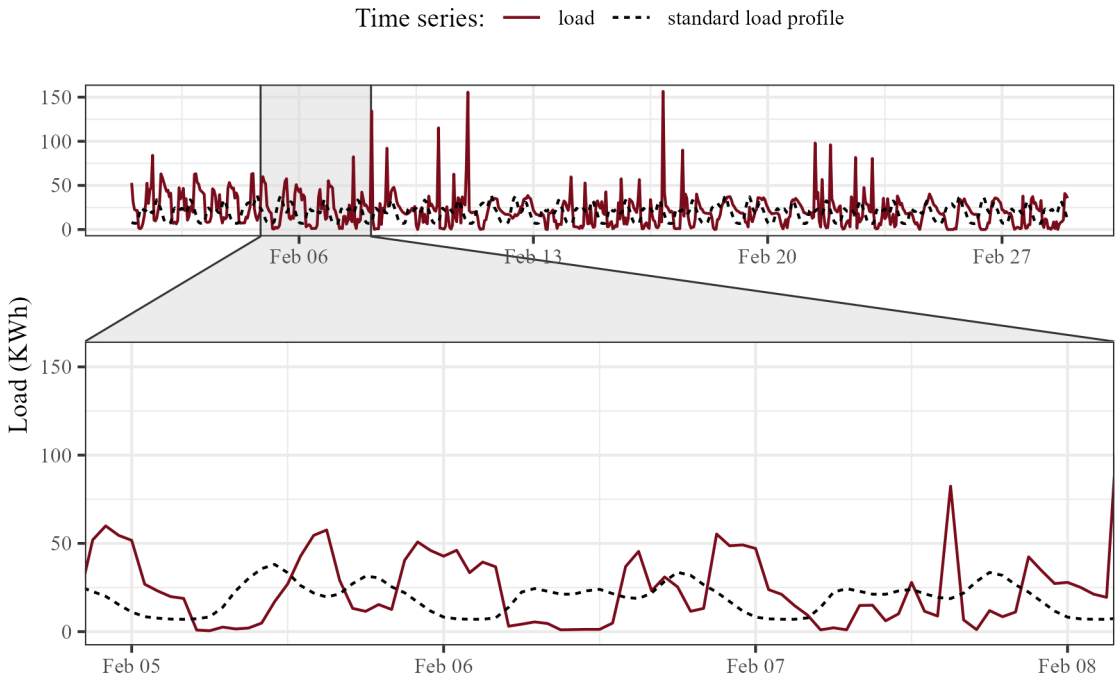

Figure 9.7: Net electricity consumption and standard load profile of the smart building (Figure 9.5). The standard profile appears to be a poor representation of the load due to an unusual consumption pattern that can be affiliated to the large photovoltaic and battery installation in the building.

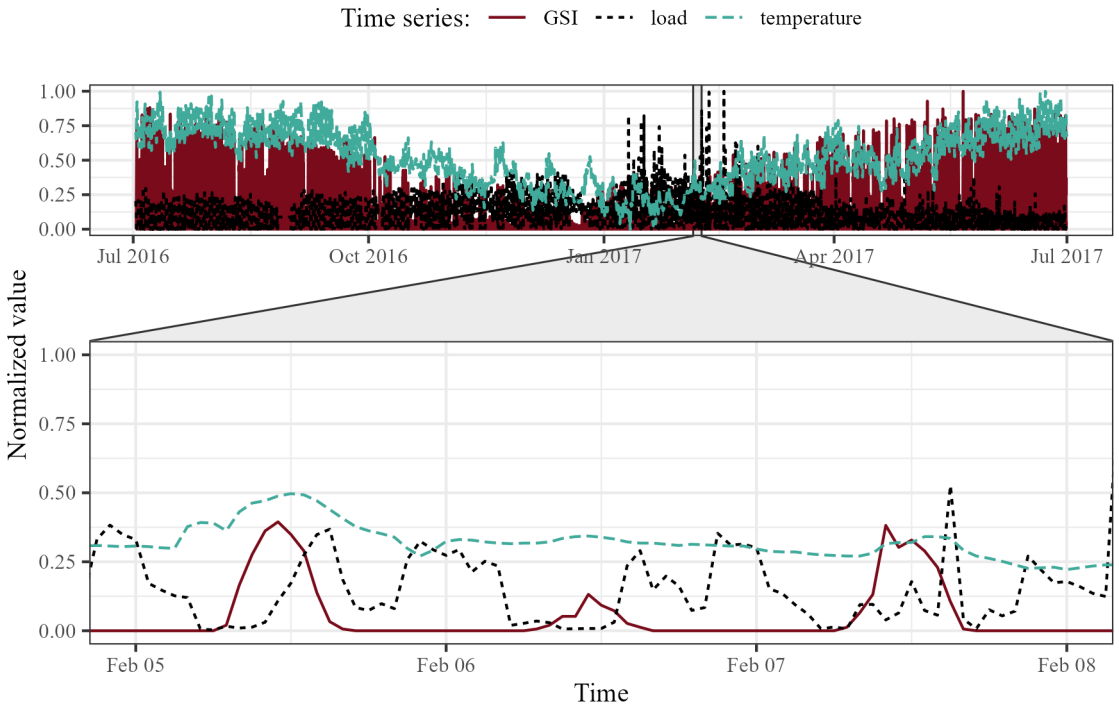

Figure 9.8: Electricity consumption of the smart building (Figure 9.5), outside ambient temperature and global solar irradiation measured at the neighboring weather station (7 km). The time series were normalized by the maximal values and resampled synchronously with 60-minute resolution.

due to the large PV and battery installation resulting in a net load pattern unusual for a residential building (Figure 9.7). For our study, we acquired the weather data from the neighboring weather station[2] [Zen]. In particular, we obtained the measurements of the outside ambient temperature and global solar irradiation which we also re-sampled with 60-minute resolution with the same sampling grid as the load time-series[3] (Figure 9.8).

We studied the dependency of electricity consumption on the outside ambient temperature and solar irradiation (Figure 9.9). The dependency on the outside temperature was small and can be explained by seasonal behavioral changes of the inhabitants. This modern building is heated thermally and is well-insulated which reduced the effect of daily temperature changes on its power demand. More notably, daily electricity consumption was affected by the solar irradiation. Therefore, we could expect the daily load curve to depend on solar irradiation and should consider it in our model. For the simulation, we assumed to have an ideal global solar irradiation forecast for the upcoming day, while in practice we would have to rely on a weather forecast[4].

[2] Weather station "Donaufeld" is located approximately 7 km away.

[3] Temperature was simply re-sampled, while for the GSI we computed the cumulative sum.

[4] This is a common assumption for a load forecasting simulation. Temperature forecasts can be very precise and we do not expect any loss of generality, similar to other researchers [CHL16, AVCMSR13, AVR16, ACV11, VCA12].

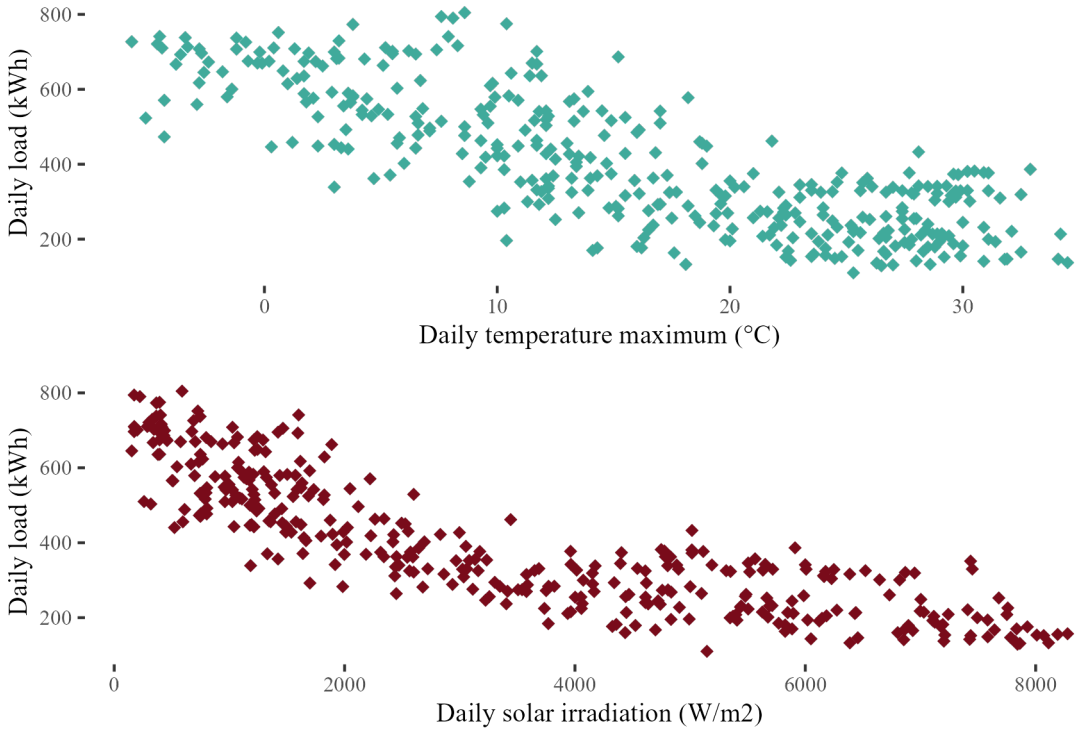

Figure 9.9: Dependency of total daily electricity consumption of the smart building (Figure 9.5) on the weather-related variables. The dependency on the outside ambient temperature was weak due to the thermal heating and insulation. At the same time, the dependency on the daily solar irradiation was more pronounced due to the large photovoltaic installation on the roof.

9.1.3 Computation Details

We simulated the loads using the MATLAB-software. On each load, the day-ahead load curve was predicted day after day on a rolling basis. The wide-scale building load forecasting simulation included the prediction of multiple local loads over numerous days and required considerable computation resources. Therefore, the computation was parallelized and run on the servers of Austrian Institute of Technology. The hyperparameters of the parametric models described in the next section that are not mentioned explicitly were left to MATLAB defaults. Data analysis was conducted with RStudio IDE.

Overall, the wide-scale day-ahead building load forecasting simulation included 283,203 daily forecasts[5] computed with each model that were evaluated in Chapter 10. Additionally, for the validation of each model we computed 30,000 daily forecasts. In particular, we validated the design decisions discussed in Section 8.2. Moreover, we used the validation dataset to parametrize various reference models discussed next.

[5] We computed 153 daily load curve forecasts for each of the 1851 loads of the extended smart meter dataset.

9.2 Reference Models

In this section, we describe the setup and validation of the existing models that we evaluated in the wide-scale day-ahead building load forecasting simulation. In particular, we selected various *reference models* that can be commonly encountered in the load forecasting literature (Chapter 5). Further, we used the validation dataset (Section 9.1.1.3) to find the most accurate setups and manually fine-tune these models before evaluating them together with the proposed functional neighbor forecaster. In this study, we compared the functional neighbor model described in the last chapter to various common approaches considering three families of models: *heuristic* (Section 9.2.1), *parametric* (Section 9.2.2), *nonparametric* (Section 9.2.3). Subsequently, we describe these reference models and use the validation dataset to pre-select the most accurate approaches. Further, we manually fine-tune their parameters and setups before applying them to the evaluation dataset for which the results are presented in the next chapter.

9.2.1 Heuristic Models

There are several load modeling heuristics that can be effective for predicting building loads. These models are often used as benchmarks, yet they can be sometimes more accurate even than more sophisticated approaches [HGZA18].

9.2.1.1 Profiling Heuristics

Standard Load Profile (SLP) Predicting power consumption using standard profiles is the method that is currently used for load forecasting in the distribution systems. It one of the oldest approaches that has been used since the establishment of wide-scale power systems. This method is remarkably accurate for larger aggregations, yet often ignored by the modern load forecasting literature.

The standard profiles are defined by the corresponding national entities. We provide some examples in Figure 9.10. The definition includes profiles for different consumer groups with the annual consumption of 1000 kWh. This method allows to predict the load in a distribution system for which we have to know its consumer group and annual consumption of the individual consumers. When given a load aggregation, the SLPs of individual loads are aggregated accordingly (e.g., 70 households, and 30 SMEs)[6].

[6] For our purposes, we normalized the SLPs by the maximum value of the corresponding load. Further, SLPs are scaled by the predicted annual consumption for the upcoming year. We assume that this prediction is ideal taking the actual annual consumption (i.e. using test set to calculate it).

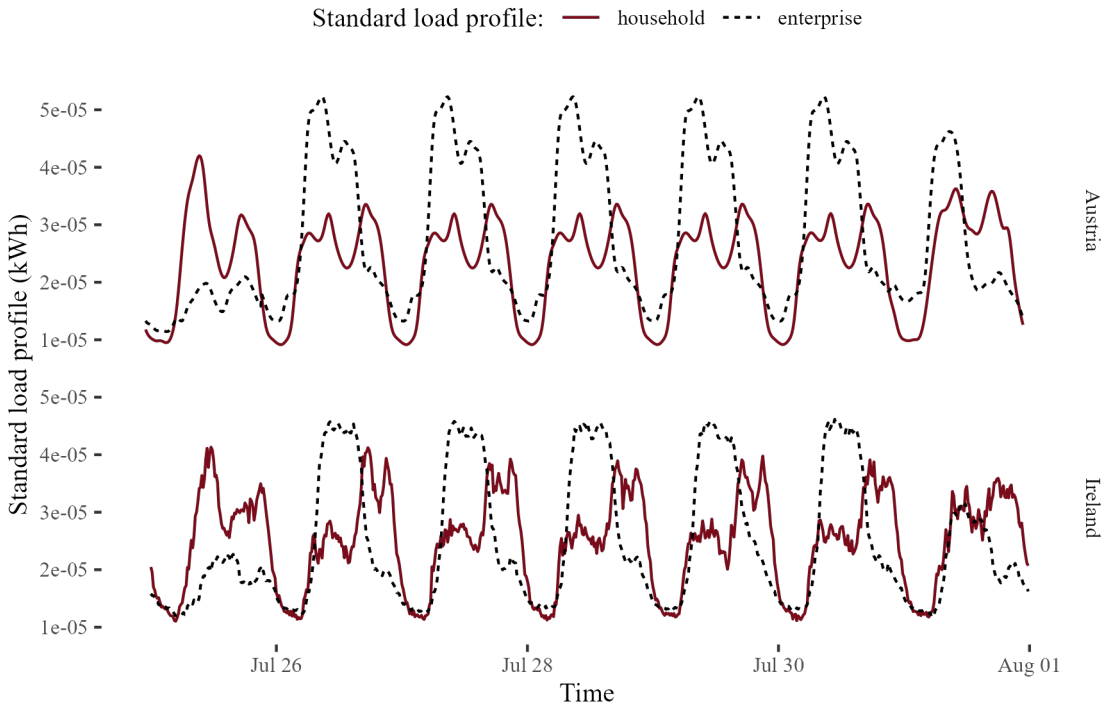

Figure 9.10: Standard load profiles of various end-consumers as defined by the national entities in Austria [Syn] and Ireland [Iri14]. The profiles are presented for the last week in July 2010 and were defined for the loads with 1000 kWh of annual consumption.

Individual Load Profile (ILP) Wide-scale introduction of smart meters allows for a trivial improvement on the SLP-heuristic by creating individual profiles for each consumer [BPT13]. Similar to the SLP-approach, after collecting historical data of the consumer, we group daily load curves by season (summer, winter, Transition) and day-type (workday, Saturday and holiday). For each group, we compute the average curve obtaining 9 ILPs for the consumer (Figure 9.11). The load is forecast by the ILP corresponding to the upcoming day.

Validation: Profiling Heuristics Forecast accuracy can be significantly improved at each level of aggregation if instead of SLPs we use the individually created profiles (Figure 9.12). Consumers can exhibit load curves that do not fit the standard load profiles. For instance, we can often find businesses that have unusual opening hours that do not follow the workday calendar. Though such deviations can be considered to some extent by defining numerous standard profiles[7], individual profiles are better at accounting for such specialties.

[7] For instance, Austrian Power Clearing and Settlement organization defines 27 profiles [Syn].

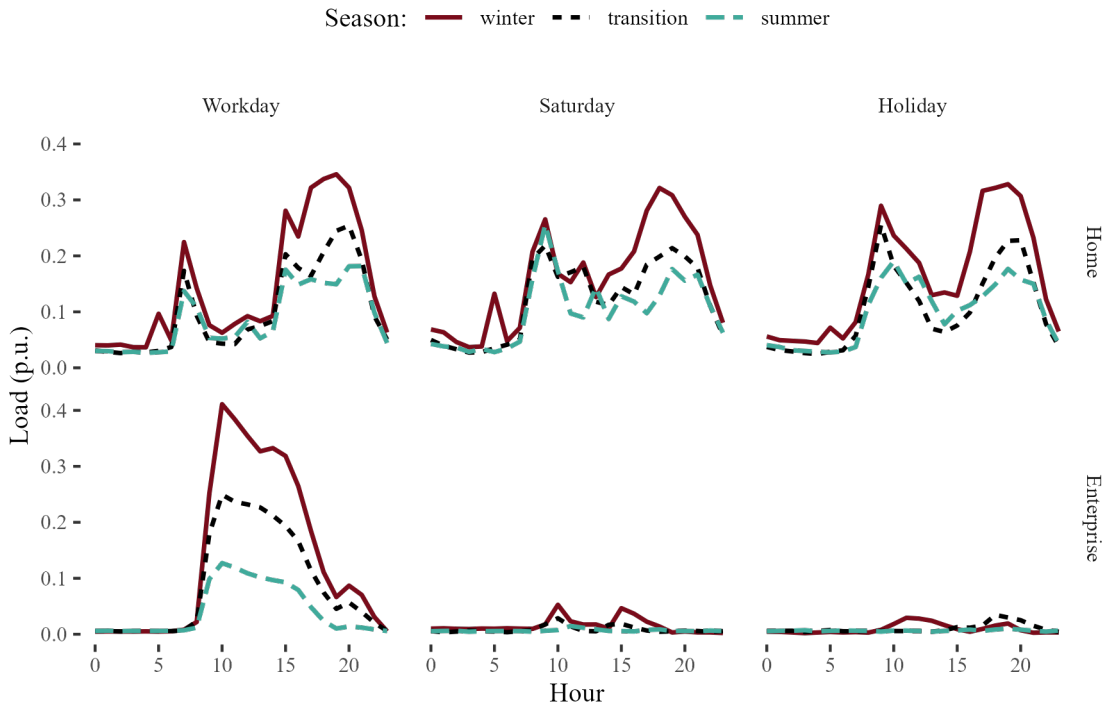

Figure 9.11: Individual load profiles computed for a commercial (top) and a residential building from the ICER smart-meter dataset [Arc16]. Each profile was calculated by averaging the historical daily load curves of the corresponding season and day-type.

9.2.1.2 Persistence Heuristics

Naive Model (D-1) With this heuristic, we take the most recent load curve as a forecast. Naive forecast is often denoted as D-1 model and is one of the most common benchmarks used in the literature [ACGW18, FRS+13].

Weekly Persistence (D-7) Assuming ideal weekly persistence of the load, we predict the consumption using the most recent observations of the same weekday. This simple method can be very effective for commercial loads as it considers weekly seasonality. Such forecast is often denoted as D-7 model and is another common benchmark together with D-1 [ACGW18, FRS+13].

Validation: Persistence Heuristics We compare naive (D-1) and weekly persistence (D-7) forecasts in Figure 9.13. Naturally, the latter is more accurate for larger loads that have stronger weekly seasonality. For enterprises and aggregations, the improvement can be substantial on Mondays, Fridays, Saturdays and Sundays. At the same time, there is little difference among the heuristics if applied in the middle of the week.

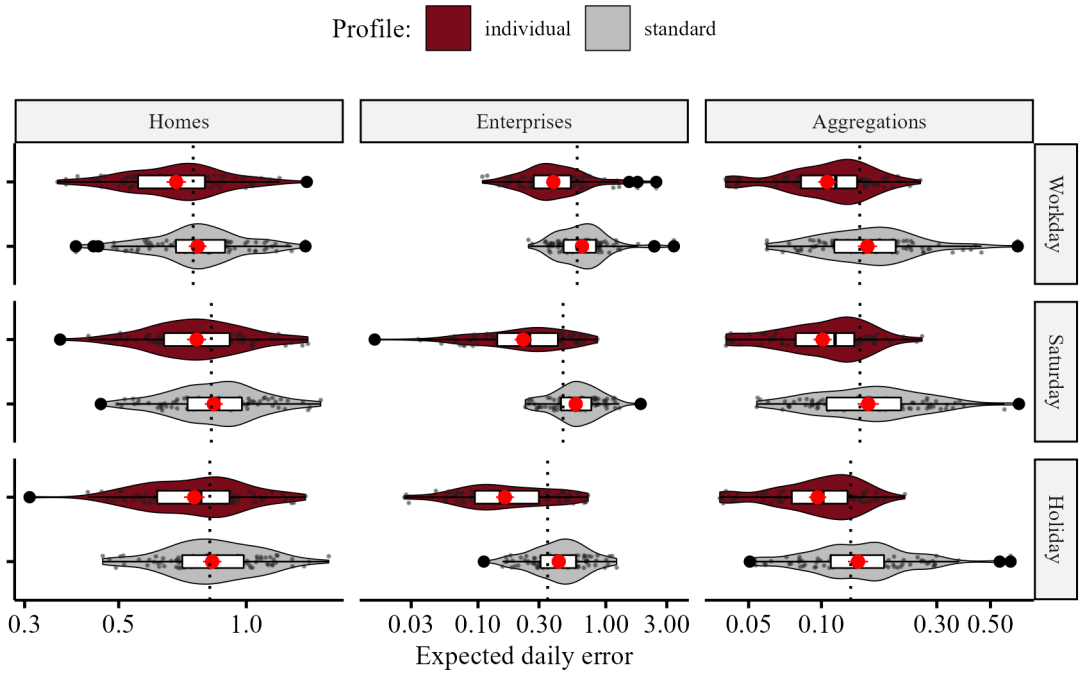

Figure 9.12: Comparison of profiling heuristic models. Standard and individual load profiles (Section 9.2.1.1) predicted the 300 loads in the validation dataset (Section 9.1.1.3) day-by-day for 100 consecutive days. For each day-type, we computed the *expected daily errors (EDE)* according to (7.15). The figure shows the 900 EDE-observations confounded on day and load type (grey dots), including the outliers (black dots). Violin and box-plots summarize the EDE-distributions in each groups. The average EDE of the model in each group (red dot) is shown together with 95%-confidence interval (red bar). We see that the individual load profile forecast was, on average, significantly ($p < 0.05$) more accurate for each type of loads and days.

9.2.2 Parametric Models

We applied various common parametric models as a reference in the wide-scale day-ahead building load forecasting simulation. Parametric regression is the most widely used approach as it includes the majority of the state-of-the-art models proposed for load forecasting (Chapter 5). There is a myriad of different parametric forecasting methodologies and for our evaluation we selected two that are the most common [ACGW18]: ARIMA and *artificial neural networks (ANN)*. In this section, we describe the reference models based on these methodologies and validate their setups for the day-ahead building load forecasting.

Parametric models require a preliminary training step which we repeat every month on the rolling basis to account for the concept change [SK12]. Model training is more efficient

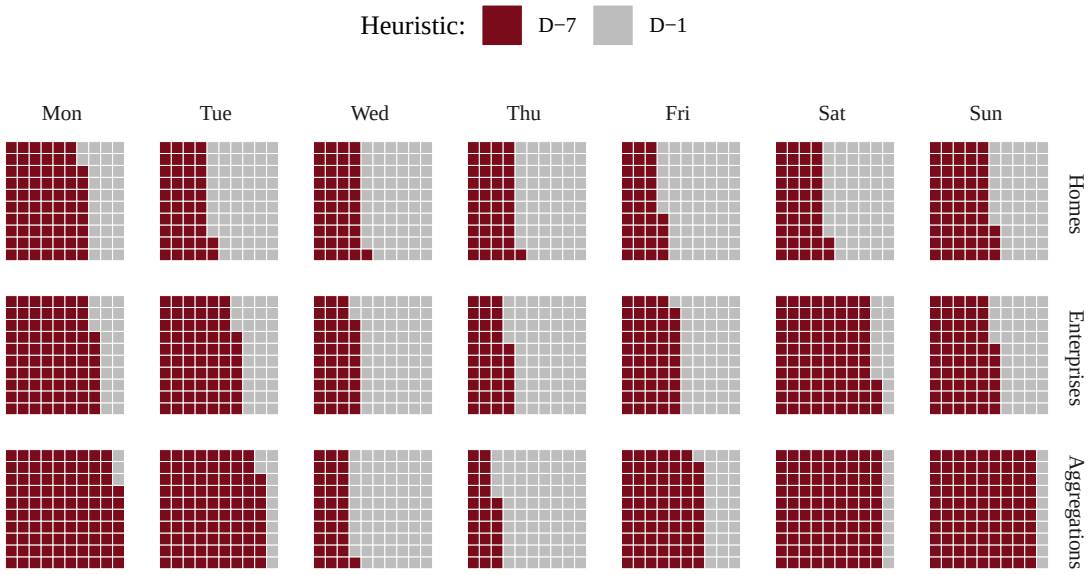

Figure 9.13: Comparison of the persistence heuristic forecasts. We applied the naive (D-1) and the weekly (D-7) persistence heuristic models (Section 9.2.1.2) to predict the 300 loads of the validation dataset (Section 9.1.1.3). Each load was predicted day-ahead for 100 consecutive days (23 April 2010 – 31 July 2010). Conditioning on load type (panel row) and weekday (panel column), we represent each individual load by a square filled depending on the model that provided the smallest expected daily error (7.15) on the days of the corresponding load type and weekday. Notably, there is a smaller difference in forecast accuracy between the heuristics in the middle of the week (Wednesday, Thursday) since the end-consumers often follow similar behavioral patterns during the week. The difference becomes more apparent around the weekend (Sunday, Monday) where the weekly seasonality becomes particularly prominent. Moreover, the weekly persistence heuristics was notably more accurate on the loads with stronger weekly seasonality (enterprises, aggregations).

with normalized inputs [DFH97b]. Following a standard practice, we apply *minmax-normalization* as follows. Each separate input $x^{(j)}$, with $j \in [1, \cdots, n_x]$ is transformed to

$$\tilde{x}^{(j)} = \frac{2\left(x^{(j)} - x^{(j)}_{\text{min}}\right)}{x^{(j)}_{\text{max}} - x^{(j)}_{\text{min}}} - 1, \tag{9.2}$$

where the smallest input value $x^{(j)}_{\text{min}}$ corresponds to $\tilde{x}^{(j)} = -1$ and the largest input $x^{(j)}_{\text{max}}$ corresponds to $\tilde{x}^{(j)} = 1$. Herewith, every normalized training input lies in the range of $[-1; 1]$. We use the same normalization constants $x^{(j)}_{\text{min}}, x^{(j)}_{\text{max}}$ for the evaluation.

Subsequently, model output must be transformed back to the range of original data. Hence, we provide the models with a pre-processing and a post-processing block (Figure 9.14).

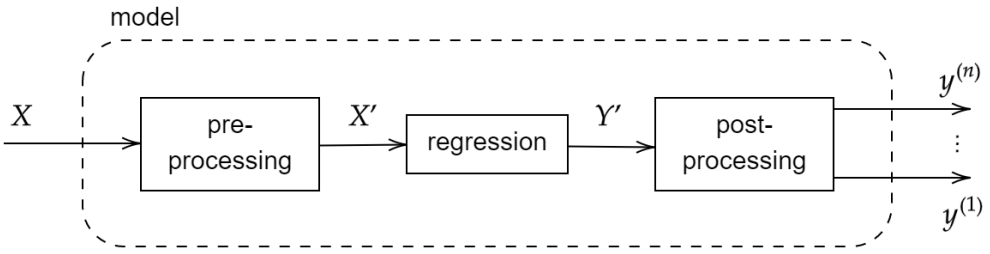

Figure 9.14: Input-output processing for parametric models.

Additionally, for recursive models[8], post-processing includes a *hold and release* unit. In their case, the output $\hat{y}$ is a scalar prediction for a single time step. For such models, post-processing allows to output the entire forecast curve at once.

9.2.2.1 Autoregressive Integrated Moving Average (ARIMA)

At higher load aggregation levels, ARIMA-models are among the most common[9]. We applied the methodology described in Section 4.1.1 to create ARIMA-models that were used as a reference. The setup of two different ARIMA-models is described in this section.

In general, ARIMA is a univariate autoregressive model that considers historical load observations as the only input. We account for the annual temperature cycle by using only two most recent months of training data. Further, we considered weekly seasonality implicitly by predicting each weekday separately. To predict the entire load curve day-ahead, we applied two different multistep strategies.

ARIMA-D Using the direct multistep strategy, a separate ARIMA was trained to predict each point of Y_{d+1}. For ease of exposition, we set up each single-step model with the same hyperparameters p, d, q described further in the text. The training output data was split into n separate series $y^{(1)}, \cdots, y^{(n)}$, so that the n'th model had different weights after the training. The forecast output by the model consisted of n separate predictions in

$$\hat{Y}_{d+1} = \left[\mathbf{r}^{(1)}\left(X_{d+1}\right), \ldots, \mathbf{r}^{(n)}\left(X_{d+1}\right)\right]. \tag{9.3}$$

We selected the three hyperparameters $p = 1, d = 1, q = 1$ through trial and error (Figure 9.15). In particular, we observed that the smallest model with $p = 1$ was the most accurate. This is consistent to the setups found in the literature (Chapter 5).

[8] In out study, recursive models include NAR, NARX, and ARIMA-R that are described further in the text.

[9] This also included the variants such as SARIMA, ARIMAX etc. See the discussion in Chapter 5.

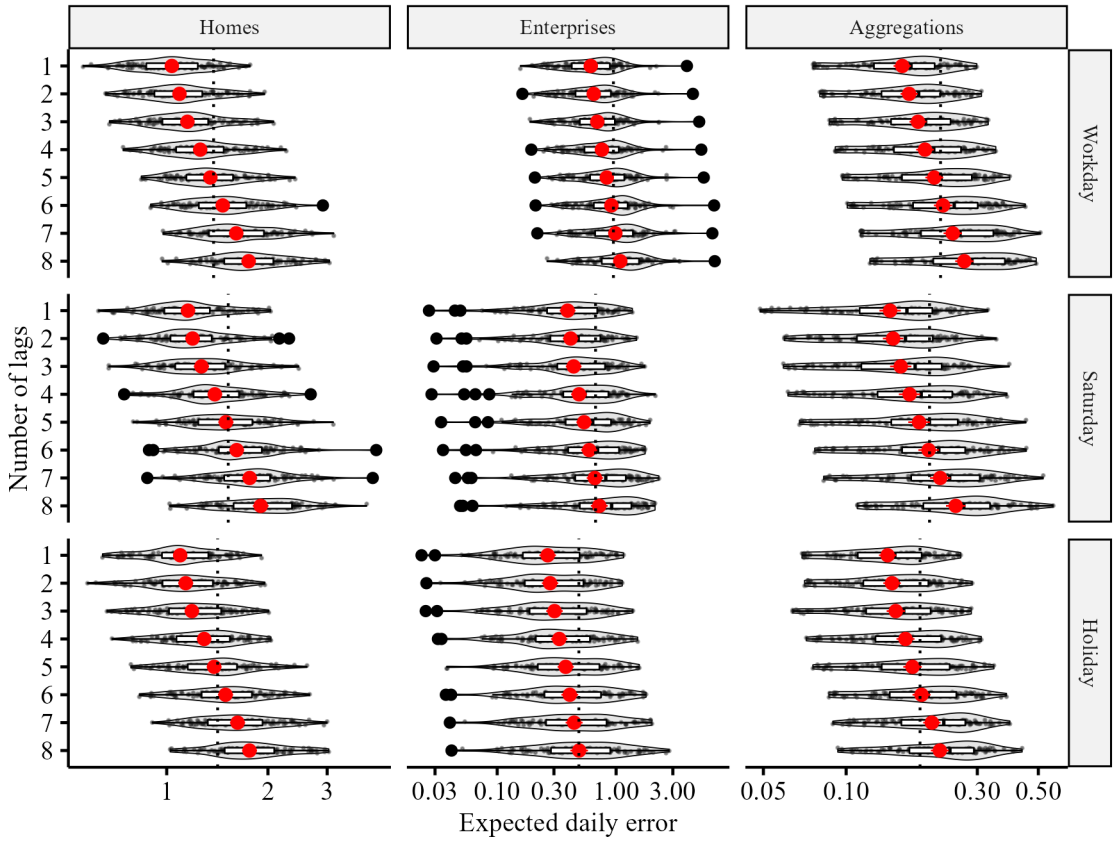

Figure 9.15: Forecast errors of different ARIMA-D-model variants. Each model variant with a different value of the parameter p (Section 9.2.2.1) predicted the 300 loads in the validation dataset (Section 9.1.1.3) day-by-day for 100 consecutive days. For each day-type, we computed the *expected daily errors (EDE)* according to (7.15). The figure shows the 900 EDE-observations confounded on day and load type (grey dots) including the outliers (black dots). Violin and box-plots summarize the error distribution in each of the groups. The average EDE of the model in each group (red dot) is shown together with 95%-confidence interval (red bar). We observed that the variant with $p = 1$ was significantly ($p < 0.05$) more accurate than other variants.

ARIMA-R Additionally, we set up a recursive ARIMA-model (ARIMA-R). It predicts the daily curve step by step while past output values are fed back to the input that considers up to p lags so that a prediction

$$\hat{y}_i = \mathbf{r}\left(y_{i-1}, \ldots, y_{i-p}\right) \tag{9.4}$$

is a function of the p preceding values $y_{i-1}, \ldots, y_{i-p}$ of the time series y.

Again, we selected the three hyperparameters $p = 24, d = 1, q = 1$ through trial and error. In particular, we observed that the most accurate model had to consider only one or two days of observations – i.e., ARIMA-R achieved the best accuracy either with $p = 24$ or $p = 48$ (Figure 9.16). This is consistent to the setups found in the literature (Chapter 5).

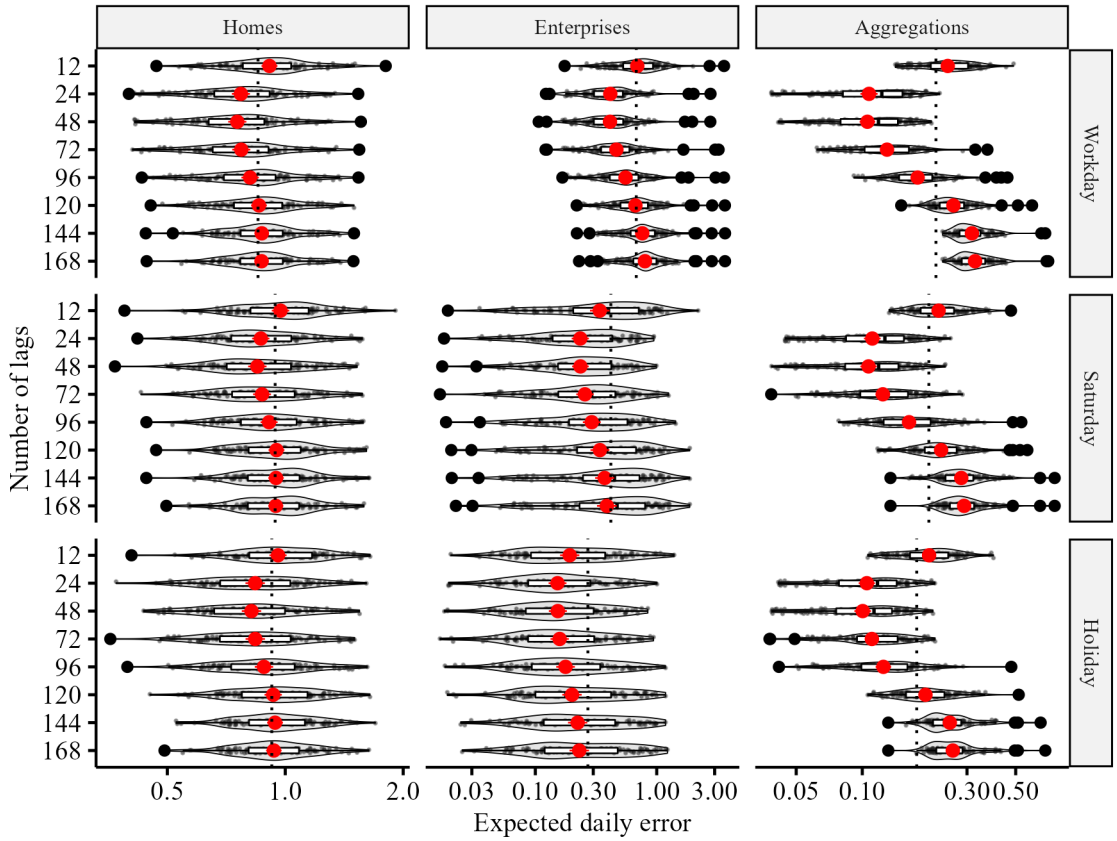

Figure 9.16: Forecast errors of the different ARIMA-R-model variants. Each model variant with a different value of the parameter p (Section 9.2.2.1) predicted the 300 loads in the validation dataset (Section 9.1.1.3) day-by-day for 100 consecutive days. For each day-type, we computed the *expected daily errors (EDE)* according to (7.15). The figure shows the 900 EDE-observations confounded on day and load type (grey dots), including the outliers (black dots). Violin and box-plots summarize the EDE-distributions in each of the groups. The average EDE of the model in each group (red dot) is shown together with 95%-confidence interval (red bar). We observed that the variant with $p = 24$ and $p = 48$ was significantly ($p < 0.05$) more accurate than other variants.

Models larger than $p = 72$ tended to overfit, especially when applied on larger and more regular loads.

Validation: ARIMA Multistep Strategy Comparing both ARIMA-models on the validation dataset, we found that ARIMA-R was significantly ($p < 0.05$) more accurate than the direct ARIMA for all load groups (Figure 9.17). Nevertheless, we used both models as a reference in the wide-scale day-ahead building load forecasting simulation.

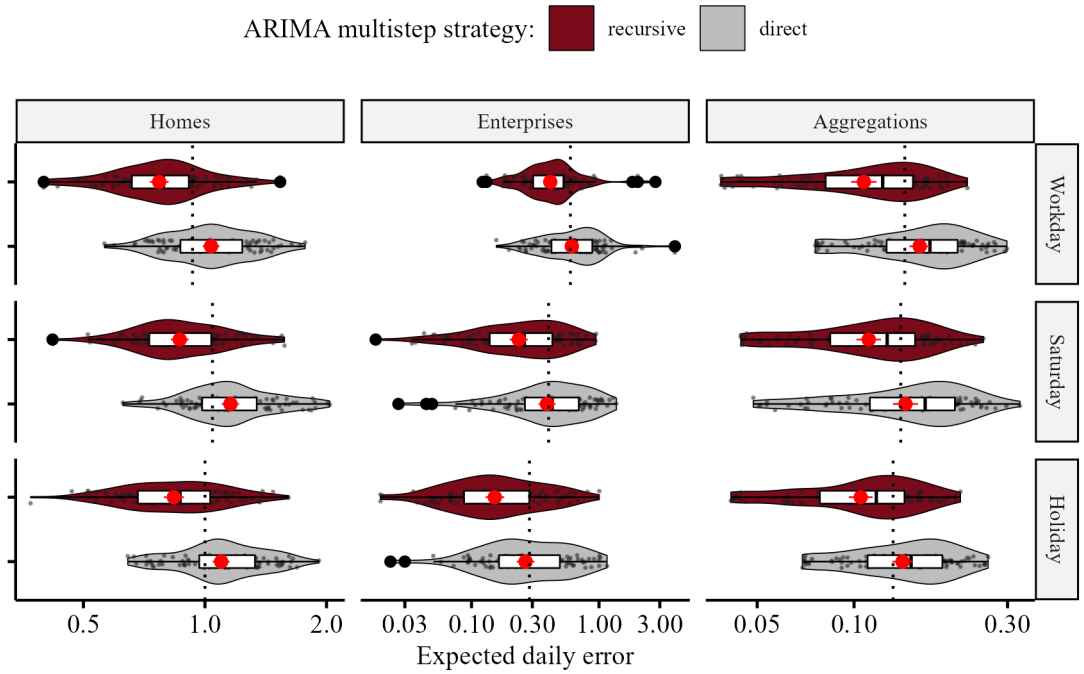

Figure 9.17: Comparison of the ARIMA-models using direct and recursive multistep strategies. The ARIMA-model (Section 9.2.2.1) using either direct (ARIMA-D) or recursive (ARIMA-R) strategy predicted the 300 loads in the validation dataset (Section 9.1.1.3) day-by-day for 100 consecutive days. For each day and load type, we computed the *expected daily errors (EDE)* according to (7.15). The figure shows the 900 EDE-observations confounded on day and load type (grey dots), including the outliers (black dots). Violin and box-plots summarize the EDE-distributions in each of the groups. The average EDE of the model in each group (red dot) is shown together with 95%-confidence interval (red bar). We observed that the recursive strategy was significantly ($p < 0.05$) more accurate than the direct multistep strategy.

9.2.2.2 Artificial Neural Networks (MLP, NARX, NAR)

Neural-network-based models became a popular parametric regression application for the load forecasting in recent years. Currently, there exist numerous different approaches in the forecasting literature (Chapter 5). Nevertheless, we have not seen any fundamental reason why any of the networks is superior for all or certain types of loads, given appropriate setup and vigor at manual fine-tuning [BZN+19]. Moreover, many of the proposed building load forecasting models appear unpractical for a wide-scale local load forecasting problem [VKS20]. In particular, we abstained from using network architectures that rely on abundant historical data or information from specific sensorial equipment (e.g. occupancy) since these data might not be widely available.

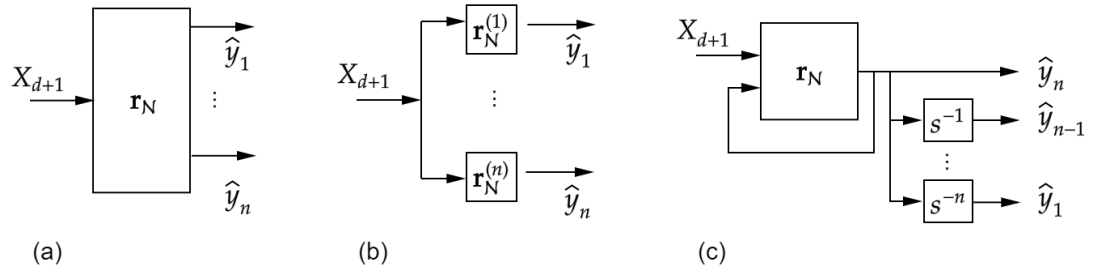

Figure 9.18: Network architectures for the day-ahead prediction. (a) MLP-D model using direct strategy; (b) MLP-M model using multi-out strategy; (c) NARX-model with external input using recursive strategy. Further description is provided in the text.

We used various network architectures[10] to create two feedforward, two recurrent and two deep neural networks for the day-ahead load forecasting (Table 9.1). In the rest of this section we discuss the choice and setup of the reference models based on neural network methodology. In particular, we used various MLP-networks to compare different multistep strategies and NAR-network to evaluate implicit seasonality modeling. These networks are discussed below.

Multi-Layer Perceptron (MLP) The feedforward network presented in Figure 4.2 is, by far, the most common architecture among numerous load forecasting applications (Chapter 5). It can theoretically model any complexity (Section 4.1.2.1) and we evaluated if it has enough modeling capacity or if we have to use more sophisticated architectures for the local load forecasting application. Since an MLP can be extended to become a recurrent network, we also used it to compare different multistep strategies (Figure 9.18).

Multilayer perceptron (MLP) is a feedforward network which can forecast the day-ahead load curve adopting either direct or multi-out strategy. Given the input

$$X = \{x^{(1)}, \cdots, x^{(n_x)}\} \text{ and } X_{d+1} = [x_1, \cdots, x_{n_x}], \tag{9.5}$$

we defined the following multivariate forecasting models:

1. MLP-D (direct multistep strategy)
2. MLP-M (multi-out multistep strategy)

In the first case, a separate MISO MLP was trained to predict each point of Y_{d+1} (Figure 9.18.a) – just as we did with ARIMA-D-model. For ease of exposition, we set up each

[10] General neural network methodology, model types and architectures were discussed in Section 4.1.2.1. We discuss the setup of neural-network-based models for a wide-scale day-ahead local forecasting in our previous publication [VKS20].

network with same inputs and hyperparameters described further in the text. The training output data was split into n separate series $y^{(1)}, \cdots, y^{(n)}$, so that the n networks had different weights after the training. The forecast output by the model consisted of n separate predictions constituting the curve

$$\hat{Y}_{d+1} = \left[\mathbf{r}_{\mathcal{N}}^{(1)}\left(X_{d+1}\right), \ldots, \mathbf{r}_{\mathcal{N}}^{(n)}\left(X_{d+1}\right)\right]. \tag{9.6}$$

In the second case, we trained one multi-out feedforward network with n_x inputs and n outputs. For a given X_{d+1}, the multivariate forecast was obtained as

$$\hat{Y}_{d+1} = \mathbf{r}_{\mathcal{N}}\left(X_{d+1}\right) \tag{9.7}$$

with one MLP $\mathbf{r}_{\mathcal{N}}$ that we had to train (Figure 9.18.b).

Nonlinear Autoregressive Model with Exogenous Input (NARX) The concept of a recurrent network, where past output values are fed back, allows to create nonlinear autoregressive time series models. In fact, an MLP-architecture can be used to set up a network that presents a multivariate *nonlinear autoregressive model with exogenous inputs (NARX)* denoted as follows:

$$\hat{y}_i = \mathbf{r}_{\mathcal{N}}\left(y_{i-1}, \ldots, y_{i-p}, X_i\right). \tag{9.8}$$

Here, a prediction $\hat{y}_i$ is calculated as a function of its p lags $y_{i-1}, \ldots, y_{i-p}$ and an exogenous input X_i (Figure 9.18.d). Note that the NARX-model corresponds to an MLP combined with recursive multistep strategy computing the day-ahead forecast.

Nonlinear Autoregressive Model (NAR) We applied another recurrent network architecture to create a univariate *nonlinear autoregressive model (NAR)* that we used as a reference in the wide-scale day-ahead building load forecasting simulation. The univariate time series model computes the prediction

$$\hat{y}_i = \mathbf{r}_{\mathcal{N}}\left(y_{i-1}, \ldots, y_{i-p}\right) \tag{9.9}$$

as a function of the p preceding values $y_{i-1}, \ldots, y_{i-p}$ (lags) of the time series y (Figure 9.19). In contrast to NARX, this model cannot consider exogenous variables and has to model the seasonality implicitly, as we explain further in the text.

Deep Neural Networks (DNN) Deep neural networks (DNN) are considered as the state of the art in machine learning research and include neural network architectures that

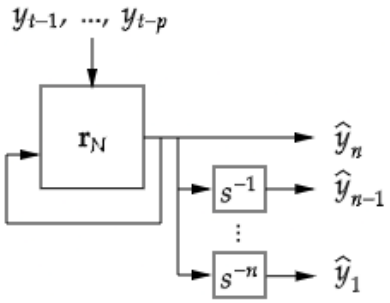

Figure 9.19: NAR-model. Further description is provided in the text.

Table 9.1: Reference models based on neural-network methodology.

	MLP-D	**MLP-M**	**NAR**	**NARX**	**DNN-2**	**DNN-3**
Type	feedforward	feedforward	recurrent	recurrent	feedforward	feedforward
Multistep strategy	direct	multi-out	recursive	recursive	multi-out	multi-out
Num. inputs	28	28	168	28	28	28
Num. outputs	1	24	1	1	24	24
Hidden layers	1	1	1	1	2	3
Hidden neurons	15	15	15	15	30	45
Num. weights	79	424	2703	463	1033	1258
Num. training data	365	365	60	365	365	365

have two or more hidden layers that might be combined with a complex topology [GBC16]. We applied two deep-neural-network models with two (DNN-2) and three (DNN-3) hidden layers that are based on the MLP architecture with multi-out strategy.

Setup of Neural-Network-Based Models In Table 9.1, we summarized the networks used in this study with corresponding number of inputs, outputs, training data (historical daily load curves) and degrees of freedom (total number of weights). Subsequently, we describe their setup and the choice of hyperparameters.

Each network modeled the daily seasonality *implicitly*. For the feedforward architectures (MLP-D, MLP-M, DNN-2, DNN-3), we input the load curve of the previous day. With the recursive architectures (NARX, NAR), we used 24 and more lags to learn the daily patterns from the historical data.

The *multivariate* networks (MLP-D, MLP-M, DNN-2, DNN-3, NARX), modeled the weekly seasonality and the annual cycle *explicitly*. The weekly seasonality was considered with a *weekday number* and a (boolean) *public holiday* variables used as inputs. Moreover, we had discovered that the annual temperature cycle can be modeled as a function of *month*

and *day* number [VKS20]. We used these variables as the network inputs and assumed that the short-term weather changes did not impact the daily load notably as it was done in [HBA+14].

Alternatively, the *univariate* network (NAR) modeled the weekly seasonality *implicitly* by considering one week of lags ($p = 168$). The annual cycle was accounted for also *implicitly*, retraining the model every month using only the most recent 60 days of data.

Training data might be limited in our application. Therefore, for a wide-scale prediction of local loads it is important to rely on the smallest viable amount of historical data. In addition to the availability issue, time series can constantly alter their regime with old data becoming irrelevant (inhabitants change, new equipment is installed). Multivariate networks required at least one year of data to learn the dependency on the month. Currently used SLPs also require total consumption over a year to scale the profile. In contrast, the NAR-model was trained only on the most recent 60 days of preceding data.

Training algorithm for each network combined the Levenberg-Marquardt algorithm with Bayesian regularization [DFH97b]. The Levenberg-Marquardt algorithm appears to be much faster than backpropagation-based approaches for moderately-sized networks [HM94]. Bayesian regularization does not keep some of the limited training data as a validation dataset in contrast to the commonly used early stopping technique [FHT08]. At the same time, it is effective preventing overfitting and improving generalization of moderately over-sized networks [DAM03]. On each load, and with each architecture, ten networks were trained and their outputs were averaged to a single forecast mitigating the stochastic nature of the training results. We retrained the networks every month using the preceding historical data for training on a rolling basis.

Activation function of the neurons must correspond to the applied normalization. With $x^{(j)} \in [-1; 1]$, each layer should have $\phi(u)$ with the same domain and range. Unfortunately, the majority of related works (Table 5.1) do not specify the activation function [MHD+13, AKZ10, HBA+14, MNGK16, POC+17, HP16, MAM16b]. Among the rest, the most common functions are *linear* [RNK16, AMM17, LSPB+12, MRCA14] and *tanh-sigmoid* [BFS+15, SLW16, KDJ+17] defined as

$$\phi(u) = \frac{2}{1 + exp(-2u)} - 1 \tag{9.10}$$

and which we used in this study since $\phi(u) \in [-1; 1], \forall u \in \mathbb{R}$.

Network size determines the predictive capacity of an ANN-model. In this study, we used the networks where each hidden layer consisted of 15 neurons. The choice is consistent with other building load forecasting applications where ANN-models often had one hidden layer with a similar number of neurons (Table 5.1). The optimal number of hidden layers

and neurons depends on the task and there is no theoretical methodology to determine these hyperparameters. It is a common practice, to choose the network size according to the experience in comparable problems and, then, to manually fine-tune the setup on a validation dataset.

Validation: Neural Network Size The NAR-model using only 60 days of data for training allowed to validate the choice of network size. This autoregressive model requires an increased complexity as it considers 168 lags, however, we observed that 15 neurons were enough to do so (Figure 9.20). Larger network variants (25 and 30 neurons) tended to overfit on aggregations that had a more regular electricity consumption time series. In contrast, the network with 15 neurons was among the most accurate in every load group. Ideally, number of neurons and other hyperparameters should be set automatically for each individual load. However, if we were to have an excessive modeling capacity, the Baysian regularization would prevent overfitting to a large extent. Therefore we can assume 15 neurons to be an adequate size for our application.

9.2.3 Nonparametric Models

We applied various common nonparametric models as a reference in the wide-scale day-ahead building load forecasting simulation. Nonparametric models such as Nadaraya-Watson estimator or K-nearest neighbors are often used as benchmarks in the load forecasting literature (Chapter 5). In this section, we describe the reference models based on nonparametric regression methodology and validate their setups for the day-ahead building load forecasting.

9.2.3.1 Nadaraya-Watson Estimator (NWE)

Nadaraya-Watson estimator (NWE) is the canonical nonparametric regression model (Section 4.2.2.1). We applied this model predicting the load curve as a weighted average of historical observations. For this reference model, we used Gaussian kernel[11] and considered the annual cycle by limiting the historical data to the 17 most recent weeks[12]. We modeled the weekly seasonality *implicitly* filtering historical observations by weekday. To model the daily seasonality, we applied the direct multistep strategy – the load curve consisting of q points was predicted by q separate univariate estimators. Therefore, each

[11] Gaussian kernel is a common choice in the relevant literature (Chapter 5).

[12] We chose this history length because ILP use the same amount of data to compute the profile of the corresponding season [BPT13]. This choice will also allow us to compare ILP with UA and NWE.

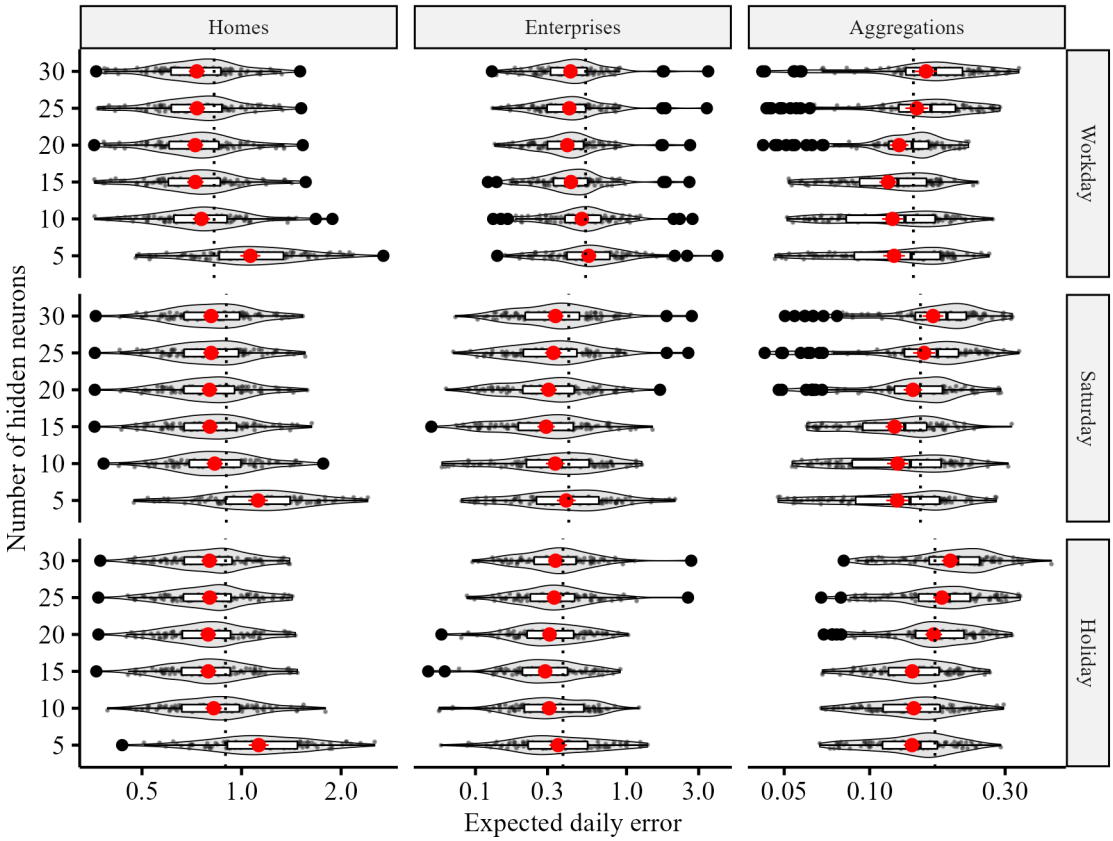

Figure 9.20: Forecast errors of the different NAR-models of different size. Each model variant with different size of the hidden layer (Section 9.2.2.2) predicted the 300 loads in the validation dataset (Section 9.1.1.3) day-by-day for 100 consecutive days. For each day-type, we computed the *expected daily error (EDE)* according to (7.15). The figure shows the 900 EDE-observations confounded on day and load type (grey dots), including the outliers (black dots). Violin and box-plots summarize the EDE-distributions in each of the groups. The average EDE of the model in each group (red dot) is shown together with 95%-confidence interval (red bar). We observed that the network with 15 hidden neurons delivered one of the best forecasts in each group.

model predicted a single point of the curve with its own fixed bandwidth b_i according to (4.57). The bandwidth was computed using (4.39).

Validation: Fixed Bandwidth Search For each model, we used Bowman's plug-in method (4.39) to select the bandwidth b_i. This common method provides an approximation for the optimal bandwidth choice. We observed that the models which set the bandwidth minimizing the leave-one-out criterion (4.44) were as accurate as those relying on the plug-in method (Figure 9.21). At the same time, the plug-in method required substantially less computation which is advantageous for a wide-scale simulation.

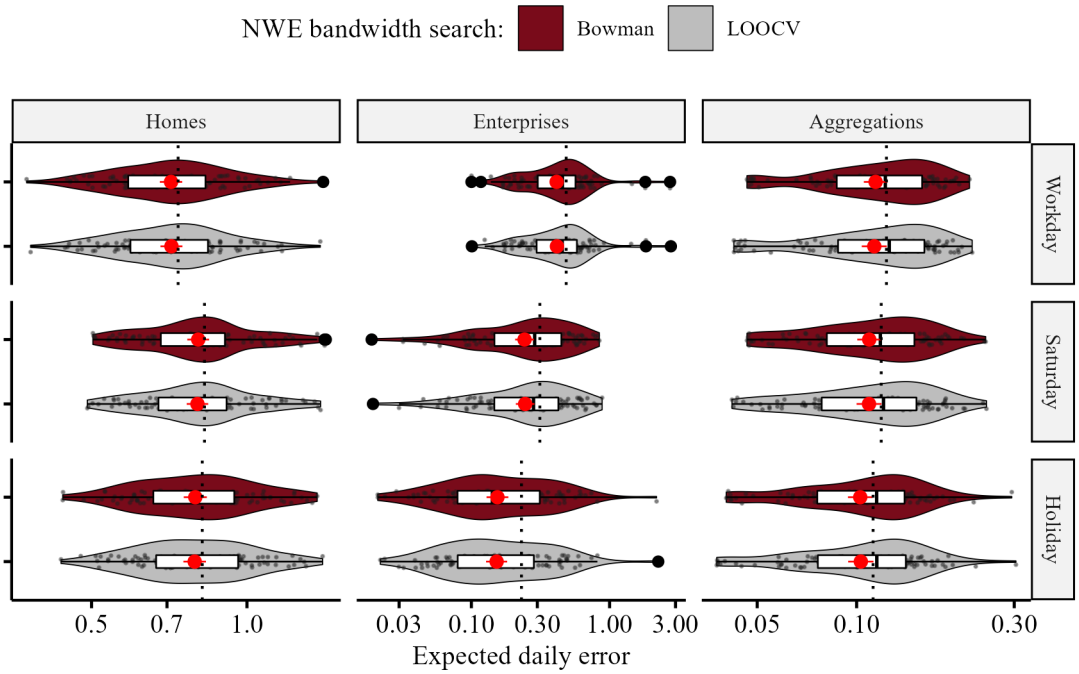

Figure 9.21: Forecast errors of the NWE-models with bandwidth found using either Bowman's plug-in method or minimizing the leave-one-out cross-validation criterion (4.44). Each model variant (Section 9.2.3.1) predicted the 300 loads in the validation dataset (Section 9.1.1.3) day-by-day for 100 consecutive days. For each day-type, we computed the EDE according to (7.15). The figure shows the 900 EDE-observations confounded on day and load type (grey dots). Violin and box-plots summarize the EDE-distributions in each of the groups. Outliers did not affect any qualitative conclusions and were removed to provide the figure panels with similar axis limits. The average EDE of the model in each group (red dot) is shown together with 95%-confidence interval (red bar). Note that both NWE-variants resulted in comparable accuracy.

Validation: Multistep Strategy We observed that the direct multistep strategy resulted in a more accurate forecast on the majority of loads in the validation dataset (Figure 9.22). We compared the direct strategy predicting the load curve by q separate models (4.57) and the multi-out strategy predicting the entire load curve with a single multivariate NWE-model that considered all q points of X^* as q inputs (4.62). In contrast to the previously discussed parametric models (ANN, ARIMA), the multi-out approach was inferior due to the curse of dimensionality that commonly occurs with nonparametric models.

Fixed bandwidth models (NWE, MNWE) were sometimes unable to forecast the load. In some cases, the local neighborhood of the query whose size is determined by the bandwidth contained no observations. This happened particularly often on households whose load is markedly volatile. In such cases, the prediction with NWE was undefined, and we relied on the D-1 heuristic instead. Alternatively, we applied the nonparametric model with variable bandwidth as a reference that avoids such situations and is described next.

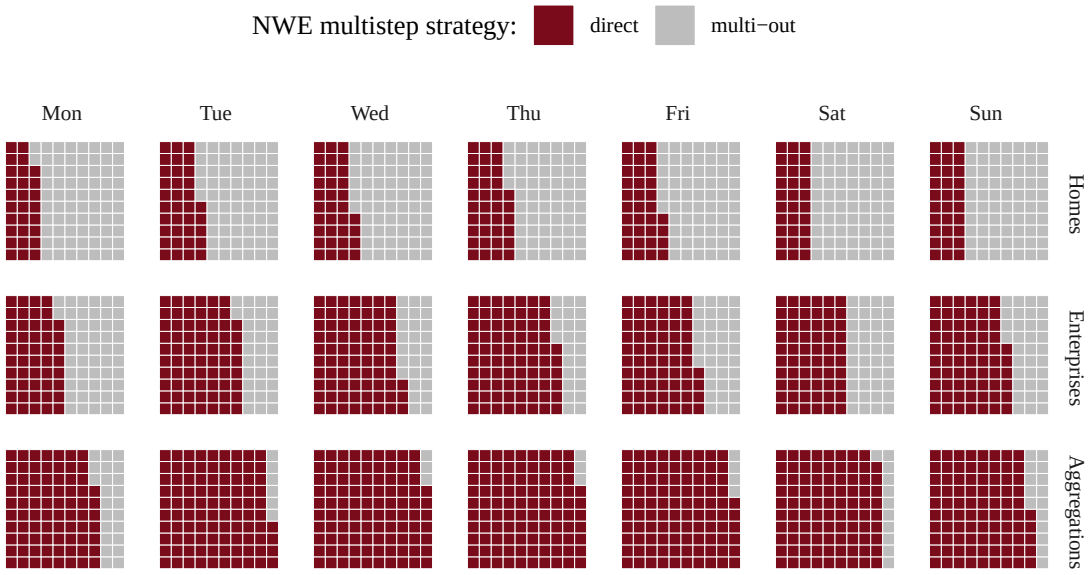

Figure 9.22: Comparison of the multistep strategies for an NWE-model with fixed bandwidth. We applied the NWE-model (Section 9.2.3.1) using direct and multi-out strategies to predict the loads in the validation dataset (Section 9.1.1.3). Each load was predicted day-ahead for 100 consecutive days (23 April 2010 – 31 July 2010). Conditioning on load type (panel row) and weekday (panel column), we represent each individual load by a square filled depending on the multistep strategy that provided the smallest expected daily error (7.15) on the days of the corresponding load type and weekday. Notably, the direct multistep strategy provided a more accurate forecast on the majority of loads.

9.2.3.2 Multivariate K-Nearest Neighbors (MKNN)

Multivariate K-nearest neighbors (MKNN) approach is a special case of the NWE where the bandwidth is variable, and the size of the local neighborhood is determined depending on the query (Section 4.2.2.2). We applied this approach creating another nonparametric model that we used as a reference in the wide-scale day-ahead building load forecasting simulation. As with the NWE-model, we used Gaussian kernel and considered the annual cycle by restricting the historical data to the 17 most recent weeks. Once again, we modeled the weekly seasonality *implicitly*, filtering the observations by weekday. To consider the daily seasonality, we applied the multi-out multistep strategy. Therefore, the load curve was predicted using a single MKNN-model (4.62) with variable bandwidth K that was found automatically for each day through leave-one-out cross-validation (4.44).

Validation: Variable Bandwidth Search The best choice of the variable bandwidth K depends on the forecast load and day (Figure 9.23). Notably, higher bias of the

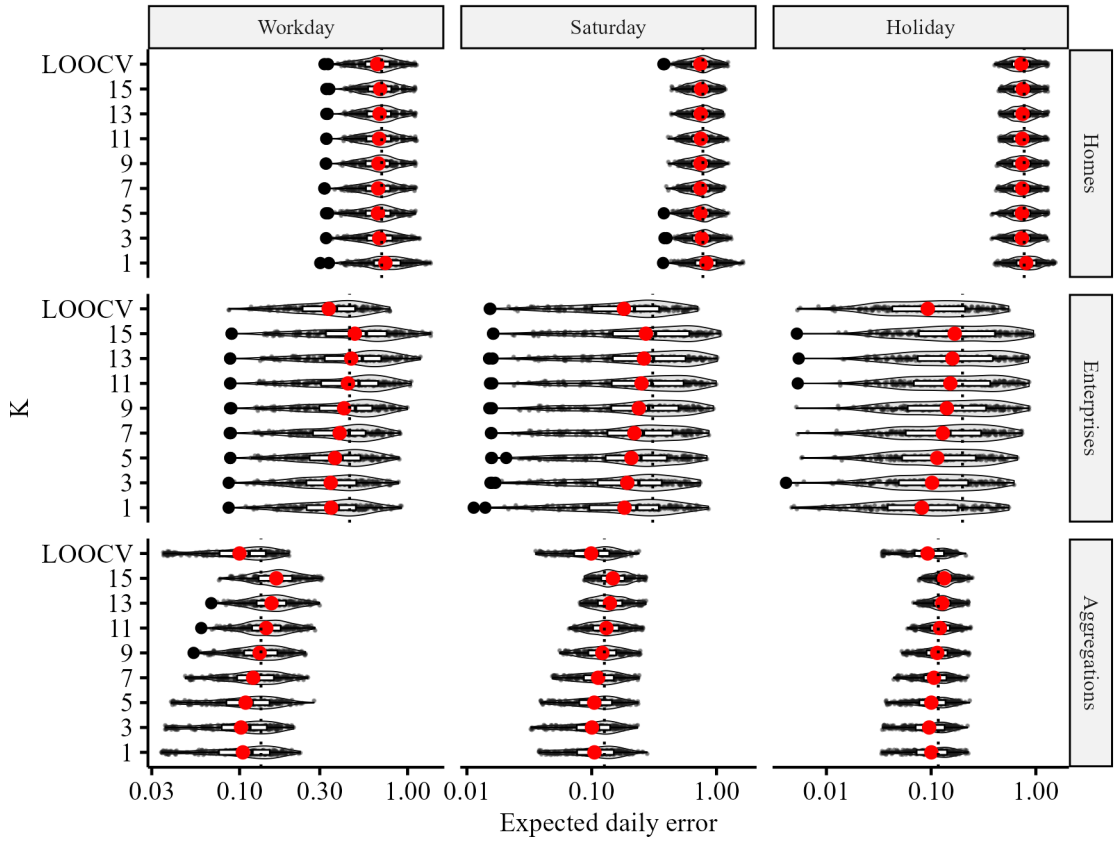

Figure 9.23: Forecast errors of MKNN-model variants with different variable bandwidth K. Each model variant (Section 9.2.3.2) predicted the 300 loads in the validation dataset (Section 9.1.1.3) day-by-day for 100 consecutive days. For each day-type, we computed the *expected daily error (EDE)* according to (7.15). The figure shows the 900 EDE-observations confounded on day and load type (grey dots), including the outliers (black dots). Violin and box-plots summarize the EDE-distributions in each of the groups. The average EDE of the model in each group (red dot) is shown together with 95%-confidence interval (red bar). We see that setting the bandwidth automatically using leave-one-out cross-validation (LOOCV) often resulted in the most accurate forecast. Here, we showed the workdays (Monday – Friday) together, but similar results can be observed when considering each workday individually.

model (i.e., $K > 1$) was preferable on single family homes due to high volatility of the load. In contrast, model variants with large bandwidth were preferred on more regular loads (enterprises, aggregations). Determining an analytical relationship between the load characteristics and optimal K is beyond the scope of this thesis. Nevertheless, we observed that finding K by minimizing the leave-one-out cross-validation criterion (4.44) prior to the forecast day made the MKNN-model more accurate on a large variety of loads.

Validation: Multistep Strategy In contrast to the NWE-model, we observed that the multi-out strategy resulted in a more accurate forecast on the majority of loads in the validation dataset (Figure 9.24). We compared it to the direct multistep strategy with which

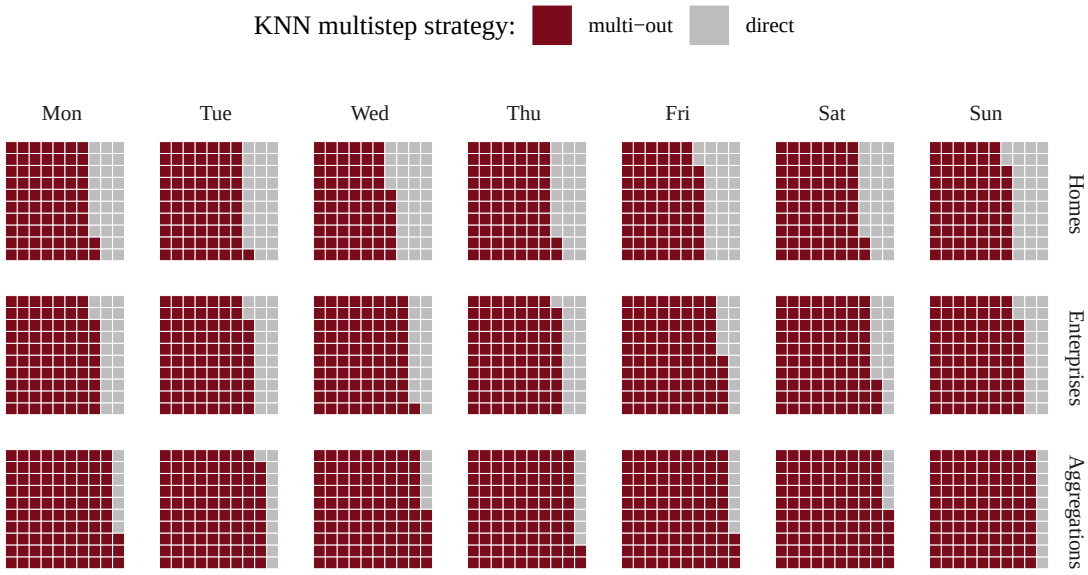

Figure 9.24: Comparison of the multistep strategies for a nonparametric model with variable bandwidth. We applied the KNN-model (Section 9.2.3.2) using direct and multi-out strategies to predict the loads in the validation dataset (Section 9.1.1.3). Each load was predicted day-ahead for 100 consecutive days (23 April 2010 – 31 July 2010). Conditioning on load type (panel row) and weekday (panel column), we represent each individual load by a square filled depending on the multistep strategy that provided the smallest expected daily error (7.15) on the days of the corresponding load type and weekday. Notably, the multi-out strategy resulted in a more accurate forecast on the vast majority of loads.

we predicted each of the q points of the load curve by a separate univariate KNN-model. Variable bandwidth excludes the possibility of an undefined forecast which substantially reduced the accuracy of the NWE-model using multi-out strategy and fixed bandwidth. As a result, we can now see that the multi-out multistep strategy considering the dependencies between the points of a daily load curve is more accurate than the direct multistep strategy.

9.2.3.3 Uniform Average (UA)

Uniform average (UA) of historical observations is a common heuristic prediction method which can also be seen as the simplest form of a nonparametric forecast[13]. We used this method as a reference model predicting the electricity consumption for the upcoming day as an average of *three* most recent daily load curves. As with the other nonparametric models described above, we modeled the weekly seasonality *implicitly* by considering

[13] Uniform average of m historical load curves corresponds to the MKNN-model with uniform kernel and $K = m$.

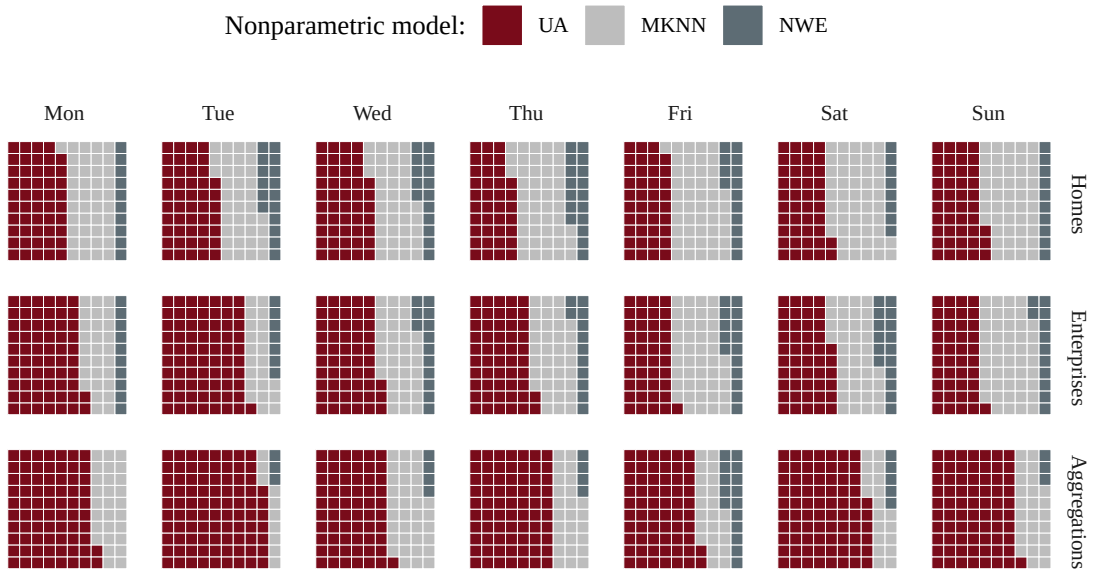

Figure 9.25: Comparison of the nonparametric reference models. The models described in Section 9.2.3 predicted the 300 loads in the validation dataset (Section 9.1.1.3). Each load was predicted day-ahead for 100 consecutive days (23 April 2010 – 31 July 2010). Conditioning on load type (panel row) and weekday (panel column), we represent each individual load by a square filled depending on the multistep strategy that provided the smallest expected daily error (7.15) on the days of the corresponding load type and weekday. Notably, the uniform average forecast had comparable accuracy to the NWE and MKNN forecasts. In fact, it was often the most accurate forecast.

only historical observations of the same weekday. We observed that the uniform average forecast had accuracy comparable to other nonparametric models (Figure 9.25). In fact, the uniform average was often the most accurate forecast particularly on aggregations and enterprises. As discussed previously, these loads often features a stronger annual cycle with which the recency of observations appears to be more relevant for the forecast rather than the ℓ^2-distance from the query that is fundamental for the other nonparametric models.

Validation: History Length We can average the load curves of same weekday or day-type. During the validation, we observed that computing the average for the days of the same day-type was often more accurate than averaging over the same weekday (Figure 9.26). In the latter case, we obtained the most accurate forecast when considering only three to five most recent observations. Therefore, it might be counter-productive to consider the observations that are older than five weeks old because of the concept change (e.g., annual cycle).

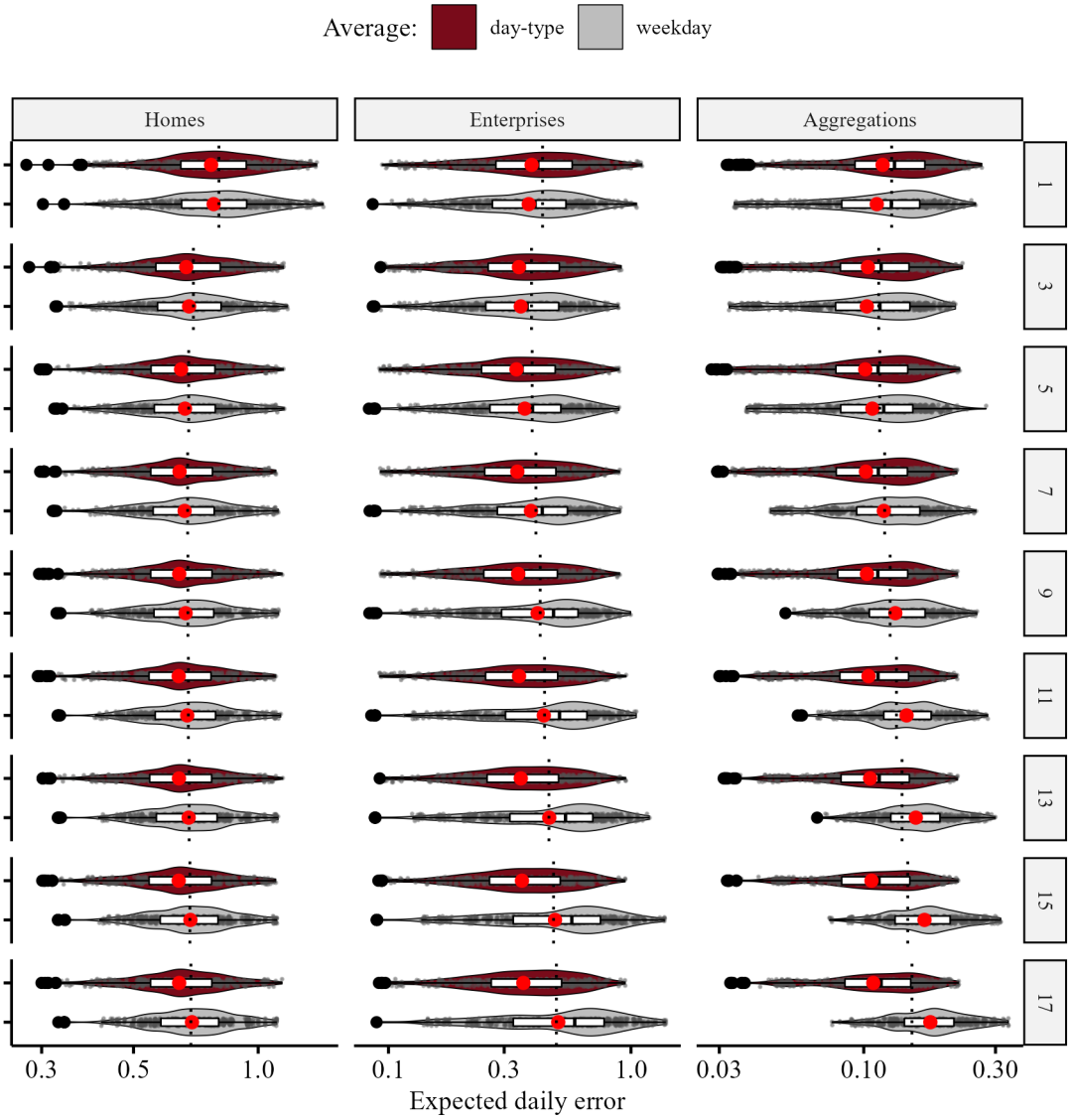

Figure 9.26: Forecast errors of the uniform average model variants using different filtering and history length. Each variant of the uniform average model (Section 9.2.3.3) predicted the load averaging over various number of historical load curve observations (panel row) of the same day-type (red) or weekday (grey). Each model variant variant predicted the 300 loads in the validation dataset (Section 9.1.1.3) day-by-day for 100 consecutive days. For each load, we computed the *expected daily error (EDE)* according to (7.15). The figure shows the 900 EDE-observations (grey dots) confounded on load type (panel column) and the number of averaged curves (panel row). Violin and box-plots summarize the EDE-distributions in each of the groups. The average EDE of the model (red dot) is shown together with 95%-confidence interval (red bar). Outliers did not affect any qualitative conclusions and were removed to provide the figure panels with similar axis limits. The figure considers only the daily errors obtained on workdays, since both filtering schemes provide the same forecast on weekends. Notably, computing the average for the days of the same day-type was in many cases significantly more accurate than averaging over the same weekday. Moreover, filtering by weekday, the most accurate forecast was obtained when averaging the load curves that are three to five weeks old.

9.3 Experiment Overview

Concluding the methodological part of the thesis, we summarize the experiments carried out in our study. Answering the research question, we conducted various simulations (Section 9.3.1). We relied on these simulations developing and evaluating the functional neighbor methodology that we are proposing for the wide-scale day-ahead building load forecasting in smart grids. In particular, we compared the corresponding *functional neighbor (FN)* and *functional neighbor extension (FNX)* forecasters to various reference models (Section 9.3.2) and assessed their accuracy with the evaluation methodology that we developed specifically for the wide-scale day-ahead building load forecasting problem considered in this work (Section 9.3.3).

9.3.1 Simulation Overview

During this study, we relied on various simulations in order to validate the design decisions for the functional neighbor forecaster and to set up the reference models (Section 9.3.1.1). Having designed the forecaster and determined an adequate setup of the reference models, we conducted a simulation evaluating the models in context of the wide-scale day-ahead building load forecasting (Section 9.3.1.2). Moreover, we conducted the smart-building load forecasting simulation (Section 9.3.1.3) validating the extension of the functional neighbor forecaster that allows to consider exogenous variables. In all load forecasting simulations conducted for this study, we used the smart-meter data with a 60-minute resolution. Further, every day-ahead forecast was done at midnight predicting at once the 24 points constituting the load curve of the upcoming day.

9.3.1.1 Validation Experiments

We conducted various validation experiments to design the functional neighbor model (Section 8.2) and to set up reference models (Section 9.2) that were compared to it. In every validation experiment, we predicted the 300 loads from three different groups (Table 9.4) constituting the validation dataset (Section 9.1.1.3). Each load was predicted day-ahead during 100 consecutive days (23rd of April 2010 – 31st of July 2010). Below, we provide the details necessary to replicate these experiments.

Distance Notion Setup In Section 8.2.2, we studied the usage of various distance notions for the functional nonparametric forecasting approach. In the corresponding simulations, we applied the functional nearest neighbor model using 17 weeks of historical data *filtered by weekday (FbW)*. Considering only the nearest neighbor allowed to eliminate

the uncertainty of the bandwidth and merger choice while focusing on the effect that the distance notion had on the forecast accuracy (Algorithm 3 with $K = 1$ and FbW).

Merger Setup In Section 8.2.3, we studied the usage of various mergers for the functional nonparametric forecasting approach. In the corresponding simulations, we applied the functional neighbor model using 17 weeks of historical data filtered by weekday. We eliminated the uncertainty of the bandwidth search by finding the optimal K among a set of variants[14] using an *ideal model selector*[15] (Algorithm 3 with $\mathbf{d}_{u=1}$-distance and FbW).

Reference Model Setup In Section 9.2, we set up and manually fine-tuned numerous reference models using the validation dataset. In each family, we selected the most accurate models that we subsequently evaluated together with the functional neighbor model in the wide-scale day-ahead building load forecasting simulation.

9.3.1.2 Wide-Scale Day-Ahead Building Load Forecasting

We simulated a wide-scale day-ahead building load forecasting using the measurements from the ICER smart-meter dataset (Section 9.1.1). The data included the power demand of hundreds residential and commercial buildings that we resampled with a 60-minute resolution. Using the original dataset, we created various load aggregations representing larger buildings (Section 9.1.1.2) and defined several residential and commercial load groups (Table 9.3). In total, our evaluation dataset consisted of 1851 local loads with power-demand time series extending from 1st of August 2010 to 31st of December 2010 (153 days in total). In this dataset, we predicted each load day-by-day applying the FN-forecaster (Algorithm 3) and various reference models (Table 9.5). For each evaluated model, the wide-scale day-ahead building load forecasting simulation provided a sample of 283,203 daily forecast errors that we analyze in the final part of the thesis.

9.3.1.3 Smart-Building Load Forecasting

We applied the functional neighbor methodology on the existing smart building within the Smart-City-Demo Aspern project (Section 9.1.2). In order to validate the FNX-model (Section 8.3), we simulated the load forecasting of a student dorm facility whose net electricity demand substantially depends on the solar irradiation due to a large PV-installation on the roof. For this smart building, we predicted the day-ahead power demand

[14] The bandwidth K was selected from the set of candidates $\{3, 5, 7, 9, 11, 13, 15, 17\}$.

[15] We used the notion of an ideal model selector as a benchmark selecting the best model variant knowing the actual error that each variant would produce – i.e., looking into the future. Hence the name *ideal*.

using the FNX-forecaster (Algorithm 4), that considered global solar irradiation as an exogenous variable, together with various reference models applied on the same building. The models predicted the day-ahead net electricity demand day-by-day on a rolling basis. Of the available twelve months of data, we used three months (1st of January 2017 – 31st of March 2017) as a validation dataset while preserving the last three months (1st of April 2017 – 31st of June 2017) for the evaluation.

9.3.2 Models

Studying how to use smart-meter data to predict day-ahead electricity consumption of individual buildings and their aggregations on a wide scale, we considered numerous data-driven models. These included the reference models that can be commonly found in the literature (Chapter 5) as well as the proposed FN- and FNX-models. The reference models for the simulations were pre-selected on a separate validation dataset and are summarized in Tables 9.5 and 9.6. Using the validation dataset, we set up those models[16] manually, to the best of our abilities, in various experiments (Section 9.3.1.1). The hyperparameters of the FN- and FNX-models were determined automatically using cross-validation ideas that are common for data-driven forecasting models [AVR16, SL15].

In the wide-scale day-ahead building load forecasting simulation (Section 9.1.1), we evaluated the reference models summarized in Table 9.5 together with the FN-forecaster (Algorithm 3). For this forecaster, we used the weighted permutation merger with triangular kernel determining the weights following the design discussion in Section 8.2.3. Further hyperparameters were set automatically for each predicted load using cross-validation (Section 8.1.2). In particular, prior to predicting the load, we used the historical data applying the leave-one-out-cross-validation approach to select the best distance notion ($\mathbf{d}_0$ or $\mathbf{d}_{\mathrm{u}=1}$) and the filtering approach. Further, we applied the out-of-sample-validation approach on the training data (17 weeks), for each forecast day, determining the best value of the bandwidth K on a daily basis.

In the smart-building load forecasting simulation, we applied the FNX-forecaster (Algorithm 4) on a smart building for which we manually selected the global solar irradiation as the most relevant feature (Section 9.1.2). The hyperparameters of the FNX-forecaster were set the same way as those of the previously discussed FN-forecaster. For comparison, we applied several reference models that we set up manually to the best of our abilities on a separate validation dataset. Next to the standard load profiles used as a benchmark in

[16] Parametric models were retrained monthly to account for the nonstationarity of the load. Moreover, the forecast of a neural-network-based model corresponded to the average prediction obtained by an ensemble of ten networks of the same architecture in order to mitigate the dispersion obtained due to the random weight initialization during the network training [VKS20].

Table 9.2: Error notions overview.

Error notion	Definition	Formula
PRMSE	(7.12)	$E(Y,\hat{Y}) := \sqrt{\frac{1}{24}\left(\min_{\pi\in\mathcal{P}(1,24)}\sum^{2}4_{i=1}\lvert Y(t_i)-\pi(\hat{Y})(t_i)\rvert^2\right)}$
ECV	(7.13)	$\mathrm{ECV}=\frac{E}{\bar{y}}*100\%$
Improvement	(7.14)	$\mathrm{R}=(1-\frac{E}{E_\mathrm{b}})*100\%$
EDE	(7.15)	$\mathrm{EDE}=\mathbb{E}\Big[E\Big]=\frac{\sum_j^m E_j}{m}$
TE	(7.16)	$\mathrm{TE}=\mathrm{median}\Big[\mathrm{EDE}\Big]$
EME	(7.17)	$\mathrm{EME}(S)=\sqrt{\frac{\alpha}{S^p}+\beta}$

our study, we applied the individual load profiles since it was the best performing heuristic approach in the wide-scale day-ahead building load forecasting simulation. Moreover, we applied multivariate parametric reference models that considered global solar irradiation as an external input. In particular, we applied the ARIMAX model which is an extension of the ARIMA-model that can consider external variables (Section 4.1.1). Additionally, we applied a DNN-architecture with three hidden layers as a reference model. The reference models for the smart-building load forecasting simulation are summarized in Table 9.6.

9.3.3 Forecast Evaluation

The forecast errors provided by the simulations underlay a substantial stochastic variation. When evaluating forecasts on a large and diverse set of buildings, we had to consider the distribution of the errors and rely on inferential statistics rather than simply quantify the average accuracy. Therefore, we analyzed the results using the forecast evaluation methodology introduced in Section 7.3 and relying on various error notions (Table 9.2).

9.3.3.1 Error Notions

Throughout our study, we used the *permuted root mean squared error (PRMSE)* as the primary error notion for quantifying the daily forecast errors (7.12). Further, we expressed the daily errors in terms of the coefficient of variation that is scale-independent and *unitless* (7.13). We often observed a strong correlation between different error notions and it can be superfluous to present the results in terms of several daily error notions [HGZA18]. At the same time, common notions such as MAPE and RMSE, that are ubiquitous in the literature, can be inapt for evaluating forecast accuracy on smaller loads. Therefore, unless stated differently, the daily forecast error is measured in terms of PRMSE allowing permutations of up to one hour (i.e., $u = 1$ for the hourly time-series resolution) and expressing the error in terms of the unitless coefficient of variation (Section 7.3.1).

Moreover, we defined various secondary error notions to quantify the overall forecast accuracy obtained on a given load or a sample of loads (Section 7.3.2). First, we defined the *expected daily error (EDE)* that corresponds to the average daily forecast error obtained on a given load (7.15). Second, we defined the *total error (TE)* quantifying the forecast error obtained in a load group (7.16). At last, predicting the loads of different size allowed us to compute the *expected model error (EME)* based on the empirical scaling law (7.17). This error notion estimates the EDE that we can expect forecasting a load of a given size (i.e., annual consumption in MWh). Herewith, we were able to evaluate a forecasting model across the building domain estimating its accuracy on the loads of all possible sizes.

A considerable variation of daily errors can reduce the sensitivity of the chosen accuracy measure, obstructing parametrization and comparison between the models. For such case, we used the relative error notion dividing the primary error E by the error of a benchmark model E_b as suggested by Hyndman et al., [HK06]. In particular, we defined the *improvement* (%) that quantifies the error reduction comparing to a benchmark (7.14). As such benchmark, we used either the basic configuration of the forecaster for functional neighbor model design (Chapter 8) or the SLP-forecast for the final model evaluation (Chapter 10). Since the improvement is normally distributed, we can summarize it in terms of mean and confidence intervals.

9.3.3.2 Statistical Tests

As a part of our evaluation methodology, we relied on various statistical tests that allowed to evaluate the prediction accuracy despite the uncertainty in the observed forecast errors. To compare different models applied on a single or several buildings, we assessed if the difference in forecast accuracy was statistically significant using the following tests:

- *Unpaired one-sided independent t-test* – provided statistical evidence verifying if a model was significantly more accurate than a benchmark using a sample of improvement observations.
- *Paired $t-$test* – allowed pair-wise comparisons of the models verifying if the average difference in improvement obtained by each model relative to a common benchmark was statistically significant.
- *Paired Wilcoxon signed rank test* – allowed pair-wise comparisons of the models verifying if the observed difference in median forecast errors was significant.

For each test, we reported the corresponding p-value and considered the results to be statistically significant for $p < 0.05$ as it is common in the statistical literature [Lav21].

Table 9.3: Summary of the load groups in the evaluation dataset. Residential load groups (A-C) include a group of 887 single family homes (A), 180 residential aggregations (B), and 180 larger residential aggregations (C). Commercial load groups (D-F) include a group of 175 single enterprises (D), 34 commercial aggregations (F), and 33 larger commercial aggregations (F). The methodology for selecting the loads for each group was described in the text (Section 9.1.1.2).

Characteristic	**A**, N = 887	**B**, N = 180	**C**, N = 180	**D**, N = 175	**E**, N = 34	**F**, N = 33
Annual consumption (MWh)						
Median	8.1	38.6	136.1	27.4	102.3	526.7
IQR	5.5 - 11.2	27.2 - 49.7	87.3 - 299.9	13.8 - 46.2	63.9 - 162.7	386.7 - 1,039.6
Minimum	0.9	7.5	65.2	2.9	23.2	233.1
Maximum	38.7	64.6	7,767.4	312.9	222.0	7,031.9
Mean load (kWh)						
Median	0.9	4.4	15.6	3.1	12.2	61.2
IQR	0.6 - 1.3	3.1 - 5.7	10.0 - 34.3	1.6 - 5.3	7.4 - 18.3	42.9 - 117.3
Minimum	0.1	0.9	7.4	0.3	2.6	25.9
Maximum	3.8	7.4	886.9	35.7	25.8	799.2
Coefficient of variation (%)						
Median	108.1	64.5	52.9	107.9	78.0	50.9
IQR	93.7 - 126.9	60.2 - 69.7	48.8 - 56.9	74.4 - 143.8	67.0 - 91.6	42.9 - 62.7
Minimum	43.1	47.4	39.8	27.0	48.8	29.4
Maximum	262.1	104.8	70.2	269.5	127.2	81.9

Table 9.4: Summary of the load groups in the validation dataset. Each group includes 100 single family homes, enterprises, or mixed aggregations. The methodology for selecting the loads for each group was described in the text (Section 9.1.1.3).

Characteristic	**Homes**, N = 100	**Enterprises**, N = 100	**Aggregations**, N = 100
Annual consumption (MWh)			
Median	8.7	29.2	499.8
IQR	8.4 - 8.9	19.8 - 39.2	304.7 - 1,390.5
Minimum	8.1	13.7	212.8
Maximum	9.4	61.7	14,157.6
Mean load (kWh)			
Median	1.0	3.2	57.5
IQR	1.0 - 1.0	2.2 - 4.3	35.0 - 158.0
Minimum	0.9	1.5	23.5
Maximum	1.2	7.1	1,611.9
Coefficient of variation (%)			
Median	106.6	107.5	42.1
IQR	99.1 - 122.0	76.5 - 132.9	37.3 - 46.6
Minimum	75.5	27.7	29.8
Maximum	157.1	269.5	67.6

Table 9.5: Reference forecasting models evaluated in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1).

Model	Description	Family	Multistep strategy	Reference
SLP	standard load profile	heuristic	multi-out	[Ber00]
ILP	individual load profile	heuristic	multi-out	[BPT13]
D-1	naive forecast	heuristic	multi-out	[ACGW18]
D-7	weekly persistence forecast	heuristic	multi-out	[HP16]
ARIMA-D	ARIMA using direct multistep strategy	parametric	direct	[**?**]
ARIMA-R	ARIMA using recursive multistep strategy	parametric	recursive	[**?**]
MLP-D	multilayer perceptron neural network (direct)	parametric	direct	[EÁBRA11]
MLP-M	multilayer perceptron neural network (multi-out)	parametric	multi-out	[ZLY19]
NAR	univariate autoregressive neural network	parametric	recursive	[POC$^+$17]
NARX	autoregressive neural network with exogenous inputs	parametric	recursive	[HP16]
DNN-2	deep neural network (MLP) with two hidden layers	parametric	multi-out	[BZN$^+$19]
DNN-3	deep neural network (MLP) with three hidden layers	parametric	multi-out	[BZN$^+$19]
UA	uniform average of three historical observations	nonparametric	multi-out	[CGS13a]
NWE	Nadaraya-Watson estimator	nonparametric	direct	[SL15]
MKNN	multivariate K-nearest neighbors	nonparametric	multi-out	[BBB†12]

Table 9.6: Reference models evaluated in the smart-building load forecasting simulation (Section 9.1.2).

Model	Description	Family	Parameters	Reference
SLP	standard load profile	heuristic	n/a	[Ber00]
ILP	individual load profile	heuristic	n/a	[BPT13]
ARIMAX	ARIMA$(1, 1, 1)$ model with exogenous input	parametric	$\beta = -2$	[AVR16]
DNN	deep neural network (MLP-M with 3 hidden layers)	parametric	$n_h = 5$	[WCHK18]

Part IV

Results

In the final part of the thesis, we provide the results relating them to the research question and contributions that were stated in the introduction to this work. In particular, we use the results of the conducted simulations to evaluate the proposed functional neighbor forecasting methodology for predicting building power demand on a wide scale (Chapter 10). Proceeding, we interpret the findings discussing their relevance and implications for the wide-scale day-ahead local load forecasting in smart grids (Chapter 11). Based on this discussion, we formulate the conclusion of our study (Chapter 12).

10 Simulation Results

In this chapter, we present the results of conducted simulation experiments[1]. For each model evaluated in the wide-scale day-ahead building load forecasting simulation, we obtained a sample of 283,203 daily forecast errors. We provide a statistical analysis of the results using the previously introduced evaluation methodology (Section 7.3) that relies on various error notions (Table 9.2). In particular, we summarize the daily forecast errors in terms of expected model error (Table 10.1), total error (Table 10.2) and the improvement relative to a forecast with a standard load profile (Table 10.3).

In a wide-scale day-ahead building load forecasting simulation, we evaluated 15 reference models of various families (Section 10.1) as well as the proposed functional neighbor model (Section 10.2). Additionally, in a smart-building load forecasting simulation, we evaluated the proposed functional neighbor model extension that considers exogenous variables on the facilities of the Smart-City-Demo-Aspern project and compared it to the best-performing reference models (Section 10.3). Subsequently, we present the simulation results before discussing them in the next chapter.

[1] We described the simulations conducted for this study in Section 9.1. In the wide-scale day-ahead building load forecasting simulation (Section 9.1.1), we forecast 1851 low-voltage loads of different size and type, on the rolling basis for 153 consecutive days. In the smart-building load forecasting simulation (Section 9.1.2), we predicted the load of a single smart building for 91 consecutive days. The simulations, the forecasting models and the evaluation methodology were summarized in Section 9.3.

Table 10.1: Expected model error of the evaluated models. The table presents the estimated parameters p, α, β for computing the forecast error that can be expected on a load of a given size according to the empirical scaling law (7.17). Further, we used the estimated parameters to compute the irreducible error E_0 (7.21) and the critical load size S_{crit} (7.23). The parameters were estimated using weighted non-linear regression to the $p < 0.001$ level of significance on the sample of daily errors obtained through the wide-scale day-ahead building load forecasting simulation (Section 9.1.1). The evaluated models were summarized in Table 9.5 while the results are discussed in the text throughout the Chapter 10.

Model	E_0	S_{crit} **(MWh)**	p	α	β
FN	0.048 (±5e-04)***	695 (±30)***	0.8 (±0)***	0.544 (±0.0019)***	0.0023 (±0)***
NWE	0.058 (±5e-04)***	569 (±23)***	0.8 (±0)***	0.586 (±0.002)***	0.0033 (±1e-04)***
ARIMA-R	0.061 (±6e-04)***	526 (±21)***	0.8 (±0)***	0.721 (±0.0026)***	0.0037 (±1e-04)***
D-7	0.065 (±6e-04)***	465 (±18)***	0.9 (±0)***	0.827 (±0.003)***	0.0042 (±1e-04)***
MKNN	0.066 (±6e-04)***	577 (±26)***	0.8 (±0)***	0.554 (±0.0021)***	0.0043 (±1e-04)***
UA	0.067 (±5e-04)***	301 (±10)***	0.9 (±0)***	0.595 (±0.0021)***	0.0045 (±1e-04)***
ILP	0.07 (±5e-04)***	319 (±11)***	0.8 (±0)***	0.582 (±0.0021)***	0.0048 (±1e-04)***
SLP	0.079 (±7e-04)***	836 (±38)***	0.7 (±0)***	0.758 (±0.0025)***	0.0063 (±1e-04)***
ARIMA-D	0.082 (±6e-04)***	353 (±11)***	0.9 (±0)***	1.074 (±0.0034)***	0.0067 (±1e-04)***
MLP-M	0.092 (±4e-04)***	159 (±4)***	0.9 (±0)***	0.929 (±0.003)***	0.0085 (±1e-04)***
DNN-2	0.094 (±4e-04)***	171 (±4)***	0.9 (±0)***	0.925 (±0.003)***	0.0089 (±1e-04)***
D-1	0.095 (±6e-04)***	271 (±10)***	0.8 (±0)***	0.827 (±0.0033)***	0.0089 (±1e-04)***
DNN-3	0.096 (±5e-04)***	180 (±5)***	0.9 (±0)***	0.866 (±0.0029)***	0.0092 (±1e-04)***
NARX	0.122 (±5e-04)***	79 (±2)***	1 (±0)***	1.347 (±0.0049)***	0.015 (±1e-04)***
NAR	0.123 (±6e-04)***	128 (±4)***	0.8 (±0)***	0.646 (±0.0029)***	0.0151 (±1e-04)***
MLP-D	0.129 (±4e-04)***	68 (±1)***	1 (±0)***	0.951 (±0.0033)***	0.0167 (±1e-04)***

Table 10.2: Total error of the models in each load group. Models summarized in Table 9.5 were evaluated in a wide-scale day-ahead building load forecasting simulation (Section 9.1.1) calculating the expected daily error (7.15) for each load. The resulting sample of 1851 errors was split into six load groups (Table 9.3). In each load group, we calculated the total error (7.16) presented in the table with the corresponding interquartile range (in brackets). The results are discussed in the text throughout the Chapter 10.

	Residential loads			Commercial loads		
	A	**B**	**C**	**D**	**E**	**F**
Heuristic models						
D-1	0.86 [0.58, 1.23]	0.44 [0.34, 0.57]	0.26 [0.18, 0.34]	0.45 [0.21, 0.88]	0.37 [0.23, 0.59]	0.19 [0.10, 0.36]
D-7	0.85 [0.58, 1.23]	0.43 [0.34, 0.57]	0.25 [0.17, 0.34]	0.35 [0.16, 0.72]	0.28 [0.18, 0.43]	0.12 [0.08, 0.19]
ILP	0.70 [0.50, 0.98]	0.36 [0.29, 0.46]	0.21 [0.15, 0.29]	0.37 [0.19, 0.70]	0.28 [0.18, 0.42]	0.12 [0.08, 0.19]
SLP	0.78 [0.60, 1.07]	0.40 [0.32, 0.50]	0.23 [0.17, 0.31]	0.66 [0.45, 0.98]	0.39 [0.29, 0.54]	0.24 [0.18, 0.32]
ARIMA models						
ARIMA-D	0.98 [0.70, 1.38]	0.49 [0.39, 0.64]	0.29 [0.20, 0.39]	0.45 [0.23, 0.86]	0.33 [0.22, 0.49]	0.15 [0.09, 0.23]
ARIMA-R	0.79 [0.55, 1.12]	0.41 [0.33, 0.54]	0.24 [0.17, 0.33]	0.34 [0.17, 0.69]	0.27 [0.18, 0.41]	0.12 [0.08, 0.19]
ANN models						
NAR	0.75 [0.52, 1.07]	0.39 [0.30, 0.50]	0.25 [0.19, 0.33]	0.38 [0.20, 0.74]	0.32 [0.21, 0.49]	0.17 [0.09, 0.30]
NARX	1.03 [0.69, 1.56]	0.44 [0.34, 0.58]	0.25 [0.18, 0.34]	0.58 [0.26, 1.14]	0.41 [0.25, 0.67]	0.18 [0.10, 0.35]
MLP-D	0.90 [0.62, 1.33]	0.39 [0.31, 0.52]	0.24 [0.17, 0.31]	0.50 [0.27, 0.91]	0.38 [0.26, 0.56]	0.22 [0.13, 0.35]
MLP-M	0.92 [0.64, 1.31]	0.42 [0.33, 0.55]	0.24 [0.17, 0.33]	0.43 [0.21, 0.84]	0.32 [0.21, 0.49]	0.15 [0.09, 0.26]
DNN-2	0.92 [0.65, 1.30]	0.43 [0.34, 0.55]	0.25 [0.18, 0.33]	0.44 [0.22, 0.84]	0.33 [0.22, 0.51]	0.16 [0.09, 0.27]
DNN-3	0.89 [0.64, 1.24]	0.42 [0.33, 0.55]	0.25 [0.18, 0.33]	0.44 [0.22, 0.84]	0.34 [0.22, 0.52]	0.17 [0.10, 0.28]
Nonparametric models						
MKNN	0.68 [0.47, 0.98]	0.37 [0.29, 0.49]	0.24 [0.17, 0.32]	0.29 [0.14, 0.63]	0.26 [0.16, 0.42]	0.12 [0.07, 0.21]
NWE	0.70 [0.49, 1.01]	0.37 [0.29, 0.49]	0.23 [0.16, 0.31]	0.31 [0.15, 0.64]	0.25 [0.16, 0.40]	0.11 [0.07, 0.18]
UA	0.72 [0.51, 1.02]	0.37 [0.29, 0.48]	0.22 [0.15, 0.29]	0.31 [0.15, 0.63]	0.24 [0.16, 0.37]	0.11 [0.07, 0.18]

Table 10.3: Accuracy improvement (%) relative to the forecast with a standard load profile. The table summarizes the daily forecast improvement (7.14) in each load group (Table 9.3) obtained by the models (Table 9.5) evaluated in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1). For each of the six load groups, we provide the expected daily improvement, 95%-confidence intervals and the corresponding significance levels ('$p < 0.1$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$).

	Residential loads			Commercial loads		
	A	**B**	**C**	**D**	**E**	**F**
Heuristic models						
D-1	-7 (±2)***	-10 (±2)***	-9 (±3)***	15 (±9)***	-6 (±13)'	-11 (±23)'
D-7	-8 (±2)***	-11 (±2)***	-8 (±3)***	30 (±8)***	18 (±11)***	33 (±13)***
ILP	10 (±1)***	8 (±1)***	7 (±2)***	33 (±7)***	22 (±11)***	38 (±12)***
ARIMA models						
ARIMA-D	-24 (±2)***	-24 (±2)***	-24 (±3)***	18 (±9)***	9 (±11)**	26 (±14)***
ARIMA-R	-1 (±1)'	-6 (±2)***	-4 (±3)***	33 (±7)***	23 (±10)***	38 (±12)***
ANN models						
NAR	3 (±2)***	2 (±2)***	-13 (±6)***	27 (±8)***	10 (±13)**	6 (±22)
NARX	-38 (±6)***	-14 (±3)***	-12 (±4)***	-4 (±13)	-17 (±19)***	-5 (±29)
MLP-D	-17 (±3)***	-1 (±3)	-5 (±4)***	11 (±10)***	-3 (±13)	-5 (±24)
MLP-M	-16 (±3)***	-6 (±2)***	-5 (±3)***	21 (±9)***	11 (±11)***	21 (±17)***
DNN-2	-15 (±3)***	-8 (±2)***	-8 (±3)***	20 (±9)***	9 (±11)**	17 (±17)***
DNN-3	-11 (±2)***	-6 (±2)***	-9 (±3)***	20 (±9)***	8 (±10)**	14 (±18)**
Nonparametric models						
MKNN	12 (±1)***	3 (±2)***	-6 (±3)***	40 (±6)***	25 (±9)***	37 (±13)***
NWE	9 (±1)***	5 (±2)***	1 (±3)	39 (±6)***	29 (±9)***	42 (±12)***
UA	8 (±1)***	6 (±2)***	6 (±2)***	40 (±6)***	30 (±9)***	42 (±11)***

Note:

Confidence intervalls (95%) and the significance levels were computed with one-sided t-test.

10.1 Reference Forecasts

In this section, we evaluate the forecasts obtained by the reference models (Table 9.5). Previously, we selected 15 different models based on an extensive literature review (Chapter 5) and set them up on a separate validation dataset (Section 9.2). Below, we present the results of the wide-scale day-ahead building load forecasting simulation obtained by the reference models belonging to heuristic (Section 10.1.1), parametric (Section 10.1.2) and nonparametric (Section 10.1.3) families. In the next section, we compare these forecasts to the predictions obtained by the functional neighbor model evaluating the proposed load forecasting approach.

10.1.1 Heuristic Forecasts

With heuristic forecasts (D-1, D-7, ILP and SLP), we observed large spread among daily errors for each load size (Figure 10.1). On the smallest loads, errors of several hundreds percent occurred often with all models. At the same time, D-1 and D-7 often had very low forecast errors, especially on the loads smaller than 100 MWh. Whenever an enterprise was closed or the inhabitants of a house were absent over several days, the daily electricity consumption was low and almost constant. For such case, the persistence heuristics could provide almost a perfect forecast. Therefore, these models were more flexible and better in adjusting to such situations than the profiling heuristics.

With all heuristic models, the *expected model error (EME)* and the variation among daily errors decreased rapidly with annual consumption, reaching the critical load size where scaling saturated and converged towards the irreducible error (Table 10.1). For larger loads, only small reduction of the error could be obtained despite increasing size.

We compare EME of the heuristic approaches (Figure 10.2). The SLP-model had the largest critical load size of 837 MWh and the irreducible error of 0.08. Forecasting larger loads with the SLP-model, reduced the error while other heuristic models had lower S_c and entered the saturation at notably smaller load sizes. At the same time, D-7 and ILP had significantly lower irreducible error (0.065 and 0.07 respectively). This indicates that these heuristics can be more accurate, especially on larger loads. In fact, we saw that the ILP and D-7 could be expected to be more accurate than the SLP-forecast on the loads larger than 10 MWh and 100 MWh respectively.

We consider the total errors within different load groups. The daily errors for each load were approximately log-normally distributed and are summarized in terms of median and *interquartile range (IQR)* in Table 10.2. For residential consumers, ILP had the smallest total error in each group. Starting with 0.7 error on single family homes (A), total error rapidly decreased to 0.21 for large residential aggregations (C). For commercial consumers

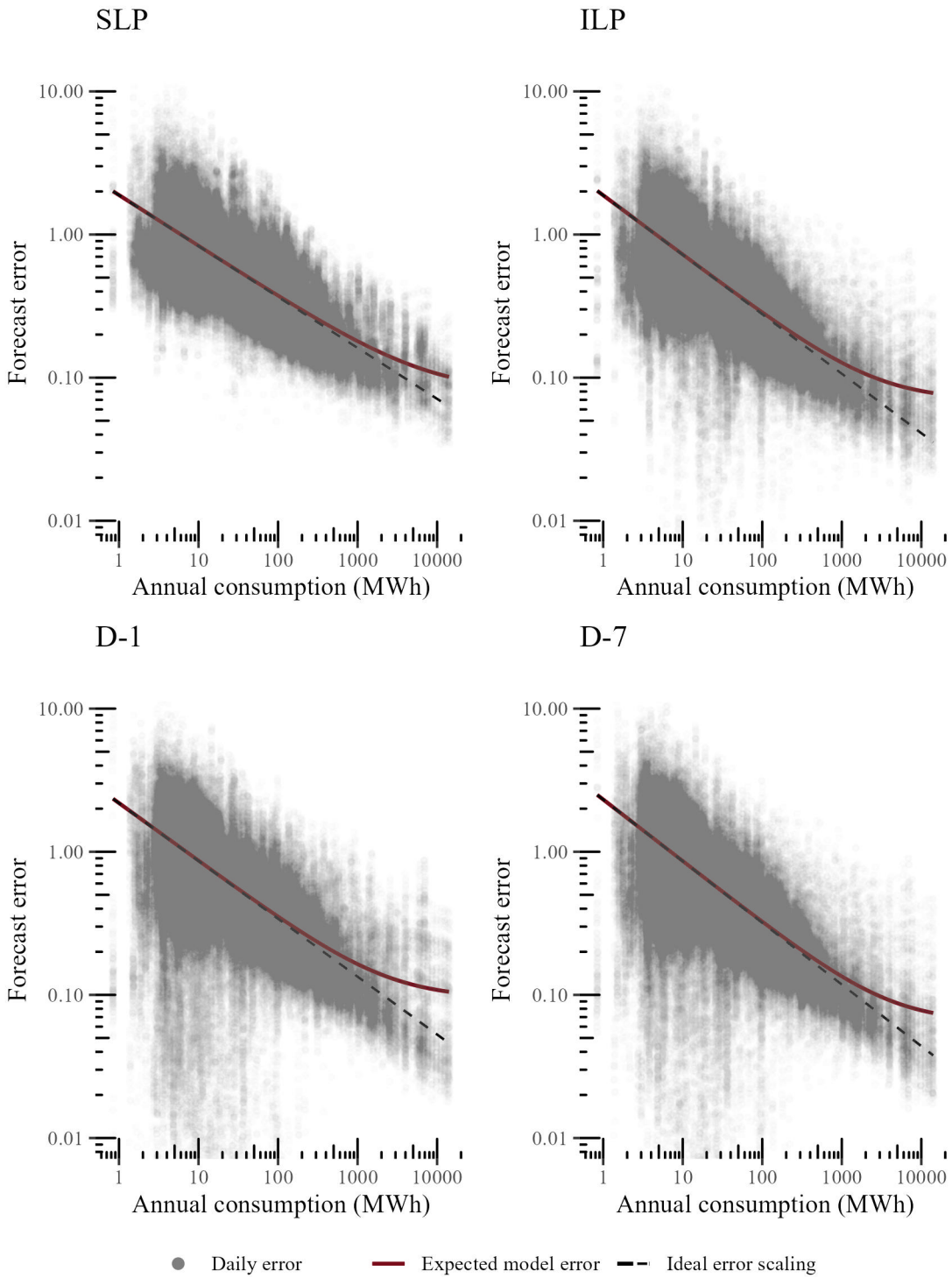

Figure 10.1: Heuristic models – forecast errors. Each panel presents the 283,203 daily errors (grey dots) obtained by a heuristic model (Table 9.5) in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1) on the loads of the specified size (annual consumption). For each model, we computed the expected model error according to the empirical scaling law (7.17) using nonlinear weighted regression (red line) and compared it to the ideal error scaling (black line). The discussion of the results is provided in the text (Section 10.1.1).

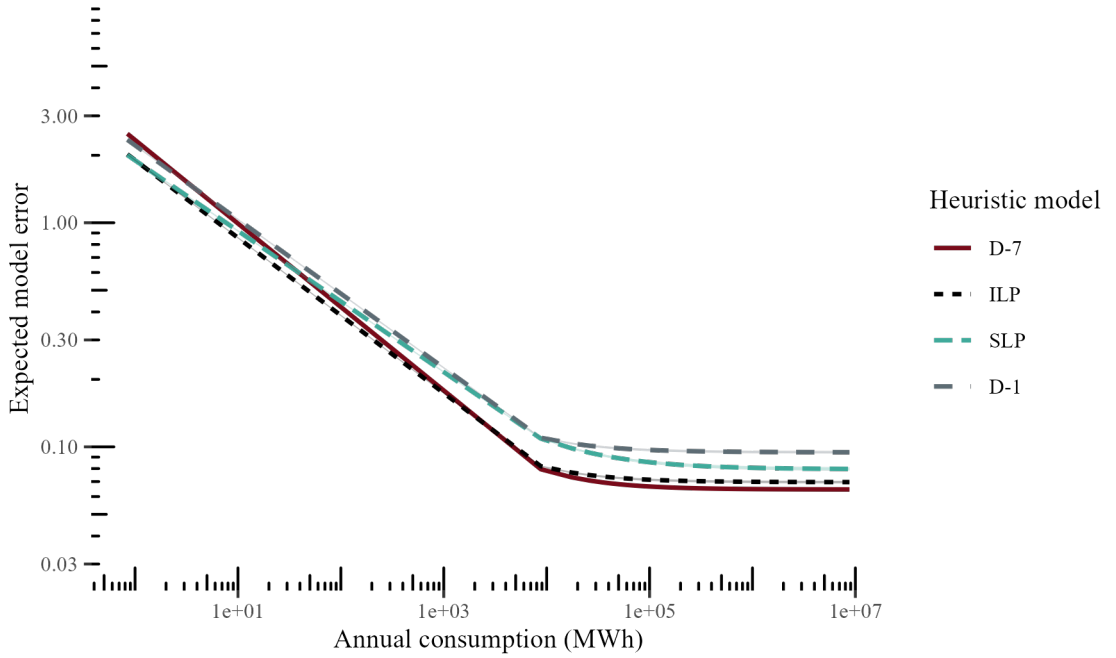

Figure 10.2: Heuristic models – *expected model error (EME)* comparison. The forecast error that we can expect from a model when predicting a load of a given size was computed applying the empirical scaling law (7.17) on the corresponding sample of 283,203 daily forecast errors obtained with each heuristic model in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1). On each sample, we used the weighted nonlinear regression estimating the parameters p, α, β of the fitted curve representing the EME on the figure. The estimated parameters are denoted in Table 10.1. Further discussion of the results in provided in the text (Section 10.1.1).

the total error was substantially lower. The most accurate forecast was D-7 which obtained 0.35 for enterprises (D) decreasing to 0.12 error for large commercial aggregations (F).

The variation of daily error in each group was reflected in the IQR. The D-1 and D-7 forecasts had the largest spread of the daily error corresponding to the widest IQR. The IQR is often asymmetrical due to the log-normal error distribution. While a load can have special days (Figure 10.1), we also had special loads where the models failed to forecast adequately.

The *expected daily error (EDE)* distribution allows us to compare the models in terms of the expected accuracy and error variation within each load group (Figure 10.3). On a log scale, EDE was approximately normally distributed (i.e., log-normal in the original scale). The EDE-distribution on larger loads was slightly skewed due to uneven distribution of loads within terms of size in these groups.

Whenever the difference between the models was not evident from Figure 10.3, we tested the significance of the pair-wise differences among the EDEs for each load. A paired

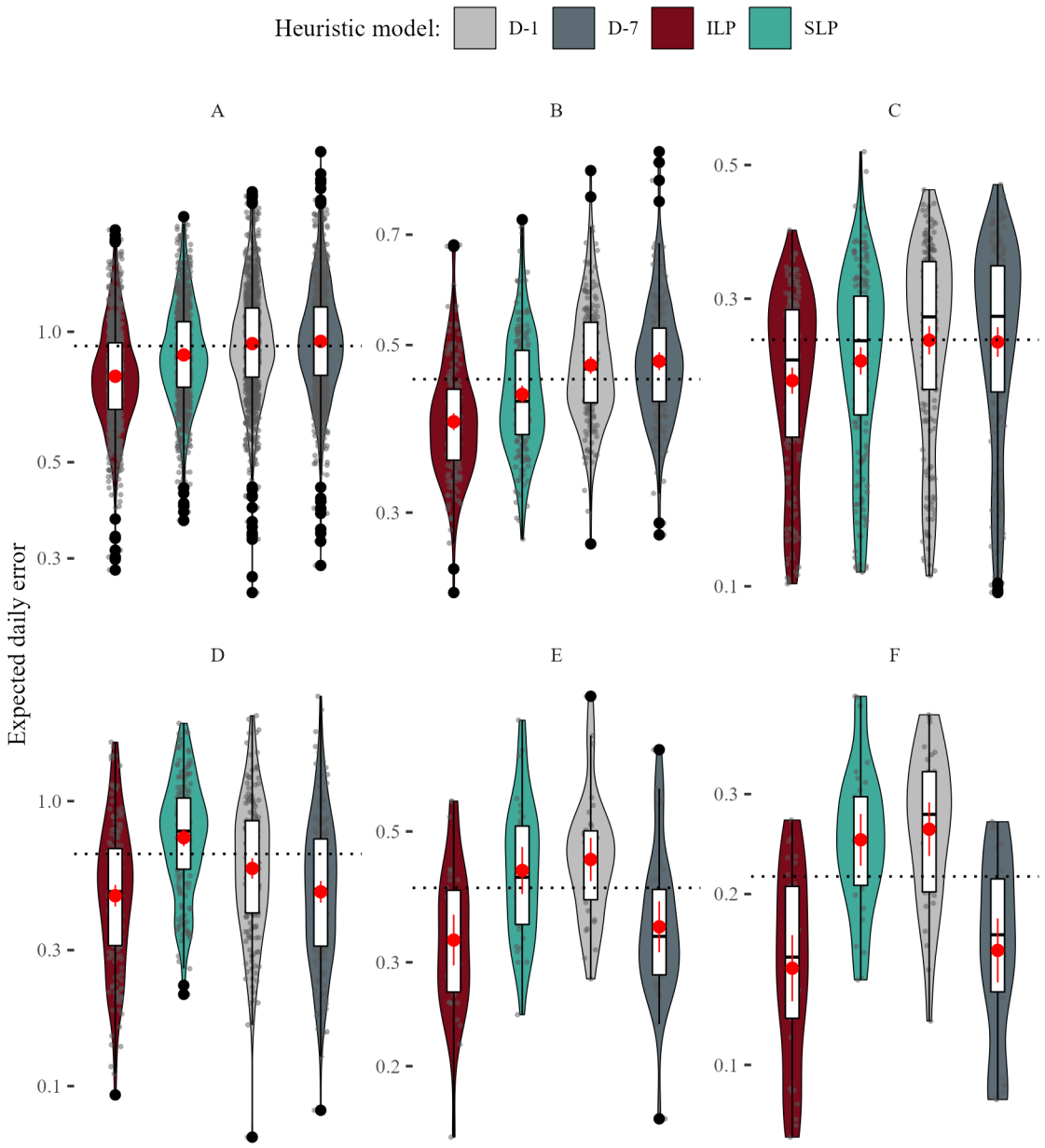

Figure 10.3: Heuristic models – *expected daily error (EDE)* distribution by load group. In a wide-scale day-ahead building load forecasting simulation (Section 9.1.1), we applied various heuristic models (Table 9.5) predicting 1851 loads of different size and type. For each load, we obtained a sample of 153 daily forecast errors (7.13) and computed the EDE (7.15) of the corresponding model. The figure presents the EDEs obtained by the models in residential (A-C) and commercial (D-F) load groups (Table 9.3). Each panel shows the values (grey dots) obtained predicting individual loads of the corresponding group and their distribution (box and violin plots). Additionally, we denoted the EDE-mean (red dot) and its 95%-confidence interval (vertical red bars) for each model. Further discussion is provided in the text (Section 10.1.1).

Wilcoxon signed rank test[2], conducted post-hoc, confirmed that the ILP was the most accurate model in most load groups (Table 10.4). The test showed that even on large residential aggregations, small total error difference between ILP and SLP of 4% was significant ($p < 0.001$). For commercial loads, the difference between D-7 and ILP was only significant on holidays ($p < 0.01$).

Overall, we observed that other heuristic models can be notably more accurate than the SLP-approach. Individual load profiles achieved the largest improvement (Table 10.3). On residential loads, the ILP-model improved the SLP forecasts by 10%. On commercial loads the improvement was several times as large. On single enterprises the forecast could be improved by 33% and up to 48% on large aggregations. The D-7-forecast had comparable improvement on commercial loads. At the same time, there were some days and loads, where no heuristics achieved any improvement comparing to the SLP-forecast.

[2] We used Wilcoxon signed rank test because the EDE-distribution is not symmetrical which prohibits the usage of the, more common, t-test.

Table 10.4: Heuristic models – total error comparison. We applied the paired Wilcoxon signed rank test on the sample of 283,203 daily forecast errors obtained in the wide-scale day.ahead local load forecasting simulation (Section 9.1.1) to evaluate the statistical significance of the total error difference between the heuristic models confounded on load group and day-type. The results are discussed in the text (Section 10.1.1).

	Model			p-values		
	D-7	**ILP**	**SLP**	**D-7 vs. ILP**	**D-7 vs. SLP**	**ILP vs. SLP**
Single family homes (A)						
Workday	0.89	0.74	0.84	<0.001	<0.001	<0.001
Saturday	1.00	0.80	0.89	<0.001	<0.001	<0.001
Holiday	1.07	0.88	0.97	<0.001	<0.001	<0.001
Residential aggregations (B)						
Workday	0.45	0.38	0.40	<0.001	<0.001	<0.001
Saturday	0.49	0.39	0.43	<0.001	<0.001	<0.001
Holiday	0.55	0.45	0.47	<0.001	<0.001	<0.001
Large residential aggregations (C)						
Workday	0.27	0.23	0.24	<0.001	<0.001	<0.001
Saturday	0.27	0.23	0.25	<0.001	<0.001	<0.001
Holiday	0.34	0.28	0.29	<0.001	<0.001	<0.001
Single enterprises (D)						
Workday	0.52	0.55	0.83	0.030	<0.001	<0.001
Saturday	0.32	0.34	0.69	0.001	<0.001	<0.001
Holiday	0.35	0.29	0.54	<0.001	<0.001	<0.001
Commercial aggregations (E)						
Workday	0.37	0.39	0.46	0.4	<0.001	<0.001
Saturday	0.24	0.23	0.34	0.087	<0.001	<0.001
Holiday	0.28	0.21	0.32	<0.001	<0.001	<0.001
Large commercial aggregations (F)						
Workday	0.17	0.16	0.24	>0.9	<0.001	<0.001
Saturday	0.12	0.12	0.24	0.005	<0.001	<0.001
Holiday	0.17	0.12	0.21	<0.001	<0.001	<0.001

Note:

Paired Wilcoxon signed rank test was used to compute p-values.

10.1.2 Parametric Forecasts

In this section, we present the forecasting results that were obtained by the reference models relying on parametric regression methodology. In particular, we evaluate the forecasts computed by the ARIMA and ANN-based models. We observed that parametric models improve the SLP-forecast only in some cases as we describe below.

10.1.2.1 ARIMA

We begin presenting the forecast errors obtained by ARIMA-models using direct (ARIMA-D) and recursive (ARIMA-R) multistep strategies. We observed that the recursive approach resulted in significantly more accurate forecasts as we show subsequently.

For both models, daily errors were widely spread at each load size (Figure 10.4). On the smallest loads, errors of several hundreds percent were observed for both models.

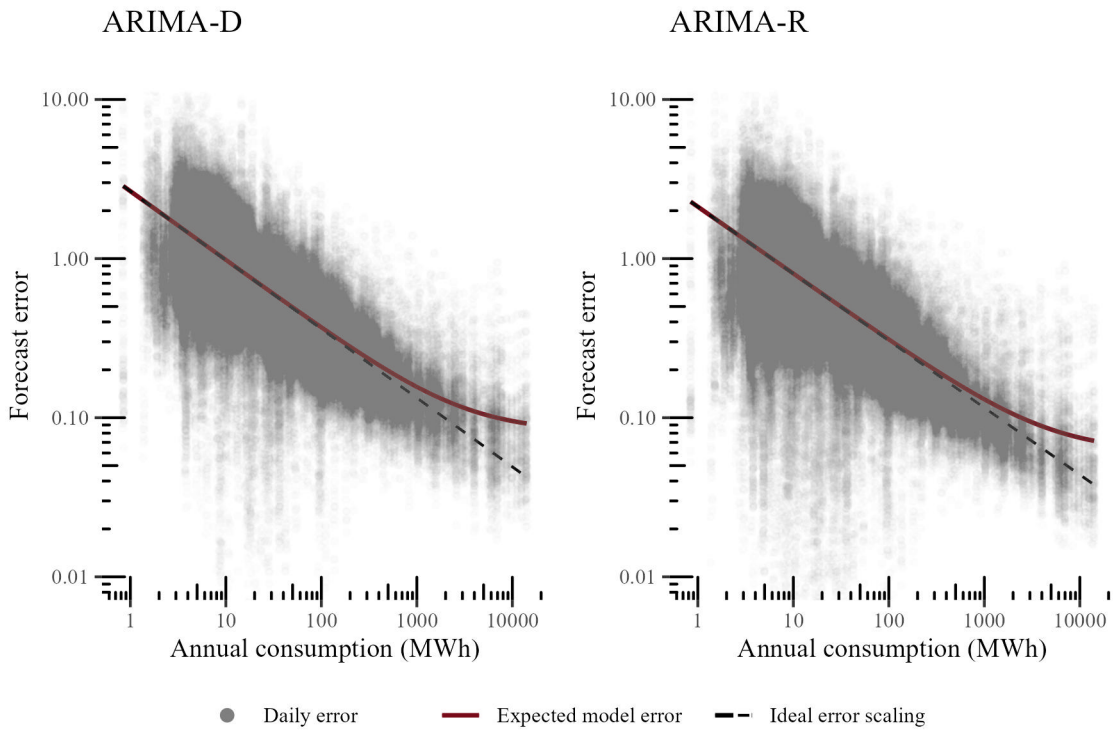

Figure 10.4: ARIMA – forecast errors. Each panel presents the 283,203 daily errors (grey dots) obtained by an ARIMA-model (Table 9.5) in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1) on the loads of the specified size (annual consumption). For each model, we computed the expected model error according to the empirical scaling law (7.17) using nonlinear weighted regression (red line) and compared it to the ideal scaling (black line). The discussion of the results is provided in the text (Section 10.1.2.1).

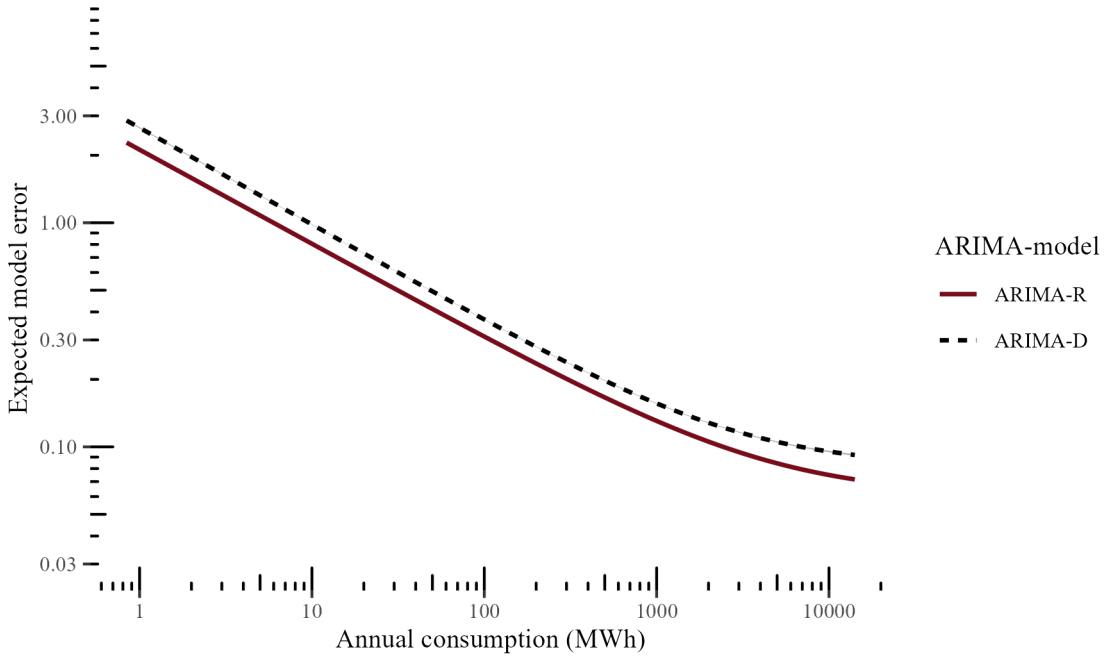

Figure 10.5: ARIMA – *expected model error (EME)* comparison. The forecast error that we can expect from a model when predicting a load of a given size was computed applying the empirical scaling law (7.17) on the corresponding sample of 283,203 daily forecast errors obtained with each ARIMA-model in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1). On each sample, we used the weighted nonlinear regression estimating the parameters p, α, β of the fitted curve representing the EME on the figure. The estimated parameters are denoted in Table 10.1. Further discussion of the results in provided in the text (Section 10.1.2.1).

At the same time, ARIMA-models often achieved very low errors[3], especially for the loads whose annual consumption was smaller than 100 MWh. This happens when the enterprise was closed or when the household inhabitants were on vacation. ARIMA-model, that considered only few lags had the flexibility to react faster to the concept change and exhibit behavior similar to persistence heuristics. Recursive multistep strategy appeared to be slightly better at this than the direct approach. The EME decreased rapidly with the consumer size until the critical load size where it saturated and converged towards the irreducible error. For larger loads, only small reduction of the error could be obtained despite increasing size.

Direct ARIMA had the critical load size of 355 MWh where the error decrease due to the aggregation began to wane before saturating at 0.08. Recursive ARIMA had higher critical load size of 529 MWh and lower irreducible error of 0.06 (Table 10.1). Comparing both

[3] Consider the forecast errors (grey dots) in the lower triangle of the Figure 10.4. These correspond to the errors that were observed on special days where an enterprise was closed or inhabitants were absent.

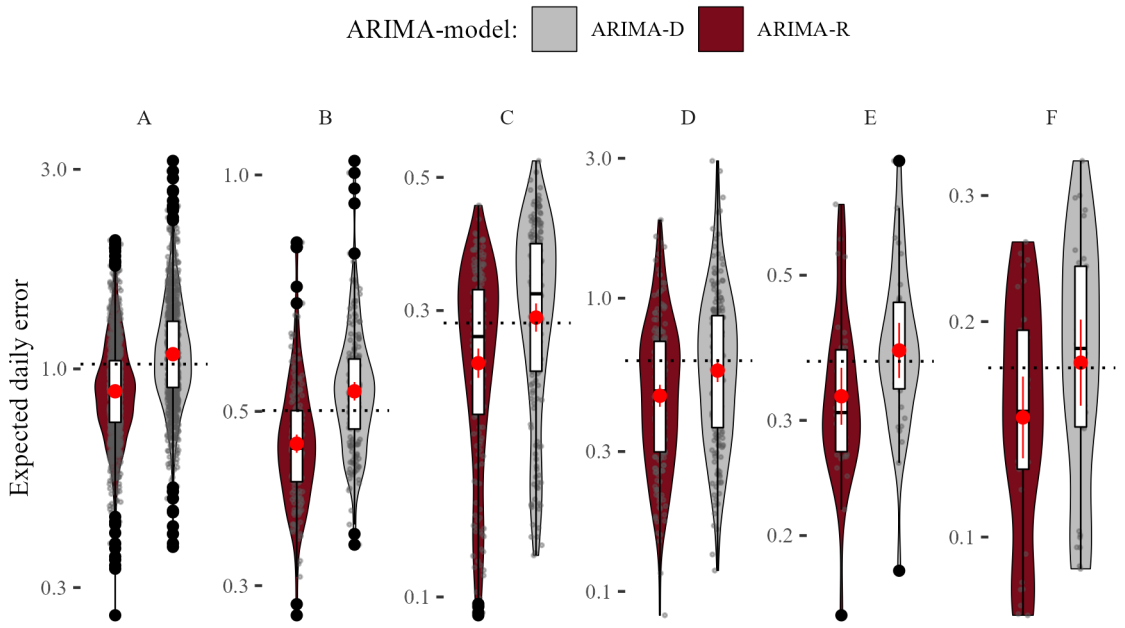

Figure 10.6: ARIMA – *expected daily error (EDE)* distribution by load group. In a wide-scale day-ahead building load forecasting simulation (Section 9.1.1), we applied various ARIMA-models (Table 9.5) predicting 1851 loads of different size and type. For each load, we obtained a sample of 153 daily forecast errors (7.13) and computed the EDE (7.15) of the corresponding model. The figure presents the EDEs obtained by the models in residential (A-C) and commercial (D-F) load groups (Table 9.3). Each panel shows the values (grey dots) obtained predicting individual loads of the corresponding group and their distribution (box and violin plots). Additionally, we denoted the expected EDE-mean (red dot) and its 95%-confidence interval (vertical red bars) for each model. Further discussion is provided in the text (Section 10.1.2.1).

EMEs, we noted that the recursive ARIMA-R could be expected to have smaller error for all load sizes (Figure 10.5). In fact, it had higher critical load size and yielded a 33% lower irreducible error which made it substantially more accurate on larger loads.

The recursive strategy used with ARIMA was 20%–30% more accurate than the direct strategy in various load groups (Table 10.2). In each group, recursive ARIMA had significantly smaller total error than the direct strategy ($p < 0.001$). The corresponding IQR of the recursive strategy was also slightly narrower. On households, the error of ARIMA-R was 0.79 and dropped to 0.24 which it achieved on large residential aggregations. In general, both ARIMA-models were much more accurate on commercial loads. On enterprises, ARIMA-R achieved 0.34 that dropped to 0.12 on large aggregations.

For each load group, we present the EDE-distribution obtained on the loads belonging to the group (Figure 10.6). Applying log scale, the distribution was approximately normal (i.e., log-normal in the original scale). For larger loads (C, F), the distribution was slightly skewed due to the uneven selection of the loads in terms of size. In all load groups,

recursive approach was, on average, significantly more accurate than the direct approach ($p < 0.001$). Additionally, there were many small residential loads where the both ARIMA-model delivered an inadequate forecast (outliers). On households, both models had EDE of around 1 and more. For commercial loads, both approaches were considerably more accurate.

We compare the accuracy of ARIMA-models to the SLP-forecast (Table 10.3). ARIMA-models improved the SLP-forecast significantly only on commercial loads ($p < 0.001$). There, the improvement was notable with 31%–48%. However, on residential loads, the ARIMA-forecast provided, on average, no improvement. In fact, direct ARIMA was notably less accurate than the SLP-forecast. Considering IQR, we saw that on, some days, ARIMA-forecast was less accurate than the SLP-forecast even on commercial loads.

10.1.2.2 Neural Networks

We made several observations considering daily forecast errors and the EME when applying neural-network-based models (MLP-D, MLP-M, NAR, NARX, DNN). We observed a large spread among daily forecast errors at each load size (Figure 10.7). On the smallest loads, considerable errors occurred often. With some architectures, we observed small errors on special days, though such observations were less often than with ARIMA-models[4]. Whenever the consumers were absent or the building was temporally closed (e.g., vacation), the monthly retrained ANN-models did not have the flexibility to quickly identify and adjust themselves for such situations. The EME and the variation decreased rapidly with the size reaching critical load size where scaling saturated and converged towards the irreducible error (Table 10.1). For larger loads, only small reduction of the error could be expected despite further aggregation.

We observed that the multi-out MLP-M-network with one hidden layer and the deep neural networks with two and three layers usually provided almost identical daily errors. As a result, the differences in the EME were almost indistinguishable while these models had similar critical load size and irreducible error (Table 10.1). This suggests that the network with one layer had enough modeling capacity for the load forecasting task. The surplus was suppressed by the Baysian regularization preventing overfitting.

Considering the EME, allowed us to compare different multistep strategies (Figure 10.8). We observed that direct (MLP-D) and the multi-out architectures (MLP-M) had similar accuracy on smaller loads and were significantly more accurate than the recursive strategy (NARX). However, the MLP-D had the lowest critical load size (68 MWh) and the error

[4] Consider errors (grey dots) in the lower triangle of the Figure 10.7. These correspond to the errors that were observed on special days where an enterprise was closed or inhabitants were absent.

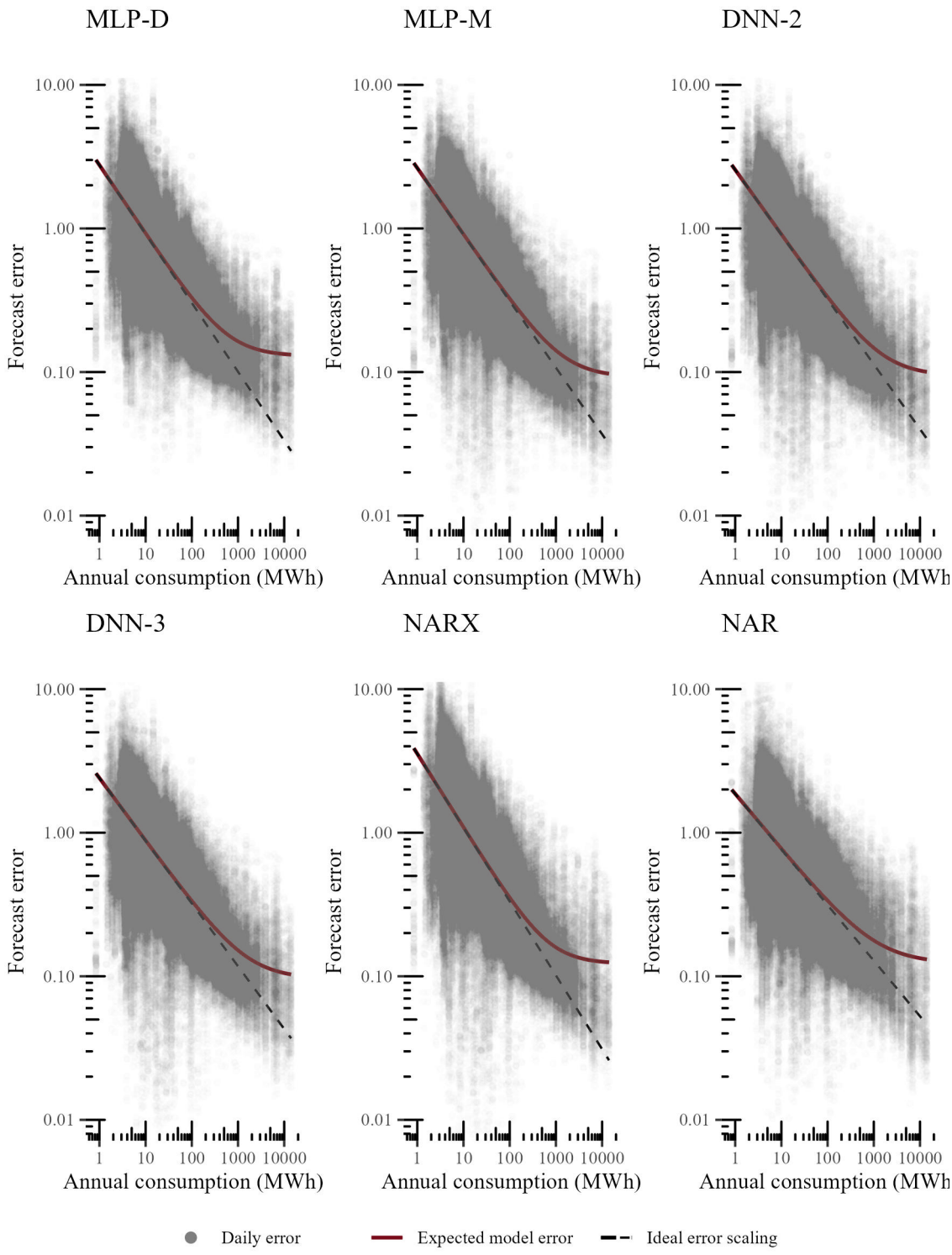

Figure 10.7: Neural networks – forecast errors. Each panel presents the 283,203 daily errors (grey dots) obtained by a neural-network-based model (Table 9.5) in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1) on the loads of the specified size (annual consumption). For each model, we computed the expected model error according to the empirical scaling law (7.17) using nonlinear weighted regression (red line) and compared it to the ideal error scaling (black line). The discussion of the results is provided in the text (Section 10.1.2.2).

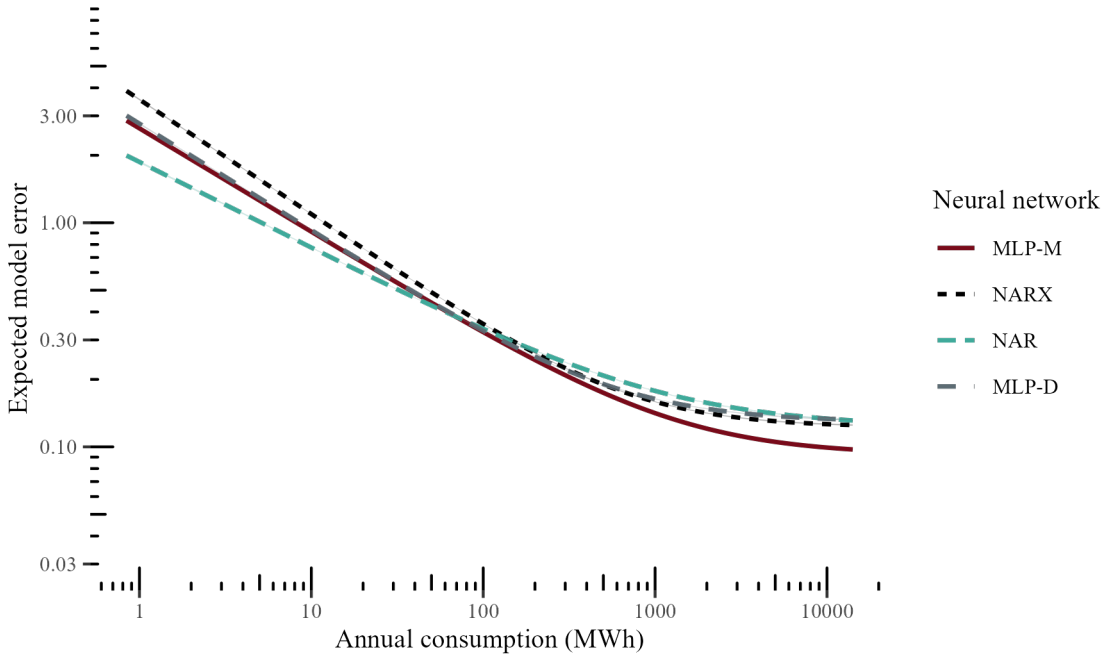

Figure 10.8: Neural networks – *expected model error (EME)* comparison. The forecast error that we can expect from a model when predicting a load of a given size was computed applying the empirical scaling law (7.17) on the corresponding sample of 283,203 daily forecast errors obtained with each neural-network-based model in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1). On each sample, we used the weighted nonlinear regression estimating the parameters p, α, β of the fitted curve representing the EME on the figure. The estimated parameters are denoted in Table 10.1. Further discussion of the results in provided in the text (Section 10.1.2.2).

saturated faster towards the larger E_0 of 0.13 than with any other network. At the same time, the MLP-M had the smallest irreducible error of 0.09 among all networks.

Moreover, we can expect NAR to be more accurate than other architectures on small loads (Figure 10.8). For middle-sized loads of around 100 MWh annual consumption, all architectures can be expected to have comparable accuracy. For the loads larger than 300 MWh, MLP-M becomes the most accurate network. Recursive architectures (NAR, NARX) had irreducible errors that were approximately 33% higher than that of multi-out networks (MLP-M, DNN). Additionally, the latter had a higher critical load size of 159–181 MWh.

Consider the total errors in different load groups (Table 10.2). On commercial loads, all models achieved substantially lower errors than on residential loads. The errors on single enterprises (D) were comparable with those on residential aggregations (B). MLP-M and DNN often had almost identical daily errors reflected by similar total errors and IQRs in each group.

The NAR had the lowest total error on smaller loads (A,B,D). On residential loads, starting with 0.75 (A), the error reduced rapidly to 0.25 on large residential aggregations (C), where all networks had comparable accuracy. On single enterprises (D), NAR was the most accurate network obtaining error of 0.38.

At the same time, the other recursive model (NARX) was the least accurate on single enterprises (D) and commercial aggregations (E) with total errors of 0.58 and 0.41 respectively. On large commercial aggregations (F), only direct model (MLP-D) was 22% less accurate. Other networks had comparable errors of 0.15–0.17.

The EDE-distribution allowed us to compare the models within the load groups in terms of error average and variation (Figure 10.9). DNNs were left out, since they had almost the same accuracy as the MLP-M. On a log scale, the distribution was approximately normal (i.e., log-normal in the original scale). The EDE-distribution on large loads (C,F) was slightly skewed due to uneven distribution of loads in terms of size in these groups.

A paired Wilcoxon signed rank test, conducted post-hoc, confirmed that the NAR and MLP-M were the most accurate models in most load groups (Table 10.5). NAR was the most accurate network on smaller loads. In particular, it was significantly more accurate ($p < 0.001$) on single family homes (A), enterprises (D) and smaller residential aggregations (B) than any other network. On commercial aggregations (E), NAR and MLP-M had similar accuracy (difference was not significant) outperforming other networks by 22%–32%. However, on larger loads (C, F), MLP-M was the most accurate network outperforming NAR by 8%–79% with high significance of the results (p<0.001). In some rare cases, MLP-D and MLP-M had very similar accuracy (A, C), but overall, MLP-M is 17%–50% more accurate ($p < 0.001$).

As we note previously, MLP-M and DNN often had almost identical daily errors. The tests confirmed that the difference between the total errors obtained with MLP-M and DNNs was either not significant or MLP-M was slightly (under 1%) more accurate in some groups with p-values ranging from $p = 0.01$ to $p = 0.1$ (Table 10.6). Only on single family homes (A), DNN with 3 layers was significantly more accurate ($p < 0.001$), but in that case, all networks had very high total error of around 0.95.

Only in some cases, we observed that neural-network-based models were more accurate than the SLP-forecast. To make a direct comparison, we computed the accuracy increase (7.14) relative to the SLP-approach for each load and day. The daily improvement is summarized in Table 10.3. All neural networks were more effective when applied to commercial loads where MLP-M and NAR significantly improved the SLP-forecast. For instance, the improvement on single enterprises (D) or commercial aggregations (E) was much larger than on large residential aggregations (C). However, on smaller residential loads (A,B) only NAR achieved a small, yet significant, 2%–3% improvement over the

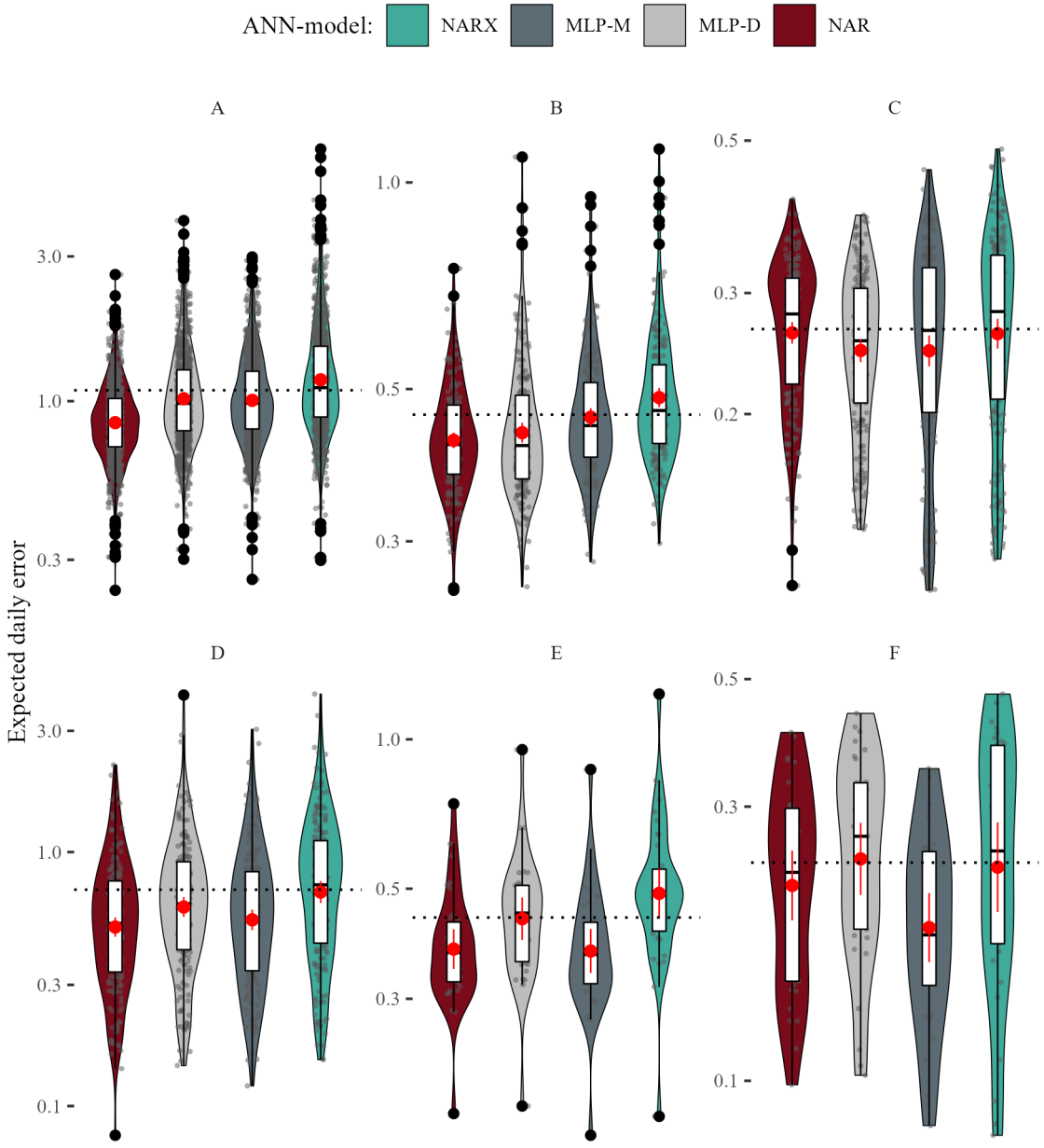

Figure 10.9: Neural networks – *expected daily error (EDE)* distribution by load group. In a wide-scale day-ahead building load forecasting simulation (Section 9.1.1), we applied various neural-network-based models (Table 9.5) predicting 1851 loads of different size and type. For each load, we obtained a sample of 153 daily forecast errors (7.13) and computed the EDE (7.15) of the corresponding model. The figure presents the EDEs obtained by the models in residential (A-C) and commercial (D-F) load groups (Table 9.3). Each panel shows the values (grey dots) obtained predicting individual loads of the corresponding group and their distribution (box and violin plots). Additionally, we denoted the expected EDE-mean (red dot) and its 95%-confidence interval (vertical red bars) for each model. Further discussion is provided in the text (Section 10.1.2.2).

SLP-forecast ($p < 0.001$). On large residential aggregations (C), no ANN-model evaluated in this study brought any significant improvement. On commercial loads (E,F), MLP-M was more accurate than the NAR-model obtaining mean improvement of 11% to 21%. Again, MLP-M and DNN had comparable improvement in each load group. At the same time, NARX and MLP-D often brought the smallest or negative improvement comparing to other architectures.

Our results allow no general conclusions about the effectiveness of neural networks for the load forecasting. While for some individual loads, an improvement exceeding 50% was obtained with various networks, there were load groups where on more than a half of the loads no improvement against currently used SLP-approach was obtained.

Table 10.5: Neural networks – total error comparison. We applied the paired Wilcoxon signed rank test on the sample of 283,203 daily forecast errors obtained in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1) to evaluate the statistical significance of the total error difference between the selected ANN-models confounded on load group and day-type. The results are discussed in the text (Section 10.1.2.2).

	Model				p-values					
	MLP-D	**MLP-M**	**NAR**	**NARX**	**MLP-M vs. MLP-D**	**MLP-M vs. NAR**	**MLP-M vs. NARX**	**MLP-D vs. NAR**	**MLP-D vs. NARX**	**NAR vs. NARX**
Single family homes (A)										
Workday	0.93	0.95	0.81	1.04	0.6	<0.001	<0.001	<0.001	<0.001	<0.001
Saturday	1.03	1.02	0.86	1.14	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Holiday	1.10	1.08	0.94	1.29	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Residential aggregations (B)										
Workday	0.40	0.43	0.40	0.44	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Saturday	0.43	0.45	0.43	0.49	<0.001	<0.001	<0.001	0.2	<0.001	<0.001
Holiday	0.47	0.51	0.48	0.56	<0.001	<0.001	<0.001	0.8	<0.001	<0.001
Large residential aggregations (C)										
Workday	0.25	0.25	0.27	0.26	<0.001	0.005	<0.001	<0.001	<0.001	0.8
Saturday	0.26	0.27	0.27	0.28	0.13	0.7	<0.001	0.005	<0.001	<0.001
Holiday	0.30	0.31	0.32	0.34	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Single enterprises (D)										
Workday	0.66	0.58	0.54	0.80	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Saturday	0.57	0.49	0.40	0.65	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Holiday	0.51	0.36	0.45	0.53	<0.001	<0.001	<0.001	<0.001	0.005	<0.001
Commercial aggregations (E)										
Workday	0.47	0.43	0.39	0.52	<0.001	0.2	<0.001	<0.001	<0.001	<0.001
Saturday	0.38	0.31	0.32	0.41	<0.001	0.2	<0.001	0.002	0.005	<0.001
Holiday	0.38	0.24	0.31	0.36	<0.001	<0.001	<0.001	<0.001	0.2	0.050
Large commercial aggregations (F)										
Workday	0.28	0.21	0.22	0.27	<0.001	0.004	<0.001	<0.001	0.8	<0.001
Saturday	0.22	0.17	0.22	0.23	<0.001	0.024	<0.001	0.080	0.4	0.005
Holiday	0.25	0.14	0.25	0.21	<0.001	<0.001	<0.001	0.9	0.011	0.031

Note:

Paired Wilcoxon signed rank test was used to compute p-values.

Table 10.6: Deep neural networks – total error comparison. We applied the paired Wilcoxon signed rank test on the sample of 283,203 daily forecast errors obtained in the wide-scale day.ahead local load forecasting simulation (Section 9.1.1) to evaluate the statistical significance of the total error difference between the selected ANN-models confounded on load group and day-type. The results are discussed in the text (Section 10.1.2.2).

	Model			p-values		
	MLP-M	**DNN-2**	**DNN-3**	**MLP-M vs. DNN-2**	**MLP-M vs. DNN-3**	**DNN-2 vs. DNN-3**
Single family homes (A)						
Workday	0.95	0.96	0.92	0.4	<0.001	<0.001
Saturday	1.02	1.01	0.98	0.5	<0.001	<0.001
Holiday	1.08	1.07	1.05	<0.001	<0.001	<0.001
Residential aggregations (B)						
Workday	0.43	0.44	0.43	<0.001	<0.001	0.003
Saturday	0.45	0.46	0.45	<0.001	0.5	<0.001
Holiday	0.51	0.51	0.50	0.064	0.2	<0.001
Large residential aggregations (C)						
Workday	0.25	0.26	0.26	<0.001	<0.001	0.054
Saturday	0.27	0.27	0.27	<0.001	<0.001	>0.9
Holiday	0.31	0.32	0.32	<0.001	<0.001	0.019
Single enterprises (D)						
Workday	0.58	0.60	0.60	<0.001	0.001	0.4
Saturday	0.49	0.50	0.50	0.3	>0.9	0.062
Holiday	0.36	0.34	0.35	0.061	0.005	0.3
Commercial aggregations (E)						
Workday	0.43	0.42	0.42	0.018	<0.001	0.020
Saturday	0.31	0.34	0.32	0.044	0.3	0.4
Holiday	0.24	0.26	0.26	0.3	0.070	0.4
Large commercial aggregations (F)						
Workday	0.21	0.23	0.23	<0.001	<0.001	0.006
Saturday	0.17	0.18	0.18	0.4	0.2	0.3
Holiday	0.14	0.15	0.15	0.020	0.051	>0.9

Note:

Paired Wilcoxon signed rank test was used to compute p-values.

10.1.3 Nonparametric Forecasts

Similar to the previously discussed model families, nonparametric reference models (MKNN, NWE, UA), also had a large spread among daily forecast errors, at each load size (Figure 10.10). On the smallest loads, considerable errors of several magnitudes of the average daily load occurred often. At the same time, all models often obtained very low errors, especially on the loads whose annual consumption was smaller than 100 MWh[5]. Whenever the consumers were absent or the building was temporally closed (e.g., vacation), nonparametric models had the flexibility to quickly identify and adjust themselves for such situations.

The EME and the variation of daily errors decreased rapidly with the annual consumption reaching critical load size where error scaling saturated and converged towards the irreducible error (Table 10.1). For larger loads, only small reduction in EME could be expected despite increasing size. Comparing the EMEs, we noted that the UA-model had substantially lower critical load size and began to saturate earlier than NWE and MKNN (Figure 10.11). While the MKNN could be expected to be more accurate on smaller loads, all models had very similar accuracy on the middle-sized and large loads. The irreducible error for all nonparametric models was around 6%, yet that of the NWE was slightly, but significantly, lower ($p < 0.001$).

In each load group, daily errors were approximately log-normally distributed and could be summarized in terms of median and IQR (Table 10.2). All nonparametric models had similar accuracy on most loads which was consistent with the EME comparison (Figure 10.11). On residential loads, the models had the largest total error of 0.68–0.72 on single family homes (A), which dropped to 0.22–0.24 for large residential aggregations (C). For commercial loads (D,E,F), the total error was notably lower. Even on single enterprises (D), daily errors were often smaller than on large residential aggregations, though the IQR was also wider.

Nevertheless, a paired Wilcoxon signed rank test, conducted post-hoc, showed that the difference between the models was mostly significant except for enterprises (D) where on workdays all models had similar accuracy (Table 10.7). At the same time, no model was consistently better for all groups and the difference was often small. For instance, MKNN was 3%–8% more accurate on single family homes (A) but was 5%–6% less accurate than the others on commercial aggregations (E, F). At the same time, UA was 2%–5% more accurate than others on residential aggregations (B, C).

[5] Consider the errors (grey dots) in the lower triangle of the Figure 10.10. These correspond to the errors that were observed on special days where an enterprise was closed or inhabitants were absent.

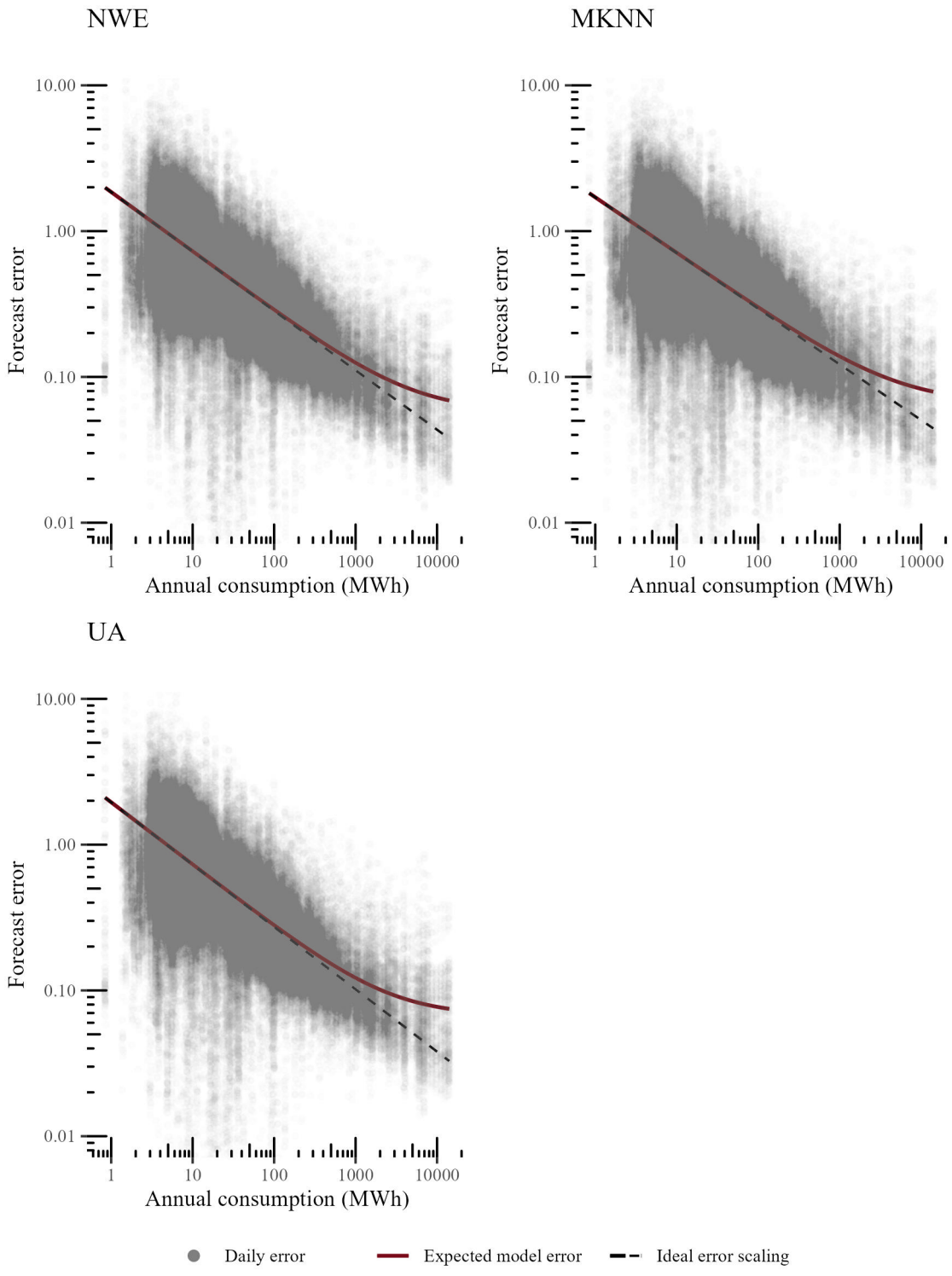

Figure 10.10: Nonparametric models – forecast errors. Each panel presents the 283,203 daily errors (grey dots) obtained by a nonparametric reference model (Table 9.5) in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1) on the loads of the specified size (annual consumption). For each model, we computed the expected model error according to the empirical scaling law (7.17) using nonlinear weighted regression (red line) and compared it to the ideal error scaling (black line). The discussion of the results is provided in the text (Section 10.1.3).

Table 10.7: Nonparametric models – total error comparison. We applied the paired Wilcoxon signed rank test on the sample of 283,203 daily forecast errors obtained in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1) to evaluate the statistical significance of the total error difference between the heuristic models confounded on load group and day-type. The results are discussed in the text (Section 10.1.3).

	Model			p-values		
	MKNN	**NWE**	**UA**	**MKNN vs. NWE**	**MKNN vs. UA**	**NWE vs. UA**
Single family homes (A)						
Workday	0.74	0.75	0.76	<0.001	<0.001	<0.001
Saturday	0.78	0.81	0.84	<0.001	<0.001	<0.001
Holiday	0.87	0.90	0.90	<0.001	<0.001	<0.001
Residential aggregations (B)						
Workday	0.40	0.39	0.38	<0.001	<0.001	<0.001
Saturday	0.41	0.41	0.41	0.5	0.019	0.018
Holiday	0.47	0.47	0.46	<0.001	<0.001	0.017
Large residential aggregations (C)						
Workday	0.26	0.25	0.23	<0.001	<0.001	<0.001
Saturday	0.26	0.25	0.24	<0.001	<0.001	<0.001
Holiday	0.32	0.30	0.29	<0.001	<0.001	<0.001
Single enterprises (D)						
Workday	0.48	0.48	0.47	0.2	0.3	0.2
Saturday	0.30	0.30	0.30	<0.001	0.2	0.074
Holiday	0.23	0.24	0.25	<0.001	<0.001	0.2
Commercial aggregations (E)						
Workday	0.35	0.34	0.33	<0.001	<0.001	0.034
Saturday	0.22	0.21	0.21	0.002	<0.001	0.002
Holiday	0.20	0.19	0.20	0.027	0.062	>0.9
Large commercial aggregations (F)						
Workday	0.16	0.15	0.15	<0.001	<0.001	0.2
Saturday	0.12	0.11	0.11	<0.001	<0.001	0.5
Holiday	0.12	0.11	0.12	<0.001	<0.001	0.4

Note:

Paired Wilcoxon signed rank test was used to compute p-values.

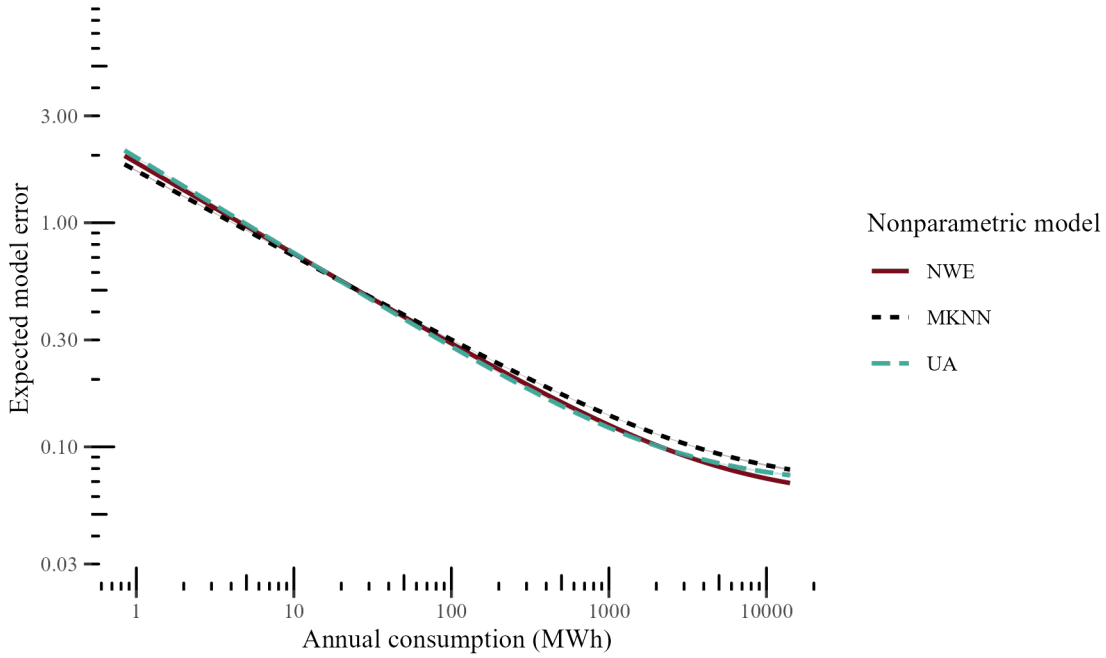

Figure 10.11: Nonparametric models – *expected model error (EME)* comparison. The forecast error that we can expect from a model when predicting a load of a given size was computed applying the empirical scaling law (7.17) on the corresponding sample of 283,203 daily forecast errors obtained with each nonparametric reference model in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1). On each sample, we used the weighted nonlinear regression estimating the parameters p, α, β of the fitted curve representing the EME on the figure. The estimated parameters are denoted in Table 10.1. Further discussion of the results in provided in the text (Section 10.1.3).

Often, nonparametric models significantly improved the SLP-forecast in the majority of the load groups (Table 10.3). When applied on single family homes (A), the models could be expected to improve SLP-forecast by 8%–12%, which was more than the improvement achieved by the best performing NAR-network. On residential loads, the improvement waned with load size as the SLP-forecast also became more accurate. For large residential aggregations (C), only UA-model achieved a significant improvement of 6% ($p < 0.001$). The improvement was more substantial on commercial loads. There, all models had average improvement of 25%–42%. On commercial loads, nonparametric models were much more accurate than the SLP-forecast.

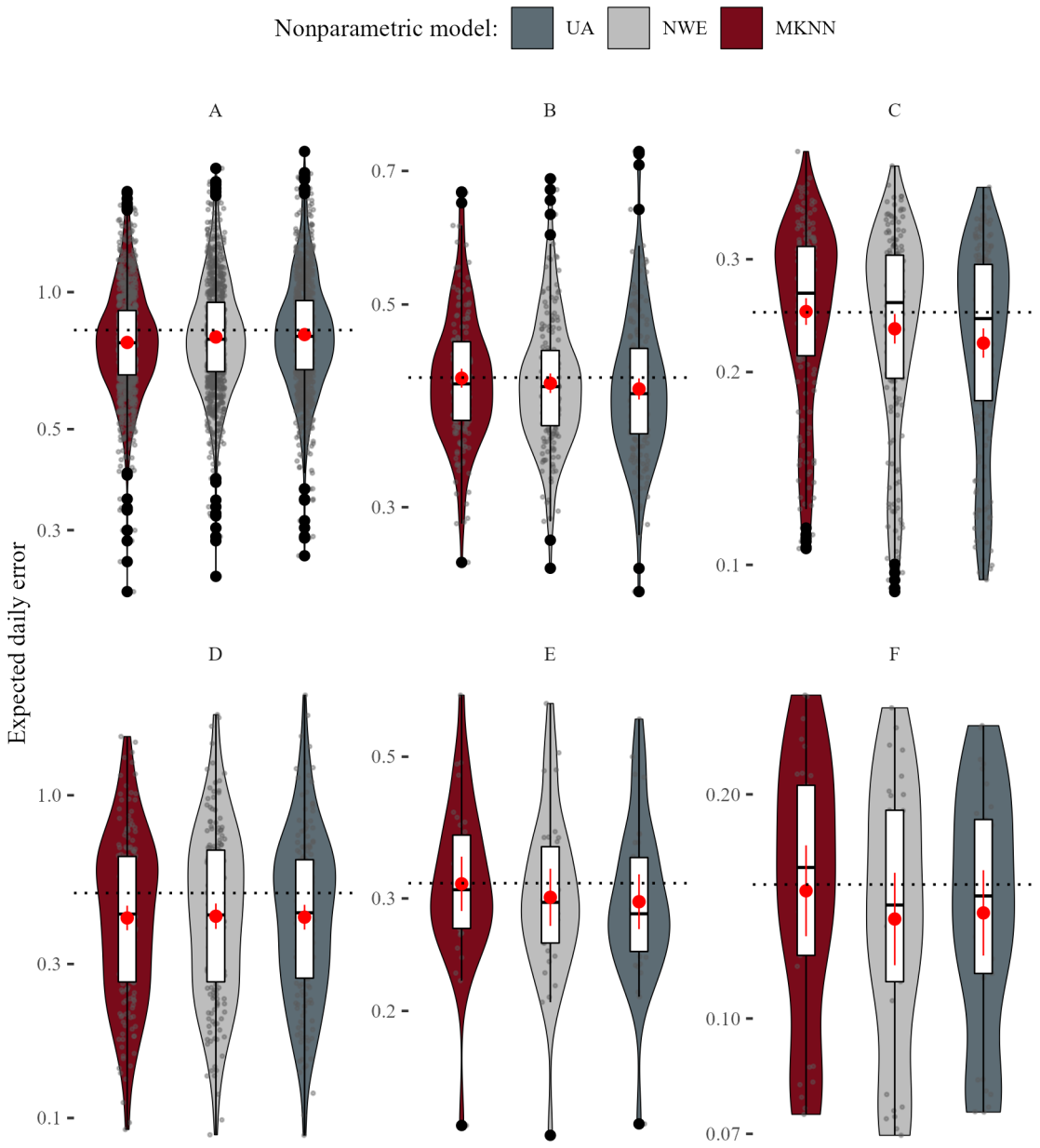

Figure 10.12: Nonparametric models – *expected daily error (EDE)* distribution by load group. In a wide-scale day-ahead building load forecasting simulation (Section 9.1.1), we applied various nonparametric models (Table 9.5) predicting 1851 loads of different size and type. For each load, we obtained a sample of 153 daily forecast errors (7.13) and computed the EDE (7.15) of the corresponding model. The figure presents the EDEs obtained by the models in residential (A-C) and commercial (D-F) load groups (Table 9.3). Each panel shows the values (grey dots) obtained predicting individual loads of the corresponding group and their distribution (box and violin plots). Additionally, we denoted the expected EDE-mean (red dot) and its 95%-confidence interval (vertical red bars) for each model. Further discussion is provided in the text (Section 10.1.3).

10.2 Functional Neighbor Forecasts

In this section, we evaluate the forecasts provided by the *functional neighbor (FN)* model proposed in Section 8.2. We present the results obtained in the wide-scale day-ahead building load forecasting simulation and compare the accuracy of our forecaster to the predictions by the previously discussed reference models. The results of the smart-building load forecasting simulation evaluating the functional neighbor extension model that considers exogenous variables are provided later in the text (Section 8.3).

As with the reference models, we observed a large spread among daily forecast errors at each load size (Figure 10.13). The largest errors were observed on the smallest loads. At the same time, on the loads between 2 MWh and 100 MWh, we often observed very

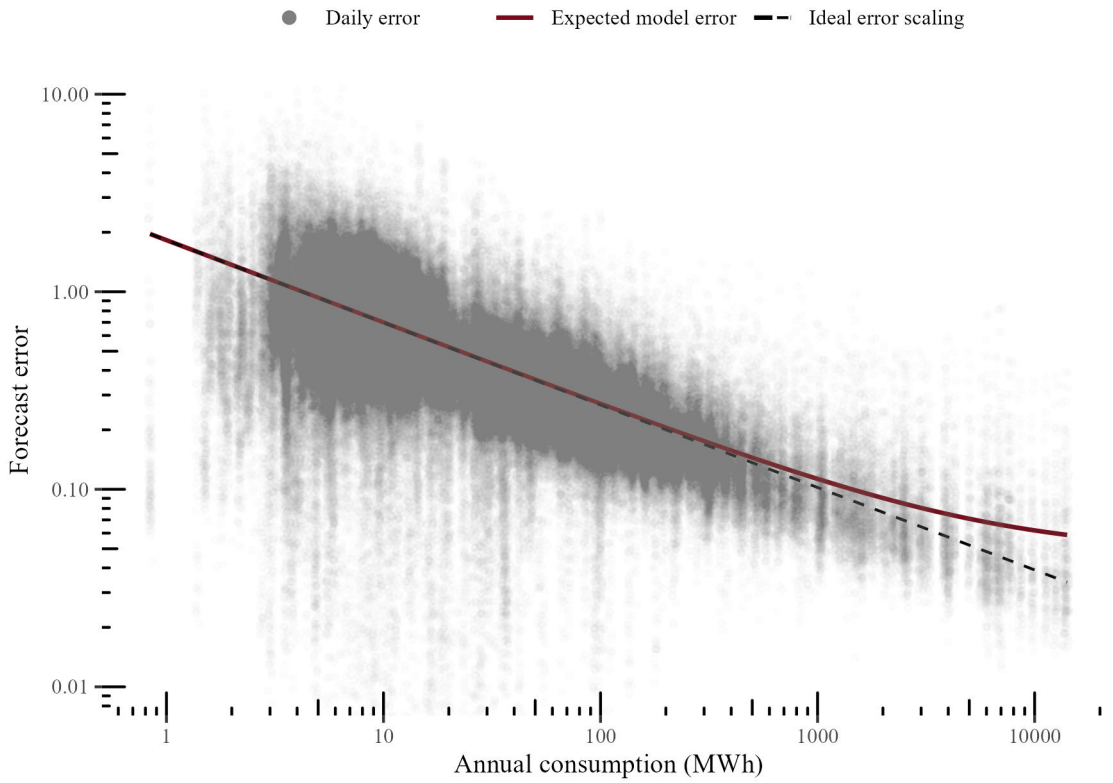

Figure 10.13: Functional neighbor model – forecast errors. The figure shows the 283,203 daily errors (grey dots) obtained by the functional neighbor model (Algorithm 3) in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1) on the loads of the specified size (annual consumption). Additionally, we computed the expected model error according to the empirical scaling law (7.17) using nonlinear weighted regression (red line) and compared it to the ideal error scaling (black line). Further discussion is provided in the text (Section 10.2).

small errors[6]. Whenever the consumers were absent or the building was temporally closed (e.g., vacation), the FN-model had the flexibility to quickly identify and adjust itself to such situations. The EME and the variation of daily errors decreased rapidly reaching the critical load size (695 MWh) where scaling saturated and converged towards the irreducible error of 0.048. For larger loads, only a small reduction of the error could be obtained despite the increasing size.

The FN-model had the smallest irreducible error compared to the reference models (Table 10.1). For the largest loads, our model could be expected to be 39% more accurate[7] than the SLP-forecast that was designed for predicting large aggregations of end-consumers and had the highest critical load size (836 MWh) among all reference models.

Extending the comparison to other heuristic models, the FN-model had notably smaller EME than any heuristic model for the loads larger than 30 MWh (large home or a small enterprise). In particular, the FN-model had 31% lower irreducible error than the best heuristic model – ILP (Figure 10.14).

Comparing to the parametric models, FN-model can be expected to have lower forecast error on loads of any size (Figure 10.14). Even on larger loads, its irreducible error was 23% smaller than that of the best parametric model (ARIMA). Notably, the FN-model was more accurate than the ANN-based models which we used as a reference in this study.

The FN-model had accuracy similar to the other nonparametric models on smaller loads. Our forecaster started to distinguish itself from those models for the loads larger than 100 MWh (e.g., an average building) for which the EME was notably smaller than that of other nonparametric models. The irreducible error of the FN-model was 17% lower than the one of the NWE and 27% less than the one of the MKNN (Figure 10.14).

We compared the EDE-distribution obtained by the FN- and SLP-models on residential (Figure 10.15) and commercial (Figure 10.16) loads. In each load group, the EDE-distribution, same as the original daily error distribution, had approximate log-normal shape (mind log scale of the plot). There was a slight left-skew for large aggregations that can be explained by uneven distribution of the loads in these groups (Section 9.1.1).

The paired difference between model errors[8] was approximately normal. For the groups where the difference between the means did not appear significant (e.g, C) considering the overlapping confidence intervals (horizontal bars), we conducted a Wilcoxon test and a paired t-test confirming that the FN-model had lower total and mean daily errors than the SLP-forecast in each group ($p < 0.001$).

[6] Consider the forecast errors (grey dots) in the lower triangle of the Figure 10.13. These correspond to the errors that were observed on special days where an enterprise was closed or inhabitants were absent.

[7] Here, we relate to the improvement of irreducible error relative to the SLP-forecast.

[8] Here, we relate to the difference in the forecast error obtained on the same load.

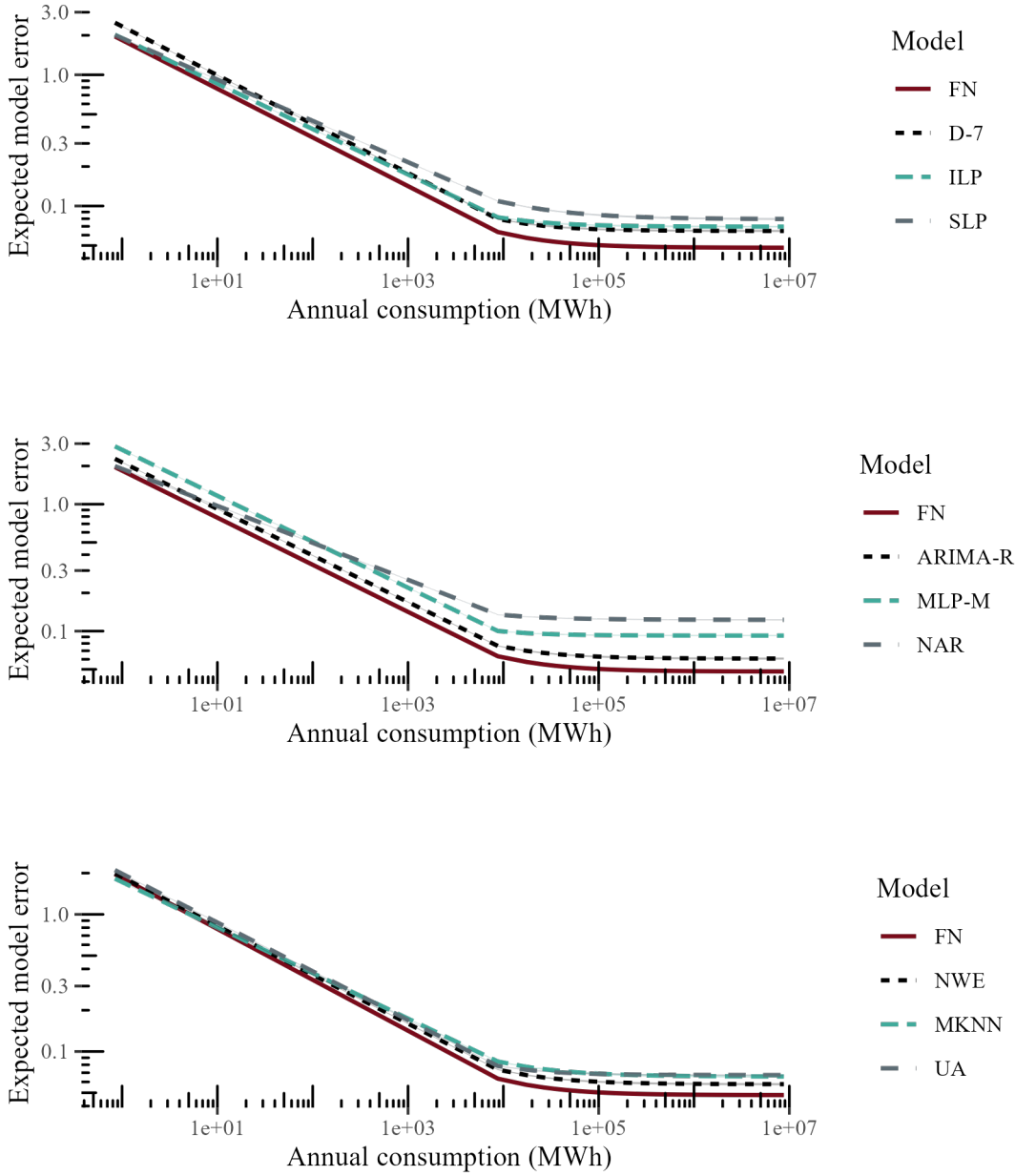

Figure 10.14: Functional neighbor model – *expected model error (EME)* comparison to the reference models. The forecast errors that we can expect from the functional neighbor model (Algorithm 3) and various reference models (Table 9.5) when predicting a load of a given size were computed applying the empirical scaling law (7.17) on the samples of 283,203 daily forecast errors obtained with each model in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1). On each sample, we used the weighted nonlinear regression estimating the parameters p, α, β of the fitted curve representing the EME on the figure. The estimated parameters are denoted in Table 10.1. Each panel compares the EME of the functional neighbor forecaster to the heuristic (top), parametric (middle) and nonparametric (bottom) reference models. Further discussion of the results is provided in the text (Section 10.2).

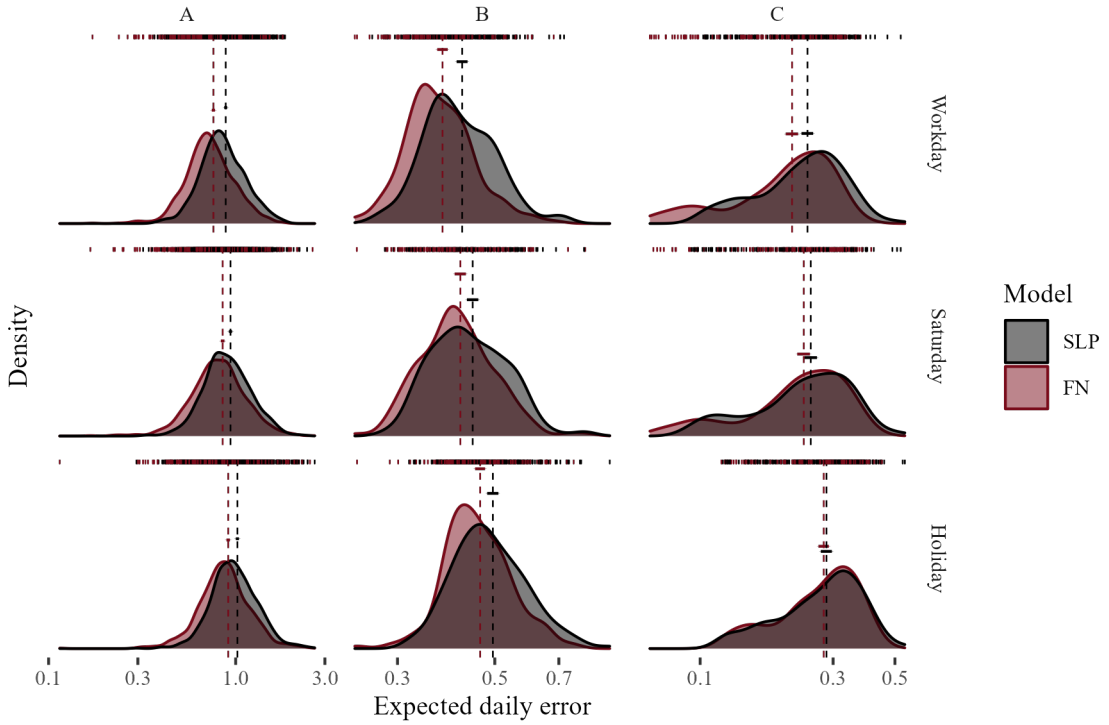

Figure 10.15: Functional neighbor model – *expected daily errors (EDE)* on residential loads. We applied the functional neighbor forecaster (Algorithm 3) and the SLP-model predicting 1247 residential loads of different size in a wide-scale day-ahead building load forecasting simulation (Section 9.1.1). For each predicted load, we computed the EDE (7.15) using the sample of 153 daily forecast errors obtained by each model. Conditioned on day-type, the panels show the EDE-distributions in residential load groups (A-C) defined in Table 9.3. Each panel shows the errors on a log scale obtained on individual loads (rugs) by the corresponding model and the probability density function. Additionally, we denoted the error mean (vertical dotted line) and its 95%-confidence interval (vertical bar). Further discussion is provided in the text.

In fact, the FN-model had the lowest total error among all reference models in every load group (Table 10.8). The difference to other models, was mostly significant ($p < 0.001$). Only in very few cases, the difference to some models (ILP, UA) was not statistically significant (Table 10.9).

For a better comparison, we consider the improvement against the SLP-forecast. On the vast majority of loads, the FN-forecast was a notable improvement of up to 100% comparing to the SLP-forecast (Figure 10.17). We saw particularly large gains of multiple dozens of percent for commercial loads and larger aggregations. For few loads, FN-model was slightly less accurate than the SLP-forecast. However, those loads were often very small – homes with annual consumption up to 20 MWh. In such cases, a deterministic forecast might not be adequate since we can have large daily errors with all models.

Overall, FN-model yielded a significant improvement comparing to the SLP-forecast in each load group (Figure 10.18). The improvement on commercial loads was three to four times higher than on residential loads. Even on single enterprises, the FN-model could

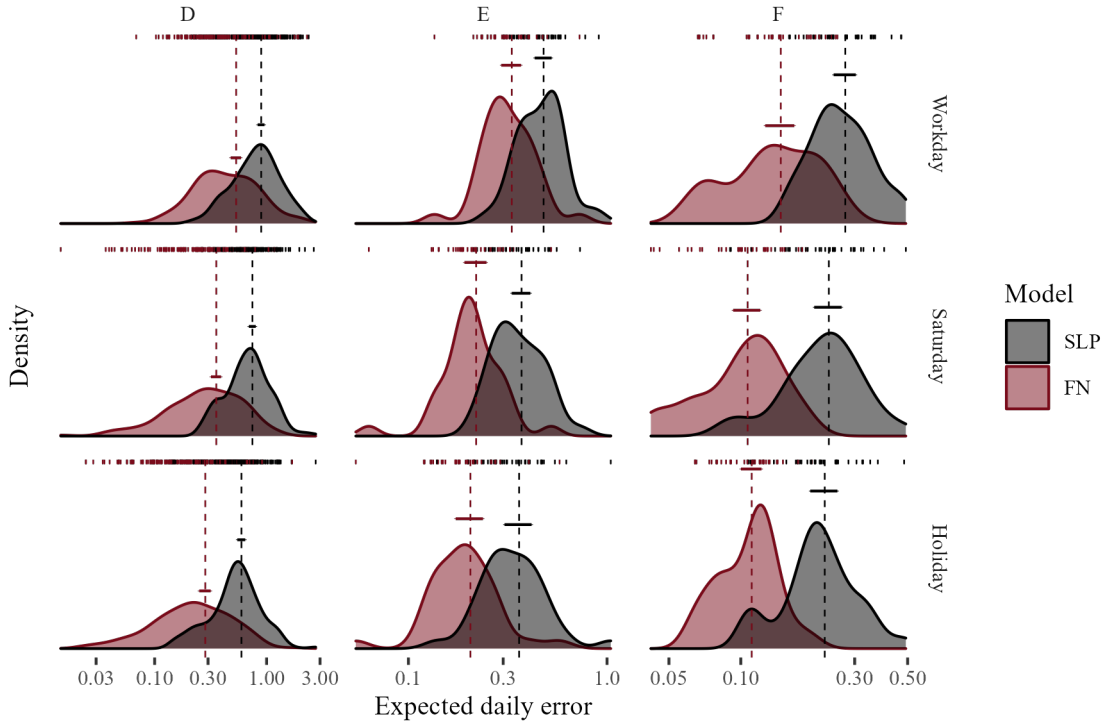

Figure 10.16: Functional neighbor model – *expected daily errors (EDE)* on commercial loads. We applied the functional neighbor forecaster (Algorithm 3) and the SLP-model predicting 242 commercial loads of different size in a wide-scale day-ahead building load forecasting simulation (Section 9.1.1). For each predicted load, we computed the EDE (7.15) using the sample of 153 daily forecast errors obtained by each model. Conditioned on day-type, the panels show the EDE-distributions in residential load groups (D-F) defined in Table 9.3. Each panel shows the errors on a log scale obtained on individual loads (rugs) by the corresponding model and the probability density function. Additionally, we denoted the error mean (vertical dotted line) and its 95%-confidence interval (vertical bar). Further discussion is provided in the text.

be expected to improve the forecast by over 43% without using any information about the enterprise such as opening hours. On residential aggregations, where the SLP-method was designed to predict particularly well, the FN-model could be expected to improve the SLP-forecast by over 10%. Even for larger aggregations (C, F) improvement of 54% and 83% was observed.

In fact, the FN-model improved the SLP-forecast more than any reference model evaluated in this study (Table 10.10). A paired t-test verified that our forecaster brought a significantly larger improvement in all load groups than any parametric (Figure 10.20), nonparametric (Figure 10.21) and heuristic (Figure 10.19) reference model. Even on single family homes, where the difference to the MKNN was under 1% it was, nevertheless, significant ($p < 0.001$). On residential aggregations (B), the difference to the most accurate heuristics (ILP) was small but could be considered significant ($p = 0.044$). Comparing the FN-model to any other reference in any other group showed an accuracy improvement with the highest significance level ($p < 0.001$).

Forecast	A	B	C	D	E	F
FN	0.76	0.39	0.24	0.41	0.27	0.13
MKNN	0.77	0.41	0.27	0.43	0.31	0.16
ILP	0.79	0.39	0.24	0.48	0.33	0.15
NWE	0.79	0.41	0.26	0.43	0.30	0.14
UA	0.80	0.40	0.24	0.43	0.28	0.15
NAR	0.85	0.42	0.28	0.52	0.37	0.23
ARIMA-R	0.88	0.45	0.27	0.48	0.31	0.15
SLP	0.87	0.42	0.26	0.78	0.42	0.25
D-7	0.94	0.47	0.28	0.49	0.33	0.17
D-1	0.93	0.46	0.28	0.58	0.45	0.28
DNN-3	0.96	0.45	0.27	0.55	0.37	0.20
MLP-M	0.99	0.44	0.26	0.54	0.37	0.18
DNN-2	0.99	0.45	0.27	0.55	0.38	0.20
MLP-D	0.98	0.41	0.26	0.63	0.45	0.27
ARIMA-D	1.07	0.53	0.32	0.57	0.38	0.18
NARX	1.11	0.47	0.28	0.74	0.49	0.25

Table 10.8: Comparison of the total errors of the forecasts in each load group. The forecasting models were evaluated in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1). For each load group defined in Table 9.3, we highlight the models that achieved the smallest (green), below average (grey) and the highest total error (7.16). Further discussion is provided in the text (Section 10.2).

Table 10.9: Functional neighbor model – total error comparison with the best reference models. We applied the paired Wilcoxon signed rank test on the sample of 283,203 daily forecast errors obtained in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1) to evaluate the statistical significance of the total error (7.16) difference between the heuristic models confounded on load group and day-type. Further discussion is provided in the text (Section 10.2).

	Model				p-values					
	FN	**ILP**	**MKNN**	**UA**	**FN vs. ILP**	**FN vs. MKNN**	**FN vs. UA**	**ILP vs. MKNN**	**ILP vs. UA**	**MKNN vs. UA**
Single family homes (A)										
Workday	0.72	0.74	0.74	0.76	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Saturday	0.80	0.80	0.78	0.84	<0.001	<0.001	<0.001	0.002	<0.001	<0.001
Holiday	0.87	0.88	0.87	0.90	0.9	0.018	<0.001	0.3	<0.001	<0.001
Residential aggregations (B)										
Workday	0.37	0.38	0.40	0.38	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Saturday	0.40	0.39	0.41	0.41	<0.001	<0.001	0.056	<0.001	<0.001	0.019
Holiday	0.45	0.45	0.47	0.46	0.040	<0.001	<0.001	<0.001	<0.001	<0.001
Large residential aggregations (C)										
Workday	0.22	0.23	0.26	0.23	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
Saturday	0.24	0.23	0.26	0.24	<0.001	<0.001	0.031	<0.001	<0.001	<0.001
Holiday	0.29	0.28	0.32	0.29	0.010	<0.001	0.036	<0.001	<0.001	<0.001
Single enterprises (D)										
Workday	0.41	0.55	0.48	0.47	<0.001	<0.001	<0.001	<0.001	<0.001	0.3
Saturday	0.30	0.34	0.30	0.30	<0.001	0.037	<0.001	<0.001	0.013	0.2
Holiday	0.22	0.29	0.23	0.25	<0.001	0.019	<0.001	<0.001	<0.001	<0.001
Commercial aggregations (E)										
Workday	0.31	0.39	0.35	0.33	<0.001	<0.001	<0.001	0.3	<0.001	<0.001
Saturday	0.21	0.23	0.22	0.21	0.009	<0.001	0.062	>0.9	<0.001	<0.001
Holiday	0.19	0.21	0.20	0.20	0.005	<0.001	0.6	0.7	0.005	0.062
Large commercial aggregations (F)										
Workday	0.14	0.16	0.16	0.15	<0.001	<0.001	<0.001	0.9	<0.001	<0.001
Saturday	0.10	0.12	0.12	0.11	0.019	<0.001	0.046	0.004	0.4	<0.001
Holiday	0.11	0.12	0.12	0.12	0.006	<0.001	>0.9	0.001	0.006	<0.001

Note:

Paired Wilcoxon signed rank test was used to compute p-values.

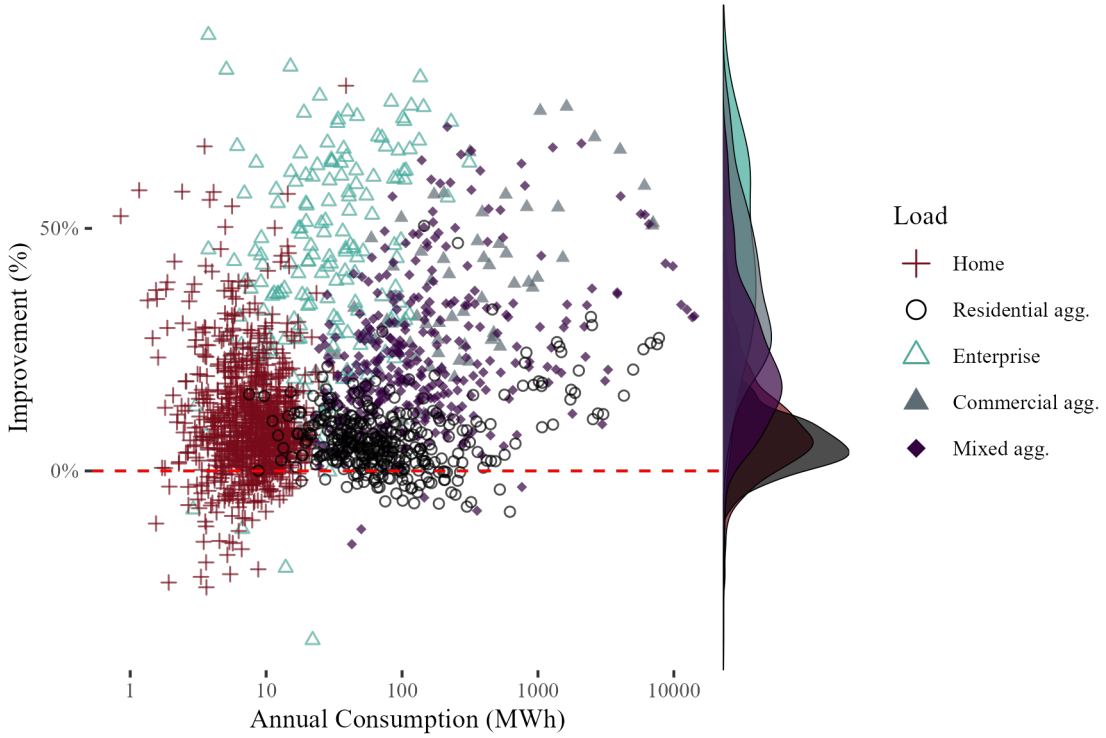

Figure 10.17: Functional neighbor model – improvement relative to the standard load profile forecast. In a wide-scale day-ahead building load forecasting simulation (Section 9.1.1), we applied the functional neighbor forecaster (Algorithm 3) and the SLP-model predicting 1851 loads of different size and type. For each load, we computed the improvement (7.14) by the functional neighbor model relative to the forecast using standard load profiles. The figure shows the improvement (%) for each predicted load denoting load size (annual consumption) and type (colors). The probability density function of the improvement obtained for the corresponding load type is presented on the right of the main plot. Further discussion is provided in the text (Section 10.2).

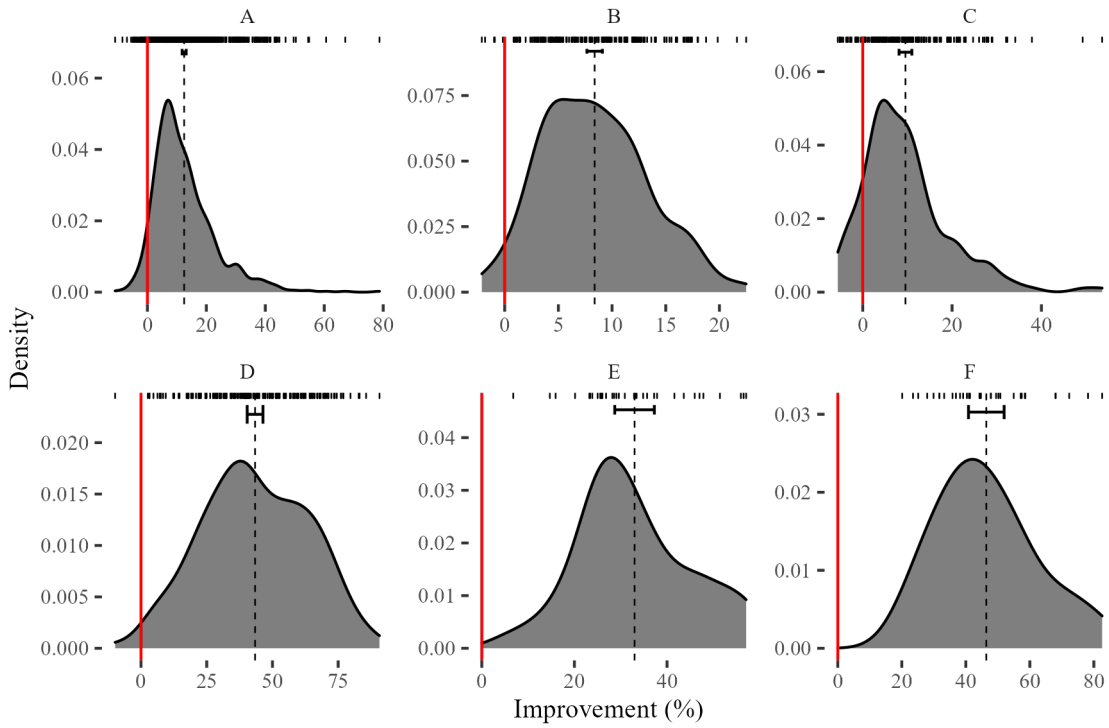

Figure 10.18: Functional neighbor model – improvement relative to the standard load profile forecast by load group. In a wide-scale day-ahead building load forecasting simulation (Section 9.1.1), we applied the functional neighbor forecaster (Algorithm 3) predicting 1851 loads of different size and type obtaining a sample of 153 daily forecast errors for each load. Additionally, we predicted the same loads using the SLP-model and used predictions as a benchmark. Relative to the benchmark, we computed the forecast improvement (7.14) for each predicted daily load curve. In the figure, the panels show the improvement in residential (A-C) and commercial (D-F) load groups (Table 9.3). Every panel shows the sampling distribution of the mean improvement for each load (rugs at the top), expected improvement in the load group (dotted vertical line) with the 95%-confidence interval (horizontal bar) and the zero-improvement line (red vertical line). Further discussion is provided in the text (Section 10.2).

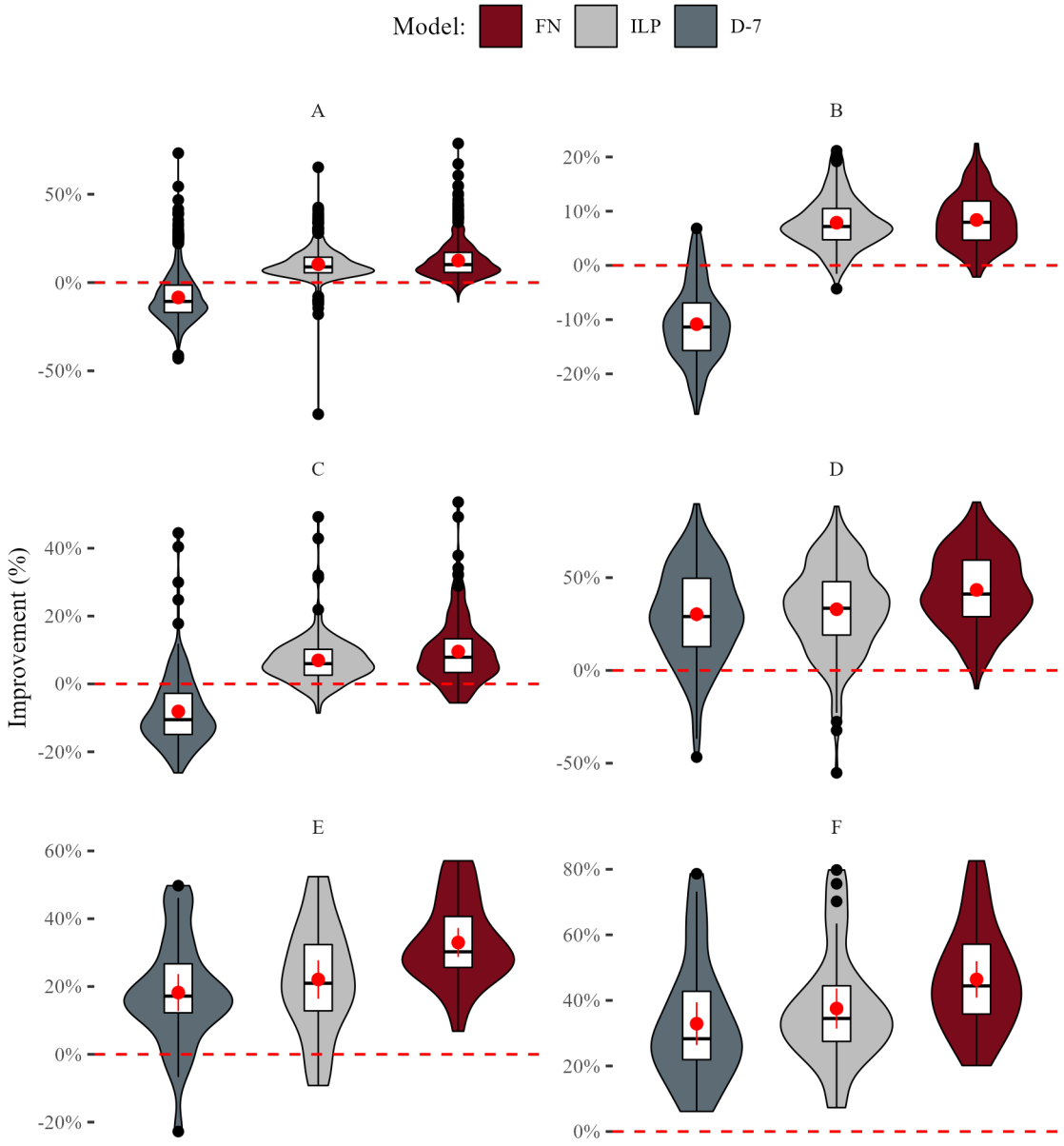

Figure 10.19: Functional neighbor model – comparison to selected heuristic models. In a wide-scale day-ahead building load forecasting simulation (Section 9.1.1), we applied the functional neighbor forecaster (Algorithm 3) and the most accurate heuristic models (Section 10.1.1) predicting 1851 loads of different size and type. Additionally, we predicted the same loads with standard load profiles and used these predictions as a benchmark. Relative to the benchmark, we computed the forecast improvement (7.14) obtained by each model. The figure presents the improvement (%) in residential (A-C) and commercial (D-F) load groups (Table 9.3). Each panel shows the distribution of the improvement in the corresponding load group (box and violin plots) with the zero-improvement mark (red dashed line). Additionally, we denoted the improvement mean (red dot) and its 95%-confidence interval (vertical red bars) in each load group. Further discussion in the text (Section 10.2).

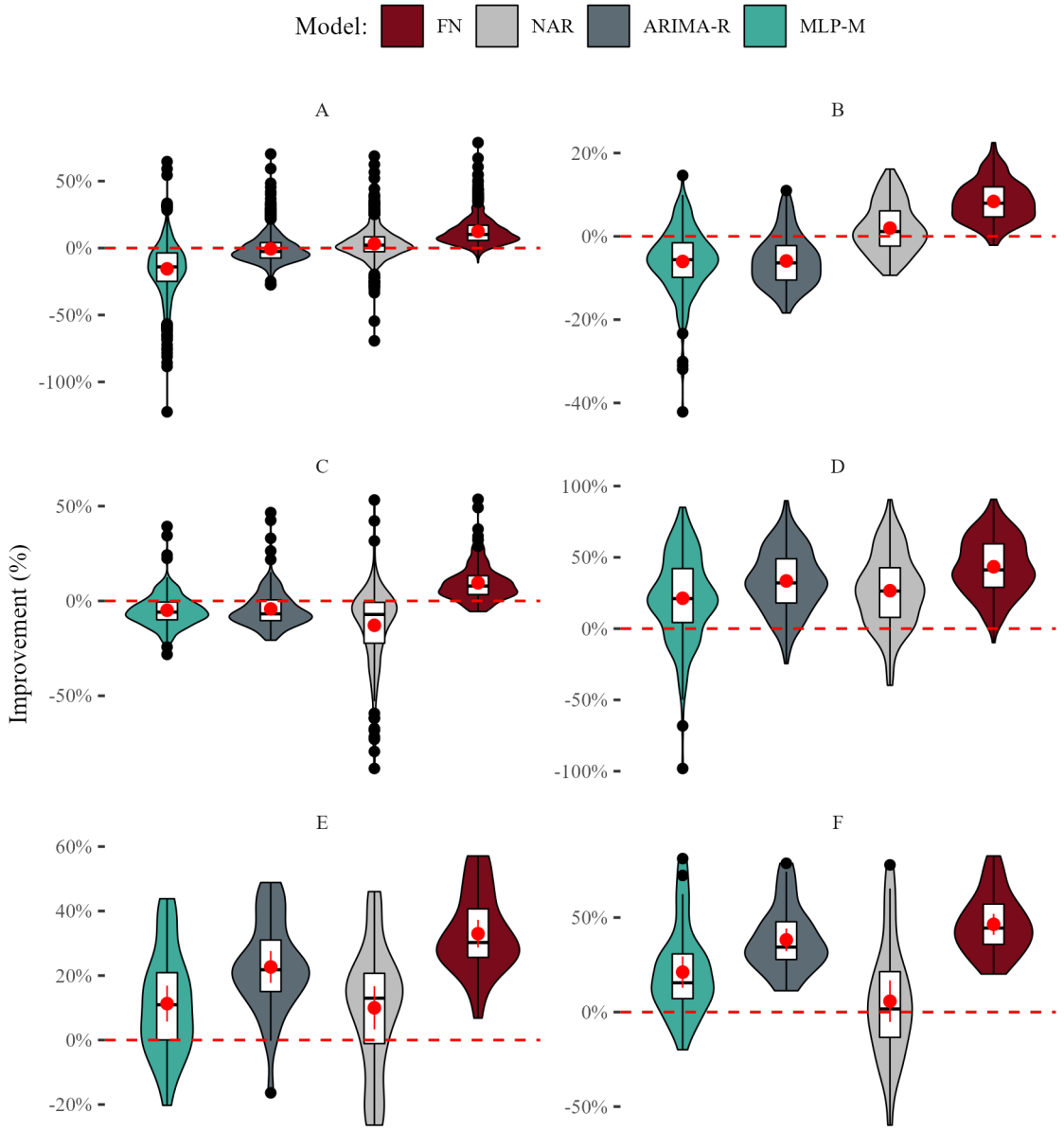

Figure 10.20: Functional neighbor model – comparison to selected parametric models. In a wide-scale day-ahead building load forecasting simulation (Section 9.1.1), we applied the functional neighbor forecaster (Algorithm 3) and the most accurate parametric models (Section 10.1.2) predicting 1851 loads of different size and type. Additionally, we predicted the same loads with standard load profiles and used these predictions as a benchmark. Relative to the benchmark, we computed the forecast improvement (7.14) obtained by each model. The figure presents the improvement (%) in residential (A-C) and commercial (D-F) load groups (Table 9.3). Each panel shows the distribution of the improvement in the corresponding load group (box and violin plots) with the zero-improvement mark (red dashed line). Additionally, we denoted the improvement mean (red dot) and its 95%-confidence interval (vertical red bars) in each load group. Further discussion in the text (Section 10.2).

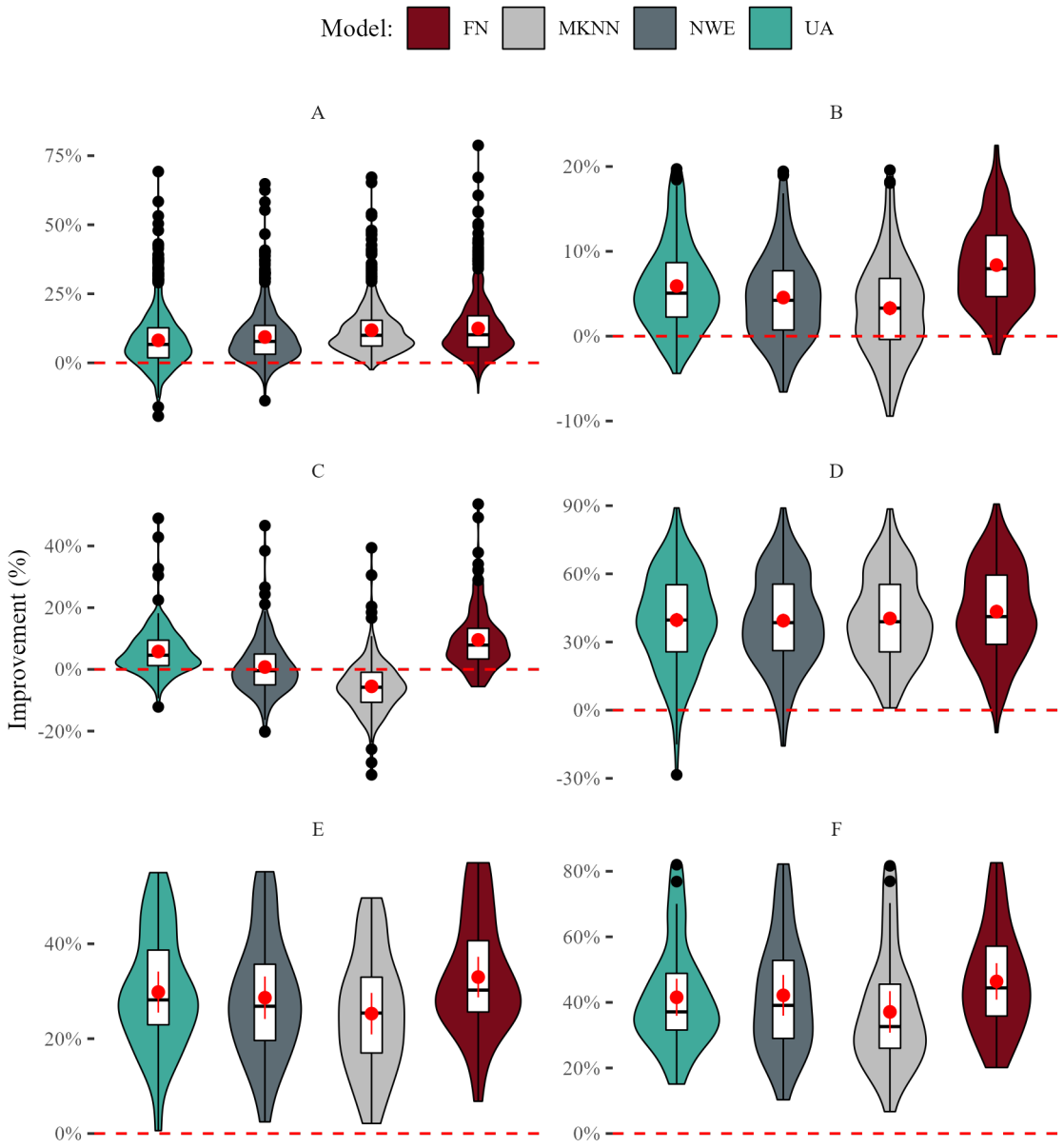

Figure 10.21: Functional neighbor model – comparison to selected nonparametric models. In a wide-scale day-ahead building load forecasting simulation (Section 9.1.1), we applied the functional neighbor forecaster (Algorithm 3) and the nonparametric models (Section 10.1.3) predicting 1851 loads of different size and type. Additionally, we predicted the same loads with standard load profiles and used these predictions as a benchmark. Relative to the benchmark, we computed the forecast improvement (7.14) obtained by each model. The figure presents the improvement (%) in residential (A-C) and commercial (D-F) load groups (Table 9.3). Each panel shows the distribution of the improvement in the corresponding load group (box and violin plots) with the zero-improvement mark (red dashed line). Additionally, we denoted the improvement mean (red dot) and its 95%-confidence interval (vertical red bars) in each load group. Further discussion in the text (Section 10.2).

Forecast	A	B	C	D	E	F
FN	12	8	10	43	33	46
ILP	10	8	7	33	22	38
MKNN	12	3	-6	40	25	37
UA	8	6	6	40	30	42
NWE	9	5	1	39	29	42
NAR	3	2	-13	27	10	6
ARIMA-R	-1	-6	-4	33	23	38
D-7	-8	-11	-8	30	18	33
D-1	-7	-10	-9	15	-6	-11
DNN-3	-11	-6	-9	20	8	14
MLP-M	-16	-6	-5	21	11	21
DNN-2	-15	-8	-8	20	9	17
MLP-D	-17	-1	-5	11	-3	-5
ARIMA-D	-24	-24	-24	18	9	26
NARX	-38	-14	-12	-4	-17	-5

Table 10.10: Comparison of median improvement relative to the SLP-forecast in each load group. The models were evaluated in the wide-scale day-ahead building load forecasting simulation (Section 9.1.1). For each load group defined in Table 9.3, we highlight the models that achieved minimal (green), below average (grey) and maximal forecast improvement (7.14) relative to the SLP-forecast. Further discussion is provided in the text (Section 10.2).

10.3 Functional Neighbor Extension Forecast

In this section, we present the results of the smart-building load forecasting simulation. The simulation was done using the load of a smart building from the Smart-City-Demo Aspern project (Section 9.1.2). The net electricity demand of this building notably depends on the solar irradiation due to a large *photovoltaic (PV)* installation on the roof. For this building, we computed the day-ahead forecast with the FNX-model that considered global solar irradiation as an exogenous variable (Algorithm 4). Together with various reference models (Table 9.6), the forecast was computed day-by-day on a rolling basis (91 consecutive days). The simulation results validate the FNX-model which allowed to consider exogenous variables within functional neighbor forecasting methodology.

We observed that the FNX-model had the smallest forecast error on the majority of the predicted days (Table 10.11). As in the wide-scale day-ahead building load forecasting simulation (Section 10.2), the daily errors were log-normally distributed and could be summarized in terms of a median and an IQR. Notably, all models were less accurate on weekends, where the power demand of a building is often more volatile than on workdays. Moreover, we observed that the multivariate models which considered the solar irradiation (FNX, DNN, ARIMAX) were substantially more accurate than the profiling heuristics (SLP, ILP) which, by design, did not consider any external variables. At the same time, the median errors and IQRs of the multivariate models were comparable which required further statistical testing before reaching any conclusion on their comparison.

Unpaired one-sided independent t-tests showed that all multivariate models significantly ($p < 0.001$) improved the SLP-forecast of our building with a PV-installation. The improvement with all these models was consistent throughout the simulated days and averaged around 50% (Table 10.12). Additionally, paired t-tests confirmed that the FNX-forecaster brought a slightly larger improvement than other models. The difference to the

Table 10.11: Smart building load forecasting simulation – summary of daily forecast errors. In a smart building simulation (Section 9.1.2), we applied the FNX-model (Algorithm 4) and various reference models (Table 9.6) predicting the 91 consecutive daily load curves of a smart building from the Smart-City-Demo Aspern project. The table summarizes the daily forecast errors in terms of the median [interquartile range] conditioning on the weekday. Further discussion is provided in the text (Section 10.3).

	Weekday						
	Mon	**Tue**	**Wed**	**Thu**	**Fri**	**Sat**	**Sun**
Model							
FNX	0.43 [0.39, 0.59]	0.43 [0.32, 0.74]	0.50 [0.40, 0.60]	0.57 [0.40, 0.78]	0.60 [0.54, 0.89]	0.83 [0.75, 0.94]	0.61 [0.53, 0.76]
DNN	0.60 [0.53, 0.68]	0.69 [0.43, 1.14]	0.65 [0.48, 0.95]	0.56 [0.43, 1.01]	0.74 [0.48, 0.97]	0.71 [0.49, 0.91]	0.74 [0.69, 0.98]
ARIMAX	0.55 [0.48, 0.82]	0.58 [0.54, 0.96]	0.78 [0.59, 0.96]	0.61 [0.52, 0.94]	0.86 [0.59, 1.12]	0.79 [0.55, 0.90]	0.94 [0.63, 1.09]
ILP	0.74 [0.66, 0.79]	0.74 [0.63, 0.91]	0.76 [0.60, 0.80]	0.78 [0.61, 0.93]	0.84 [0.78, 0.92]	0.86 [0.74, 1.03]	0.95 [0.90, 1.04]
SLP	1.26 [1.17, 1.40]	1.22 [1.12, 1.31]	1.15 [1.09, 1.26]	1.20 [1.14, 1.31]	1.22 [1.16, 1.34]	1.36 [1.21, 1.48]	1.41 [1.29, 1.47]

Table 10.12: Smart building load forecasting simulation – improvement relative to the standard load profile forecast. In a smart building simulation (Section 9.1.2), we applied the FNX-model (Algorithm 4) and various reference models (Table 9.6) predicting the 91 consecutive daily load curves of a smart building from the Smart-City-Demo Aspern project. For each predicted daily load curve, we computed the improvement (7.14) relative to the SLP-forecast. The table summarizes the improvement (%) in terms of the mean [median] and the results of a paired t-test evaluating the statistical significance of the differences between the models. Further discussion is provided in the text (Section 10.3).

	Model			p-values		
	ARIMAX	**FNX**	**DNN**	**DNN vs. FNX**	**DNN vs. ARIMAX**	**FNX vs. ARIMAX**
Weekday						
Mon	52 [50]	62 [64]	51 [57]	0.050	0.8	0.073
Tue	45 [49]	57 [62]	33 [48]	0.009	0.2	0.005
Wed	26 [28]	50 [54]	37 [41]	0.035	0.2	<0.001
Thu	37 [46]	44 [53]	31 [54]	0.12	0.5	0.4
Fri	27 [29]	40 [51]	35 [37]	0.15	0.3	0.086
Sat	43 [41]	35 [36]	41 [53]	0.5	0.7	0.3
Sun	35 [35]	48 [50]	41 [44]	0.3	0.4	0.045

Note:

Paired t-test was used to compute p-values.

ARIMAX-forecast was more substantial and often significant ($p < 0.05$). It appeared that the FNX-model was more accurate at modeling the nonlinearities of the relationship between solar irradiation and the net consumption of the building. The difference in accuracy between FNX and DNN was smaller, yet also significant on the majority of workdays ($p < 0.05$).

Overall, the results of the smart-building load forecasting simulation validated the functional neighbor extension approach that considers external variables within the proposed functional neighbor forecasting methodology. We observed that our methodology provided forecast comparable, and often more accurate, than complex parametric models (ARIMAX, DNN) which are commonly used to predict the load of smart buildings.

11 Discussion

This dissertation aims to provide an alternative to the standard load profiles for the day-ahead load forecasting of buildings on a wide scale. There are numerous methods for predicting the aggregated consumption at the power system level. At the same time, only few studies exist focusing on predicting local low-voltage loads – the electricity demand of individual buildings connected to the distribution grid. Commonly, the research concentrates on fine-tuning a complex parametric model for a given building. The model is set up manually for a one-step ahead prediction (intraday) and parametrized using explicit knowledge of the building. The forecast is usually evaluated on a single or a small set of loads which impedes any conclusions about a possible wide-scale application of the model. A wide-scale application of a forecaster requires to predict numerous local loads of different size and type without a possibility for any manual adjustment of the model. Importantly, the local loads can be highly volatile and diverse requiring the forecast evaluation on a statistically relevant sample of the end-consumers. In total, a model for predicting the day-ahead building loads on a wide scale requires a fully-automated forecasting method and an extensive evaluation that we provide in this dissertation.

With our study, we establish the wide-scale local load forecasting as a subfield of research within distribution system operation (*Contribution 1*). To this end, we formulated the before-the-meter forecasting problem (Chapter 7) and consolidated the existing knowledge on local load forecasting. In particular, we presented a unified view on data-driven load forecasting in distribution systems combining the perspectives of statistical learning theory, classical and functional time series analysis (Chapter 4). Herewith, we provided a classification of the existing building load forecasting methods and their critical review placing them into the context of a wide-scale application (Chapter 5).

Furthermore, we formulated and applied a methodology to evaluate and compare forecasting models for the wide-scale application, which we demonstrate evaluating the most common existing forecasting techniques (*Contribution 2*). We used an extensive public smart-meter dataset to simulate wide-scale day-ahead building load forecasting using numerous reference models (Chapter 9). The results included hundreds of thousands of daily load forecasts which allowed us to draw several statistically founded conclusions about common data-driven load models (Chapter 10). To the best of our knowledge, this is the first such evaluation in context of a wide-scale day-ahead forecasting on a statistically

relevant sample of local loads – a challenging domain where historical data is limited, manual adjustment is not possible and the loads can be highly volatile and diverse.

The main practical outcome of this dissertation is a load forecaster based on the novel functional neighbor methodology (*Contribution 3*). We had considered the insights from the aforementioned reference model evaluation and developed a method specifically for the wide-scale day-ahead load forecasting in a distribution grid (Chapter 8). The corresponding data-driven model requires no manual setup and can be universally applied to buildings of any size and type. Our forecaster requires three months of historical load measurements and can optionally consider further data inputs available in a smart grid. Applying the aforementioned evaluation methodology, we demonstrated that the functional neighbor model is significantly more accurate than standard load profiles and more sophisticated approaches based on classical time series analysis and machine learning (Chapter 10).

In this chapter, we discuss the results of our study relating them to the research question and contributions that were stated in the introductory part of the thesis. We interpret the findings and show their relevance for the wide-scale day-ahead local load forecasting (Section 11.1). Focusing on implications for designing predictive models for wide-scale applications on buildings, we connect our contributions to the existing scholarly work on load forecasting. Further, we explain how the functional neighbor methodology proposed in this thesis advances the load forecasting capabilities in smart grids (Section 11.2). Completing the discussion, we acknowledge the limitations of our study and highlight various paths for future developments in the field of wide-area local load forecasting (Section 11.3). Overall, this chapter provides the arguments required for concluding the dissertation.

11.1 Wide-Scale Day-Ahead Local Load Forecasting

This study presents a unified view on data-driven distribution system load forecasting. In particular, we combined the perspectives of statistical learning theory and time series analysis on designing predictive models for the load forecasting applications (Chapter 4). Studying the existing literature, we identified three different families of forecasting approaches (Chapter 5) and evaluated various reference models from each family (Chapter 9) in a wide-scale day-ahead building load forecasting simulation (Chapter 10). In this section, we analyze the simulation results that allow us to compare the approaches between each other and discuss implications to designing predictive models for a wide-scale day-ahead local load forecasting.

11.1.1 Evaluation of Forecasts in Context of a Wide-Scale Application

Our results imply that the load forecasting studies where models are compared on a single or a small group of buildings allow only limited and case specific conclusions about the evaluated methods. With our research, we demonstrate the importance of statistical analysis for evaluating and comparing the models for wide-scale local load forecasting. In the conducted day-ahead building load forecasting simulation, daily forecast errors varied substantially depending on the building. For instance, we observed a variation of over 100% on smaller buildings. In these circumstances, single number quantifying the forecast error (e.g., average) can be insufficient to evaluate model accuracy. Instead, we need to rely on further descriptive statistics representing the error distribution.

Contrary to a common preconception, we observed that daily errors are distributed neither normally nor symmetrically which was reflected by the interquartile range (Table 10.2). For the most smaller loads, we observed daily error outliers reflecting the days with unusual consumption. Moreover, we had several smaller loads where the models failed to forecast adequately. The presence of the outliers has to be considered when applying descriptive statistics. In this context, median is a more appropriate statistic than the average for describing daily error distribution.

The accuracy of all models depended on the time-series characteristics that are linked to the building size and type – i.e., accuracy of the forecasting methods varied substantially depending on the load. In particular, our results confirm the empirical scaling law [SR18] that describes how the forecast error decreases with load size. Additionally, we found that the error also substantially depends on building type. The models evaluated in this study were notably more accurate predicting the electricity consumption of commercial buildings rather than residential buildings of the same size. Commercial loads closely follow workday calendar and have more regular usage patterns over the day than residential loads. Our results extend the findings of Sevlian et al., [SR18] who claim that the load size alone determines the scaling of the forecast error.

Moreover, our results explain conflicting conclusions that may be obtained when comparing the models based on the existing literature (Chapter 5). We observed that different models can be expected to have similar accuracy, especially on the loads of an average size with annual consumption of around 100 MWh. Therefore, a load forecasting model for a wide-scale application must be evaluated on a diverse set of loads of different size and type. Any observed difference in model accuracy must be verified on various levels of load aggregation and checked for statistical significance due to the error variation.

In this study, we proposed a methodology (Section 7.3) combining descriptive and inferential statistics to evaluate and compare the forecasting models in a wide-scale application. For such application, we need to estimate the forecast error that we can expect on a building

of a given size and quantify the accuracy across the building domain. This suggests the necessity to consider *expected model error (EME)* (7.17) and its scaling as a part of the model evaluation as we did in this study. The empirical scaling law [SR18] allowed us to compute the EME for the buildings of all sizes. Most importantly, it allowed us to compare the models not only in terms of the expected error on a particular load or a sample of loads, but also estimate the expected accuracy on the loads that are not part of our dataset.

Additionally, we have to use inferential statistics and hypothesis testing to draw any comparisons between the models. Even when we compare two models on a single building, we have to assess if the difference in the forecast errors is statistically significant or is due to the natural variation of daily consumption patterns. Notable error variation can lead to case-based conclusions when comparing the models between each other. For a comprehensive comparison, we have to evaluate the models not only on single loads or groups of loads, but across all sizes and compute the irreducible errors (7.21) for each model that describe until which point the forecast error can be reduced following the load aggregation.

11.1.2 Reference Model Comparison

The results of the wide-scale day-ahead building load forecasting simulation allow to compare various heuristic, parametric and nonparametric reference models that are common in the forecasting literature. We forecast numerous loads of different size and type and evaluated the models in context of a wide-scale application using the aforementioned evaluation methodology.

The forecast using standard load profiles had the largest critical load size of 837 MWh and the irreducible error of 0.08. This is consistent with the praxis, where the error of 0.1 is expected for aggregations exceeding 400 households [Ber00]. At the same time, other heuristics (D-7, ILP) had significantly lower irreducible error (0.07). Consequently, these heuristics can be more accurate, especially on larger loads. In fact, we saw that the ILP and D-7 can be expected to be more accurate than the SLP-forecast on the loads larger than 10 MWh and 100 MWh respectively. Among all heuristic models, ILP had the smallest error in each load group. Hence, we can replace SLP with ILP and expect a notable accuracy increase.

Our results suggest that parametric regression might not be the best approach for the load forecasting in a wide-scale application. Parametric regression is the most common load forecasting methodology and various authors demonstrated the effectiveness of corresponding models when predicting the consumption of a single building or a small group of buildings (Chapter 5). The models presented in the literature often rely on manual fine-tuning and parametrization that is not possible in context of a wide-scale application.

In our wide-scale load forecasting simulation, we observed that only in some cases (mostly commercial loads), a parametric model was more accurate than currently used SLPs. In particular, ARIMA-models were only effective for commercial loads and failed to improve the forecast on residential buildings because of the nonlinearities of the loads. At the same time, none of the evaluated neural network architectures were consistently better than others on loads of all sizes. While NAR performed best on small loads, MLP-M was more accurate on larger loads. On the middle-sized loads, EME of different architectures was similar.

More generally, we observed that heuristic models were often more accurate than sophisticated parametric approaches. Nonstationarity of the building loads imposes a major limitation on common machine learning methods. We repeatedly observed that even DNNs could be outperformed by simple heuristic forecasts. We explain the weak performance of the applied neural networks by the nonstationarity of local loads. Following the ANN-methodology, a trained network forecasts unseen data, assuming that the statistical properties of the process generating the data remain constant. Once the regression function is estimated, a network does not adapt to the change in data characteristics which often arises with local loads. In such case, historical data quickly becomes irrelevant undermining the network training. The fact that the most accurate neural network model used the least amount of training data supports this hypothesis (NAR-model using only 2 months of data). On the other hand, nonparametric regression appears to be more flexible and accurate for day-ahead predictions of building loads.

More than the reference models of any other family, nonparametric models significantly improved the SLP-forecast in the majority of the load groups (Table 10.3). On single family homes, these models can be expected to improve the SLP-forecast by 8%–12%, which is more than the best performing NAR-network could achieve. Yet, on residential loads, the improvement wanes with load size as the SLP-forecast becomes more accurate. For large residential aggregations (C), only UA-model achieved a significant improvement of 6% ($p < 0.001$). The improvement was more substantial on commercial loads. There, nonparametric models improved the SLP-forecast by 25%–42% on average. The EME also shows that nonparametric models have smaller irreducible error than the SLP-forecast and are better at predicting larger loads.

11.1.3 Practical Implications for Load Forecaster Design

The evaluation and comparison of numerous forecasting models provided various practical insights for predicting day-ahead building load curves on a wide scale. First, our results suggest that the models predicting the time series one-step ahead, should not be adopted directly for multistep forecasts. We considered various multistep strategies (Section 7.2.2)

and observed that the direct approach was the least accurate strategy. Predicting each point of the load curve separately, was significantly worse than any other approach. Second, we observed that forecast accuracy did not necessarily improve with model complexity. For instance, modeling capacity of a neural network depends on its size, yet we did not observe any accuracy improvement despite the increase in network size. It seems that the nonstationarity of the predicted time series remains a fundamental limitation for the corresponding forecast accuracy.

The first insight is important because one-step ahead models for intraday predictions are far more common in the forecasting literature. For instance, we showed that a traditional ARIMA-model applied directly (ARIMA-D) was significantly less accurate than when the same approach was applied recursively (ARIMA-R). Similarly, we observed that the ANN using the direct strategy (MLP-D) was significantly less accurate than the same network using the recursive (NARX) or multi-out (MLP-M) strategy. In fact, our results provide substantial evidence that the multi-out strategy is the most effective for adapting one-step ahead models for multistep day-ahead predictions.

The second insight is based on our simulation results which imply that it is often senseless to use complex models to predict local loads. Increased modeling capacity did not increase the forecast accuracy in a wide-scale load forecasting application considered in this study. In particular, we observed that the accuracy of a neural network model with just one hidden layer of neurons was almost indistinguishable from the more complex DNNs with two and more layers. This suggests that a single layer network had enough modeling capacity for the local load forecasting task. The surplus in modeling capacity available in the DNNs was discarded by the Baysian regularization preventing overfitting.

Advanced model architectures like the DNNs are being intensively researched and considered for the load forecasting applications [AMM17,HCR18,KC19,Sch15,SXL18,SLW16]. Such complex parametric models can sometimes accurately forecast a given load, however, our results showed that the nonstationarity of the predicted time series remains a fundamental challenge. For instance, we observed that even larger neural networks (DNNs) did not have the flexibility to consider a concept change in the load (e.g., special days when the building becomes vacated). We suggest that a neural network or any other parametric model should be retrained daily to account for the nonstationarities of the load. Before considering a complex parametric model (e.g. DNN) for the wide-scale building load forecasting application, we need to see compelling evidence that such model is adequate for predicting local loads without manual setup.

At the same time, we observed that the simplest ANN-architecture (NAR) yielded the most accurate forecast of small loads among neural networks. It modeled inherent load seasonalities (Section 8.1.1) implicitly which simplified the architecture. Modeling the

dependency on the calendar explicitly (e.g., MLP-M, NARX) only provided accuracy improvement on larger loads where such dependency is more pronounced.

Alternatively, our results motivate the usage of nonparametric regression approach for the day-ahead local load forecasting. Nonparametric models were notably more accurate than any parametric forecasters on the majority of loads of all sizes. Furthermore, they significantly improved the SLP-forecast on residential loads where parametric models failed to do so. Hypothesizing, nonparametric methodology appears to be more robust against the nonstationarity of the local loads. While the most research concentrates on parametric regression, our study challenges this approach in the context of wide-scale load forecasting in the distribution systems.

11.2 Functional Neighbor Forecasting Methodology

We considered the insights discussed above while developing a novel methodology for wide-scale day-ahead building load forecasting. Our method is based on functional nonparametric regression and addresses the main issues of a nonparametric modeling: curse of dimensionality and the difficulty to consider exogenous variables. Simulation results demonstrated that the proposed functional neighbor forecaster is more accurate than any of the evaluated reference models and can substantially improve the forecasting capabilities in distribution systems for predicting the load of single buildings and aggregations.

The functional neighbor forecaster that we propose in this dissertation (Chapter 8), was designed specifically for a wide-scale application and requires no manual setup. Modularity and flexibility of the corresponding algorithm allows the forecaster to be applied to various loads of different type and size. The load seasonalities are modeled implicitly which allows to reduce the amount of required historical data. As a result, our forecaster was the most accurate in each load group and had the smallest irreducible error comparing to the 15 different reference models from the literature. Statistical analysis of the results allows us to expect the functional neighbor forecaster to be more accurate than any of the common models on buildings loads of any size.

Most importantly, our simulation results (Chapter 10.2) suggest that the proposed forecaster is a better alternative to the currently used SLP-method wherever smart-meter data is available. On almost every load, the functional neighbor method significantly improved the forecast comparing to the SLP-prediction by up to 100%. Notably, the improvement on commercial loads was three to four times higher than on residential loads. Hence, our method can drastically improve the SLP-forecast without using any knowledge about the business (e.g., opening hours). Moreover, our forecaster had irreducible error that is 39% lower than that of the SLP-method which was designed to deliver accurate predictions of

aggregated loads. Consequently, the functional neighbor model is not only more accurate on the vast majority of single loads, but also on aggregations of any size. Therefore, if we replace currently used SLPs with the functional neighbor model, we can expect significantly more accurate forecasts in the distribution systems.

Our forecasting method can have various practical applications in smart grids. Following the mass adoption of smart-meters, functional neighbor methodology can replace the traditionally used SLPs and notably improve the forecasting capabilities of the distribution system operators. Moreover, the proposed forecasting methodology allows further improvements such as incorporating external inputs. We proposed an extension (FNX) that can do so and demonstrated that it was effective in considering exogenous variables when predicting the load of a smart building (Section 10.3). The FNX-model was tested on a building where the daily load curve depends substantially on solar irradiation and was (though insignificantly) more accurate than the more common DNN and ARIMAX models. Considering exogenous variables, such as weather or scheduled control-signals for the next day, enables to predict the day-ahead load of a modern building equipped with photovoltaic panels, electrical heating or air-conditioning systems, and flexible loads, facilitating wide-scale demand response applications.

Improved forecast accuracy can have substantial practical advantages. When applied to numerous buildings, a forecast improvement of a single percent can lead to sizable cost savings [SSM16]. In contrast, an inapt forecast can result in severe problems for the distribution system. With an increasing share of distributed generation and storage, a congestion becomes more probable while a substantial prediction error imperils the countermeasures (e.g., predictive control). This makes accurate short-term forecasts fundamental for the operation of smart buildings and grids. At the same time, smart grids allow new business models for the European electricity market [NA16]. Those are often based on pooling together smaller consumers and producers to virtual power plants and selling the aggregated load flexibility. The operators of such entities often rely on the day-ahead predictions of the consumers in the pool to optimize the cost of supply, anticipate energy purchases, and estimate the amount of flexibility which they can monetize [NHG17, YFLL19]. In total, improved accuracy and flexibility of the functional neighbor forecaster facilitates various smart-grid applications which can increase the efficiency of the existing distribution system infrastructure and aid accommodating decentralized renewable energy generators.

11.3 Limitations and Future Research

Finalizing the discussion, we acknowledge the limitations of our study and highlight the paths for further development of the building load forecasting field. Our future research

will focus on overcoming the limitations of the current study and improving functional neighbor methodology facilitating its practical applications in smart grids.

For ease of exposition, we made several methodological choices when simulating the wide-scale day-ahead building load forecasting using a public smart-meter dataset. As a result, the simulation deviated from a practical application in several ways. First, the load curve for the upcoming day was predicted at midnight assuming that the measurements of the most recent day are fully available. In practice, a day-ahead forecast might be required earlier[1]. In such case, we can use recursion or other multistep strategy (Section 7.2.2) to extend the forecaster horizon depending on the application. Second, we predicted the load curves with hourly resolution to reduce the amount of data processed by the simulator. However, modern smart meters can provide a higher load measurement resolution given that power system operators have the adequate computational resources to process the data. At last, we used a public smart-meter dataset that is not fully representative for all building loads. All smart meters were from the same region (Greater Dublin Area) and included only single family homes, small businesses as well as middle-sized enterprises. However, we can expect further building types such as industrial and institutional facilities to feature similar workday calendar dependency and recurrent weekly patterns present in commercial loads included in the chosen dataset. The absence of a representative smart-meter dataset that is publicly available remains an important obstacle for further research on a wide-scale local load forecasting.

Despite those limitations, our simulation provided an empirical evidence that the parametric regression might be an ill-chosen methodology for wide-scale load forecasting (Section 10.1). However, our study does not allow a more definitive conclusion. Such conclusion would require to prove that there is a fundamental limitation of a parametric approach for predicting low-voltage loads or, at least, to evaluate all the numerous parametric models proposed for building load forecasting. Instead, we evaluated only the most common methods (ANN, ARIMA) observing that those were notably less accurate than simple heuristic models for predicting local loads. Our hypothesis about load nonstationarity being such a limitation remains to be formally proven in the future.

Considering the functional neighbor methodology, the proposed extension allowing to consider external inputs (Section 8.3) is yet to be evaluated in a wide-scale smart-building load forecasting simulation. As it is common in the building load forecasting literature, we validated the proposed extension against various reference models on a smart building from the Smart-City-Demo Aspern project. In our smart-building load forecasting simulation (Section 9.1.2), the dataset consisted of a single load and was very small comparing to the wide-scale forecasting simulation with which we validated the basic functional neighbor

[1] For instance, at 12:00 in the European electricity market balancing.

methodology. Moreover, the inputs affecting the net consumption were identified manually knowing the energy equipment installed in the building. The manual input selection does not impede a practical application of the FNX-model, since larger energy equipment has to be registered by the grid operator. Nevertheless, at the moment there is no publicly available dataset which contains smart-meter data of numerous smart buildings together with the measurements of relevant exogenous variables that could be used for a wide-scale load forecasting simulation of smart buildings.

Authors proposing a novel method and evaluating it against the existing techniques are subject to confirmation bias. In the building load forecasting literature, it is often unclear if the same relative comparison of the forecasting models would hold if the researchers would had put same effort to setting up and fine-tuning the reference models as they did for the model they are proposing. To minimize the confirmation bias, we evaluated our model on numerous loads of different type and size, without any manual setup. Moreover, we used a public smart-meter dataset to facilitate the replication of the wide-scale building load forecasting simulation results. Most importantly, we relied on the extensive validation of the reference models (Section 9.2) to assure that those are set up to the best of our abilities.

The modularity of the functional neighbor methodology facilitates future improvements. In particular, we will investigate further distance notions (Section 8.2.2) for assessing the similarity of load curves and develop corresponding mergers (Section 8.2.3). Moreover, further research is needed to improve model selector in a nonstationary environment that can emphasize relevant training data and discard special days when selecting the bandwidth and other global parameters of the functional neighbor model. Additionally, we will develop an automated feature selection module that will allow the FNX-model to select the most relevant inputs for a given smart building in cases where installed energy equipment is unknown.

Further, we will study the relation between load measurement resolution, forecast horizon and model accuracy. In particular, we will extend the functional neighbor methodology to enable forecasts with different horizons. This will allow smart-grid operators to adjust the horizon for their application and facilitate intraday forecasts using our methodology if so required. We will demonstrate the usage of the functional neighbor forecaster in a practical smart-grid application and estimate the economic potential due to the accuracy improvement comparing to the currently used standard load profiles.

Regarding the wide-scale local load forecasting in general, researchers can consider alternative forecasting methodologies. For instance, traditional parametric models can be adapted for nonstationary environments as it was demonstrated in other fields [KM18]. Alternatively, ensemble models can be a promising approach for predicting diverse loads across the building domain. At the same time, a probabilistic forecast might be more adequate for the loads of single households where the largest prediction errors occur due

to the increased volatility and nonstationarity of the electricity consumption. Probabilistic forecasting methods providing a prediction band with corresponding confidence intervals started to appear in the literature [AT14] and might be a promising extension of the functional nonparametric regression approach.

Model evaluation is an important issue for the field of wide-scale local load forecasting maturing as a field. Any model for a wide-scale local load forecasting application must be validated on numerous loads of different type, size and geographic location. Currently, there is a no public smart-meter dataset which can become a standard for model evaluation. As it is done in other fields[2], such dataset can be used for benchmarking facilitating model comparison. At the moment, building load forecasting models are often evaluated on private datasets which impedes the replication and comparison to other models. We are convinced, that creating a public smart-meter dataset that contains a representative group of buildings of different size, type, energy equipment and geographic location will advance wide-scale local load forecasting as a field.

[2] For instance, ImageNet dataset [DDS+09] greatly advanced the research on image recognition and machine learning.

12 Conclusion

This dissertation investigates the usage of smart-meter data for predicting the day-ahead electricity consumption of buildings and their aggregations on a wide scale. We propose a novel functional neighbor forecasting method that allows power system operators to predict the day-ahead load curves of any individual low-voltage end-consumer equipped with a smart meter and an aggregation thereof, without manual setup of the forecaster. More generally, we establish the wide-scale day-ahead local load forecasting as an area of research in power system operation. In particular, we consolidated the fragmented knowledge providing a unified view on the subfield and evaluated the existing load forecasting models in context of a wide-scale application on local loads. After conducting extensive simulations and statistical analysis of numerous forecasts, we conclude that the functional neighbor forecaster, proposed in this dissertation, can be expected to be significantly more accurate than the existing load forecasting methods that are commonly found in the literature. In praxis, our forecaster can replace the currently used standard load profiles for predicting low-voltage loads and notably increase the forecasting capabilities in the distribution systems.

Load forecasting is a traditional subfield of power engineering. There exist various methods that can predict the day-ahead consumption very accurately at the high voltage level. At the low-voltage level, distribution system operators estimate that buildings consume the electricity according to predefined and standard load profiles. These profiles give a reasonable approximation for larger aggregations of low-voltage loads, while detailed load measurements of single end-consumers are not required. The ongoing wide-area installation of smart meters enables the usage of data-driven models for load forecasting at the level of individual buildings. Various research projects have demonstrated that accurate day-ahead building load forecasts can improve the distribution system operation while providing a foundation for various smart-grid applications.

The need for novel building load forecasting methods is reflected in the intensifying research efforts. At present, numerous propositions focus on predicting the electricity consumption of a single building rather than developing a model that can replace standard load profiles in a wide-scale application. The existing works approach building load forecasting either from the time series analysis perspective, creating a model of the underlying stochastic process, or from the machine learning perspective focusing on approximating the regression function using historical load measurements. Both perspectives allow to

create models that, given manual setup and fine-tuning, have been shown to improve the accuracy of a building load forecast comparing to the standard load profiles. However, there is no forecasting method that can predict building power demand on a wide scale – a method that can be applied to numerous individual buildings of different size and type without any explicit knowledge of the building or a possibility for manual setup of the forecaster.

This study establishes wide-scale day-ahead local load forecasting as a research subfield of smart grid operation and consolidates the relevant knowledge on data-driven load forecasting. Focusing on a wide-scale application on low-voltage loads and aggregations, we formulated the forecasting problem and provided a methodology for evaluating the models in this context. In particular, we formulated the prediction task as a problem of computing the multistep forecast of the load curve before-the-meter (Section 7.2.3). Assuming area-wide installation of smart meters, this problem can be approached with data-driven machine learning models common in other areas of application. At the same time, the time-series nature of the load-measurement data must be considered, which makes the usage of conventional regression models more challenging. Moreover, model evaluation must be based on statistical analysis of the forecast due to notable stochastic variation among the loads. To this end, we introduced a methodology that combines various descriptive and inferential statistics to evaluate a forecaster in a wide-scale application context.

With this methodology, we evaluated the most common models from the load forecasting literature. Summarizing the findings, our study provides an empirical evidence that heuristic and nonparametric approaches are better suited for a wide-scale day-ahead local load forecasting than parametric regression techniques which are far more common. In particular, we demonstrated that simple heuristic and nonparametric models are more accurate for predicting local loads than sophisticated parametric models (e.g., ARIMA, ANN). Nonstationarity of the load imposes a major limitation on parametric models as those require extensive training data that is subject to concept change which is often present in local load time-series. At the same time, nonparametric models, requiring less historical data, delivered significantly more accurate forecasts on the vast majority of loads.

Combining nonparametric and functional regression approaches, we developed a forecasting algorithm specifically for a wide-scale application, that does not require any manual setup and can be used on local loads such (i.e., buildings and their aggregations) of different type and size. Moreover, our forecaster can consider exogenous variables (e.g., weather, control signals) that affect the demand of individual buildings which are increasingly equipped with photovoltaic panels and load flexibilities participating in demand response. We validated our method by simulating a wide-scale day-ahead building load forecasting using an extensive public smart-meter dataset. Statistical analysis of the results indicates

that a distribution system operator can expect our forecaster to be notably more accurate than other common methods existing to this day, on local loads of any size. In particular, we demonstrated that the functional neighbor forecast is often twice as accurate as the standard load profiles when predicting the load of individual buildings. Even for the largest loads, our method can be expected to be at least 39% more accurate than standard load profiles that were designed to predict larger aggregations of end-consumers. Therefore, replacing standard load profiles by our method, we can expect a significant improvement of the forecasting capabilities in distribution systems.

Our future research will aim at further accuracy improvement on smaller loads and practical application of the functional neighbor forecaster in smart grids. We will investigate the effect of smart-meter resolution and extend our methodology to facilitate variable forecast horizon. Herewith, smart-grid operators will be able to use our forecaster for their particular application. Moreover, we foresee an accuracy improvement from using a more advanced distance notion to quantify the similarity of load curves and a model selector suited to work with nonstationary data for setting model parameters. Further, we will develop an automated input selection procedure allowing the FNX-model to determine the most relevant inputs from the available data and validate this model in a wide-scale smart-building load forecasting simulation. More generally, the wide-scale local load forecasting field can be advanced by creating the standard smart-meter dataset including a representative and diverse set of building loads.

Given a mass adoption of smart meters, the functional neighbor methodology presented in this dissertation can replace the currently used standard load profiles for predicting day-ahead distribution system loads and notably increase forecasting capabilities of the power system operators. Improved accuracy and flexibility of our method can provide a foundation for various smart-grid applications relying on day-ahead forecasts. Such applications will increase the efficiency of the distribution system infrastructure and aid accommodating decentralized renewable energy generators in the distribution grids.

Part V

Appendix

List of Figures

List of Tables

Bibliography

[AAD+17] Fatima Amara, Kodjo Agbossou, Yves Dubé, Sousso Kelouwani, Alben Cardenas, and Jonathan Bouchard. Household electricity demand forecasting using adaptive conditional density estimation. *Energy and Buildings*, 156:271–280, December 2017.

[AAH+20] Abdullah Hamed Al-Badi, Razzaqul Ahshan, Nasser Hosseinzadeh, Reza Ghorbani, and Eklas Hossain. Survey of Smart Grid Concepts and Technological Demonstrations Worldwide Emphasizing on the Oman Perspective. *Applied System Innovation*, 3(1):5, January 2020.

[ABDM76] R. S. Anderssen, R. P. Brent, D. J. Daley, and P. A. P. Moran. Concerning $\int_0^1 \cdots \int_0^1 (X_1^2 + \cdots + X_k^2)^{1/2} dx_1 \cdots dx_k$ and a Taylor Series Method. *SIAM Journal on Applied Mathematics*, 30(1):22–30, January 1976.

[AC10] Sylvain Arlot and Alain Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4(0):40–79, 2010.

[AC13] Fahad H. Al-Qahtani and Sven F. Crone. Multivariate k-nearest neighbour regression for time series data - A novel algorithm for forecasting UK electricity demand. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, August 2013.

[ACGW18] Tanveer Ahmad, Huanxin Chen, Yabin Guo, and Jiangyu Wang. A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand: A review. *Energy and Buildings*, 165:301–320, April 2018.

[ACV11] Germán Aneiros-Pérez, Ricardo Cao, and Juan M. Vilar-Fernández. Functional methods for time series prediction: A nonparametric approach. *Journal of Forecasting*, 30(4):377–392, July 2011.

[ADDPAL20] Joud Al Dakheel, Claudio Del Pero, Niccolò Aste, and Fabrizio Leonforte. Smart buildings features and key performance indicators: A review. *Sustainable Cities and Society*, 61:102328, October 2020.

[AFKV08] Ahmed Ait-Saïdi, Frédéric Ferraty, Rabah Kassa, and Philippe Vieu. Cross-validated estimations in the single-functional index model. *Statistics*, 42(6):475–494, December 2008.

[AHA+14] A.S. Ahmad, M.Y. Hassan, M.P. Abdullah, H.A. Rahman, F. Hussin, H. Abdullah, and R. Saidur. A review on applications of ANN and SVM for building electrical energy consumption forecasting. *Renewable and Sustainable Energy Reviews*, 33:102–109, May 2014.

[AHU20] Sk Abdul Aleem, S. M. Suhail Hussain, and Taha Selim Ustun. A Review of Strategies to Increase PV Penetration Level in Smart Grids. *Energies*, 13(3):636, February 2020.

[AIJH21] Abdul Azeem, Idris Ismail, Syed Muslim Jameel, and V. R. Harindran. Electrical Load Forecasting Models for Different Generation Modalities: A Review. *IEEE Access*, 9:142239–142263, 2021.

[AKS14] Omid Ardakanian, Negar Koochakzadeh, and Rayman Preet Singh. Computing Electricity Consumption Profiles from Household Smart Meter Data. *EDBT/ICDT Workshops*, 14:140–147, 2014.

[AKZ10] Nima Amjady, Farshid Keynia, and Hamidreza Zareipour. Short-Term Load Forecast of Microgrids by a New Bilevel Prediction Strategy. *IEEE Transactions on Smart Grid*, 1(3):286–294, December 2010.

[AMM17] Kasun Amarasinghe, Daniel L. Marino, and Milos Manic. Deep neural networks for energy load forecasting. In *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pages 1483–1488. IEEE, June 2017.

[AMS97] Christopher G. Atkeson, Andrew W. Moore, and Stefan Schaal. Locally Weighted Learning for Control. In David W. Aha, editor, *Lazy Learning*, pages 75–113. Springer Netherlands, Dordrecht, 1997.

[AN02] Hesham K. Alfares and Mohammad Nazeeruddin. Electric load forecasting: Literature survey and classification of methods. *International Journal of Systems Science*, 33(1):23–34, January 2002.

[ANHS13] Khalid Alkhatib, Hassan Najadat, Ismail Hmeidi, and Mohammed K Ali Shatnawi. Stock Price Prediction Using K-Nearest Neighbor (kNN) Algorithm. *International Journal of Business, Humanities and Technology*, 3(3):32–44, 2013.

[APS06] Anestis Antoniadis, Efstathios Paparoditis, and Theofanis Sapatinas. A functional wavelet-kernel approach for time series prediction. *Journal*

of the Royal Statistical Society: Series B (Statistical Methodology), 68(5):837–857, November 2006.

[APS09] Anestis Antoniadis, Efstathios Paparoditis, and Theofanis Sapatinas. Bandwidth selection for functional time series prediction. *Statistics & Probability Letters*, 79(6):733–740, March 2009.

[Arc16] Irish Social Science Data Archive. Irish Commision for Energy Regulation. http://www.ucd.ie/issda/data/commissionforenergyregulationcer/, September 2016.

[Arm85] Jon Scott Armstrong. *Long-Range Forecasting*. Wiley New York ETC., 1985.

[Arm01] Jon Scott Armstrong. *Principles of Forecasting: A Handbook for Researchers and Practitioners*, volume 30. Springer Science & Business Media, 2001.

[Asp] Aspern Smart City Research. https://www.ascr.at/en/.

[AT14] Siddharth Arora and James W. Taylor. Forecasting electricity smart meter data using conditional kernel density estimation. *Omega*, December 2014.

[AV06] Germán Aneiros-Pérez and Philippe Vieu. Semi-functional partial linear regression. *Statistics & Probability Letters*, 76(11):1102–1110, June 2006.

[AV08] Germán Aneiros-Pérez and Philippe Vieu. Nonparametric time series prediction: A semi-functional partial linear modeling. *Journal of Multivariate Analysis*, 99(5):834–857, May 2008.

[AVCMSR13] German Aneiros, Juan M. Vilar, Ricardo Cao, and Antonio Munoz San Roque. Functional Prediction for the Residual Demand in Electricity Spot Markets. *IEEE Transactions on Power Systems*, 28(4):4201–4208, November 2013.

[AVR16] Germán Aneiros, Juan Vilar, and Paula Raña. Short-term forecast of daily curves of electricity demand and price. *International Journal of Electrical Power & Energy Systems*, 80:96–108, September 2016.

[BA97] Adrian W Bowman and Adelchi Azzalini. *Applied Smoothing Techniques for Data Analysis: The Kernel Approach With S-Plus Illustrations (Oxford Science Publications)*. Oxford University Press, 1997.

[BAB12] A. Badri, Z. Ameli, and A.Motie Birjandi. Application of Artificial Neural Networks and Fuzzy logic Methods for Short Term Load Forecasting. *Energy Procedia*, 14:1883–1888, 2012.

[BBB†12] Matthew Brown, Chris Barrington-Leigh, and Zosia Brown †. Kernel regression for real-time building energy analysis. *Journal of Building Performance Simulation*, 5(4):263–276, July 2012.

[BBK+14] Richard Berk, Justin Bleich, Adam Kapelner, Jaime Henderson, Geoffrey Barnes, and Ellen Kurtz. Using regression kernels to forecast a failure to appear in court. *arXiv preprint arXiv:1409.1798*, 2014.

[BBTLB13] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. Machine Learning Strategies for Time Series Forecasting. In Wil van der Aalst, John Mylopoulos, Michael Rosemann, Michael J. Shaw, Clemens Szyperski, Marie-Aude Aufaure, and Esteban Zimányi, editors, *Business Intelligence*, volume 138, pages 62–77. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[BCO18] Ralf Becker, Adam Clements, and Robert O'Neill. A Multivariate Kernel Approach to Forecasting the Variance Covariance of Stock Market Returns. *Econometrics*, 6(1):7, February 2018.

[BD91] Peter J. Brockwell and Richard A. Davis. *Time Series: Theory and Methods*. Springer Series in Statistics. Springer New York, New York, NY, 1991.

[BD10] Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer Texts in Statistics. Springer, New York, NY, 2. ed., corr. at 8. printing edition, 2010.

[Ber00] VDEW Bericht. Lastprofilverfahren zur Beliferung und Abrechnung von Kleinkunden in Deutschland. 2000.

[BFS+15] A. Bagnasco, F. Fresi, M. Saviozzi, F. Silvestro, and A. Vinci. Electrical consumption forecasting in hospital facilities: An application case. *Energy and Buildings*, 103:261–270, September 2015.

[BHK18] Christoph Bergmeir, Rob J. Hyndman, and Bonsoo Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120:70–83, April 2018.

[Bis94] Chris M Bishop. Neural networks and their applications. *Neural networks*, 65(6):30, 1994.

[BJRL15] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.

[BM15a] Eric M Burger and Scott J Moura. Building Electricity Load Forecasting via Stacking Ensemble Learning Method with Moving Horizon Optimization. *UC Berkeley: Energy, Controls, and Applications Lab*, 2015.

[BM15b] Eric M. Burger and Scott J. Moura. Gated ensemble learning method for demand-side electricity load forecasting. *Energy and Buildings*, 109:23–34, December 2015.

[BPF11] C. E. Borges, Y. K. Penya, and I. Fernandez. Optimal combined short-term building load forecasting. In *2011 IEEE PES Innovative Smart Grid Technologies*, pages 1–7. IEEE, November 2011.

[BPT13] Dustin Baranek, Alexander Probst, and Stefan Tenbohlen. Optimierung der Lastprognose mittels Smart Meter Daten. *IEEE PES Bielefeld*, pages 23–34, 2013.

[BRF16] M.A. Rafe Biswas, Melvin D. Robinson, and Nelson Fumo. Prediction of residential building energy consumption: A neural network approach. *Energy*, 117:84–92, December 2016.

[BRXK02] Steven M. Boker, Jennifer L. Rotondo, Minquan Xu, and Kadijah King. Windowed cross-correlation and peak picking for the analysis of variability in the association between behavioral time series. *Psychological Methods*, 7(3):338–355, 2002.

[BSRP08] C. V. K. Bhanu, G. Sudheer, C. Radhakrishna, and V. Phanikanth. Day-ahead Electricity Price forecasting using Wavelets and Weighted Nearest Neighborhood. In *2008 Joint International Conference on Power System Technology and IEEE Power India Conference*, pages 1–4. IEEE, October 2008.

[BTBAS12] Souhaib Ben Taieb, Gianluca Bontempi, Amir F. Atiya, and Antti Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications*, 39(8):7067–7083, June 2012.

[BTD+18] Mehrad Bastani, Aristotelis E. Thanos, Haluk Damgacioglu, Nurcin Celik, and Chun-Hung Chen. An evolutionary simulation optimization framework for interruptible load management in the smart grid. *Sustainable Cities and Society*, 41:802–809, August 2018.

[Bun82] Derek W Bunn. Short-Term Forecasting: A Review of Procedures in the Electricity Supply Industry. *Journal of the Operational Research Society*, 33(6):533–545, 1982.

[Bur17] Eric Burger. *Building Energy Modeling and Control Methods for Optimization and Renewables Integration.* PhD thesis, UC Berkeley, 2017.

[BZN+19] Mathieu Bourdeau, Xiao Qiang Zhai, Elyes Nefzaoui, Xiaofeng Guo, and Patrice Chatellier. Modeling and forecasting building energy consumption: A review of data-driven techniques. *Sustainable Cities and Society*, 48:101533, July 2019.

[CBWS16] Moulay Larbi Chalal, Medjdoub Benachir, Michael White, and Raid Shrahily. Energy planning and forecasting approaches for supporting physical improvement strategies in the building sector: A review. *Renewable and Sustainable Energy Reviews*, 64:761–776, October 2016.

[CCL04] B.-J. Chen, M.-W. Chang, and C.-J. Lin. Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001. *IEEE Transactions on Power Systems*, 19(4):1821–1830, November 2004.

[CCVO98] W. Charytoniuk, M.S. Chen, and P. Van Olinda. Nonparametric regression based short-term load forecasting. *IEEE Transactions on Power Systems*, 13(3):725–730, Aug./1998.

[CD14] T. Chai and R. R. Draxler. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3):1247–1250, June 2014.

[CGS13a] Nathaniel Charlton, Danica Vukadinovic Greetham, and Colin Singleton. Graph-based algorithms for comparison and prediction of household-level energy use profiles. In *2013 IEEE International Workshop on Inteligent Energy Systems (IWIES)*, pages 119–124, Vienna, Austria, November 2013. IEEE.

[CGS13b] Nathaniel Charlton, Danica Vukadinovic Greetham, and Colin Singleton. On minimum cost local permutation problems and their application to smart meter data. *University of Reading, Tech. Rep. Mathematics Report Series*, 2:2013, 2013.

[Cha14] Mohamed Chaouch. Clustering-Based Improvement of Nonparametric Functional Time Series Forecasting: Application to Intra-Day Household-Level Load Curves. *IEEE Transactions on Smart Grid*, 5(1):411–419, January 2014.

[CHC95] M.Y. Cho, J.C. Hwang, and C.S. Chen. Customer short term load forecasting by using ARIMA transfer function model. In *Proceedings 1995 International Conference on Energy Management and Power Delivery EMPD'95*, volume 1, pages 317–322. IEEE, 1995.

[CHL16] A.E. Clements, A.S. Hurn, and Z. Li. Forecasting day-ahead electricity load using a multiple equation time series approach. *European Journal of Operational Research*, 251(2):522–530, June 2016.

[CO17] Kristen S. Cetin and Zheng O'Neill. Smart Meters and Smart Devices in Buildings: A Review of Recent Progress and Influence on Electricity Use and Peak Demand. *Current Sustainable/Renewable Energy Reports*, 4(1):1–7, March 2017.

[CPB09] Yacine Chakhchoukh, Patrick Panciatici, and Pascal Bondon. Robust estimation of SARIMA models: Application to short-term load forecasting. In *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, pages 77–80, Cardiff, United Kingdom, August 2009. IEEE.

[CPR19] Mengmeng Cai, Manisa Pipattanasomporn, and Saifur Rahman. Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. *Applied Energy*, 236:1078–1088, February 2019.

[ČS20] Pavel Čížek and Serhan Sadıkoğlu. Robust nonparametric regression: A review. *WIREs Computational Statistics*, 12(3), May 2020.

[CSB+15] Changqing Cheng, Akkarapol Sa-Ngasoongsong, Omer Beyca, Trung Le, Hui Yang, Zhenyu (James) Kong, and Satish T.S. Bukkapatnam. Time series forecasting for nonlinear and non-stationary processes: A review and comparative study. *IIE Transactions*, 47(10):1053–1071, October 2015.

[CSVM16] C.F. Calvillo, A. Sánchez-Miralles, J. Villar, and F. Martín. Optimal planning and operation of aggregated distributed energy resources with market participation. *Applied Energy*, 182:340–357, November 2016.

[CSZ+15a] Hamed Chitsaz, Hamid Shaker, Hamidreza Zareipour, David Wood, and Nima Amjady. Short-term electricity load forecasting of buildings in microgrids. *Energy and Buildings*, 99:50–60, July 2015.

[CSZ+15b] Hamed Chitsaz, Hamid Shaker, Hamidreza Zareipour, David Wood, and Nima Amjady. Short-term electricity load forecasting of buildings in microgrids. *Energy and Buildings*, 99:50–60, July 2015.

[Cyb89] G Cybenkot. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[CZAS12] Giuseppe Tommaso Costanzo, Guchuan Zhu, Miguel F. Anjos, and Gilles Savard. A System Architecture for Autonomous Demand Side Load Management in Smart Buildings. *IEEE Transactions on Smart Grid*, 3(4):2157–2165, December 2012.

[DAM03] Fernando Morgado Dias, Ana Antunes, and Alexandre Manuel Mota. Regularization versus early stopping: A case study with a real system. In *2nd IFAC Conference Control Systems Design, Bratislava, República Eslovaca*, 2003.

[DBDBM+78] Carl De Boor, Carl De Boor, Etats-Unis Mathématicien, Carl De Boor, and Carl De Boor. *A Practical Guide to Splines*, volume 27. springer-verlag New York, 1978.

[DBW15] Ni Ding, Yvon Bésanger, and Frédéric Wurtz. Next-day MV/LV substation load forecaster using time series method. *Electric Power Systems Research*, 119:345–354, February 2015.

[DCL05] Bing Dong, Cheng Cao, and Siew Eang Lee. Applying support vector machines to predict building energy consumption in tropical region. *Energy and Buildings*, 37(5):545–553, May 2005.

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, Miami, FL, June 2009. IEEE.

[DFH97a] F. Dan Foresee and M.T. Hagan. Gauss-Newton approximation to Bayesian learning. In *Proceedings of International Conference on Neural Networks (ICNN'97)*, volume 3, pages 1930–1935, Houston, TX, USA, 1997. IEEE.

[DFH97b] F. Dan Foresee and M.T. Hagan. Gauss-Newton approximation to Bayesian learning. In *Proceedings of International Conference on Neural Networks (ICNN'97)*, volume 3, pages 1930–1935, Houston, TX, USA, 1997. IEEE.

[DFV06] Sophie Dabo-Niang, Frideric Ferraty, and Philippe Vieu. Mode estimation for functional random variable and its application for curve classification. *Far East Journal of Theoretical Statistics*, 18(1):93–119, January 2006.

[Dia09] Marcos Alvarez Diaz. Evidence against the Spanish stock market efficiency using the Nearest Neighbour method and a cluster forecasting technique. *International Journal of Monetary Economics and Finance*, 2(1):16, 2009.

[Dira] Directive (EU) 2018/ 2001 of the European Parlament and of the Council - of 11 December 2018 - on the promotion of the use of energy from renewable sources. page 128.

[Dirb] Directive (EU) 2019/ 944 of the European Parlament and of the Council - of 5 June 2019 - on common rules for the internal market for electricity and amending Directive 2012/ 27/ EU. page 75.

[DSDSMS18] Katia Gregio Di Santo, Silvio Giuseppe Di Santo, Renato Machado Monaro, and Marco Antonio Saidel. Active demand side management for households in smart grids using optimization and artificial intelligence. *Measurement*, 115:152–161, February 2018.

[DSKDSS15] Katia Gregio Di Santo, Eduardo Kanashiro, Silvio Giuseppe Di Santo, and Marco Antonio Saidel. A review on smart grids and experiences in Brazil. *Renewable and Sustainable Energy Reviews*, 52:1072–1082, December 2015.

[EÁBRA11] Guillermo Escrivá-Escrivá, Carlos Álvarez-Bel, Carlos Roldán-Blay, and Manuel Alcázar-Ortega. New artificial neural network prediction method for electrical consumption forecasting based on building end-uses. *Energy and Buildings*, 43(11):3112–3119, November 2011.

[ECo] E-Control Marktregeln: SLP definionen. http://www.apcs.at/apcs/regelwerk/aktuelle_version/soma-strom-kapitel-6-jan-2016-v3.4.pdf.

[EPC18] Eunice Espe, Vidyasagar Potdar, and Elizabeth Chang. Prosumer Communities and Relationships in Smart Grids: A Literature Review, Evolution and Future Directions. *Energies*, 11(10):2528, September 2018.

[ET20] European Commission. Directorate General for Energy. and Tractebel Impact. *Benchmarking Smart Metering Deployment in the EU-28: Final Report*. Publications Office, LU, 2020.

[EU14] EU. Smart grids and meters. https://ec.europa.eu/energy/topics/markets-and-consumers/smart-grids-and-meters_en, July 2014.

[Eur18] European Commission. Directorate General for Research and Innovation. *Final Report of the High-Level Panel of the European Decarbonisation Pathways Initiative.* Publications Office, LU, 2018.

[Eur20] European Commission. Joint Research Centre. *State-of-the-Art for Assessment of Solar Energy Technologies 2019.* Publications Office, LU, 2020.

[FBP11] Ivan Fernandez, Cruz E. Borges, and Yoseba K. Penya. Efficient building load forecasting. pages 1–8. IEEE, September 2011.

[FCDM+16] Gabriella Ferruzzi, Guido Cervone, Luca Delle Monache, Giorgio Graditi, and Francesca Jacobone. Optimal bidding in a Day-Ahead energy market for Micro Grid under uncertainty in renewable energy production. *Energy*, 106:194–202, July 2016.

[Fer11] Frédéric Ferraty, editor. *Recent Advances in Functional Data Analysis and Related Topics*. Contributions to Statistics. Physica-Verlag HD, Heidelberg, 2011.

[FGSC19] Seyedeh Fallah, Mehdi Ganjkhani, Shahaboddin Shamshirband, and Kwok-wing Chau. Computational Intelligence on Short-Term Load Forecasting: A Methodological Overview. *Energies*, 12(3):393, January 2019.

[FGSV13] F. Ferraty, A. Goia, E. Salinelli, and P. Vieu. Functional projection pursuit regression. *TEST*, 22(2):293–320, June 2013.

[FGV02] Frédéric Ferraty, Aldo Goia, and Philippe Vieu. Functional nonparametric model for time series: A fractal approach for dimension reduction. *Test*, 11(2):317–344, December 2002.

[FHT08] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer series in statistics Springer, Berlin, 2008.

[FLV06] Frédéric Ferraty, Ali Laksaci, and Philippe Vieu. Estimating Some Characteristics of the Conditional Distribution in Nonparametric Functional Models. *Statistical Inference for Stochastic Processes*, 9(1):47–76, May 2006.

[FM94] Jianqing Fan and James S. Marron. Fast Implementations of Nonparametric Curve Estimators. *Journal of Computational and Graphical Statistics*, 3(1):35–56, March 1994.

[FM07] J. Nuno Fidalgo and Manuel A. Matos. Forecasting portugal global load with artificial neural networks. In Joaquim Marques de Sá, Luís A. Alexandre, Włodzisław Duch, and Danilo Mandic, editors, *Artificial Neural Networks – ICANN 2007*, pages 728–737, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[FMV07] Frédéric Ferraty, André Mas, and Philippe Vieu. Nonparametric Regression of Functional Data: Inference and Practical Aspects. *Australian & New Zealand Journal of Statistics*, 49(3):267–286, September 2007.

[FPV03] Frédéric Ferraty, Agnès Peuch, and Philippe Vieu. Modèle à indice fonctionnel simple. *Comptes Rendus Mathematique*, 336(12):1025–1028, June 2003.

[FR11] Frédéric Ferraty and Yves Romain, editors. *The Oxford Handbook of Functional Data Analysis*. Oxford Univ. Press, Oxford, 2011.

[FRS+13] Aurélie Foucquier, Sylvain Robert, Frédéric Suard, Louis Stéphan, and Arnaud Jay. State of the art in building modelling and energy performances prediction: A review. *Renewable and Sustainable Energy Reviews*, 23:272–288, July 2013.

[FV03] F. Ferraty and Ph. Vieu. Functional nonparametric statistics: A double infinite dimensional framework. In *Recent Advances and Trends in Nonparametric Statistics*, pages 61–76. Elsevier, 2003.

[FV04] F. Ferraty and P. Vieu. Nonparametric models for functional data, with application in regression, time series prediction and curve discrimination. *Journal of Nonparametric Statistics*, 16(1-2):111–125, February 2004.

[FV06] Frédéric Ferraty and Philippe Vieu. *Nonparametric Functional Data Analysis Theory and Practice*. Springer, New York, 2006.

[FVKV12] F. Ferraty, I. Van Keilegom, and P. Vieu. Regression when both response and predictor are functions. *Journal of Multivariate Analysis*, 109:10–28, August 2012.

[Gam] Gamma Function – from Wolfram MathWorld. https://mathworld.wolfram.com/GammaFunction.html.

[GAWY17] Iman Ghalehkhondabi, Ehsan Ardjmand, Gary R. Weckman, and William A. Young. An overview of energy demand forecasting methods published in 2005–2015. *Energy Systems*, 8(2):411–447, May 2017.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[GDPB+20] Maryam Gholamzadehmir, Claudio Del Pero, Simone Buffa, Roberto Fedrizzi, and Niccolo Aste. Adaptive-predictive control strategy for HVAC systems in smart buildings – A review. *Sustainable Cities and Society*, 63:102480, December 2020.

[GDS19] Maedeh Ghorbanian, Sarineh Hacopian Dolatabadi, and Pierluigi Siano. Big Data Issues in Smart Grids: A Survey. *IEEE Systems Journal*, 13(4):4158–4168, December 2019.

[Gee11] Gery Geenens. Curse of dimensionality and related issues in nonparametric functional regression. *Statistics Surveys*, 5(0):30–43, 2011.

[GFI18] Fivos Galatoulas, Marc Frere, and Christos Ioakimidis. An Overview of Renewable Smart District Heating and Cooling Applications with Thermal Storage in Europe:. In *Proceedings of the 7th International Conference on Smart Cities and Green ICT Systems*, pages 311–319, Funchal, Madeira, Portugal, 2018. SCITEPRESS - Science and Technology Publications.

[GKS13] Lazaros Gkatzikis, Iordanis Koutsopoulos, and Theodoros Salonidis. The Role of Aggregators in Smart Grid Demand Response Markets. *IEEE Journal on Selected Areas in Communications*, 31(7):1247–1257, July 2013.

[GRW79] Theodor A. Gasser, Murray Rosenblatt, and Workshop Smoothing Techniques for Curve Estimation, editors. *Smoothing Techniques for Curve Estimation: Proceedings of a Workshop Held in Heidelberg, April 2 - 4, 1979*. Number 757 in Lecture Notes in Mathematics. Springer, Berlin, 1979.

[GT18] A. N. Gorban and I. Y. Tyukin. Blessing of dimensionality: Mathematical foundations of the statistical physics of data. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 376(2118):20170237, April 2018.

[GZ05] Pedro A. González and Jesús M. Zamarreño. Prediction of hourly energy consumption in buildings based on a feedback artificial neural network. *Energy and Buildings*, 37(6):595–601, June 2005.

[HA18] Rob J Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, 2018.

[Ham95] James D Hamilton. Time series analysis. *Economic Theory. II, Princeton University Press, USA*, pages 625–630, 1995.

[Har96] Jeffrey D. Hart. Some automated methods of smoothing time-dependent data. *Journal of Nonparametric Statistics*, 6(2-3):115–142, January 1996.

[HBA+14] Luis Hernández, Carlos Baladrón, Javier Aguiar, Lorena Calavia, Belén Carro, Antonio Sánchez-Esguevillas, Francisco Pérez, Ángel Fernández, and Jaime Lloret. Artificial Neural Network for Short-Term Load Forecasting in Distribution Systems. *Energies*, 7(3):1576–1598, March 2014.

[HC15] Filmon G. Habtemichael and Mecit Cetin. Short-term traffic flow rate forecasting based on identifying similar traffic patterns. *Transportation Research Part C: Emerging Technologies*, September 2015.

[HCC95] Hong-Tzer Yang, Chao-Ming Huang, and Ching-Lien Huang. Identification of ARMAX model for short term load forecasting: An evolutionary programming approach. pages 325–330. IEEE, 1995.

[HCR18] Cristina Heghedus, Antorweep Chakravorty, and Chunming Rong. Energy Load Forecasting Using Deep Learning. pages 146–151. IEEE, May 2018.

[HFS14] Christian Hirsch, Lucas Friedrich, and Hartmut Schmeck. Kurzfristige Lastprognose von Einzelhaushalten. In *VDE-Kongress 2014*. VDE VERLAG GmbH, 2014.

[HGP15] Barry Hayes, Jorn Gruber, and Milan Prodanovic. Short-Term Load Forecasting at the local level using smart meter data. In *2015 IEEE Eindhoven PowerTech*, pages 1–6. IEEE, June 2015.

[HGZA18] Stephen Haben, Georgios Giasemidis, Florian Ziel, and Siddharth Arora. Short term load forecasting and the effect of temperature at the low voltage level. *International Journal of Forecasting*, December 2018.

[HK06] Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, October 2006.

[HK12] Lajos Horváth and Piotr Kokoszka. *Inference for Functional Data with Applications*, volume 200 of *Springer Series in Statistics*. Springer New York, New York, NY, 2012.

[HKV19] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, editors. *Automated Machine Learning: Methods, Systems, Challenges*. The Springer Series on Challenges in Machine Learning. Springer International Publishing, Cham, 2019.

[HLC+97a] Wolfgang Härdle, Helmut Lütkepohl, Rong Chen, Wolfgang Hardle, and Helmut Lutkepohl. A Review of Nonparametric Time Series Analysis. *International Statistical Review / Revue Internationale de Statistique*, 65(1):49, April 1997.

[HLC+97b] Wolfgang Härdle, Helmut Lütkepohl, Rong Chen, Wolfgang Hardle, and Helmut Lutkepohl. A Review of Nonparametric Time Series Analysis. *International Statistical Review / Revue Internationale de Statistique*, 65(1):49, April 1997.

[HM94] M.T. Hagan and M.B. Menhaj. Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, Nov./1994.

[HP16] Barry Patrick Hayes and Milan Prodanovic. State Forecasting and Operational Planning for Distribution Network Energy Management Systems. *IEEE Transactions on Smart Grid*, 7(2):1002–1011, March 2016.

[HPS] Paul G Hoel, Sidney C Port, and Charles J Stone. *Introduction to Probability Theory, 1971*. Houghton Mifflin Company.

[HPS01] H.S. Hippert, C.E. Pedreira, and R.C. Souza. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on Power Systems*, 16(1):44–55, Feb./2001.

[HSG16] Stephen Haben, Colin Singleton, and Peter Grindrod. Analysis and Clustering of Residential Customers Energy Behavioral Demand Using Smart Meter Data. *IEEE Transactions on Smart Grid*, 7(1):136–144, January 2016.

[Hub64] Peter J. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, March 1964.

[HWMS04] Wolfgang Härdle, Axel Werwatz, Marlene Müller, and Stefan Sperlich. *Nonparametric and Semiparametric Models*. Springer Series in Statistics. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[HWVA13] Samuel Humeau, Tri Kurniawan Wijaya, Matteo Vasirani, and Karl Aberer. Electricity load forecasting for residential customers: Exploiting aggregation and correlation between households. In *2013 Sustainable Internet and ICT for Sustainability (SustainIT)*, pages 1–6, Palermo, Italy, October 2013. IEEE.

[HWVG+14] Stephen Haben, Jonathan Ward, Danica Vukadinovic Greetham, Colin Singleton, and Peter Grindrod. A new error measure for forecasts of household-level, high resolution electrical energy consumption. *International Journal of Forecasting*, 30(2):246–256, April 2014.

[Iri14] Irish Standard Load Profiles. https://rmdservice.com/standard-load-profiles/, March 2014.

[IYIO14] Yumiko Iwafune, Yoshie Yagita, Takashi Ikegami, and Kazuhiko Ogimoto. Short-term forecasting of residential building load for distributed energy management. In *2014 IEEE International Energy Conference (ENERGYCON)*, pages 1197–1204. IEEE, May 2014.

[JAW+12] Fahad Javed, Naveed Arshad, Fredrik Wallin, Iana Vassileva, and Erik Dahlquist. Forecasting for demand response in smart grids: An analysis on use of anthropologic and structural data and short term multiple loads forecasting. *Applied Energy*, 96:150–160, August 2012.

[JMC14] Jorjeta G. Jetcheva, Mostafa Majidpour, and Wei-Peng Chen. Neural network model ensembles for building-level electricity load forecasts. *Energy and Buildings*, 84:214–223, December 2014.

[Jor19] A. Rezaee Jordehi. Optimisation of demand response in electric power systems, a review. *Renewable and Sustainable Energy Reviews*, 103:308–319, April 2019.

[JSCT14] Rishee K. Jain, Kevin M. Smith, Patricia J. Culligan, and John E. Taylor. Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy. *Applied Energy*, 123:168–178, June 2014.

[JV86] Roy Jonker and Ton Volgenant. Improving the Hungarian assignment algorithm. *Operations Research Letters*, 5(4):171–175, October 1986.

[JWHT13] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*, volume 103 of *Springer Texts in Statistics*. Springer New York, New York, NY, 2013.

[KAS13] Rajesh Kumar, R.K. Aggarwal, and J.D. Sharma. Energy analysis of a building using artificial neural network: A review. *Energy and Buildings*, 65:352–358, October 2013.

[KB17] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, January 2017.

[KC18] Sandeep Kakran and Saurabh Chanana. Smart operations of smart grids integrated with distributed generation: A review. *Renewable and Sustainable Energy Reviews*, 81:524–535, January 2018.

[KC19] Tae-Young Kim and Sung-Bae Cho. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy*, 182:72–81, September 2019.

[KCBG13] Anna Magdalena Kosek, Giuseppe Tommaso Costanzo, Henrik W. Bindner, and Oliver Gehrke. An overview of demand side management control

schemes for buildings in smart grids. In *2013 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, pages 1–9. IEEE, August 2013.

[KDJ+17] Weicong Kong, Zhao Yang Dong, Youwei Jia, David J. Hill, Yan Xu, and Yuan Zhang. Short-Term Residential Load Forecasting based on LSTM Recurrent Neural Network. *IEEE Transactions on Smart Grid*, pages 1–1, 2017.

[KH88] Kung and Hwang. An algebraic projection analysis for optimal hidden units size and learning rates in back-propagation learning. In *IEEE International Conference on Neural Networks*, pages 363–370 vol.1, San Diego, CA, USA, 1988. IEEE.

[KH94] Jan F Kreider and Jeff S Haberl. Predicting hourly building energy use: The great energy predictor shootout–Overview and discussion of results. Technical report, American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., Atlanta, GA (United States), 1994.

[KM] Vitaly Kuznetsov and Mehryar Mohri. Learning Theory and Algorithms for Forecasting Non-stationary Time Series. page 9.

[KM18] Vitaly Kuznetsov and Mehryar Mohri. Theory and Algorithms for Forecasting Time Series. *arXiv:1803.05814 [cs]*, March 2018.

[KMS+16a] Ahsan Raza Khan, Anzar Mahmood, Awais Safdar, Zafar A. Khan, and Naveed Ahmed Khan. Load forecasting, dynamic pricing and DSM in smart grid: A review. *Renewable and Sustainable Energy Reviews*, 54:1311–1322, February 2016.

[KMS+16b] Ahsan Raza Khan, Anzar Mahmood, Awais Safdar, Zafar A. Khan, and Naveed Ahmed Khan. Load forecasting, dynamic pricing and DSM in smart grid: A review. *Renewable and Sustainable Energy Reviews*, 54:1311–1322, February 2016.

[KW52] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pages 462–466, 1952.

[Lav21] Michael Laver. David Spiegelhalter: The Art of Statistics: Learning from Data: (Pelican, 2020), pp. 426, ISBN: 978-0241258767. *Society*, 58(3):241–243, June 2021.

[LBH+16] Thomas M. Lawrence, Marie-Claude Boudreau, Lieve Helsen, Gregor Henze, Javad Mohammadpour, Doug Noonan, Dieter Patteeuw, Shanti

Pless, and Richard T. Watson. Ten questions concerning integrating smart buildings into the smart grid. *Building and Environment*, 108:273–283, November 2016.

[LGC+12] N. Leemput, F. Geth, B. Claessens, J. Van Roy, R. Ponnette, and J. Driesen. A case study of coordinated electric vehicle charging for peak shaving on a low voltage grid. In *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pages 1–7, 2012.

[LGT98] Steve Lawrence, C Lee Giles, and Ah Chung Tsoi. What size neural network gives optimal generalization? Convergence properties of back-propagation. Technical report, 1998.

[Lie13] Dominik Liebl. Modeling and forecasting electricity spot prices: A functional data perspective. *The Annals of Applied Statistics*, 7(3):1562–1592, September 2013.

[LNVR07] Daniel J. Levitin, Regina L. Nuzzo, Bradley W. Vines, and J. O. Ramsay. Introduction to functional data analysis. *Canadian Psychology/Psychologie canadienne*, 48(3):135–155, 2007.

[LS01] W.V. Li and Q.-M. Shao. Gaussian processes: Inequalities, small ball probabilities and applications. In *Stochastic Processes: Theory and Methods*, volume 19 of *Handbook of Statistics*, pages 533–597. Elsevier, 2001.

[LSE+07] Alicia Troncoso Lora, Jesus M. Riquelme Santos, Antonio Gomez Exposito, Jose Luis MartÍnez Ramos, and Jose C. Riquelme Santos. Electricity Market Price Forecasting Based on Weighted Nearest Neighbors Techniques. *IEEE Transactions on Power Systems*, 22(3):1294–1301, August 2007.

[LSPB+12] J. Llanos, D. Sáez, R. Palma-Behnke, A. Núñez, and G. Jiménez-Estévez. Load profile generator and load forecasting for a renewable based microgrid using self organizing maps and neural networks. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2012.

[LSS+02a] Alicia Troncoso Lora, Jesús Riquelme Santos, José Riquelme Santos, Antonio Gómez Expósito, and José Luís Martínez Ramos. A Comparison of Two Techniques for Next- Day Electricity Price Forecasting. In Hujun Yin, Nigel Allinson, Richard Freeman, John Keane, Simon Hubbard, Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen, editors, *Intelligent Data Engineering and Automated Learning — IDEAL 2002*, volume 2412, pages 384–390. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[LSS+02b] Alicia Troncoso Lora, Jose Riquelme Santos, Jesus Riquelme Santos, Jose Luis Martinez Ramos, and Antonio Gomez Exposito. Electricity Market Price Forecasting: Neural Networks versus Weighted-Distance k Nearest Neighbours. In Abdelkader Hameurlain, Rosine Cicchetti, Roland Traunmüller, Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen, editors, *Database and Expert Systems Applications*, volume 2453, pages 321–330. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.

[LXT16] Guodong Liu, Yan Xu, and Kevin Tomsovic. Bidding Strategy for Microgrid in Day-Ahead Market Based on Hybrid Stochastic/Robust Optimization. *IEEE Transactions on Smart Grid*, 7(1):227–237, January 2016.

[MA14] J. Steve Marron and Andrés M. Alonso. Overview of object oriented data analysis: An overview of object oriented data analysis. *Biometrical Journal*, 56(5):732–753, September 2014.

[MAM16a] Daniel L. Marino, Kasun Amarasinghe, and Milos Manic. Building energy load forecasting using Deep Neural Networks. pages 7046–7051. IEEE, October 2016.

[MAM16b] Daniel L. Marino, Kasun Amarasinghe, and Milos Manic. Building energy load forecasting using Deep Neural Networks. In *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, pages 7046–7051, Florence, Italy, October 2016. IEEE.

[MBC16] Lubna Mariam, Malabika Basu, and Michael F. Conlon. Microgrid: Architecture, policy and future trends. *Renewable and Sustainable Energy Reviews*, 64:477–489, October 2016.

[MDRH+18] Benjamin Manrique Delgado, Reino Ruusu, Ala Hasan, Simo Kilpeläinen, Sunliang Cao, and Kai Sirén. Energetic, Cost, and Comfort Performance of a Nearly-Zero Energy Building Including Rule-Based Control of Four Sources of Energy Flexibility. *Buildings*, 8(12):172, December 2018.

[MHD+13] Andrei Marinescu, Colin Harris, Ivana Dusparic, Siobhan Clarke, and Vinny Cahill. Residential electrical demand forecasting in very small scale: An evaluation of forecasting methods. In *2013 2nd International Workshop on Software Engineering Challenges for the Smart Grid (SE4SG)*, pages 25–32. IEEE, May 2013.

[MK10] Khosrow Moslehi and Ranjit Kumar. A Reliability Perspective of the Smart Grid. *IEEE Transactions on Smart Grid*, 1(1):57–64, June 2010.

[MMA+22] Arash Moradzadeh, Behnam Mohammadi-Ivatloo, Mehdi Abapour, Amjad Anvari-Moghaddam, and Sanjiban Sekhar Roy. Heating and Cooling

Loads Forecasting for Residential Buildings Based on Hybrid Machine Learning Applications: A Comprehensive Review and Comparative Analysis. *IEEE Access*, 10:2196–2215, 2022.

[MMG18] Paige A Mynhoff, Elena Mocanu, and Madeleine Gibescu. Statistical learning versus deep learning: Performance comparison for building energy prediction methods. In *IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6, 2018.

[MNGK16] Elena Mocanu, Phuong H. Nguyen, Madeleine Gibescu, and Wil L. Kling. Deep learning for estimating building energy consumption. *Sustainable Energy, Grids and Networks*, 6:91–99, June 2016.

[Mor78] Jorge J. Moré. The Levenberg-Marquardt algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, volume 630, pages 105–116. Springer Berlin Heidelberg, Berlin, Heidelberg, 1978.

[MPB+15] Joaquim Massana, Carles Pous, Llorenç Burgas, Joaquim Melendez, and Joan Colomer. Short-term load forecasting in a non-residential building contrasting models and attributes. *Energy and Buildings*, 92:322–330, April 2015.

[MR89] I. Moghram and S. Rahman. Analysis and evaluation of five short-term load forecasting techniques. *IEEE Transactions on Power Systems*, 4(4):1484–1491, Nov./1989.

[MRCA14] R. Mena, F. Rodríguez, M. Castilla, and M.R. Arahal. A prediction model based on neural networks for the energy consumption of a bioclimatic building. *Energy and Buildings*, 82:142–155, October 2014.

[MRRJ11] Antti Mutanen, Maija Ruska, Sami Repo, and Pertti Jarventausta. Customer Classification and Load Profiling Method for Distribution Systems. *IEEE Transactions on Power Delivery*, 26(3):1755–1763, July 2011.

[MST02] G Mihalakakou, M Santamouris, and A Tsangrassoulis. On the energy consumption in residential buildings. *Energy and Buildings*, 34(7):727–736, August 2002.

[MTAR15] Francisco Martínez-Álvarez, Alicia Troncoso, Gualberto Asencio-Cortés, and José Riquelme. A Survey on Data Mining Techniques Applied to Electricity-Related Time Series Forecasting. *Energies*, 8(12):13162–13193, November 2015.

[Mus14] Joseph Muscat. *Functional Analysis*. Springer International Publishing, Cham, 2014.

[NA16] Eva Niesten and Floortje Alkemade. How is value created and captured in smart grids? A review of the literature and an analysis of pilot projects. *Renewable and Sustainable Energy Reviews*, 53:629–638, January 2016.

[Nad64] E. A. Nadaraya. On Estimating Regression. *Theory of Probability & Its Applications*, 9(1):141–142, January 1964.

[NB10] Guy R. Newsham and Benjamin J. Birt. Building-level occupancy data to improve ARIMA-based electricity use forecasts. In *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building - BuildSys '10*, page 13, Zurich, Switzerland, 2010. ACM Press.

[NF08] Alberto Hernandez Neto and Flávio Augusto Sanzovo Fiorelli. Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption. *Energy and Buildings*, 40(12):2169–2176, January 2008.

[NHG17] Seyyed Mostafa Nosratabadi, Rahmat-Allah Hooshmand, and Eskandar Gholipour. A comprehensive review on microgrid and virtual power plant concepts employed for distributed energy resources scheduling in power systems. *Renewable and Sustainable Energy Reviews*, 67:341–363, January 2017.

[NL14] Duong Tung Nguyen and Long Bao Le. Optimal Bidding Strategy for Microgrids Considering Renewable Energy and Building Thermal Dynamics. *IEEE Transactions on Smart Grid*, 5(4):1608–1620, July 2014.

[OLM+18] Pol Olivella-Rosell, Pau Lloret-Gallego, Íngrid Munné-Collado, Roberto Villafafila-Robles, Andreas Sumper, Stig Ottessen, Jayaprakash Rajasekharan, and Bernt Bremdal. Local Flexibility Market Design for Aggregators Providing Multiple Flexibility Services at Distribution Network Level. *Energies*, 11(4):822, April 2018.

[PBF11a] Yoseba K. Penya, Cruz E. Borges, and Ivan Fernandez. Short-term load forecasting in air-conditioned non-residential Buildings. In *2011 IEEE International Symposium on Industrial Electronics*, pages 1–6. IEEE, September 2011.

[PBF11b] Yoseba K. Penya, Cruz E. Borges, and Ivan Fernandez. Short-term load forecasting in non-residential Buildings. In *IEEE Africon'11*, pages 1–6. IEEE, September 2011.

[PKG14a] Omid Palizban, Kimmo Kauhaniemi, and Josep M. Guerrero. Microgrids in active network management – part II: System operation, power quality

and protection. *Renewable and Sustainable Energy Reviews*, 36:440–451, August 2014.

[PKG14b] Omid Palizban, Kimmo Kauhaniemi, and Josep M. Guerrero. Microgrids in active network management—Part I: Hierarchical control, energy storage, virtual power plants, and market participation. *Renewable and Sustainable Energy Reviews*, 36:428–439, August 2014.

[PMA17] Jose Portela, Antonio Munoz, and Estrella Alonso. Forecasting functional time series with a new Hilbertian ARMAX model: Application to electricity price forecasting. *IEEE Transactions on Power Systems*, pages 1–1, 2017.

[POC+17] Alexandru Pîrjan, Simona-Vasilica Oprea, George Căruțașu, Dana-Mihaela Petroșanu, Adela Bâra, and Cristina Coculescu. Devising Hourly Forecasting Solutions Regarding Electricity Consumption in the Case of Commercial Center Type Consumers. *Energies*, 10(11):1727, October 2017.

[PS13] Efstathios Paparoditis and Theofanis Sapatinas. Short-Term Load Forecasting: The Similar Shape Functional Time-Series Predictor. *IEEE Transactions on Power Systems*, 28(4):3818–3825, November 2013.

[RB93] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *IEEE International Conference on Neural Networks*, pages 586–591, San Francisco, CA, USA, 1993. IEEE.

[RCC14] Filipe Rodrigues, Carlos Cardeira, and J.M.F. Calado. The Daily and Hourly Energy Consumption and Load Forecasting Using Artificial Neural Network Method: A Case Study Using a Set of 93 Households in Portugal. *Energy Procedia*, 62:220–229, 2014.

[RD91] J. O. Ramsay and C. J. Dalzell. Some Tools for Functional Data Analysis. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(3):539–561, July 1991.

[REG15] Matej Rejc, Alfred Einfalt, and Tobias Gawron-Deutsch. Short-term aggregated load and distributed generation forecast using fuzzy grouping approach. In *2015 International Symposium on Smart Electric Distribution Systems and Technologies (EDST)*, pages 212–217. IEEE, September 2015.

[RHG09] James Ramsay, Giles Hooker, and Spencer Graves. *Functional Data Analysis with R and MATLAB*. Springer New York, New York, NY, 2009.

[RHW86] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986.

[RK15] Muhammad Qamar Raza and Abbas Khosravi. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renewable and Sustainable Energy Reviews*, 50:1352–1372, October 2015.

[RNK16] Seunghyoung Ryu, Jaekoo Noh, and Hongseok Kim. Deep Neural Network Based Demand Side Short Term Load Forecasting. *Energies*, 10(1):3, December 2016.

[RS02] James O. Ramsay and Bernard W. Silverman, editors. *Applied Functional Data Analysis: Methods and Case Studies*. Springer Series in Statistics. Springer New York, New York, NY, 2002.

[RS05] J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer Series in Statistics. Springer, New York, NY, 2. ed edition, 2005.

[RZ19] Jason Runge and Radu Zmeureanu. Forecasting Energy Use in Buildings Using Artificial Neural Networks: A Review. *Energies*, 12(17):3254, August 2019.

[SC08] Shiliang Sun and Qiaona Chen. Kernel Regression with a Mahalanobis Metric for Short-Term Traffic Flow Forecasting. In Colin Fyfe, Dongsup Kim, Soo-Young Lee, and Hujun Yin, editors, *Intelligent Data Engineering and Automated Learning – IDEAL 2008*, volume 5326, pages 9–16. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[Sch15] Juergen Schmidhuber. Deep Learning in Neural Networks: An Overview. *Neural Networks*, 61:85–117, January 2015.

[SGQ+19] Hongbin Sun, Qinglai Guo, Junjian Qi, Venkataramana Ajjarapu, Richard Bravo, Joe Chow, Zhengshuo Li, Rohit Moghe, Ehsan Nasr-Azadani, Ujjwol Tamrakar, Glauco N. Taranto, Reinaldo Tonkoski, Gustavo Valverde, Qiuwei Wu, and Guangya Yang. Review of Challenges and Research Opportunities for Voltage Control in Smart Grids. *IEEE Transactions on Power Systems*, 34(4):2790–2801, July 2019.

[SHF20] Ying Sun, Fariborz Haghighat, and Benjamin C.M. Fung. A review of the-state-of-the-art in data-driven approaches for building energy prediction. *Energy and Buildings*, 221:110022, August 2020.

[Sia14] Pierluigi Siano. Demand response and smart grids—A survey. *Renewable and Sustainable Energy Reviews*, 30:461–478, February 2014.

[SJW13] S. Subbayya, J. G. Jetcheva, and Wei-Peng Chen. Model selection criteria for short-term microgrid-scale electricity load forecasts. In *2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–6. IEEE, February 2013.

[SK12] Masashi Sugiyama and Motoaki Kawanabe. *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*. The MIT Press, March 2012.

[SL15] Ismail Shah and Francesco Lisi. Day-ahead electricity demand forecasting with nonparametric functional models. pages 1–5. IEEE, May 2015.

[SLW16] Guang Shi, Derong Liu, and Qinglai Wei. Energy consumption prediction of office buildings based on echo state networks. *Neurocomputing*, 216:478–488, December 2016.

[SPS16] A. K. Srivastava, Ajay Shekhar Pandey, and Devender Singh. Short-term load forecasting methods: A review. pages 130–138. IEEE, March 2016.

[SR14] Raffi Sevlian and Ram Rajagopal. A model for the effect of aggregation on short term load forecasting. In *IEEE Power and Energy Society General Meeting*, 2014.

[SR18] Raffi Sevlian and Ram Rajagopal. A scaling law for short term load forecasting on varying levels of aggregation. *International Journal of Electrical Power & Energy Systems*, 98:350–361, June 2018.

[SS12] L. Suganthi and Anand A. Samuel. Energy models for demand forecasting—A review. *Renewable and Sustainable Energy Reviews*, 16(2):1223–1240, February 2012.

[SSM16] Chao Sun, Fengchun Sun, and Scott J. Moura. Nonlinear predictive energy management of residential buildings with photovoltaics & batteries. *Journal of Power Sources*, 325:723–731, September 2016.

[STH+15] Bruce Stephen, Xiaoqing Tang, Poppy R. Harvey, Stuart Galloway, and Kyle I. Jennett. Incorporating Practice Theory in Sub-Profile Models for Short Term Aggregated Residential Load Forecasting. *IEEE Transactions on Smart Grid*, pages 1–8, 2015.

[Sto74] M. Stone. Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, January 1974.

[Sto82] Charles J. Stone. Optimal Global Rates of Convergence for Nonparametric Regression. *The Annals of Statistics*, 10(4):1040–1053, December 1982.

[SWKO02] Brian L Smith, Billy M Williams, and R Keith Oswald. Comparison of parametric and nonparametric models for traffic flow forecasting. *Transportation Research Part C: Emerging Technologies*, 10(4):303–321, August 2002.

[SXL18] Heng Shi, Minghao Xu, and Ran Li. Deep Learning for Household Load Forecasting—A Novel Pooling Deep RNN. *IEEE Transactions on Smart Grid*, 9(5):5271–5280, September 2018.

[Syn] Synthetische Lastprofile APCS - Power Clearing & Settlement. https://www.apcs.at/de/clearing/technisches-clearing/lastprofile.

[Tai14] Souhaib Ben Taieb. *Machine Learning Strategies for Multi-Step-Ahead Time Series Forecasting*. PhD thesis, PhD Thesis, Universit Libre de Bruxelles, Belgium, 2014.

[TB13] S.R. Twanabasu and B.A. Bremdal. Load forecasting in a smart grid oriented building. In *22nd International Conference and Exhibition on Electricity Distribution (CIRED 2013)*, pages 0907–0907, Stockholm, Sweden, 2013. Institution of Engineering and Technology.

[TLRSR+04] Alicia Troncoso Lora, Jesús Manuel Riquelme Santos, José Cristóbal Riquelme, Antonio Gómez Expósito, and José Luís Martínez Ramos. Time-Series Prediction: Application to the Short-Term Electric Energy Demand. In Ricardo Conejo, Maite Urretavizcaya, José-Luis Pérez-de-la-Cruz, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, and Gerhard Weikum, editors, *Current Topics in Artificial Intelligence*, volume 3040, pages 577–586. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[TMA16] Likewin Thomas, Manoj Kumar M V, and Annappa B. Discovery of optimal neurons and hidden layers in feed-forward Neural Network. In *2016 IEEE International Conference on Emerging Technologies and Innovative Business Practices for the Transformation of Societies (EmergiTech)*, pages 286–291, Mauritius, August 2016. IEEE.

[TVM02] A.C. Tsakoumis, S.S. Vladov, and V.M. Mladenov. Daily load forecasting based on previous day load. In *Th Seminar on Neural Network Applications in Electrical Engineering*, pages 83–86. IEEE, 2002.

[ÜVCS+15] Diana Ürge-Vorsatz, Luisa F. Cabeza, Susana Serrano, Camila Barreneche, and Ksenia Petrichenko. Heating and cooling energy trends and drivers in buildings. *Renewable and Sustainable Energy Reviews*, 41:85–98, January 2015.

[Vap91] V. Vapnik. Principles of risk minimization for learning theory. In *Proceedings of the 4th International Conference on Neural Information Processing Systems*, NIPS'91, pages 831–838, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

[Vap10] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Information Science and Statistics. Springer New York, New York, NY, 2., nd ed. softcover version of original hardcover edition 2000 edition, 2010.

[Vap13] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer science & business media, 2013.

[VCA12] Juan M. Vilar, Ricardo Cao, and Germán Aneiros. Forecasting next-day electricity demand and price using nonparametric functional methods. *International Journal of Electrical Power & Energy Systems*, 39(1):48–55, July 2012.

[VK16a] Oleg Valgaev and Friederich Kupzog. Building power demand forecasting. *it - Information Technology*, 58(1):37–43, February 2016.

[VK16b] Oleg Valgaev and Friederich Kupzog. Building power demand forecasting using K-nearest neighbors model - initial approach. In *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pages 1055–1060, Xi'an, China, October 2016. IEEE.

[VKS16] Oleg Valgaev, Friedrich Kupzog, and Hartmut Schmeck. Low-voltage power demand forecasting using K-nearest neighbors approach. In *2016 IEEE Innovative Smart Grid Technologies - Asia (ISGT-Asia)*, pages 1019–1024, Melbourne, Australia, November 2016. IEEE.

[VKS17a] Oleg Valgaev, Friederich Kupzog, and Hartmut Schmeck. Building power demand forecasting using K-nearest neighbours model – practical application in Smart City Demo Aspern project. *CIRED - Open Access Proceedings Journal*, 2017(1):1601–1604, October 2017.

[VKS17b] Oleg Valgaev, Friederich Kupzog, and Hartmut Schmeck. Designing K-nearest neighbors model for low voltage load forecasting. In *2017 IEEE Power & Energy Society General Meeting*, pages 1–5, Chicago, IL, July 2017. IEEE.

[VKS17c] Oleg Valgaev, Friederich Kupzog, and Hartmut Schmeck. Outlining Ensemble K-Nearest Neighbors Approach for Low-Voltage Power Demand Forecasting. In *Proceedings of the Eighth International Conference on Future Energy Systems*, pages 268–270, Shatin Hong Kong, May 2017. ACM.

[VKS20] Oleg Valgaev, Friederich Kupzog, and Hartmut Schmeck. Adequacy of neural networks for wide-scale day-ahead load forecasts on buildings and distribution systems using smart meter data. *Energy Informatics*, 3(1):28, December 2020.

[VPLZ19] A. Vaccaro, I. Pisica, L.L. Lai, and A.F. Zobaa. A review of enabling methodologies for information processing in smart grids. *International Journal of Electrical Power & Energy Systems*, 107:516–522, May 2019.

[VS78] V N Vapnik and A R Stefanyuk. Nonparametric methods for restoring the probability densities. *Avtomatika i telemekhanika*, 8:38–52, 1978.

[Was04] Larry Wasserman. *All of Statistics*. Springer Texts in Statistics. Springer New York, New York, NY, 2004.

[Wat64] Geoffrey S. Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A (1961-2002)*, 26(4):359–372, 1964.

[WCHK18] Yi Wang, Qixin Chen, Tao Hong, and Chongqing Kang. Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges. *IEEE Transactions on Smart Grid*, pages 1–1, 2018.

[WCM16] Jane-Ling Wang, Jeng-Min Chiou, and Hans-Georg Müller. Functional Data Analysis. *Annual Review of Statistics and Its Application*, 3(1):257–295, June 2016.

[Wei] Eric W. Weisstein. Lipschitz Function. https://mathworld.wolfram.com/LipschitzFunction.html.

[WNJ12] Yongli Wang, Dongxiao Niu, and Li Ji. Short-term power load forecasting based on IVL-BP neural network technology. *Systems Engineering Procedia*, 4:168–174, 2012.

[WVHA15] Tri Kurniawan Wijaya, Matteo Vasirani, Samuel Humeau, and Karl Aberer. Cluster-based aggregate forecasting for residential electricity demand using smart meter data. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 879–887. IEEE, October 2015.

[YBDS17] B. Yildiz, J.I. Bilbao, J. Dore, and A.B. Sproul. Recent advances in the analysis of residential electricity consumption and applications of smart meter data. *Applied Energy*, 208:402–427, December 2017.

[YFLL19] Songyuan Yu, Fang Fang, Yajuan Liu, and Jizhen Liu. Uncertainties of virtual power plant: Problems and countermeasures. *Applied Energy*, 239:454–470, April 2019.

[YHAG19] Jingpeng Yue, Zhijian Hu, Amjad Anvari-Moghaddam, and Josep M. Guerrero. A Multi-Market-Driven Approach to Energy Scheduling of Smart Microgrids in Distribution Networks. *Sustainability*, 11(2):301, January 2019.

[YMW05] Fang Yao, Hans-Georg Müller, and Jane-Ling Wang. Functional Data Analysis for Sparse Longitudinal Data. *Journal of the American Statistical Association*, 100(470):577–590, June 2005.

[YOHS18] Xing Yan, Yusuf Ozturk, Zechun Hu, and Yonghua Song. A review on price-driven residential demand response. *Renewable and Sustainable Energy Reviews*, 96:411–419, November 2018.

[YRZ05] Jin Yang, Hugues Rivard, and Radu Zmeureanu. On-line building energy prediction using adaptive artificial neural networks. *Energy and Buildings*, 37(12):1250–1259, December 2005.

[YSP+14] Jiahai Yuan, Jiakun Shen, Li Pan, Changhong Zhao, and Junjie Kang. Smart grids in China. *Renewable and Sustainable Energy Reviews*, 37:896–906, September 2014.

[ZDZ+22] Jizhong Zhu, Hanjiang Dong, Weiye Zheng, Shenglin Li, Yanting Huang, and Lei Xi. Review and prospect of data-driven techniques for load forecasting in integrated energy systems. *Applied Energy*, 321:119269, September 2022.

[Zen] Zentralanstalt für Meteorologie und Geodynamik — ZAMG. https://www.zamg.ac.at/cms/de/aktuell.

[ZHB18] Yang Zhang, Tao Huang, and Ettore Francesco Bompard. Big data analytics in smart grids: A review. *Energy Informatics*, 1(1):8, December 2018.

[ZL15] Jing-ting Zhong and Shuai Ling. Key Factors of K-Nearest Neighbours Nonparametric Regression in Short-Time Traffic Flow Forecasting. In Ershi Qi, Jiang Shen, and Runliang Dou, editors, *Proceedings of the*

21st International Conference on Industrial Engineering and Engineering Management 2014, pages 9–12. Atlantis Press, Paris, 2015.

[ZLY19] Aaron Zeng, Sheng Liu, and Yao Yu. Comparative study of data driven methods in building electricity use prediction. *Energy and Buildings*, 194:289–300, July 2019.

[Zuo00] Zuordnung der VDEW-Lastprofile zum Kundengruppenschlüssel. *VDEW Materialien*, 2000.