

New Data-Driven Probabilistic Forecasting Methods with Applications in Energy Systems

Zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)
von der KIT-Fakultät für Maschinenbau des
Karlsruher Instituts für Technologie (KIT)

genehmigte
Dissertation
von

M.Sc. Jorge Ángel González Ordiano

Tag der mündlichen Prüfung: 20. März 2019
Hauptreferent: apl. Prof. Dr.-Ing. Ralf Mikut
Korreferenten: Prof. Dr.-Ing. Veit Hagemeyer,
Prof. Dr.-Ing. Peter Bretschneider

"We wish to pursue the truth no matter where it leads. But to find the truth, we need imagination and skepticism both. We will not be afraid to speculate. But we will be careful to distinguish speculation from fact."

Carl Sagan

Kurzfassung

Um die Nachhaltigkeit der heutigen Energiesysteme zu steigern, muss die Installation von erneuerbaren Energiesystemen verstärkt werden. Leider sind viele dieser Systeme wetterabhängig, was das Bilanzieren von Energieerzeugung und Last erschwert. Prognosemodelle sind deshalb für die Regelung und Einsatzplanung von Energiesystemen notwendig. Derzeit beschreiben nur wenige Prognosemodelle die Prognoseunsicherheit, die für eine optimale Entscheidungsfindung eine wichtige Rolle spielt.

Probabilistische Prognosen bieten die Möglichkeit, eine solche Prognoseunsicherheit zu beschreiben. Ein gängiger Ansatz zur Beschreibung der Prognoseunsicherheit sind Quantil-Regressionen, die keine Annahmen des Verteilungstyps benötigen. Allerdings muss für diese Art von Regressionen eine spezielle nicht differenzierbare Kostenfunktion minimiert werden, was die Berechnung von Quantil-Regressionen mit komplexeren Ansätzen, wie beispielsweise künstlichen neuronalen Netzen, erschwert. Zusätzlich erfordern einige Optimierungs- und Einsatzplanungsalgorithmen andere Arten von Unsicherheitsbeschreibungen (z.B. eine parametrische Verteilung oder Szenarien).

Die vorliegende Arbeit versucht die obengenannten Probleme durch die Entwicklung neuer datengetriebener probabilistischer Prognosemethoden zu lösen. Eine der in dieser Arbeit vorgestellten Methoden ermöglicht die Berechnung von Quantil-Regressionen, ohne die nicht differenzierbare Kostenfunktion explizit zu minimieren. Diese Methode vereinfacht somit die Berechnung von Quantil-Regressionen unter Nutzung von komplexen datengetriebenen Verfahren. Außerdem stellt diese Dissertation ver-

schiedene Ansätze zur Umwandlung der Quantil-Regressionen in andere Arten von Unsicherheitsbeschreibungen vor. Diese Beschreibungen sind zum Beispiel Intervallprognosen, nichtparametrische und/oder parametrische Verteilungsprognosen und Szenario-Prognosen. Weiterhin beschreibt die vorliegende Dissertation neue Konzepte zur Schätzung der Gesamtverteilung von zwei zukünftigen Zeitreihenwerten und zur Ermittlung von hierarchischen Prognosen.

Alle genannten Methoden wurden in einer Open-Source-Toolbox implementiert und unter Verwendung von simulierten und realen Energiedaten validiert. Die Validierungsergebnisse zeigen, dass die entwickelten Methoden die Unsicherheit zukünftiger Energiewerte mit hoher Genauigkeit schätzen und somit für die Regelung und Einsatzplanung eines Energiesystems eingesetzt werden können.

Abstract

The desire of making the existing energy systems more sustainable, calls for an increase installation of renewable power systems. Nevertheless, the fact that some renewable power systems are completely weather-dependent complicates the necessary balancing between energy demand and supply. For such reason, forecasting models able to provide accurate information about the future generation and load have become quite useful for the control and scheduling of energy systems. However, most types of models do not quantify in any way the forecast uncertainty; information that is necessary for an optimal decision-making.

Probabilistic forecasts offer the possibility of describing such forecast uncertainty. A commonly used approach for describing the uncertainty of future energy time series values are quantile regressions. Their popularity stems from the fact that they do not require a distribution type assumption. Nevertheless, they are trained using a non-differentiable cost function that may complicate their training with more complex approaches, for instance, artificial neural networks. In addition, some optimization and scheduling algorithms necessitate other types of uncertainty descriptions (e.g., a parametric distribution or scenarios), yet there is a lack, to the best of the author's knowledge, of straightforward methods that transform the commonly used quantile regressions into those other types.

To tackle the above mentioned problems, the current thesis describes a series of new data-driven probabilistic forecasting methods. For instance, a method that allows the training of quantile regressions without the need of minimizing the original non-differentiable cost function. This opens the

door to the effortless obtainment of quantile regressions using more complex approaches. Likewise, the current thesis also presents different straightforward approaches for transforming quantile regressions into other types of uncertainty descriptions, i.e. interval forecasts, non-parametric and/or parametric distribution forecasts, and scenario forecasts. Moreover, two new concepts are also presented: one for describing the correlation of future time series values and another for obtaining coherent hierarchical forecasts with such description.

To conclude, the methods presented in the current thesis have been implemented as part of an open-source toolbox and have also been validated within an energy-related context using both simulated and real-world data. The validation results demonstrate that the developed approaches are able to accurately estimate the uncertainty of future energy time series values and hence can be used for the control and scheduling of energy systems.

Dedication

First and foremost, I would like to thank my direct supervisor apl. Prof. Dr.-Ing. Ralf Mikut, for his support, guidance, and countless constructive meetings these past couple of years. Your advice is the foundation in which the present thesis has been built. Likewise, I would like to acknowledge the support of Prof. Dr.-Ing. Veit Hagenmeyer for not only cosupervising my thesis, but also for given me the opportunity of conducting my research as part of the Institute for Automation and Applied Informatics. Additional thanks to Prof. Dr.-Ing. Peter Bretschneider for taking the time to read and evaluate the present work. Special thanks to PD Dr.-Ing. Markus Reischl, who by supervising my Bachelor's thesis jump started my research career so many years ago. My sincere thanks go also to Lutz Gröll, who helped me with several concepts and equations found in the present thesis.

I would also like to extend my thanks to all my colleagues at the Automated Image and Data Analysis group, specifically, Simon Waczowicz, Andreas Bartschat, Marian Turowski, Nicole Ludwig, Tim Scherr, Moritz Böhland, Johannes Stegmaier, Benjamin Schott, and Wolfgang Doneit. Thank you for the great working atmosphere that always allowed for discussion of both scientific and non-scientific topics alike; you have made the past years a lot of fun. I also want to acknowledge my fellow PhDs at our institute. Special thanks to Eric Braun, Jianlei Liu, and Riccardo Appino, without our discussions and collaboration most of what is written in the present thesis would not exist. In addition, I would also like to thank Theresa Schlegel and Sebastian Sauer for providing me with insights about some of the data used

herein. Special thanks to the Helmholtz Association ES2050 project, for the financial support provided.

Last but not least, I would like to thank my parents whose unconditional love and support are the reasons why I am standing here today. I cannot be thankful enough for the world of opportunities your hard work have given me and I hope to make you proud not only with this work, but in each of my future endeavors. At the same time, I want to thank my brother, my girlfriend, and her family for their support and care. No matter the distance I know I can count on you. Your love is the force that drives me through thick and thin. Thank you from the bottom of my heart.

Por último, quiero dar las gracias a mis padres. Su amor incondicional y apoyo son las razones por las que hoy me encuentro aquí. No hay manera suficiente de agradecerles las oportunidades que su arduo trabajo me han dado. Espero no sólo estén orgullosos de este trabajo, sino también de cada uno de mis decisiones futuras. Al mismo tiempo, quisiera agradecer a mi hermano, a mi novia y a su familia por su apoyo y atención. Sin importar la distancia, yo sé que puedo contar con ustedes. El amor de todos ustedes es la fuerza que me impulsa cada día. De todo corazón muchas gracias.

Karlsruhe, December 2018

Jorge Ángel González Ordiano

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Theoretical Background	4
1.2.1	Random Variables	4
1.2.2	Quantile Regressions	7
1.2.3	Time Series Forecasting	10
1.2.4	Probabilistic Forecasting	14
1.3	State-of-the-Art in Energy-Related Forecasting	17
1.4	Open Questions	27
1.5	Objectives	29
2	Probabilistic Forecasting	31
2.1	Overview	31
2.2	Methods	32
2.2.1	Quantile Regressions based on the Nearest Neighbors Quantile Filter	33
2.2.2	Interval Forecast	39
2.2.3	Non-Parametric Distribution Forecast	42
2.2.4	Parametric Distribution Forecast	47
2.2.5	Scenario Forecast	51
2.3	Evaluation Values	56
2.4	Experiments	60
2.4.1	Preliminary Information	61

2.4.2	Quantile Regressions based on the NNQF and Non-Parametric Distribution Forecast	64
2.4.3	Interval Forecast	75
2.4.4	Parametric Distribution Forecast	82
2.4.5	Scenario Forecast	90
2.5	Novelty and Remaining Questions	93
3	Hierarchical Probabilistic Forecasting	97
3.1	Overview	97
3.2	Joint Distribution Forecast	99
3.3	Hierarchical Probabilistic Forecast	102
3.4	Novelty and Remaining Questions	106
4	Software Implementation	109
4.1	Overview	109
4.2	SciXMiner Forecasting Extension	110
4.2.1	Quantile Regressions based on the NNQF	111
4.2.2	Interval Forecast	113
4.2.3	Parametric Distribution Forecast	115
4.2.4	Scenario Forecast	117
4.2.5	Hierarchical Probabilistic Forecast	120
4.3	Deployment without SciXMiner	121
4.4	Novelty and Remaining Questions	124
5	Application	125
5.1	Overview	125
5.2	Application on Real Data	125
5.2.1	Data	125
5.2.2	Experiments	127
5.2.3	Data Mining Techniques	129
5.2.4	Results Experiment 1	132
5.2.5	Results Experiment 2 (GEFCom14)	135

5.2.6	Results Experiment 2 (KIT)	142
5.3	Application in Control and Scheduling	146
5.4	Novelty and Remaining Questions	150
6	Summary and Outlook	153
A	Probabilistic Forecasting	157
A.1	Load Time Series Benchmark Generator	157
A.2	Calculation of an Empirical Quantile	160
A.3	Calculation of the Maximal and Minimal Non-Parametric Distribution Values	161
A.4	Approximation of the Continuous Ranked Probability Score	162
A.5	Experimental Results	163
B	Hierarchical Probabilistic Forecasting	171
B.1	Load Time Series Benchmark Generator	171
C	Application	173
C.1	KIT Dataset Preprocessing	173
C.2	GEFCom14 Training and Test Data	174
C.3	Methods for Identifying Night Values	174
C.4	Results Experiment 1	176
C.5	Results Experiment 2	177
C.5.1	Results Experiment 2 (GEFCom14)	177
C.5.2	Results Experiment 2 (KIT)	183

List of Symbols

Symbol	Description
$\hat{\cdot}$	Superscript denoting estimated values
$\tilde{\cdot}$	Superscript denoting the dependence of a value on the nearest neighbors quantile filter
$\hat{\sim}$	Superscript denoting an estimate of a value dependent on the nearest neighbors quantile filter
α	Smoothing parameter
α_i	Factors defining the influence that each component of the simulated time series has; with $i = 1, \dots, 5$
ϵ	Regression error
$\epsilon[\cdot]$	ARIMAX model residual value
ϵ_r	Regularization value
ϵ_{th}	Threshold defining the maximal distance that an input vector can have to \mathbf{x}_n for it to be considered a nearest neighbor
$\boldsymbol{\theta}$	Vector containing regression parameters
$\boldsymbol{\theta}_f$	Vector containing forecasting model parameters
$\boldsymbol{\theta}_{(q)}$	Vector containing quantile regression parameters
$\hat{\boldsymbol{\theta}}$	Vector containing estimated regression parameters
$\hat{\boldsymbol{\theta}}_{(q)}$	Vector containing estimated quantile regression parameters
$\tilde{\boldsymbol{\theta}}_{(q)}$	Vector containing parameters of a quantile regression based on the nearest neighbors quantile filter
$\hat{\tilde{\boldsymbol{\theta}}}_{(q)}$	Vector containing estimated parameters of a quantile regression based on the nearest neighbors quantile filter
ζ_i	Parameters defining $F_{\zeta_i}(y \mathbf{x})$
$\hat{\zeta}_i$	Parameters defining $F_{\hat{\zeta}_i}(y \mathbf{x})$
$\hat{\tilde{\zeta}}_i$	Parameters defining $F_{\hat{\tilde{\zeta}}_i}^{pref}(y \mathbf{x})$
ζ_{ij}	j^{th} element of ζ_i ; with $j = 1, \dots, N_p$
$\hat{\zeta}_{ij}$	j^{th} element of $\hat{\zeta}_i$; with $j = 1, \dots, N_p$

Symbol	Description
$\hat{\zeta}_j$	j^{th} element of $\hat{\zeta}$; with $j = 1, \dots, N_p$
a_i	ARIMAX model parameter; with $i = 0, \dots, H_1$
\mathbf{b}_j	ARIMAX model parameters; with $j = 0, \dots, H_1$
C	Computational effort
$C(\cdot)$	Copula function
c_l	ARIMAX model parameter; with $l = 0, \dots, H_1$
$D_w[k]$	Observation of a dummy variable time series used defining if a day is a weekday or not
$D_f[k]$	Observation of a dummy variable time series used defining if a day is a weekend or a holiday
$d(\cdot, \cdot)$	Distance function
$E(\cdot)$	Expected value of a random variable
$\hat{E}(y \mathbf{x})$	Conditional expected value obtained from $\hat{F}(y \mathbf{x})$
E_L	Energy content of the simulated time series
$F(\cdot)$	Cumulative distribution function
$F(y \mathbf{x})$	Conditional distribution of y given \mathbf{x}
$\hat{F}(y \mathbf{x})$	Estimated non-parametric conditional distribution of y given \mathbf{x}
$\hat{F}(y_a \mathbf{x})$	Estimated non-parametric distribution of y_a conditioned on \mathbf{x}
$\hat{F}(y_b \mathbf{x})$	Estimated non-parametric distribution of y_b conditioned on \mathbf{x}
$\hat{F}(y_{a+b} \mathbf{x})$	Estimated distribution of y_{a+b} conditioned on \mathbf{x}
$\hat{F}(y_a, y_b \mathbf{x})$	Estimated joint distribution of y_a and y_b conditioned on \mathbf{x}
$\hat{F}^{-1}(u \mathbf{x})$	Inverse function of $\hat{F}(y \mathbf{x})$
$F_{\hat{\zeta}_i}(y \mathbf{x})$	i^{th} parametric conditional distribution to be compared to $\hat{F}(y \mathbf{x})$; with $i = 1, \dots, N_{pd}$
$F_{\hat{\zeta}_i}(y \mathbf{x})$	i^{th} parametric conditional distribution to be compared to $\hat{F}(y \mathbf{x})$; with parameters estimated from $\hat{F}(y \mathbf{x})$ and $i = 1, \dots, N_{pd}$
$F_{\hat{\zeta}}^{\text{pref}}(y \mathbf{x})$	Parametric distribution best fitting $\hat{F}(y \mathbf{x})$
$\hat{F}_e(y \mathbf{x})$	Estimate of the cumulative distribution function of y given \mathbf{x} for $y \in (\hat{y}_{qL}, \bar{y}]$
$\hat{F}_e^{-1}(u \mathbf{x})$	Estimate of the inverse cumulative distribution function of y given \mathbf{x} for $u \in (q_L, 1]$
$\hat{F}_m(y \mathbf{x})$	Estimate of the cumulative distribution function of y given \mathbf{x} for $y \in (\hat{y}_{q1}, \hat{y}_{qL}]$

Symbol	Description
$\hat{\hat{F}}_m^{-1}(u \mathbf{x})$	Estimate of the inverse cumulative distribution function of y given \mathbf{x} for $u \in (q_1, q_L]$
$\hat{\hat{F}}_s(y \mathbf{x})$	Estimate of the cumulative distribution function of y given \mathbf{x} for $y \in [\underline{y}, \hat{\hat{y}}_{q_1}]$
$\hat{\hat{F}}_s^{-1}(u \mathbf{x})$	Estimate of the inverse cumulative distribution function of y given \mathbf{x} for $u \in [0, q_1]$
$f(\cdot)$	Probability density function
$f(\cdot, \boldsymbol{\theta})$	Parametric regression model
$f(\cdot, \boldsymbol{\theta}_f)$	Time series forecasting model
f_{clock}	Processor's clock rate
G	Solar irradiation
\hat{G}	Forecast solar irradiation
$\hat{G}_s[k]$	Observation of a time series containing forecast surface solar radiation
$\hat{G}_t[k]$	Observation of a time series containing forecast top net radiation
$\hat{G}_{\text{th}}[k]$	Observation of a time series containing forecast surface thermal radiation
$g_{ij}(\cdot)$	Function for calculating ζ_{ij} or $\hat{\zeta}_{ij}$
H	Forecast horizon
H_1	Number of time lags
H_s	Number of timesteps representing a season of a time series of interest
$I(\cdot)$	Indicator function
i	Index variable
J_n^{pref}	Nearest neighbors' index set
J_{np}	Index set of possible nearest neighbors
j	Index variable
K	Length of a finite time series
k	Timestep
L	Number of quantile regressions/interval forecasts created
l	Index variable
M	Number of dependent random variables
m	Index variable
N	Number of input and desired output observations in a training set
N_{cores}	Number of processing cores

Symbol	Description
N_{NN}	Number of nearest neighbors
N_P	Number of parameters defining a parametric distribution
N_{pd}	Number of parametric distributions
N_{prc}	Training data percentage
N_S	Number of scenarios
N_T	Number of input and desired output observations in a test set
n	Index variable
n_s	Number of seasons of a time series used to calculate $P_{\max}[k]$, $P_{\text{mean}}[k]$, and $P_{\min}[k]$
n_T	Minimal number of values in a segment in which the test set is divided
$P[k]$	Observation of a time series containing measured power/load values
$P_L[\cdot]$	Observation of a simulated load time series $\{P_{Li}[k]; k = 1, \dots, K\}$
$P_{Li}[\cdot]$	Observation of a simulated load time series $\{P_{Li}[k]; k = 1, \dots, K; i = 1, \dots, 3\}$
$P_{LS}[\cdot]$	Observation of the sum of time series $\{P_{Li}[k]; k = 1, \dots, K; i = 1, \dots, 3\}$
$P_{L,T}[\cdot]$	Trend component of $P_L[\cdot]$
$P_{L,P}[\cdot]$	Periodic component of $P_L[\cdot]$
$P_{L,Pa}[\cdot]$	Pattern component of $P_L[\cdot]$
$P_{L,S}[\cdot]$	Stochastic component of $P_L[\cdot]$
$P_{L,X}[\cdot]$	Exogenous component of $P_L[\cdot]$
$P_{\max}[k]$	Observation of a time series containing the maximal values obtained over several seasons of $\{P[k]; k = 1, \dots, K\}$
$P_{\text{mean}}[k]$	Observation of a time series containing the mean values obtained over several seasons of $\{P[k]; k = 1, \dots, K\}$
$P_{\min}[k]$	Observation of a time series containing the min values obtained over several seasons of $\{P[k]; k = 1, \dots, K\}$
p	Index variable
p_{opt}	Index of the largest J_{np}
Q_C	Average continuous ranked probability score
$Q_{C,n}$	Continuous ranked probability score of the n^{th} test set observation
$Q_{IRD,(q_u,q_l)i}$	Average reliability deviation of a single quantile estimate obtained on the i^{th} segment of a test set; with $i = 1, \dots, S_T$

Symbol	Description
Q_{IS}	Average interval score
$Q_{IS,(q_u,q_l)}$	Average interval score of a single interval forecast
Q_{IW}	Average interval width
$Q_{IW,(q_u,q_l)}$	Average interval width of a single interval forecast
Q_{MIRD}	Average modified interval reliability deviation
$Q_{MIRD,(q_u,q_l)}$	Average modified interval reliability deviation of a single interval forecast
Q_{MRD}	Average modified reliability deviation
$Q_{MRD,(q)}$	Average modified reliability deviation of a single quantile estimate
$Q_{MSE,i}$	Average mean squared error obtained by the i^{th} parametric distribution compared to $\hat{F}(y \mathbf{x})$; with $i = 1, \dots, N_{pd}$
Q_{PL}	Average pinball-loss
$Q_{PL,(q)}$	Average pinball-loss of a single quantile estimate
Q_{RD}	Average reliability deviation
$Q_{RD,(q)}$	Average reliability deviation of a single quantile estimate
$Q_{RD,(q)}i$	Average reliability deviation of a single quantile estimate obtained on the i^{th} segment of a test set; with $i = 1, \dots, S_T$
q	Probability of a random variable being lower or equal to its q -quantile; with $q \in (0, 1)$
q_l	Probability of a random variable being lower or equal to a quantile defining a lower bound of an interval with a probability $(q_u - q_l)$ of containing it
q_l	Probability of the l^{th} quantile used to obtain a non-parametric distribution forecast
q_{NN,n_i}	Probability associated to the i^{th} element of $\mathbf{y}_{NN,n}$ used during the calculation of the empirical quantiles; with $i = 1, \dots, N_{NN}$
q_u	Probability of a random variable being lower or equal to a quantile defining an upper bound of an interval with a probability $(q_u - q_l)$ of containing it
r_{prc}	Noise percentage
S	Number of features inside of an input vector \mathbf{x}
S_T	Number of segments in which a test set is divided
s	Index variable
T_a	Ambient temperature
\hat{T}_a	Forecast ambient temperature

Symbol	Description
t	Computation time
$\hat{\mathbf{u}}_{a b}$	Vector containing uniform distributed realizations obtained by applying the probability integral transform on estimated quantiles of y_a conditioned on y_b
$\hat{\mathbf{u}}_{b a}$	Vector containing uniform distributed realizations obtained by applying the probability integral transform on estimated quantiles of y_b conditioned on y_a
U	Uniformly distributed random variable
U_m	Uniformly distributed random variable dependent on other $M - 1$ uniformly distributed random variables; with $m = 1, \dots, M : M \geq 2$
u	Realization of U
u_{ij}	Realization of a uniformly distributed random variable used to obtain the j^{th} element of a the i^{th} scenario forming a scenario forecast; with $j = 1, \dots, H$ and $i = 1, \dots, N_S$
u_m	Realization of U_m
$\hat{u}_{a b,i}$	Element contained in $\hat{\mathbf{u}}_{a b}$; with $i = 1, \dots, 2L^2$
$\hat{u}_{b a,i}$	Element contained in $\hat{\mathbf{u}}_{b a}$; with $i = 1, \dots, 2L^2$
$\mathbf{u}[\cdot]$	Vector containing observations of various exogenous time series
$\text{Var}(\cdot)$	Variance
$\hat{\text{V}}\text{ar}(y \mathbf{x})$	Conditional variance obtained from $\hat{F}(y \mathbf{x})$
w_d	Wind direction
\hat{w}_d	Forecast wind direction
w_s	Wind speed
\hat{w}_s	Forecast wind speed
\mathbf{X}	Input matrix containing N observations of the input vector
\mathbf{x}	Input vector
\mathbf{x}_i	Input vector considered not to be a nearest neighbor of \mathbf{x}_n
\mathbf{x}_j	Input vector considered to be a nearest neighbor of \mathbf{x}_n
\mathbf{x}_n	n^{th} observation of a regression's input vector with $n = 1, \dots, N$
x_s	s^{th} feature of an input vector \mathbf{x} with $s = 1, \dots, S$
$x_{n,s}$	s^{th} feature of an input vector \mathbf{x}_n with $s = 1, \dots, S$
\mathbf{y}	Vector containing N observations of the desired output
$\mathbf{y}_{\text{NN},n}$	Vector containing the output values of the nearest neighbors of \mathbf{x}_n

Symbol	Description
$\tilde{Y}(q)$	Desired output vector obtained via the nearest neighbors quantile filter
$\hat{Y}(q_u, q_l)$	Interval forecast formed by \hat{y}_{q_u} and \hat{y}_{q_l}
$\hat{Y}_{S,i}$	Possible time series scenario forming a scenario forecast; with $i = 1, \dots, N_S$
y	Desired output
y_a	Desired output correlated to y_b
y_b	Desired output correlated to y_a
y_{a+b}	Sum of y_a and y_b
$y(q)$	q -quantile of a regression's desired output value y
\underline{y}	Lower threshold of y
\underline{y}_a	Lower threshold of y_a
\underline{y}_b	Lower threshold of y_b
\bar{y}	Upper threshold of y
\bar{y}_a	Upper threshold of y_a
\bar{y}_b	Upper threshold of y_b
\hat{y}	Estimate of a regression's desired output value
$\tilde{y}(q), n$	Empirical q -quantile of the values contained in $\mathbf{y}_{NN,n}$
$\hat{y}(q)$	Estimate of $y(q)$
$\hat{y}(q)$	Estimate of $y(q)$ given by a quantile regression based on the NNQF
$\hat{y}_a(q)$	Estimate of the q -quantile y_a
$\hat{y}_b(q)$	Estimate of the q -quantile y_b
$\hat{y}_{a b}(q)$	Estimate of the q -quantile y_a conditioned on y_b
$\hat{y}_{a b}(q)j$	Estimate of the q -quantile y_a conditioned on the j^{th} estimated quantile of y_b ; with $j = 1, \dots, L$
$\hat{y}_{b a}(q)$	Estimate of the q -quantile y_b conditioned on y_a
$\hat{y}_{b a}(q)j$	Estimate of the q -quantile y_b conditioned on the j^{th} estimated quantile of y_a ; with $j = 1, \dots, L$
$\hat{y}(q), n$	Estimate of the q -quantile of y_n
$y(q_1)$	Quantile defining the lower bound of an interval with a probability $(q_u - q_l)$ of containing a regression's desired output value y
$\hat{y}(q_1)$	Estimate of $y(q_1)$
$y(q_u)$	Quantile defining the upper bound of an interval with a probability $(q_u - q_l)$ of containing a regression's desired output value y

Symbol	Description
$\hat{y}_{(q_u)}$	Estimate of $y_{(q_u)}$
$\hat{y}_{(q_u),n}$	Estimate of the q_u -quantile of y_n
$\hat{y}_{(q_l),n}$	Estimate of the q_l -quantile of y_n
y_{\min}	Global minimum defining the lowest value a trained quantile regression may take
y_n	n^{th} observation of a regression's desired output value y with $n = 1, \dots, N$
y_{Smax}	Upper threshold not to be surpassed by a scenario
y_{Smin}	Lower threshold not to be surpassed by a scenario
$\hat{y}_{\text{S},ij}$	j^{th} element of a possible time series scenario forming a scenario forecast; with $j = 1 \dots, N_{\text{S}}$
$y[\cdot]$	Observation of a time series $\{y[k]; k = 1, \dots, K\}$
$y_{\text{a}}[\cdot]$	Observation of a time series $\{y_{\text{a}}[k]; k = 1, \dots, K\}$
$y_{\text{b}}[\cdot]$	Observation of a time series $\{y_{\text{b}}[k]; k = 1, \dots, K\}$
$y_{\text{T}}[\cdot]$	Trend-cycle component of a time series observation $y[\cdot]$
$y_{\text{S}}[\cdot]$	Seasonal component of a time series observation $y[\cdot]$
$y_{\text{R}}[\cdot]$	Remainder of a time series observation $y[\cdot]$
$\hat{y}[\cdot]$	Forecast value of a time series $\{y[k]; k = 1, \dots, K\}$
Z	Random variable
Z_m	Random variable with $m = 1, \dots, M; M \geq 2$ dependent on other $M - 1$ random variables
z	Realization of a random variable Z
z_m	Realization of a random variable Z_m
$z(q)$	q -quantile of a random variable Z
$z(q_l)$	Quantile defining the lower bound of an interval with a probability $(q_u - q_l)$ of containing the random variable
$z(q_u)$	Quantile defining the upper bound of an interval with a probability $(q_u - q_l)$ of containing the random variable

1 Introduction

1.1 Motivation

The increasing concern of damaging the planet through the continuous release of greenhouse gases into the atmosphere [1] calls for a thorough transformation of the current energy grids. This transformation is exemplified by the German “Energiewende” and its two main goals [2]: (i) increasing the share of renewables in the overall power consumption and (ii) reducing Germany’s primary energy demand. For instance, the share of renewables in the gross national electricity consumption (which was 31.7% in 2016) is planned to be increased to 80% by the year 2050 [3]. Nonetheless, increasing the overall share of renewables requires the additional consideration of both the heating and transportation sectors. For the sake of illustration, Figure 1.1 depicts historical and future percentage values of the final energy consumption of the heating and transport sectors, as well as the gross electricity consumption.

Some of the solutions to achieve the goals of the “Energiewende” include, but are not limited to the expansion of energy storage capacities, the market and system integration of renewables, the increase in energy efficiency of private households, and the use of alternative fuels (e.g., hydrogen) [2, 5]. Furthermore, the coupling of sectors (as e.g., the electrical, heating, and transportation sectors) also plays an important role in achieving the desired goals [6, 7]. The transformation of the energy grids, however, comes hand in hand with an increase in their complexity [8]. This complexity has motivated an increase in the integration of information and communication tech-

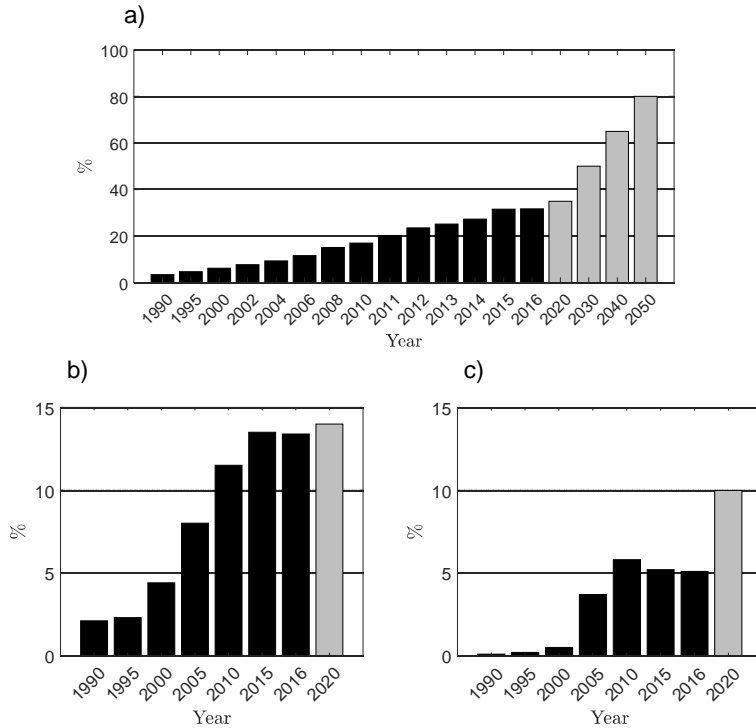


Figure 1.1: Historical (black bars) and future (gray bars) planned percentage of renewables in the German: a) gross electricity consumption; b) heating sector; c) transport sector. The figure is based on [3, 4].

nologies (ICTs), i.e. the development of the Smart Grid [9, 10], and a great number of energy research projects like: the Energy Lab 2.0¹ [10, 11], the Helmholtz Association Energy System 2050² and Storage and Cross-Linked Infrastructures projects³, and the Kopernikus projects⁴.

¹ elab2.kit.edu/english/index.php

² helmholtz.de/en/research/energy/energy_system_2050

³ helmholtz.de/en/research/energy/storage_and_cross_linked_infrastructures

⁴ kopernikus-projekte.de/en/home

A problem that arises when integrating volatile renewable power systems (e.g., wind and photovoltaic (PV) power plants) into the electrical grid is that they complicate the necessary balancing of electricity demand and supply, due to their intermittency [12, 13]. Thereupon, both demand and supply forecast have become necessary in enabling the correct planning and scheduling of the electrical grid [14] – for instance, via demand side management (DSM) [15, 16]. Furthermore, natural gas and district heating grids also benefit from load forecasts, as they allow the operators to ensure the grids’ stability and good performance [17, 18]. Moreover, conversion technologies like “Power to Gas” [19] and “Power to Heat” [20, 21] can be used to feed heating and gas grids with excess renewable energy. Such coupling increases the need for accurate volatile renewable power forecasts. Additionally, accurate load forecasts also assist energy utility companies (especially in competitive energy markets [22]) in decisions, like electricity generation and purchasing, as well as infrastructure planning [23, 24].

Time series forecasting models are a tool used for predicting the future of values whose change over time is of interest, for instance, sales, stock prices, and – most importantly in the present thesis context – renewable power generation and load [25, 26]. The majority of forecasting models deliver a time series value that according to their criteria is to be expected at a future moment in time [27], i.e. a point forecast, yet are unable to quantify the forecast uncertainty. As all forecasts are in some degree uncertain, the description of their uncertainty is necessary if the robustness of optimization and decision-making procedures is to be assured and increased [28]. Probabilistic forecasts offer a possibility of quantifying the uncertainty [28], for example, by offering a description of the future value’s cumulative distribution function or by providing intervals with a certain probability of the future value being within. Moreover, there is a current interest towards the development of energy-related probabilistic forecasting techniques [29], fueled by the increasing installation of volatile renewable power generation systems. For such reason, the present thesis presents a number of new data-

driven probabilistic forecasting approaches that can be utilized in an energy-related context, yet have at the same time a generalization capacity (i.e. they can be used to forecast other types of values).

The next sections in this chapter present theoretical background information as well as a description of the state-of-the-art in energy-related forecasting. Additionally, some open questions that the present contribution tries to address, as well as various objectives of the present thesis are also mentioned. Please notice that parts of the current chapter are an extended version of [30].

1.2 Theoretical Background

The present section briefly outlines the basic concepts that are necessary for a thorough understanding of this thesis. First, concepts of probability theory, as well as regression and quantile regression theory are introduced. Thereafter, information concerning time series forecasting is also presented.

1.2.1 Random Variables

A random variable, Z , is a function that maps the outcome of a chance experiment to a real number [31, 32, 33]. Random variables are mostly categorized as discrete or continuous depending on the values they are able to take. While the set of values that discrete random variables can take is finite or can be listed as an infinite sequence, the set of continuous random variables is an entire interval of numbers in a defined domain [31, 32, 33]; continuous random variables are the ones relevant in the present thesis.

A property of continuous random variables is the fact that the probability of them being equal to a single one of all their possible realizations, z , is always equal to zero, i.e. $\Pr(Z = z) = 0$ [31, 32]. In addition to moments like the expected value, $E(Z)$, or the variance, $\text{Var}(Z)$, other tools used for the description of continuous random variables are cumulative distribution

functions (CDFs) and their derivatives – assuming they exist –, probability density functions (PDFs). A CDF⁵, $F(z)$, is a monotonic non-decreasing function that maps a specific realization, z , to the probability of a random variable Z being lower or equal to z [34], i.e.

$$F(z) = \int_{-\infty}^z f(z^*) dz^* = \Pr(Z \leq z), \quad (1.1)$$

with $f(\cdot)$ as the random variable's PDF. Furthermore, values $z_{(q)}$ for which the following inequalities

$$\Pr(Z \leq z_{(q)}) \leq q \text{ and } \Pr(Z \geq z_{(q)}) \geq 1 - q; q \in (0, 1) \quad (1.2)$$

hold are defined as the q -quantiles of the random variable Z . Additionally, a quantile $z_{(q)}$ can also be defined as the value that solves the following optimization problem [35]:

$$z_{(q)} = \underset{z}{\operatorname{argmin}} \mathbb{E} \left(\begin{cases} (q-1) \cdot (Z-z) & , \text{if } Z < z \\ q \cdot (Z-z) & , \text{else} \end{cases} \right). \quad (1.3)$$

If Z has a continuous and strictly increasing CDF, Equation (1.2) reduces itself to

$$\Pr(Z \leq z_{(q)}) = q; q \in (0, 1). \quad (1.4)$$

In such cases an interval with a given probability of Z laying within can be defined using an upper and a lower quantile, i.e. $z_{(q_1)}$ and $z_{(q_u)}$. Such property can be attributed to the fact that the following equation holds:

$$\Pr(z_{(q_1)} \leq Z \leq z_{(q_u)}) = q_u - q_1; z_{(q_1)} < z_{(q_u)}. \quad (1.5)$$

⁵ CDF functions are normally written with their random variable as index, e.g., $F_Z(z)$. Nonetheless, for the sake simplicity, the use of the random variable as index is avoided in the present thesis.

Moreover, if the CDF of a continuous random variable Z is again continuous and strictly increasing, it can be used to transform a sample of a uniformly distributed random variable, $U \sim U(0, 1)$, into a sample of Z or vice versa [31]. Such transformation is called the probability integral transform (PIT) and it can be written as:

$$z = F^{-1}(u). \quad (1.6)$$

This transformation is mostly used for simulating samples of a given CDF.

CDFs can also be used to describe more than a single random variable [31, 32]. For example, the joint probability distribution of variables Z_1 and Z_2 can be written as

$$F(z_1, z_2) = \int_{-\infty}^{z_1} \int_{-\infty}^{z_2} f(z_1^*, z_2^*) dz_1^* dz_2^* = \Pr(Z_1 \leq z_1, Z_2 \leq z_2), \quad (1.7)$$

with $F(z_1) = F(z_1, \infty)$ and $F(z_2) = F(\infty, z_2)$ being referred to as the corresponding marginal CDFs. In addition, the variables Z_1 and Z_2 are said to be dependent, for example, if Z_2 – by taking a specific value – influences the possible outcomes of Z_1 . This dependence is captured by the conditional distribution function [32]:

$$F(z_1|z_2) = \Pr(Z_1 \leq z_1|Z_2 = z_2). \quad (1.8)$$

Furthermore, a common problem when looking at several dependent random variables is the estimation of their joint CDF. A possibility of solving this problem is the use of copulas. A Copula is a mathematical tool able to capture the dependence structure of random variables based only on their marginal CDFs [36, 37, 38]. The use of copulas is based on Sklar's theorem

that is defined for the case of $M \in \mathbb{N}_{>0}$ dependent random variables, Z_m (with $m = 1, \dots, M$ and $M \geq 2$), as

$$F(z_1, \dots, z_M) = C(u_1, \dots, u_M) = C(F(z_1), \dots, F(z_M)), \quad (1.9)$$

with C being the utilized copula. Equation (1.9) shows, that (i) the estimation of the dependence structure of random variables is based on Equation (1.6) and that (ii) a copula is a joint distribution function with uniform marginal CDFs. Moreover, Sklar's theorem can be extended to allow the determination of conditional joint CDFs [39] as shown for example, in the following equation

$$F(z_1, \dots, z_{M-1}|z_M) = C(F(z_1|z_M), \dots, F(z_{M-1}|z_M)|z_M). \quad (1.10)$$

Note:

In the remainder of the present thesis both random variables (e.g., Z) and their realizations (e.g., z) are going to be denoted using only lower case letters, thus they are not going to be differentiated. This notation is consistent with the one used in [35].

1.2.2 Quantile Regressions

Regression is a supervised learning approach [40] that is defined as the process of training – via a data mining technique – a data-driven model to estimate an output value y given an input vector $\mathbf{x} = [x_1, \dots, x_s, \dots, x_S]^T$ (with $s = 1, \dots, S$ and $S \in \mathbb{N}_{>0}$ being the number of used features) [41]. Regression models are divided into parametric and non-parametric, with the former being the ones relevant in the present thesis. Parametric regression models can be described using the equation,

$$y = f(\mathbf{x}; \boldsymbol{\theta}) + \epsilon, \quad (1.11)$$

in which $f(\cdot; \boldsymbol{\theta})$ represents the model, ϵ the error, and $\boldsymbol{\theta}$ the model parameters. Furthermore, both y and \mathbf{x} can also be described as random variables [42].

The obtainment of a regression model, just as every supervised learning approach, requires a training set consisting of $N \in \mathbb{N}_{>0}$ different observations [42]. These observations are comprised of N desired output values, y_n , and their corresponding input vectors $\mathbf{x}_n = [x_{n1}, \dots, x_{ns}, \dots, x_{nS}]^T$ (with $n = 1, \dots, N$), which are in turn contained inside of a desired output vector \mathbf{y} and an input matrix \mathbf{X} , i.e.:

$$\begin{aligned}\mathbf{y} &= [y_1, \dots, y_n, \dots, y_N]^T \\ \mathbf{X} &= [\mathbf{x}_1, \dots, \mathbf{x}_n, \dots, \mathbf{x}_N]^T.\end{aligned}\tag{1.12}$$

The goal of a parametric regression is to determine an optimal set of parameters, $\hat{\boldsymbol{\theta}}$, by minimizing the value of a cost function on the given training set [43]. Afterwards, the optimal parameters can be used to obtain an estimate of the desired output, i.e. \hat{y} :

$$\hat{y} = f(\mathbf{x}; \hat{\boldsymbol{\theta}}).\tag{1.13}$$

A normally used cost function (e.g., in case of a linear regression) is the sum of squared errors; its minimization results in \hat{y} being an estimate of the conditional expected value of y given the input \mathbf{x} , i.e. $\hat{y} = \hat{\mathbb{E}}(y|\mathbf{x})$ [35].

The type of regressions that are relevant in the present thesis are quantile regressions. A quantile regression estimates, instead of the conditional expected value, the conditional q -quantile of the desired output y , i.e. a value $y_{(q)}$ for which the following equation holds:

$$\Pr(y \leq y_{(q)} | \mathbf{x}) = q.\tag{1.14}$$

Quantile regressions are defined in the present thesis as:

$$\hat{y}_{(q)} = f(\mathbf{x}; \hat{\boldsymbol{\theta}}_{(q)}), \quad (1.15)$$

with $\hat{y}_{(q)}$ being the q -quantile's estimate and $\hat{\boldsymbol{\theta}}_{(q)}$ being the quantile regression estimated parameters. Furthermore, the parameters of a classical quantile regression are obtained by minimizing the sum of pinball-losses [44], i.e.:

$$\hat{\boldsymbol{\theta}}_{(q)} = \underset{\boldsymbol{\theta}_{(q)}}{\operatorname{argmin}} \sum_{n=1}^N \begin{cases} (q-1) \cdot (y_n - f(\mathbf{x}_n; \boldsymbol{\theta}_{(q)})) & , \text{ if } y_n < f_{(q)}(x_n; \boldsymbol{\theta}_{(q)}) \\ q \cdot (y_n - f(\mathbf{x}_n; \boldsymbol{\theta}_{(q)})) & , \text{ else} \end{cases}, \quad (1.16)$$

in which a pinball-loss is described by the function depicted in Figure 1.2. The fact that a quantile regression can be trained by minimizing the sum of pinball-losses is derived from Equation (1.3) [35].

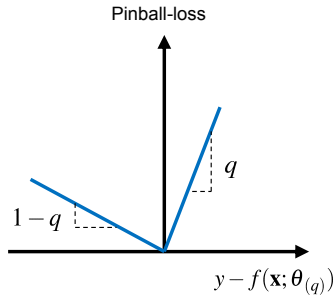


Figure 1.2: Pinball-loss function

In addition, pairs of quantile regressions can also be used to estimate intervals with a given conditional probability of containing y . For instance, quantile regressions that will form the interval's upper and lower bound, i.e. $y_{(q_u)}$ and $y_{(q_l)}$ respectively, can be used to estimate an interval with a probability $(q_u - q_l)$ of containing the future value (analogously as in Equa-

tion (1.5)). Likewise, the combination of several quantile regressions can also be used as an estimate of the conditional probability distribution of y , i.e. $F(y|\mathbf{x})$ [45]. It needs to be mentioned, however, that since quantile regressions are created independently of one another the problem of quantile crossing may exist [35]. Quantile crossing refers to the problem of a quantile regression estimating a lower quantile (e.g., 0.1) being greater than one estimating a greater one (e.g., 0.15), or vice versa. Since the quality of regression models is highly dependent on their training data, the use of methods as the ones described in [46] is recommended in cases in which the training data exhibits low quality.

1.2.3 Time Series Forecasting

A finite time series $\{y[k]; k = 1, \dots, K\}$ can be described as a set of $K \in \mathbb{N}_{>0}$ observations measured at equidistant points in time [47]. These observations form an orderly sequence in which the position of each observation is based on its corresponding timestep value k . A time series forecasting model uses some available information to estimate the unknown future of a desired time series at a forecast horizon $H \in \mathbb{N}_{>0}$. For example, a forecasting model able to estimate a future time series value using current and past values of the time series in question and from several exogenous time series can be given as:

$$\hat{y}[k + H] = f(y[k], \dots, y[k - H_1], \mathbf{u}^T[k], \dots, \mathbf{u}^T[k - H_1]; \boldsymbol{\theta}_f); k > H_1. \quad (1.17)$$

In the previous equation the vector $\boldsymbol{\theta}_f$ describes the forecasting model parameters, the vector $\mathbf{u}[k]$ contains the observations of various exogenous time series at timestep k , and the value $H_1 \in \mathbb{N}_0$ represents the number of used time lags.

Two of the most common time series forecasting techniques are exponential smoothing and the auto-regressive integrated moving average (ARIMA)

models [25]. A simple exponential smoothing model is given as a weighted average of past observations of the desired time series⁶, i.e.:

$$\hat{y}[k + H] = \begin{cases} y[1] & , \text{ if } k = 1 \\ \sum_{i=0}^{k-2} \alpha(1 - \alpha)^i y[k - i] + (1 - \alpha)^{k-1} y[1] & , \text{ else} \end{cases}, \quad (1.18)$$

with $\alpha \in [0, 1]$ being the smoothing parameter. As Equation (1.18) shows, the weights decay exponentially as the observations get older, therefore giving more recent observations a greater influence [14, 25]. Additionally, since the simple exponential smoothing in Equation (1.18) uses only past observations to conduct its forecast, it can also be defined as an auto-regressive time series model. More complex and non-linear exponential smoothing variants can be found in [48].

An autoregressive (AR) integrated (I) moving average (MA) model, i.e. ARIMA, is a generalization for non-stationary time series of the autoregressive moving average (ARMA) [47] model, which can be further divided into its auto-regressive (AR) and moving average part (MA). These models are based on the idea that time series are realizations of a stochastic process [49]. Moreover, the ARIMA model and all of its simplifications allow the use of exogenous time series as input; this inclusion is denoted by the

⁶ Equation (1.18) delivers accurate forecasts for $H > 1$ only if the forecast time series has no trend or seasonal component.

letter X at the end of their names, e.g., ARIMAX [50]. For example, an ARIMAX model of first order difference can be given as:

$$\begin{aligned} \hat{y}[k + H] = & \hat{y}[k + H - 1] + \sum_{i=0}^{H_1} a_i \Delta y[k - i] + \sum_{j=0}^{H_1} \mathbf{b}_j^T \mathbf{u}[k - j] \\ & + \sum_{l=0}^{H_1} c_l \epsilon[k - l]; k > H_1 + 1, \text{ with} \\ \epsilon[k] = & \hat{y}[k] - y[k]; \Delta y[k] = y[k] - y[k - 1], \end{aligned} \tag{1.19}$$

with a_i , \mathbf{b}_j and c_l representing the model parameters. It needs to be mentioned, that while Equation (1.19) defines the ARIMA(X) model for a generic forecast horizon H , it is traditionally used for $H = 1$ (i.e. $\hat{y}[k + H - 1] = y[k]$). In such case, forecasts for greater forecast horizons are obtained using the ARIMA(X) model iteratively and setting all unknown residuals (i.e. $\epsilon[k + 1]$, $\epsilon[k + 2]$, etc.) equal to zero, as exemplified in [25]. Interested readers are referred to [47] and [51] for additional information regarding ARIMA(X).

According to [25] time series can be decomposed into three distinct components: a trend-cycle component, a seasonal component, and a remainder. The way in which the various components form the original time series depends on the used assumption. For instance if an additive decomposition is assumed, an observation of a given time series can be described as:

$$y[k] = y_T[k] + y_S[k] + y_R[k]; \tag{1.20}$$

with $y_T[k]$ representing the trend-cycle component, $y_S[k]$ the seasonal component, and $y_R[k]$ the remainder. These components can be used to obtain models, like the Holt-Winters model [48] (i.e. an expansion of the traditional exponential smoothing technique that estimates a trend and a seasonality) or the seasonal ARIMA(X) model (SARIMA(X)).

Since linear models can be inadequate for some real-world applications [49], non-linear techniques, as e.g., artificial neural networks (ANN) [14] and support vector regressions (SVR) [52], have also been found useful in forecasting [53, 54]. Notice that models obtained by non-linear techniques can also be classified as NARIMA(X) models (with the letter N denoting their non-linearity), depending on their utilized input values [55].

Time series forecasting models can be in general separated into white-box, data-driven (i.e. black-box), and gray-box models [30]. The first use known relations, expert knowledge, etc. to determine the relation between the used inputs and the future of the time series of interest, the second try to estimate such relation by applying data mining techniques (e.g., linear regressions, ANNs), and the third are a combination of the previous two.

In the present thesis, forecasting models are going to be generalized as regressions, since they map – just as a regression – a given input to an estimate of a desired output. For example, to estimate Equation (1.17) using regression data mining techniques, both the input and desired output have to be defined as:

$$\begin{aligned} y &:= y[k + H] \\ \mathbf{x} &:= [y[k], \dots, y[k - H_1], \\ &\quad \mathbf{u}^T[k], \dots, \mathbf{u}^T[k - H_1]]^T; k \in [H_1 + 1, K - H]. \end{aligned} \quad (1.21)$$

Likewise, the forecasting model parameters (cf. Equation (1.17)) become then estimated regression parameters, i.e. $\boldsymbol{\theta}_f := \hat{\boldsymbol{\theta}}$.

Readers interested in other forecasting approaches, as e.g., state space models, autoregressive conditional heteroscedasticity (ARCH) models, generalized ARCH (GARCH) models, deep learning, Gaussian process regression, techniques using compressed sensing, etc., are referred to [49], [56], [57], [58], and [59].

In addition, forecasts describing the future of various time series and their aggregation at different aggregation levels, i.e. hierarchical forecasts, are

also relevant in the present thesis. Their importance is attributed to the fact that the aggregation of single time series (as e.g., load time series of individual households) results in values whose forecast is in some cases of interest (e.g., a substation's load time series). Hierarchical forecasts are mostly divided into bottom-up and top-down [60]. The former start by obtaining forecasts at the lowest aggregation level and then aggregating them accordingly, while the latter begin with a forecast for the highest aggregation level that is later distributed to the lower levels. Of course, there is also the possibility of independently forecasting each time series at every aggregation level. Nonetheless, such approach does not assure the obtainment of coherent hierarchical forecasts, i.e. that the forecast of time series at higher aggregation levels actually represent the sum of the ones at the lower levels.

1.2.4 Probabilistic Forecasting

The difficulty to perfectly predict the future makes every forecast inherently uncertain [61, 62]. Most time series forecasting models deliver for a given forecast horizon a single expected value (i.e. a point-forecast [63, 64]), but are unable to quantify their own forecast uncertainty. The quantification of this uncertainty is nonetheless of relevance in optimal decision-making [28] and in stochastic optimization [62].

Probabilistic forecasts offer a possibility of describing this uncertainty, for instance, by estimating the variance, the PDF, the CDF, or some quantiles of a future time series value [62]. Such methods are divided into parametric and non-parametric approaches; the former assume that the uncertainty follows a known distribution function – from which its defining parameters are to be determined –, while the latter do not. For the sake of illustration, Figure 1.3 depicts visual examples of various probabilistic forecasting methods described next, using the definitions given in Equation (1.21).

A Gaussian process [34, 43, 66, 67, 68, 69] is a clear example of a parametric probabilistic forecasting approach, since it assumes the forecast un-

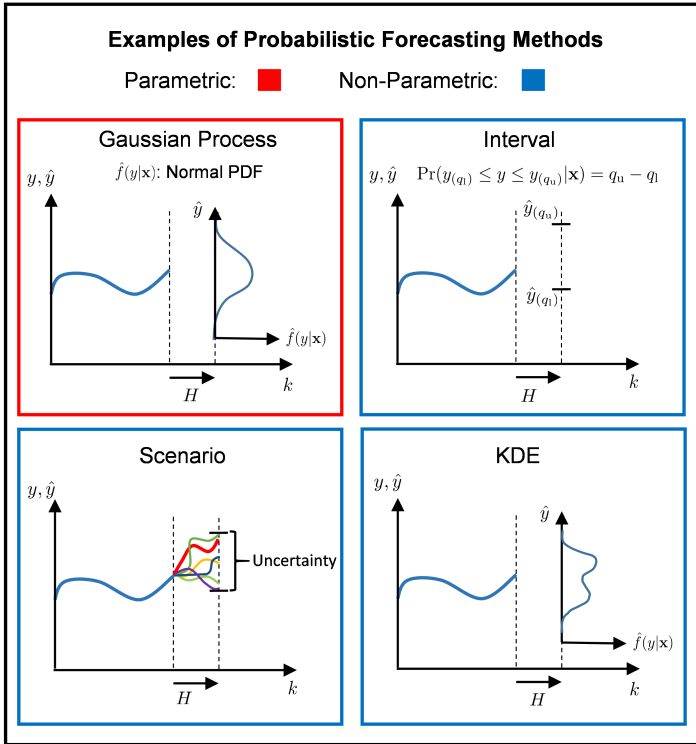


Figure 1.3: Examples of probabilistic forecasting methods (adapted from some figures in [65])

certainty to be normally distributed and hence estimates for each of its point forecasts (i.e. expected value) the corresponding variance. Using the Gaussian distribution assumption, the expected value, and the variance, a Gaussian process is able to fully describe the distribution of the forecast uncertainty. Examples of Gaussian processes used for time series forecasting can be found in [70] and [71].

Intervals created using pairs of either parametric or non-parametric quantile regressions are clear examples of non-parametric probabilistic fore-

casts [44, 72, 73, 74, 75, 76, 77]. These intervals are characterized by having a given probability of containing the future time series values. Additionally as previously stated, several quantile regressions can also be used to estimate the whole CDF of a future time series value conditioned on the models' input [45].

Kernel density estimators (KDE) offer another possibility of describing the forecast uncertainty by providing a smooth non-parametric estimation of the PDF of a future time series value, as e.g., in [78], [79], and [80].

Scenarios forecasts formed by combining several possible time series scenarios [81] are another approach for describing the forecast uncertainty. Scenario forecasts are traditionally used in weather forecasting, in which they are referred to as ensemble forecasts and are obtained using numerical weather models with varying initial conditions [82, 83]. It is important to mention that even though scenarios forecasts are commonly not considered probabilistic forecasts (as in [62]), the present thesis still classifies them as such. This classification is based on the following two reasons. First, the uncertainty of the forecast at a forecast horizon H can be represented – in a non-parametric fashion – by the spread of values that the different scenarios take. Second, the values of the different scenarios can be used to estimate probability values like, the mean, the variance, and some quantiles, which in turn can be used to further describe the uncertainty or to estimate a PDF or a CDF [84].

Currently there is an interest in describing the joint uncertainty of future values of two or more time series or of several future values of a single time series using information of their joint probability distribution. This may be relevant for instance, in the creation of scenario or coherent hierarchical probabilistic forecasts [85]. Unfortunately, the obtainment of the joint distributions is not a trivial task, as the various future values are not necessarily statistically independent.

1.3 State-of-the-Art in Energy-Related Forecasting

The present section describes the state-of-the-art in energy-related forecasting, for both load and volatile renewable power (i.e. wind and photovoltaic power). For the sake of illustration, Figure 1.4 depicts the state of maturity of various point and probabilistic energy forecasting research fields.

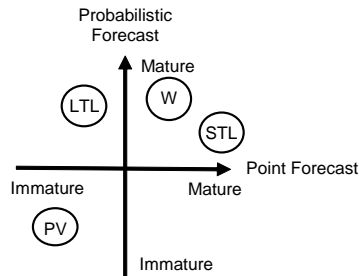


Figure 1.4: Maturity of point and probabilistic energy forecasting research fields; STL: short-term (two weeks ahead or shorter) load forecasting; LTL: long-term (a few months to a few decades ahead) load forecasting; PV: photovoltaic power forecasting; W: wind power forecasting (based on [29] and adapted from [30])

Energy time series are highly influenced by properties that differentiate them from many other time series; properties that have to be taken into account if accurate forecasts are to be obtained. A clear example thereof is the weather dependency of volatile renewable power, e.g., PV power depends mostly on solar irradiation [86], while wind power on wind speed [87]. The relation between weather and volatile renewable power is shown for instance, in the Osterwald equation [88, 89] (commonly utilized to estimate a PV cell's maximal possible power) and in the fundamental wind power equation [90]. Likewise, weather dependencies can also be observed in load time series, especially in heat load time series [91]. These can be attributed to human behavior that not only influences load depending on the weather, but also depending on the time of day and the day of the week. The effect

that human behavior has on load is visualized in Figure 1.5; it shows heat maps depicting normalized averages of households' daily electrical loads separated into months and weekdays.

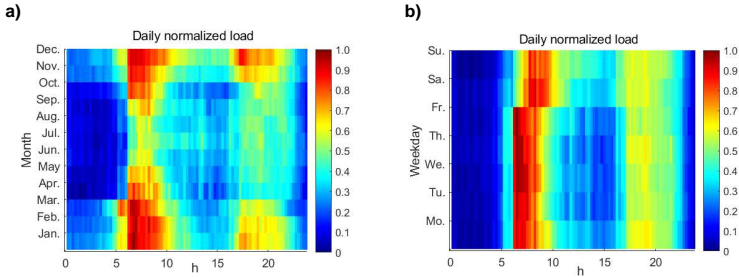


Figure 1.5: Heat maps depicting normalized average daily household electrical loads across a) several months and b) different weekdays. The data comes from the FIXED price group households in the Olympic Peninsula Project [92] (adapted from [30])

Figure 1.5 shows that electrical load time series possess seasonal components, similarly to PV power time series. Whereas the daily seasonality of PV power is governed by the Earth's day and night cycle, the seasons (i.e. the daily, weekly, and yearly seasons) of load are instead strongly dependent on people's behavior. Such seasonal components can be utilized, for instance, to obtain forecasts by using or averaging past observations [93, 94] or to obtain time series that, when used as further input values, may increase the models' accuracy [95]. Moreover, exogenous time series may help in estimating behaviors that repeat after non-fixed periods of time. For instance, averaging the power or load of previous days whose conditions are similar according to some exogenous time series values (e.g., weather, day of the week) may result in an acceptable forecast, just as in [80]. Apart from the previously described properties, the quality of the used time series (regarding missing values, outliers, etc.) needs also to be taken into account when creating forecasting models. For this reason, preprocessing methods able to test and improve the quality of the used time series are of major importance.

Since a thorough description of this type of approaches surpasses the scope of the present thesis, interested readers are referred to [3, 96, 97] for further information thereof.

In the literature, several categorizations of energy forecasting approaches exist. The most common one uses the forecast horizon to categorize forecasting approaches as short-, medium-, or long-term. Table 1.1 offers examples present in the literature of this common categorization⁷. As the examples show, the categorization based on forecast horizons is not consistent. Thereupon, the concepts of short-, medium-, and long-term are not utilized further, instead the forecast horizons are mentioned explicitly. Furthermore, Figure 1.6 shows examples of possible applications that forecasts can have based on their forecast horizons.

The selection of the approach and data utilized when conducting load forecasts is highly influenced by the desired forecast horizon. For instance, ANNs, SVMs, and time series models (e.g., ARIMA, regression models) are – according to [23] – normally used for forecast horizons of up to a day. In comparison, end-use models [113], econometric models [114], and their combination are commonly used for greater horizons. End-use models use extensive information about the users and their equipment to estimate the future load, while econometric models relate energy demand to macro-economic variables. As previously mentioned, human routine exerts a great influence on energy consumption, thereupon variables representing such behavior, as e.g., calendar functions [115], are frequently employed as input for the models. Likewise, the use of variables representing socio-economic growth should be considered when obtaining monthly and yearly load forecasts for regions whose socio-economic conditions could rapidly change [101]. Weather data, such as temperature, have also been found useful when obtaining load forecasts. Forecast weather data can be easily obtained from weather services for forecast horizons of up to a week. How-

⁷ Various works classify forecasting approaches also as very short-term. Nevertheless in the present thesis, such a categorization is considered part of the short-term category.

	Author	Forecast Horizon				
		$H \leq 1d$	$1d < H \leq 1w$	$1w < H \leq 1M$	$1M < H \leq 1Y$	$1Y < H$
Load	Ahmad [98]	Short		Medium		Long
	Feinberg [23]	Short		Medium		Long
	Hahn [99]	Short		Medium		Long
	MCSHarry [100]	Short		Medium	Medium & Long	Long
	Raza [54]	Short		-	Medium	Long
	Ghiassi [101]	Short		-	Medium	
	Alfares [102]	Short	Medium			Long
	Takiyar [24]	Short	Medium			Long
	Almashaie [103]	Short	Short & Medium	Medium		Long
	Metaxiotis [104]	Short	Medium	Medium & Long		Long
	Volatile Renewable Power	Foley [105]	Short	Short & Medium	-	
Monteiro [106]		Short	Short & Medium	-		
Wan [107]		Short & Medium		Long		
Soman [108]		Short & Medium	Long			
Wang [109]		Short	Long	-		
Huang [110]		Short		Long		-
Ogliari [111]		Short		Medium		Long
Kostylev [112]		OD		Medium		Long

Table 1.1: Categorization of energy-related forecasting approaches based on their forecast horizon H (adapted from [30])

Forecast Horizon: hour (h), day (d), week (w), month (M), year (Y)

ever, if greater forecast horizons are required, other methods of obtaining future weather data are necessary [116]. The fact that weather forecasts for great forecast horizons may be extremely uncertain is an important aspect to consider prior to their use. Furthermore, information about e.g., planned urban changes and expansions, dismantling, or maintenance of large industrial complexes can be helpful at increasing the accuracy of a given load forecasting model.

In contrast to load forecasting, volatile renewable power forecasting approaches can be divided into physical approaches, statistical (in the present thesis referred to as data-driven) approaches, and their combination [106, 117, 118, 119]. The models of the first approach correspond to white-box models, of the second to data-driven models, and of the last one to gray-box

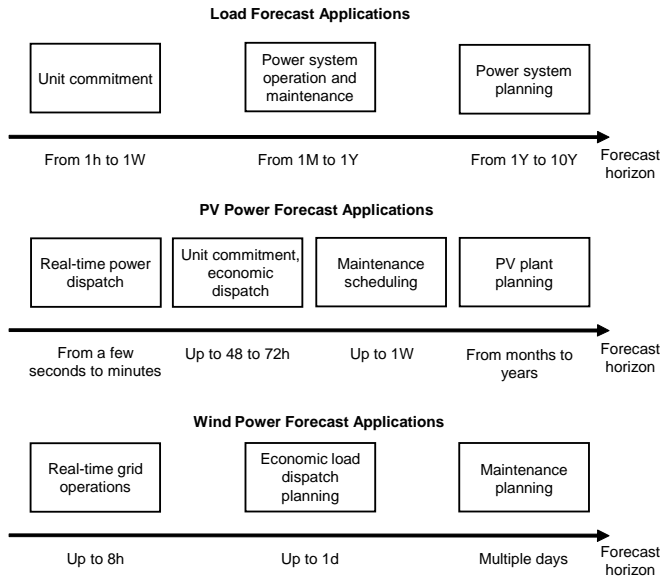


Figure 1.6: Examples of energy time series forecast applications based on their forecast horizon [54, 107, 109] (adapted from [30])

models. According to [120], pure data-driven forecasting approaches are considered to be more effective for short horizons, while approaches using more physical information are better in case of longer horizons. Nevertheless, a clear definition of short and long horizons is missing in [120].

Physical approaches start with some available forecast weather data, as e.g., numerical weather predictions (NWP) of solar irradiation in case of PV power [86] or of wind speed in case of wind power [87]. The forecast weather data is then transformed into values that directly influence the volatile renewable power generation (e.g., solar irradiation on the PV modules or wind speed at the wind turbines). These values are then converted into power forecast with the help of models based on physical relations, e.g., the fundamental wind power equation. An example of a physical approach

can be found in [121]. Since physical models do not require historical power generation data to work, they can be used to obtain volatile renewable power forecasts prior to the construction of a renewable energy system [117]. Nevertheless, it can be argued that physical models are not true forecasting techniques, as they only transform forecast weather data into other values.

In comparison, data-driven approaches (e.g., ANNs) train models able to directly obtain power forecasts from past generated power data and/or other possible inputs like historical NWP data [106, 109, 119]. Furthermore, data-driven models do not need to explicitly model specific properties of the renewable energy systems, since such specific properties are implicitly contained in the used data. Some examples of these properties are: local shadowing caused by neighboring buildings and/or vegetation in the case of PV power and the influence of the geographical location on wind power production.

The fact that the volatile renewable power and certain weather parameters are highly correlated have made weather data – specifically forecast weather data – a common used input. Nonetheless, it has been shown that accurate PV power forecasts can be obtained through pure autoregressive models, when considering short horizons [119]. Evidence thereof is shown for horizons of one and two hours in [94] and for horizons of 24 hours in [93] and [122]. Additionally, pure autoregressive models can also be used to forecast wind power generation [123], nonetheless it is recommended that NWP data is used as input for horizons greater than three to six hours [106]. Please note that autoregressive approaches for forecast horizons larger than a couple of hours come with quality restrictions, some of which are thoroughly outlined in [93].

Tables 1.2 and 1.3 contain examples of both point and probabilistic forecasting approaches found in the literature. The tables contain: the author of the respective article, the utilized technique, the forecast value (electrical load (E), heat load (H), gas load (G), photovoltaic power (PV), and wind

Author	Technique	Forecast Horizon	Forecast	Resolution	Input
Bacher [124]	Adaptive linear Time Series Model	1-42h	H	1h	$T_a, \hat{T}_a, w_s, \hat{w}_s, \hat{G}, \hat{C}$, Hour
Bouzerdoum [125]	Hybrid SARIMA SVM Model	1h	PV	1h	Autoregressive
Catalaou [123]	ANN, Wavelet Transform	3h	W	15min	Autoregressive
Chaouachi [126]	ANN Ensemble	1d	PV	1h	G, T_a , vapor pressure, humidity, cloud coverage, sunshine duration
Ding [127]	ARMAX, ANN, SVR	15, 30, 60min	E	15, 30, 60min	Autoregressive, activity information, hour, day
Dozauer [17]	Time Series Model	1w	H	1h	\hat{T}_a , social component
Duran [120]	AR and ARX	6,12,24h	W	1h	Autoregressive, \hat{w}_s
Fan [128]	Semi-Parametric Additive Model	1d	E	30min	Autoregressive, calendar function, T_a
Ghiassi [101]	Dynamic ANN	1, 3, 6, 12M	E	1M	Autoregressive, monthly cooling degree days
Gonzalez Ordiano [93]	Polynomial Regression, ANN	1d	PV	15min	Autoregressive, hour
Hong [53]	SVR with Immune Algorithm	1Y	E	1Y	Autoregressive
Idowu [91]	SVR	1, 3, 6, 12, 18, 24h	H	1h	Autoregressive, T_a, \hat{T}_a , hour, supply temperature, flow rate, supply and return temperature difference
Kandil [129]	Knowledge-Based Expert System	1Y	E	1Y	Knowledge base
Li [130]	ARMAX	1d	PV	1d	Autoregressive, daily average of \hat{T}_a , highest and lowest \hat{T}_a , \hat{w}_s, \hat{w}_d , forecast values of: dew temperature, precipitation amount, humidity, insolation duration, air pressure
Lin [131]	Evolutionary Seasonal Decomposition Least-Square SVR	1M	PV	1M	Autoregressive
Liu [132]	Least Squares SVM	1M	G	1d	Autoregressive, \hat{T}_a , calendar function
Mellit [133]	ANN Ensemble with Day-Type Classification	1d	PV	1h	Autoregressive, \hat{G}, \hat{T}_a
Senjyu [134]	Autoregressive ANN	1h	E	1h	Autoregressive, T_a
Stelmas [135]	ANNs with Fuzzy Logic	1-48h	W	1h	Autoregressive, \hat{w}_s, \hat{w}_d , hour

Author	Technique	Forecast	Forecast Horizon	Resolution	Input
Tiao [136]	NARX ANN	PV	1d	1h	Autoregressive, highest and lowest $\hat{T}_{a,t}$, dty-type index, Forecast Clear Sky Radiation
Taylor [137]	ANN with Ensemble Weather Predictions	E	1-10d	1d	Autoregressive, effective temperature, effective illumination, cooling influence of wind, calendar function
Yang [138]	SVR with Vector Quantization using Self-Organizing Maps	PV	1d	1h	Autoregressive, G , T_a , month, weather description, precipitation probability
Yu [18]	Genetic Algorithm and ANN	G	1d	1d	Autoregressive, maximum, minimum, and average T_a , future weather conditions, calendar function

Table 1.2: Examples of point forecasting approaches found in the literature (adapted from [30])

Forecast horizon & resolution: minutes (min), hour (h), day (d), week (w), month (M), year (Y)

Input description: Autoregressive: past load/power values, T_a : ambient temperature, $T_{a,t}$: forecast ambient temperature

w_s : wind speed, \hat{w}_s : forecast wind speed, w_d : forecast Wind Direction, \hat{w}_d : forecast wind direction,

G : solar irradiation, \hat{G} : forecast solar irradiation

Author	Technique	Forecast	Forecast Horizon	Resolution	Input
Bessa [139]	Time Adaptive Quantile-Copula	W	6-48h	1h	Autoregressive, \hat{w}_d, \hat{w}_s
Gaillard [73]	Quantile Regression with Generalized Additive Models	E	1M	1h	Time of year, day-type, T_a
Hahn [140]	KDE and Quantile Regression	E	1M	1h	Autoregressive, hour of the week, day of the year, T_a
Hong [141]	Multiple Linear Regression with Weather Ensemble	E	1Y	1M	T_a , monthly peak T_a , gross state product, month of the year, weekday, hour of the day, day of the year
Huang [142]	Gradient Boosting and k-Nearest-Neighbor Regression	PV	2-4h	1h	Autoregressive, \hat{G} , \hat{T}_a
Juban [79]	KDE	W	0-60h	3h	Autoregressive, \hat{w}_d, \hat{w}_s , horizon
Juban [44]	Quantile Regression with Non-Linear Feature Transformation	PV/W	1d	1h	Day of the week, and hour of the day, PV, \hat{T}_a , forecast values of: total column liquid and ice water, surface pressure relative humidity, total cloud cover, surface thermal and solar radiation downward, total precipitation, top net solar radiation, and azimuthal and meridional wind speed W: \hat{w}_d, \hat{w}_s , and forecast azimuthal and meridional wind speed,
Kou [143]	Online Model Selection and Warped Gaussian Process	W	15min-6h	15min	w_s, \hat{w}_s
Liu [144]	Quantile Regression Averaging on Sister Forecasts	E	1d	1h	Month of the year, day of the week, hour, T_a
Nagy [145]	Valid Ensemble of Quantile Regression Forest and Stacked Random Forest	PV/W	2-4h	1h	PV: hour of the day, day of the year, month, T_a , and forecast values of: total column liquid and ice water, relative humidity, total cloud cover, surface pressure, and solar radiation downward, total precipitation, top net solar radiation, and azimuthal and meridional wind speed W: \hat{w}_d, \hat{w}_s , and forecast azimuthal and meridional wind speed,
Nielsen [76]	Quantile Regression and Basis Functions	W	18-36h	NM	Power's point forecasts, horizon, forecast air density and friction velocity, \hat{w}_d, \hat{w}_s , number of hours, since 06Z
Pinson [81]	Ensemble Forecast and Adaptive Kernel Dressing	W	1-48h	1h	\hat{w}_d, \hat{w}_s (ensemble forecast)

Author	Technique	Forecast	Forecast Horizon	Resolution	Input
Pinson [146]	Parametric	W	10min	10min	Autoregressive
Sonjati [147]	Higher Order Markov Chains	PV	15min	15min	Autoregressive, G , T_a
Xie [148]	Forecast Combination and Residual Simulation	E	1M	1h	Autoregressive, day of the week, month, T_a
Zhang [80]	k-Nearest-Neighbor and KDE	PV	2-4h	1h	\hat{G} , relative humidity, total column ice water, total column liquid water, hour

Table 1.3: Examples of probabilistic forecasting approaches found in the literature

Forecast Horizon & Resolution: minutes (min), hour (h), day (d), week (w), month (M), year (Y)

Input Description: Autoregressive: past load/power values, T_a : ambient temperature, \hat{T}_a : forecast ambient temperature
 w_s : wind speed, \hat{w}_s : forecast wind speed, w_d : forecast Wind Direction, \hat{w}_d : forecast wind direction,
 G : solar irradiation, \hat{G} : forecast solar irradiation

power (W)), the forecast horizon, the temporal resolution in which the forecasts are given, and the used input data.

Further information regarding the state-of-the-art for both point and probabilistic forecasting can be found in the reviews given in [54], [102], [114], [149] regarding load (specifically electrical load forecasting), in [62], [87], [105], [106], [108], [109], [150] regarding wind power, and in [107], [112], [117], [119], [151] regarding PV power.

Since the forecast of renewable power generation and load is not the only one relevant in an energy-related context (e.g., forecast of energy prices is relevant for energy trading decisions [152]), interested readers are referred to [153] and [154] for information regarding energy price forecasting, to [155], [156], and [157] for an example on the influence that prices have on load time series forecasts, and to [158] for an approach that forecasts time series formed as a combination of both renewable power and load (for instance, a time series measured at a low voltage substation).

1.4 Open Questions

Regardless of the extensive amount of probabilistic forecasting related articles found in the literature, there are still a number of open questions that need to be addressed. For example:

- Quantile regressions are useful at obtaining non-parametric probabilistic forecasts. Nonetheless, the obtainment of classical parametric quantile regressions with more complex data mining techniques (e.g., ANN and SVR) is hindered by the required minimization of the non-differentiable sum of pinball-losses; the lack of differentiability may lead to problems when using gradient based optimization [159] and to a higher computational effort. Moreover, the effort of obtaining classical quantile regressions is further increased by the fact that the training algorithms of the used data mining techniques have to be

altered in order for them to minimize the sum of pinball-losses (cf. Equation (1.16)); as shown in [159, 160, 161].

- Non-parametric probabilistic forecasts have the advantage of avoiding assumptions regarding the distribution of the forecast uncertainty. However, there are still optimization algorithms that require a parametric description of the uncertainty (e.g., [162]). Therefore, methods able to transform non-parametric probabilistic forecasts (and their lack of assumptions) into parametric probabilistic forecasts are of interest.
- Scenario forecasts are required by some optimization algorithms (e.g., [163]). However, their creation commonly requires an explicit description of the correlation structure between neighboring time series values, which complicates their obtainment. Thereupon, the development of a method able to create scenario forecasts without the description of the correlation of neighboring values is of interest.
- Forecasts of joint distributions are necessary in the creation of coherent hierarchical probabilistic forecasts and in the description of the dependence of forecast neighboring time series values. Yet a non-parametric standard approach for their creation has not been established.
- Probabilistic forecasts are inherently more complex than point forecasts, for this reason the former are not as commonly used in an energy forecasting context. Therefore, there appears to be a lack of experience for deploying probabilistic forecasts in real-world applications.

1.5 Objectives

Based on the open questions outlined in the previous section, the main objectives of the present thesis are:

1. The development of a method able to efficiently create parametric quantile regressions without minimizing the sum of pinball-losses.
2. To derivation of an approach that based on a number of quantile regression is able to find the best corresponding parametric probabilistic forecast, thus avoiding the need of making a distribution assumption beforehand.
3. The creation of a heuristic that generates scenario forecasts without an explicit description of the correlation between neighboring time series values.
4. To development of a simple technique for the creation of non-parametric joint distribution forecasts for two correlated future values.
5. The derivation of a method for the obtainment of coherent hierarchical probabilistic forecasts for any number of aggregated time series by sequentially using joint distribution forecasts of only two values at a time.
6. The presentation of real-world use-cases for the current thesis probabilistic forecasting approaches.

Chapter 2 tackles the first three objectives by first, presenting an approach that based on a new nearest neighbors quantile filter is able to obtain quantile regressions. Chapter 2 then describes various methods that based on the acquired quantile regressions are able to obtain a series of probabilistic forecasts: interval forecasts, non-parametric distribution forecast, parametric distribution forecasts, and scenario forecasts. Analogously, Chapter 3

offers solutions for the fourth and fifth objective by presenting a concept for obtaining joint distribution forecasts of two correlated time series and a simple algorithm that uses those joint distributions to create coherent hierarchical forecasts. Additionally, Chapter 4 details the software implementation of all new developed methods and Chapter 5 shows the results obtained by applying the new developed methods on various real datasets, thus tackling the last objective. Lastly, Chapter 6 offers the conclusion and outlook of the present thesis.

2 Probabilistic Forecasting

2.1 Overview

The current chapter is divided in four main sections: the first thoroughly describes the probabilistic forecasting methods developed as part of the present thesis (Section 2.2), the second outlines the evaluation values used to assess the quality of forecasts given by the new methods (Section 2.3), the third presents a series of experiments that offer insight into certain properties of the described approaches (Section 2.4), and the fourth discusses the novelty of the presented approaches as well as some questions to be answered in future related works (Section 2.5).

Note that all examples and results presented in the current chapter are obtained using the open-source MATLAB toolbox SciXMiner [164] and various simulated load time series created using the same benchmark generator as in [3, 156]. More information regarding the implementation of the methods in SciXMiner and the benchmark generator can be found in Chapter 4 and Appendix A.1, respectively. Furthermore, it is assumed that the simulated time series do not present anomalies (i.e. missing values, outliers, etc.), hence no preprocessing methods are applied to them. This assumption is supported by the fact that the benchmark generator's options for adding anomalies to the time series are not used. Finally, please note that parts of the current chapter are extensions of the work presented in [65], [95], and [165].

2.2 Methods

The present section begins by describing a new method able to obtain quantile regressions without minimizing the sum of pinball-losses (cf. Equation (1.16)), i.e. the nearest neighbors quantile filter (NNQF). Thereafter, a description of four separate probabilistic forecasting approaches is given. These approaches create – based on quantile regressions and the NNQF – interval forecasts, non-parametric distribution forecasts, parametric distribution forecasts, or scenario forecasts. For the sake of clarity, Figure 2.1 depicts each of the described techniques and their dependence to one another.

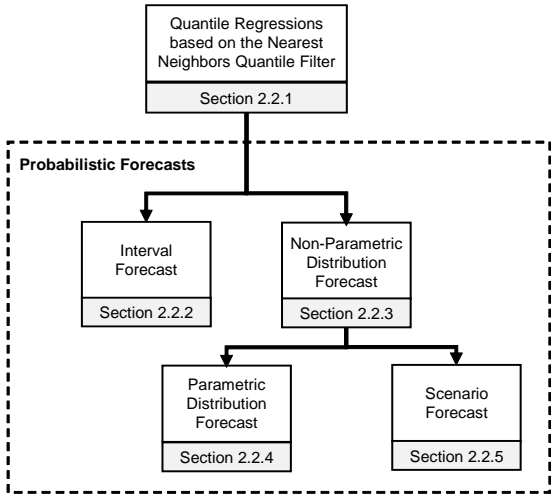


Figure 2.1: Techniques described in the present section and their relation to one another

Additionally, to allow a better understanding of the presented approaches, a series of examples are given across the whole section using a simulated

load time series, referred to as $\{P_L[k]; k = 1, \dots, K\}$. For more information about the obtainment of the simulated load time series please refer to Appendix A.1. Figure 2.2 depicts the simulated time series; as it can be seen, it represents one year of hourly load measurements (i.e. $K = 8760$).

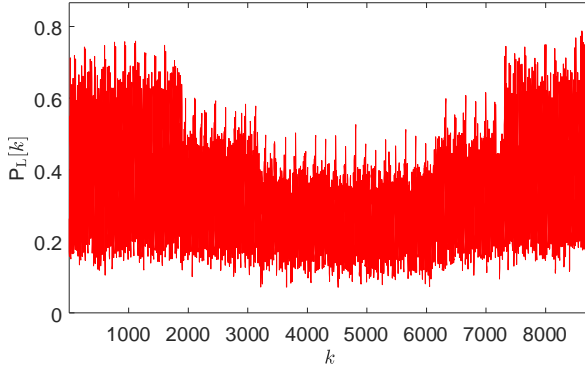


Figure 2.2: Simulated load time series used as an example in Section 2.2

2.2.1 Quantile Regressions based on the Nearest Neighbors Quantile Filter

The presented method begins with a training set composed of an input matrix \mathbf{X} and its corresponding desired output vector \mathbf{y} (cf. Equation (1.12)), for which the following assumptions are made:

- *The training set is large enough to accurately represent the uncertainty of y given \mathbf{x} .*
- *The conditional distribution of y given a specific \mathbf{x}_n is accurately represented by the output values of the nearest neighbors of \mathbf{x}_n .*

Under these assumptions, the new nearest neighbors quantile filter (NNQF) is able to modify the training set in such a way that a quantile regression model can be trained without Equation (1.16).

The NNQF starts by finding the nearest neighbors of \mathbf{x}_n . Such procedure consists in determining the nearest neighbors' index set $J_n^{\text{pref}} \subset \{1, \dots, N\}$ by solving the following optimization problem:

$$\begin{aligned}
 p_{\text{opt}} &= \underset{p}{\text{argmax}} \text{card}\{J_{np}\} \\
 \text{s.t. } J_{np} &: \max_{j \in J_{np}} \{d(\mathbf{x}_n, \mathbf{x}_j)\} \leq \min\{\min_{i \notin J_{np}} \{d(\mathbf{x}_n, \mathbf{x}_i)\}, \epsilon_{\text{th}}\} \\
 &\text{card}\{J_{np}\} \leq N_{\text{NN}} \\
 J_n^{\text{pref}} &= J_{np_{\text{opt}}}.
 \end{aligned} \tag{2.1}$$

In the previous equations $d(\cdot, \cdot)$ represents a distance function (defined by a respective distance measure¹ as, e.g., the Euclidean distance), $N_{\text{NN}} \in \mathbb{N}_{>0}$ is the number of searched nearest neighbors, and ϵ_{th} is a threshold defining the maximal distance that a nearest neighbor \mathbf{x}_j can have to \mathbf{x}_n . The process of finding J_n^{pref} outlined in Equation (2.1) can easily be described in three main steps:

1. Find index sets J_{np} for which the following constraints hold: the number of elements in J_{np} has to be lower than or equal to N_{NN} and the greatest distance of $\mathbf{x}_j; j \in J_{np}$ to \mathbf{x}_n cannot surpass the lowest distance of $\mathbf{x}_i; i \notin J_{np}$ to \mathbf{x}_n nor ϵ_{th} .
2. Determine the index p_{opt} , i.e. the index p of the J_{np} set with the greatest number of elements².
3. Define the index set $J_{np_{\text{opt}}}$ as the index set of the nearest neighbors J_n^{pref} .

¹ The present thesis uses the same distance measure as in [95], i.e. a weighted Euclidean distance with the inverse of the features variance as weights

² Finding more than one p_{opt} is avoided by implementing the NNQF to only find index sets J_{np} with indexes sorted by nearest neighbors' distance in an ascending order. Additionally, the implementation needs to consider only index sets J_{np} in which the indexes of nearest neighbors with the same distance are sorted in an ascending order.

Using J_n^{pref} the NNQF defines then a vector containing the nearest neighbors output values, i.e. $\mathbf{y}_{\text{NN},n} = \{y_n; n \in J_n^{\text{pref}}\}$, from which the empirical q -quantile $\tilde{y}_{(q),n}$ is calculated using a procedure given in Definition 5 of [166] and Method 10 of [167] (cf. Appendix A.2). Lastly, by repeating the procedure for all $n = 1, \dots, N$ the NNQF is able to define a vector

$$\tilde{\mathbf{y}}_{(q)} = [\tilde{y}_{(q),1}, \dots, \tilde{y}_{(q),N}]^T \quad (2.2)$$

that together with the original input matrix \mathbf{X} forms the modified training set.

The modified training set (i.e. \mathbf{X} and $\tilde{\mathbf{y}}_{(q)}$) can be used afterwards to train a regression model with a given data mining technique and its unmodified training algorithm (e.g., a linear regression trained with the least squares method). The obtained model is then capable of estimating the conditional empirical q -quantile defined by the used nearest neighbors and is described in the present thesis as

$$\hat{\tilde{y}}_{(q)} = f(\mathbf{x}; \hat{\boldsymbol{\theta}}_{(q)}). \quad (2.3)$$

In the previous equation, the tilde-superscript denotes that the regression is trained using $\tilde{\mathbf{y}}_{(q)}$. For the sake of illustration, Figure 2.3 depicts the previous described approach.

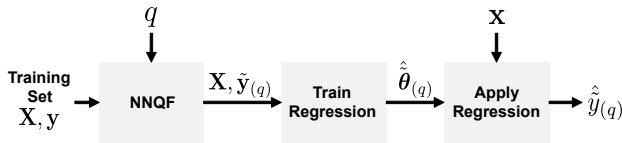


Figure 2.3: Estimation of a future value's quantile using a quantile regression based on the NNQF; 1. The training set is modified using the NNQF; 2. A regression model is trained using the modified training set; 3. The trained regression can be applied to estimate a quantile of a future value given an input vector \mathbf{x}

The use of the NNQF in the creation of quantile regressions possesses the following advantages: first, the creation of quantile regressions for every $q \in (0, 1)$ is possible, since q is defined as a free parameter of the NNQF. Secondly, the fact that the NNQF is decoupled from the actual training process allows quantile regressions to be obtained with any regression data mining technique (e.g., linear regression, ANN, SVR). Lastly, the necessity of saving the original training set and of conducting the nearest neighbors calculation during the application of the quantile regressions is eliminated, since the nearest neighbors are only used by the NNQF prior to the models' training.

For the sake of illustration, Example 1 uses the simulated load time series (cf. Figure 2.2) to depict the creation of quantile regressions based on the NNQF.

Example 1: Quantile Regressions based on the NNQF

The present example uses a simulated load time series (cf. Figure 2.2) to show the creation of quantile regressions that estimate the quantiles of a value 24 timesteps into the future based only on the current time series value. In other words, the desired output and the input of the created quantile regressions (cf. Equation (1.21)) are defined as:

$$\begin{aligned} y &:= P_L[k + H]; H = 24 \\ x &:= P_L[k]; \end{aligned} \tag{2.4}$$

notice that the input is in the present case one-dimensional and thus is not written in bold. The use of only the current time series value as input is to allow the upcoming depiction of the input and output values.

The training set obtained using the definitions given in Equation (2.4) is depicted in Figure 2.4:

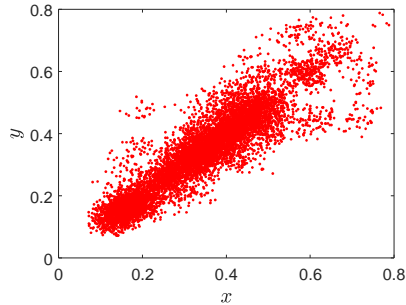


Figure 2.4: Example training set coming from the time series shown in Figure 2.2

Thereafter, the NNQF can be used to define a new training set (cf. Equation (2.2)) with which a quantile regression will be trained. For instance, Figure 2.5 depicts two training sets obtained by filtering the previously shown training data – with $N_{\text{NN}} = 50$ and ϵ_{th} set to infinity – for $q = 0.1$ and $q = 0.9$, respectively.

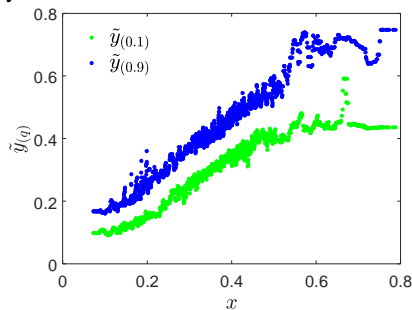


Figure 2.5: Example training set after its filtering with the NNQF

Finally, the filtered data can be used to obtain quantile regressions using a specific data mining technique. The data shown in Figure 2.5 appears to suggest that a non-linear regression model would be better at describing the relationship between input values and desired quantiles. Therefore, an ANN (traditional multi-layer perceptron) with six hidden neurons is selected as the data mining technique to be used. Figure 2.6 shows the two quantile regressions obtained.

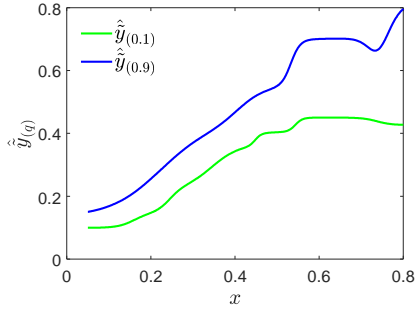


Figure 2.6: Quantile regressions based on the NNQF trained on the example

Furthermore, Figure 2.7 depicts forecasts given by the trained quantile regressions.

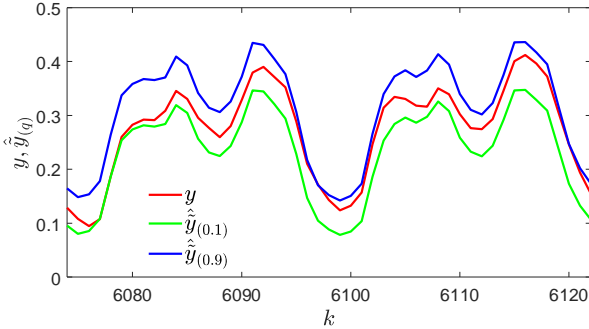


Figure 2.7: Time series obtained with the quantile regressions shown in Figure 2.6

Note that the implementation of the NNQF can be divided in two main procedures: the computationally expensive calculation of the nearest neighbors, including the determination of the vectors $\mathbf{y}_{NN,n}$ and the more computationally cheap calculation of the elements of $\tilde{\mathbf{y}}_{(q)}$. Therefore, an optimized implementation of the NNQF is recommended when creating several quantile regressions. This optimized version conducts the first procedure only once and uses the determined $\mathbf{y}_{NN,n}$ vectors to calculate the $\tilde{\mathbf{y}}_{(q)}$ vectors that are needed. The optimized implementation is the one used in the present thesis.

Finally, the presented approach raises the following questions:

- How accurate are quantile regressions based on the NNQF?
- What influence does the number of nearest neighbors have on the obtained quantile regressions?
- How do quantile regressions based on the NNQF fare in comparison to quantile regressions obtained by minimizing the sum of pinball-losses (cf. Equation (1.16))?
- What is the impact that the amount of training data has on the obtained quantile regressions?

To answer the previous questions, a series of experiments are conducted in Section 2.4.2.

2.2.2 Interval Forecast

Interval forecasts with a desired probability ($q_u - q_l$) of containing a future time series value can be obtained using quantile regressions based on the NNQF (cf. Section 2.2.1). The creation of an interval forecast begins with the obtainment of two quantile regressions that will form the upper and lower interval bounds, i.e. $\hat{y}_{(q_u)}$ and $\hat{y}_{(q_l)}$, respectively. Figure 2.8 depicts the estimation of $\hat{y}_{(q_u)}$ and $\hat{y}_{(q_l)}$. As it can be seen, the process consists in applying the procedure depicted in Figure 2.3 twice in order to train the quantile regressions defining the upper and lower bounds separately. Afterwards, using a given input vector \mathbf{x} , the regressions can be applied to estimate $\hat{y}_{(q_u)}$ and $\hat{y}_{(q_l)}$. Finally, the combination of both values results in the desired interval forecast that is described in the present thesis as:

$$\hat{\mathbf{y}}_{(q_u, q_l)} = [\hat{y}_{(q_u)}, \hat{y}_{(q_l)}]^T ; q_u > q_l . \quad (2.5)$$

As it is shown in Equation (2.3), interval forecasts are just the concatenation of quantile regressions; this means that their obtainment requires almost

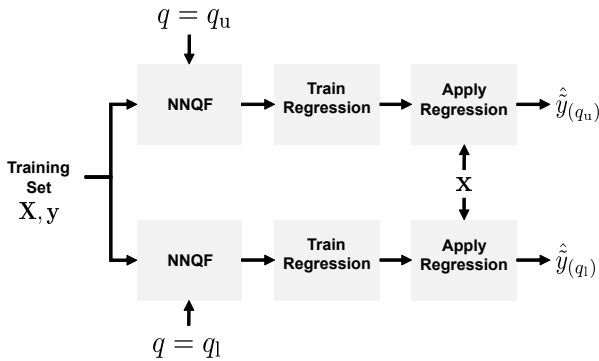


Figure 2.8: Obtainment of an interval forecast using quantile regressions based on the NNQF

no additional calculations if the quantile regressions have been previously computed.

Using the simulated load time series (cf. Figure 2.2), Example 2 shows the principles behind the obtainment of interval forecasts.

Example 2: Interval Forecast

The goal of the present example is to create interval forecasts that describe the uncertainty of a value 24 hours into the future of the simulated load time series (cf. Figure 2.2) using quantile regressions obtained with the NNQF (cf. Equation (2.1) with $N_{NN} = 50$ and ϵ_{th} set to infinity) and the following desired output, input vector, and probabilities:

$$\begin{aligned}
 y &:= P_L[k + H]; H = 24 \\
 \mathbf{x} &:= [P_L[k], \dots, P_L[k - H_1]]^T; k \in [H_1 + 1, K - H], H_1 = 168 \quad (2.6) \\
 q &\in \{0.10, 0.20, 0.30, 0.40, 0.60, 0.70, 0.80, 0.90\}.
 \end{aligned}$$

The previous equation shows that the quantile regressions estimate quantiles of a value 24 hours into the future using all load values of the previous week. The use of a different input vector as in Example 1 has two reasons: first, the

visualization of the quantile regressions in feature space is not required and second, the consideration of the weekly seasonality of the load time series is desired. Prior to the quantile regressions' training, a forward feature selection that chooses four relevant features with which all quantile regressions are to be trained is applied. In other words, all quantile regressions have the same features but vary in their fitted parameters (cf. Section 2.4). Additionally, the data mining technique used for training is a polynomial model with a maximal degree of two.

After training them, the quantile regressions are defined as either an upper or a lower bound. This distinction results in two sets of corresponding q_u and q_l values, i.e.:

$$\begin{aligned} q_u &\in \{0.6, 0.7, 0.8, 0.9\} \\ q_l &\in \{0.4, 0.3, 0.2, 0.1\}. \end{aligned} \quad (2.7)$$

Using the defined upper and lower bounds, four interval forecasts – centered on the median – with 20%, 40%, 60%, and 80% probability of the future value laying within can be created (cf. Equation (2.5)). For the sake of illustration, Figure 2.9 shows examples of the obtained interval forecasts.

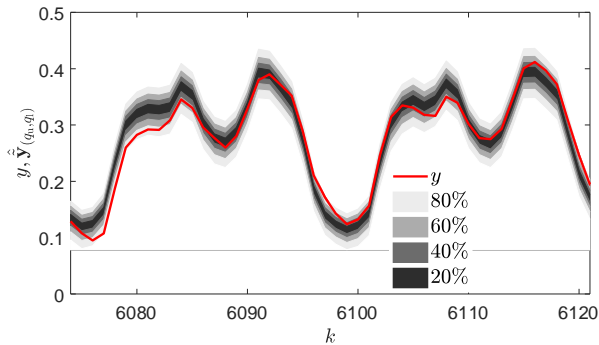


Figure 2.9: Example of interval forecasts

The obtainment of interval forecasts using quantile regressions based on the NNQF raises a series of questions that are examined with more detail in Section 2.4.3. These questions are:

- How accurate are interval forecasts formed by quantile regressions based on the NNQF?
- What influence does the number of nearest neighbors has on the obtained interval forecasts?
- What is the impact that the amount of training data has on the obtained interval forecasts?
- How do different forecast horizons influence interval forecasts?

2.2.3 Non-Parametric Distribution Forecast

Apart from interval forecasts, another type of probabilistic forecasts that can be derived from quantile regressions are non-parametric distribution forecasts, i.e. a non-parametric estimate of the conditional CDF of y given \mathbf{x} . These forecasts are obtained using quantile regressions based on the NNQF and the following assumptions:

- *The conditional CDF is strictly monotonic increasing.*
- *The conditional CDF equals zero for y values equal to or lower than a lower threshold \underline{y} .*
- *The conditional CDF equals one for y values equal to or greater than an upper threshold \bar{y} .*

Note that the definition of both thresholds may vary depending on the case. For instance, they can be set to be dependent of \mathbf{x} or to be always the same value. Furthermore, \underline{y} and \bar{y} can also be defined as minus and plus infinity. The specifics of how the thresholds are defined in the present thesis can be found in Appendix A.3.

Under these assumptions, the method begins with the obtainment of $L \in \mathbb{N}_{>0}$ separate quantile regressions $\{\hat{y}_{(q_l)}; l = 1, \dots, L \text{ and } q_1 < \dots < q_L\}$. Please notice that to fulfill the assumption regarding the CDF's monotony,

quantile crossing between the quantile regressions needs to be avoided, i.e. $\hat{y}_{(q_1)} < \dots < \hat{y}_{(q_L)}$. In the present thesis several measures to avoid quantile crossing are used. In the case of training multiple linear regressions (e.g., polynomial models), the risk of quantile crossing is reduced by training them via the following constrained least squares method, starting with the lowest quantile:

$$\begin{aligned} \underset{\tilde{\theta}_{(q_l)}}{\operatorname{argmin}} \quad & \sum_{n=1}^N (\tilde{y}_{(q_l),n} - f(\mathbf{x}_n; \tilde{\theta}_{(q_l)}))^2 \\ \text{s. t.} \quad & \begin{cases} f(\mathbf{x}; \tilde{\theta}_{(q_l)}) \geq y_{\min} & \text{if } q_l = q_1 \\ f(\mathbf{x}; \tilde{\theta}_{(q_l)}) \geq f(\mathbf{x}; \tilde{\theta}_{(q_{l-1})}) & \text{else ;} \end{cases} \end{aligned} \quad (2.8)$$

with y_{\min} representing a global minimum that should not be surpassed. Additionally, to further reduce the risk of quantile crossing, all quantile regressions, including the ones trained with non-linear regression techniques (e.g., ANNs), are subjected to the next constraint during their application:

$$\hat{y}_{(q_l)} = \begin{cases} \max(\hat{y}_{(q_l)}, y_{\min}) & \text{if } q_l = q_1 \\ \max(\hat{y}_{(q_l)}, \hat{y}_{(q_{l-1})}) + I(\hat{y}_{(q_l)} \leq \hat{y}_{(q_{l-1})}) \cdot \epsilon_r & \text{else .} \end{cases} \quad (2.9)$$

In the previous equation, ϵ_r is a regularization value that avoids neighboring quantiles to be equal and thus creates a set of quantiles that fulfill the desired monotony condition. Note that y_{\min} and ϵ_r are free parameters that can be defined differently depending on the application cases.

After obtaining the various quantile regressions, a first estimate of the CDF can be made by interpolating between the estimated quantiles. In the present thesis a linear interpolation is used, i.e.³:

$$\hat{F}_m(y|\mathbf{x}) = \sum_{l=2}^L \left(\frac{q_l - q_{l-1}}{\hat{y}_{(q_l)} - \hat{y}_{(q_{l-1})}} (y - \hat{y}_{(q_{l-1})}) + q_{l-1} \right) \cdot I(\hat{y}_{(q_{l-1})} < y \leq \hat{y}_{(q_l)}); \quad (2.10)$$

with $\hat{F}_m(y|\mathbf{x})$ representing the estimate of the conditional CDF for $y \in (\hat{y}_{(q_1)}, \hat{y}_{(q_L)}]$ and $I(\cdot)$ being an indicator function that is equal to one if its condition is fulfilled and equal to zero otherwise. Figure 2.10a shows the idea behind the previously described step.

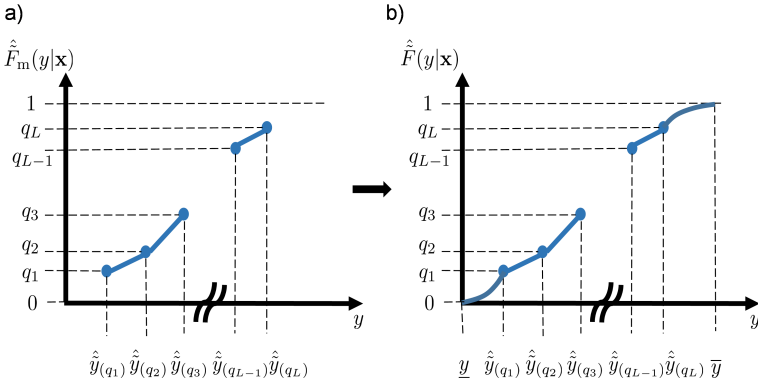


Figure 2.10: Principles behind the non-parametric distribution forecast (adapted from [65])

As shown by Figure 2.10a, Equation (2.10) cannot describe the behavior of the conditional CDF for y values lower than $\hat{y}_{(q_1)}$ and greater than $\hat{y}_{(q_L)}$. However, the assumptions made allow for the definition of functions $\hat{F}_s(y|\mathbf{x})$ and $\hat{F}_e(y|\mathbf{x})$ that combined with $\hat{F}_m(y|\mathbf{x})$ are able to estimate the

³ In other cases more complex interpolation approaches can be applied.

whole CDF. This combination is made possible by making sure that the functions fulfill the conditions shown in the next equation:

$$\begin{aligned} \hat{F}_s(\underline{y}|\mathbf{x}) = 0, \hat{F}_s(\hat{y}_{(q_1)}|\mathbf{x}) = q_1, \frac{d\hat{F}_s(y|\mathbf{x})}{dy} > 0 \text{ for } y \in (\underline{y}, \hat{y}_{(q_1)}) \\ \hat{F}_e(\bar{y}|\mathbf{x}) = 1, \hat{F}_e(\hat{y}_{(q_L)}|\mathbf{x}) = q_L, \frac{d\hat{F}_e(y|\mathbf{x})}{dy} > 0 \text{ for } y \in (\hat{y}_{(q_L)}, \bar{y}) \end{aligned} \quad (2.11)$$

Table 2.1 contains possible $\hat{F}_s(y|\mathbf{x})$ and $\hat{F}_e(y|\mathbf{x})$ functions. Please note, that the exponential functions in Table 2.1 are the ones selected to be used in the present thesis.

	Linear	Exponential
$\hat{F}_s(y \mathbf{x})$	$\frac{q_1}{\hat{y}_{(q_1)} - \underline{y}}(y - \underline{y})$	$\frac{q_1}{\exp(\hat{y}_{(q_1)}) - \exp(\underline{y})} \cdot (\exp(y) - \exp(\underline{y}))$
$\hat{F}_e(y \mathbf{x})$	$\frac{1 - q_L}{\bar{y} - \hat{y}_{(q_L)}}(y - \hat{y}_{(q_L)}) + q_L$	$1 - \frac{1 - q_L}{\exp(-\hat{y}_{(q_L)}) - \exp(-\bar{y})} \cdot (\exp(-y) - \exp(-\bar{y}))$

Table 2.1: Examples of starting and ending parts of the non-parametric CDF estimate; note that the linear functions only fulfill the conditions in cases in which the threshold values are finite

Finally, using Equation (2.10) and functions fulfilling the conditions outlined in Equation (2.11), the final estimate of the conditional CDF can be given as:

$$\begin{aligned} \hat{F}(y|\mathbf{x}) = \hat{F}_s(y|\mathbf{x}) \cdot I(\underline{y} \leq y \leq \hat{y}_{(q_1)}) + \hat{F}_m(y|\mathbf{x}) \\ + \hat{F}_e(y|\mathbf{x}) \cdot I(\hat{y}_{(q_L)} < y \leq \bar{y}) + I(y > \bar{y}) . \end{aligned} \quad (2.12)$$

Figure 2.10b depicts the principle of the previous equation. Additionally, Example 3 shows the obtainment of non-parametric distribution forecasts based on Equation (2.12).

Example 3: Non-Parametric Distribution Forecast

The present example shows the creation of non-parametric distribution forecasts for values 24 hours in the future of the simulated times series (cf. Figure 2.2). In order to do so, quantile regressions based on the NNQF for the following desired output, input vector, and probabilities are created:

$$\begin{aligned}
 y &:= P_L[k + H]; H = 24 \\
 \mathbf{x} &:= [P_L[k], \dots, P_L[k - H_1]]^T; k \in [H_1 + 1, K - H], H_1 = 168 \quad (2.13) \\
 q &\in \{0.01, 0.02, \dots, 0.99\}.
 \end{aligned}$$

As in Example 2, the quantile regressions estimate the values of the future load using the values of the previous week. Moreover, the same data mining technique as in Example 2 is also used.

Using the obtained quantile regressions as well as Equation (2.12), estimates of non-parametric CDFs of a future value can be made. For instance, Figure 2.11 depicts examples of non-parametric CDFs estimates obtained for three separate future time series values of the simulated load time series ($k = 6086, 6098, 6116$).

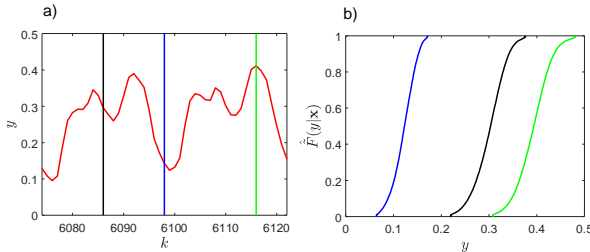


Figure 2.11: Examples of non-parametric distribution forecasts; the different colors of the vertical lines (a) and the CDFs (b) are used to make clear which CDF corresponds to which future value (adapted from [65])

The estimation of the whole conditional CDF allows for a better understanding of the forecast uncertainty. Likewise and most importantly in the present thesis, is the starting point in the estimation of the probabilistic forecasts outlined in Sections 2.2.4 and 2.2.5, as well as in Chapter 3.

A question regarding the presented method still remains, i.e.:

How well are the non-parametric distribution forecasts able to describe the future values' uncertainty?

Thereupon, experiments to answer the previous question are presented in Section 2.4.2 .

2.2.4 Parametric Distribution Forecast

As presented in Section 2.2.3, quantile regressions can be used to estimate a non-parametric conditional CDF of y given an input \mathbf{x} . While the non-parametric estimate may be advantageous in some cases, there are still optimization algorithms that require a parametric description of the forecast uncertainty (e.g., [162]). Therefore, the present section presents an approach able to find a parametric distribution that best fits Equation (2.12)'s non-parametric estimate. This method is based on the next assumption:

The available non-parametric estimate is the one that best approximates the true conditional CDF.

The goal of the present method – under the given assumption – is to find a parametric distribution that best resembles the given non-parametric estimate. In order to do so, the present approach compares $N_{\text{pd}} \in \mathbb{N}_{>0}$ parametric distributions $\{F_{\zeta_i}(y|\mathbf{x}); i = 1, \dots, N_{\text{pd}}\}$ to the non-parametric estimate. Note that in the present thesis, the type of parametric distributions used (e.g., Normal, Beta, Gamma) is given by its $N_{\text{p}} \in \mathbb{N}_{>0}$ defining parameters $\zeta_i = [\zeta_{i1}, \dots, \zeta_{iN_{\text{p}}}]^T$.

The method begins by estimating various central and/or non-central moments of the non-parametric conditional CDF (cf. Equation (2.12)), for instance the expected value and variance:

$$\begin{aligned} \hat{\mathbb{E}}(y|\mathbf{x}) &= \int_{\underline{y}}^{\hat{y}_{(q_1)}} \frac{d\hat{F}_s(y|\mathbf{x})}{dy} \cdot y \, dy + \int_{\hat{y}_{(q_1)}}^{\hat{y}_{(q_L)}} \frac{d\hat{F}_m(y|\mathbf{x})}{dy} \cdot y \, dy \\ &\quad + \int_{\hat{y}_{(q_L)}}^{\bar{y}} \frac{d\hat{F}_e(y|\mathbf{x})}{dy} \cdot y \, dy \end{aligned} \quad (2.14)$$

$$\begin{aligned} \hat{\mathbb{V}}\text{ar}(y|\mathbf{x}) &= \int_{\underline{y}}^{\hat{y}_{(q_1)}} \frac{d\hat{F}_s(y|\mathbf{x})}{dy} \cdot y^2 \, dy + \int_{\hat{y}_{(q_1)}}^{\hat{y}_{(q_L)}} \frac{d\hat{F}_m(y|\mathbf{x})}{dy} \cdot y^2 \, dy \\ &\quad + \int_{\hat{y}_{(q_L)}}^{\bar{y}} \frac{d\hat{F}_e(y|\mathbf{x})}{dy} \cdot y^2 \, dy - \hat{\mathbb{E}}(y|\mathbf{x})^2 \end{aligned} \quad (2.15)$$

Thereafter, the estimated moments, as well as the lower and upper threshold values – \underline{y} and \bar{y} – are used to calculate the parameters of the conditional parametric CDFs tested. These parameters are given as:

$$\hat{\zeta}_{ij} = g_{ij}(\hat{\mathbb{E}}(y|\mathbf{x}), \hat{\mathbb{V}}\text{ar}(y|\mathbf{x}), \dots, \underline{y}, \bar{y}). \quad (2.16)$$

In the previous equation, $g_{ij}(\cdot)$ describes a function that calculates ζ_{ij} using various moments, as well as \underline{y} and \bar{y} . Table 2.2 presents examples of functions $g_{ij}(\cdot)$ for four separate conditional parametric CDFs. Furthermore, the hat and tilde above ζ_{ij} in Equation (2.16) represent the fact that it is obtained using estimates based on the NNQF. Note that the present approach differentiates itself from the well-known method of moments [168] by estimating the moments not from a number of independent and identically distributed (i.i.d.) samples, but rather from the non-parametric CDF estimate. Likewise, the availability of a non-parametric CDF instead of several i.i.d. samples is the reason why the parameters are not estimated using the traditional maximum likelihood method [31]. Furthermore, the fact that the parameters are

not fitted, for example via a non-linear least squares approach, but instead are determined through straightforward calculations (cf. Equation (2.16)) stems mainly from the desire of keeping to a minimum the computational effort required.

		Normal	Uniform
		Name	Name
		$g_{ij}(\cdot)$	$g_{ij}(\cdot)$
ζ_{i1}	Mean	$E(y \mathbf{x})$	a $E(y \mathbf{x}) - \sqrt{3 \cdot \text{Var}(y \mathbf{x})}$
ζ_{i2}	Variance	$\text{Var}(y \mathbf{x})$	b $E(y \mathbf{x}) + \sqrt{3 \cdot \text{Var}(y \mathbf{x})}$
		Beta	Gamma
		Name	Name
		$g_{ij}(\cdot)$	$g_{ij}(\cdot)$
ζ_{i1}	Shape 1	$\frac{\zeta_{i3}^2 \cdot (1 - \zeta_{i3})}{\zeta_{i4}} - \zeta_{i3}$	Shape $\frac{\zeta_{i3}^2}{\text{Var}(y \mathbf{x})}$
ζ_{i2}	Shape 2	$\frac{\zeta_{i3} \cdot (1 - \zeta_{i3})^2}{\zeta_{i4}} - (1 - \zeta_{i3})$	Scale $\frac{\text{Var}(y \mathbf{x})}{\zeta_{i3}}$
ζ_{i3}	-	$E(y \mathbf{x}) \cdot I(\underline{y} \geq 0 \wedge \bar{y} \leq 1) + \frac{E(y \mathbf{x}) - \underline{y}}{\bar{y} - \underline{y}} \cdot I(\underline{y} < 0 \vee \bar{y} > 1)$	- $E(y \mathbf{x}) - \underline{y} \cdot I(\underline{y} < 0)$
ζ_{i4}	-	$\text{Var}(y \mathbf{x}) \cdot I(\underline{y} \geq 0 \wedge \bar{y} \leq 1) + \frac{\text{Var}(y \mathbf{x})}{(\bar{y} - \underline{y})^2} \cdot I(\underline{y} < 0 \vee \bar{y} > 1)$	- -

Table 2.2: Examples of how the parameters of four different parametric distributions are obtained.

Afterwards, each parametric CDF is compared to the non-parametric estimate using the mean squared error (MSE), defined as:

$$Q_{\text{MSE},i} = \frac{1}{L} \sum_{l=1}^L (\hat{F}(\hat{y}_{(q_l)}|\mathbf{x}) - F_{\hat{\zeta}_i}(\hat{y}_{(q_l)}|\mathbf{x}))^2; i = 1, \dots, N_{\text{pd}}, \quad (2.17)$$

where $Q_{\text{MSE},i}$ represent the MSE of the i^{th} parametric CDF tested. Finally, the parametric CDF with the lowest MSE is selected as the best fit, $F_{\hat{\zeta}}^{\text{pref}}(y|\mathbf{x})$ (with $\hat{\zeta} = [\hat{\zeta}_1, \dots, \hat{\zeta}_{N_p}]^T$). For the sake of illustration, Example 4 shows the obtainment of parametric distribution forecasts using the described approach.

Example 4: Parametric Distribution Forecast

The present example is based on the same non-parametric distribution forecasts presented in Example 3. Using the available non-parametric estimates and the method outlined in Equation (2.17), a parametric description of the uncertainty can be obtained. Furthermore, four different parametric CDFs are used for comparison, i.e. Beta, Gamma, Normal, and Uniform.

The three parametric CDFs that best fit the CDFs shown in Figure 2.11 (i.e. the ones for $k = 6086, 6098, 6116$) are depicted in Figure 2.12. Additionally, the type of the selected parametric CDFs, their corresponding parameters, and the squared root of their obtained MSE (i.e. RMSE) are contained in Table 2.3.

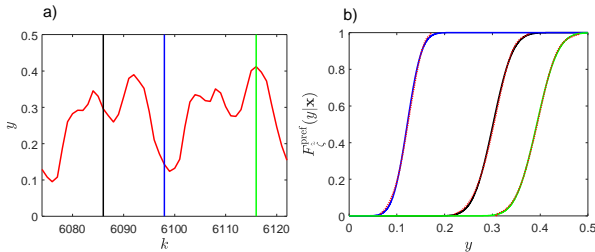


Figure 2.12: Examples of parametric distribution forecasts; the different colors of the vertical lines and the CDFs (a) are used to make clear which CDF (b) corresponds to which future value; the dotted red curves represent the non-parametric estimates being fitted (adapted from [65])

CDF Color	$F_{\hat{\xi}}^{\text{Pref}}(y \mathbf{x})$			RMSE [%]
	Type	$\hat{\xi}_1$	$\hat{\xi}_2$	
Black	Normal	0.3013	0.0327	1.38
Blue	Normal	0.1217	0.0239	1.61
Green	Normal	0.3929	0.0349	0.60

Table 2.3: Type and parameters of the parametric CDFs shown in Figure 2.12

As the low RMSE values and Figure 2.12 show, the present approach is able to find parametric CDFs that accurately fit the given non-parametric estimates.

There are two main reasons for which Equation (2.17) is based on the mean sum of squared errors and not on empirical distribution function statis-

tics (edf-statistic), like the Kolmogorov-Smirnov test [169, 170, 171]. First, edf-statistics assume that the non-parametric CDF is an empirical distribution function created from a number of i.i.d. samples, which is not the case, since the non-parametric CDF is based on Equation (2.12). Second, edf-statistics test – with a given significance level – if samples forming an empirical distribution function stem from a **single** given reference distribution. Therefore, they are considered inadequate for comparing the non-parametric CDF to **several** parametric candidates from which a best fitting one is to be chosen.

Lastly, finding the best fitting parametric CDF using the present approach, raises the following questions:

- Is the sum of squared errors of the parametric CDFs considered to be the best fit close to the optimal result (i.e. one obtained for instance, by CDFs with parameters using a non-linear least squares approach)?
- Do the parametric distribution forecasts describe the future values' uncertainty accurately compared to the non-parametric estimates with which they are obtained?

These questions are explored further in Section 2.4.4.

2.2.5 Scenario Forecast

As mentioned in Section 1.2.4, a scenario forecast is formed by a collection of time series scenarios; which are formed in turn by possible values that $y[k+1], \dots, y[k+H]$ may take. Important to consider when creating realistic time series scenarios is the temporal correlation between neighboring time series values; as for example, the estimate of $y[k+1]$ conditions the possible values that $y[k+2]$ may take. Since the temporal correlation is in most cases unknown, a joint distribution function needs to be estimated, which is not trivial. Therefore, the present section offers an alternative based on a

heuristic that uses quantile regressions and the probability integral transform (PIT, cf. Equation (1.6)).

The presented approach begins by training $L \in \mathbb{N}_{>0}$ separate quantile regressions $\{\hat{y}_{(q_l)}; l = 1, \dots, L \text{ and } q_1 < \dots < q_L\}$ with an input vector and a desired output (cf. Equation (1.21)) defined as:

$$\begin{aligned} y &:= y[k + 1] \\ \mathbf{x} &:= [y[k], \dots, y[k - H_1], \\ &\quad \mathbf{u}^T[k], \dots, \mathbf{u}^T[k - H_1]]^T; k \in [H_1 + 1, K - 1]. \end{aligned} \quad (2.18)$$

These definitions allow the quantile regression to implicitly consider the temporal correlation between neighboring values, since they estimate the quantiles of $y[k + 1]$ conditioned on an input vector that includes $y[k]$. It is important to mention that the following steps are conducted under the following assumption:

If exogenous time series are used as input, their values for the next H timesteps for which the scenarios are to be created are always available⁴.

Afterwards, the quantile regressions, as well as Equation (2.12) can be used to estimate the inverse of the conditional CDF, i.e. $\hat{F}^{-1}(u|\mathbf{x})$:

$$\begin{aligned} \hat{F}^{-1}(u|\mathbf{x}) &= \hat{F}_s^{-1}(u|\mathbf{x}) \cdot I(0 \leq u \leq q_1) + \hat{F}_m^{-1}(u|\mathbf{x}) \\ &\quad + \hat{F}_e^{-1}(u|\mathbf{x}) \cdot I(q_L < u \leq 1) \end{aligned} \quad (2.19)$$

with $u \sim U(0, 1)$ representing a random variable that is uniformly distributed between zero and one. Note that inputting a realization of u into $\hat{F}^{-1}(u|\mathbf{x})$ results in the obtainment of a possible realization of y conditioned on \mathbf{x} (cf. Equation (1.6)) [39]. This property is used to create the scenario forecast, as it is shown in the following paragraph.

⁴ The exogenous times series used may contain corresponding forecast values, as e.g., forecast ambient temperature and forecast solar irradiation.

After obtaining $\hat{F}^{-1}(u|\mathbf{x})$, a set of $N_s \in \mathbb{N}_{>0}$ vectors $\{\hat{\mathbf{y}}_{S,i}; i = 1, \dots, N_s\}$ is defined as the scenario forecast, with each vector $\hat{\mathbf{y}}_{S,i} = [\hat{y}_{S,i1}, \dots, \hat{y}_{S,iH}]^T$ representing a possible time series scenario for the next H timesteps. Hence, the entries $\hat{y}_{S,i,j}$ correspond to possible realizations of $y[k+j]$ and are given using the equation:

$$\hat{y}_{S,i,j} = \begin{cases} y_{S\max} & , \text{ if } \hat{F}^{-1}(u_{ij}|\mathbf{x}) > y_{S\max} \\ y_{S\min} & , \text{ if } \hat{F}^{-1}(u_{ij}|\mathbf{x}) < y_{S\min} \\ \hat{F}^{-1}(u_{ij}|\mathbf{x}) & , \text{ else} \end{cases} ; j = 1, \dots, H \quad (2.20)$$

$$\text{s.t. } y[k-p] := \hat{y}_{S,i(j-p-1)}; j > 1, p = 0, \dots, (j-2).$$

In the previous equation, $y_{S\max}$ and $y_{S\min}$ are upper and lower thresholds that should not be surpassed, while the values u_{ij} represent possible realizations of u , sampled independently of one another. In other words, the process of creating a single scenario consists in estimating the inverse CDF of a value a single timestep into the future. Then by inputting a realization u in the inverse CDF, a possible realization of the future value can be obtained. Thereafter, the estimated value is delayed and used as input to estimate the inverse CDF of another future value one timestep into the future. This process is repeated for the length of the desired scenario. Finally, a scenario forecast is obtained by repeating the procedure for creating a single scenario N_s times. For the sake of illustration, Figure 2.13 depicts the previously described process in four main steps, which are:

1. The creation of a non-parametric distribution forecast for a value one timestep into the future using an input vector containing the most recent observation and the application of the PIT to obtain a possible future value realization
2. The repetition of the first step defining the previously estimated future value as the latest available observation of the time series

3. The creation of a single scenario by repeating the second step until H values are estimated
4. The obtainment of a scenario forecast by repeating the three previous steps until the number of desired scenarios, i.e. N_S , have been calculated

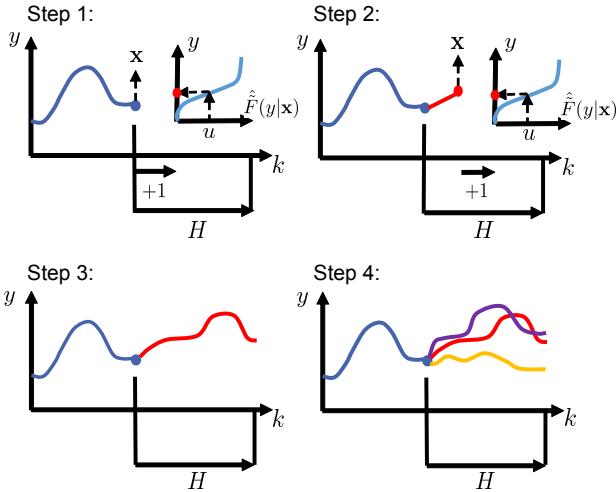


Figure 2.13: Steps for creating a scenario forecast (adapted from [65])

Finally, Example 5 presents the creation of a scenario forecast for the simulated load time series shown in Figure 2.2.

Example 5: Scenario Forecast

The goal of the present example is to estimate a scenario forecast with $H = 24$. In order to do so, several quantile regressions are created using the same

data mining technique as in Example 2 and the following desired output, input vector, and probabilities:

$$\begin{aligned}
 y &:= P_L[k + H]; H = 1 \\
 \mathbf{x} &:= [P_L[k], \dots, P_L[k - H_1]]^T; k \in [H_1 + 1, K - H], H_1 = 168 \quad (2.21) \\
 q &\in \{0.01, 0.02, \dots, 0.99\} .
 \end{aligned}$$

Thereafter, the inverse of the non-parametric CDF function can be obtained using Equation (2.19). Using the process outlined in Equation (2.20) scenario forecasts can be obtained. For instance, Figure 2.14 depicts a scenario forecast with $N_s = 100$, $H = 24$, $y_{S\max} = 1$, and $y_{S\min} = 0$.

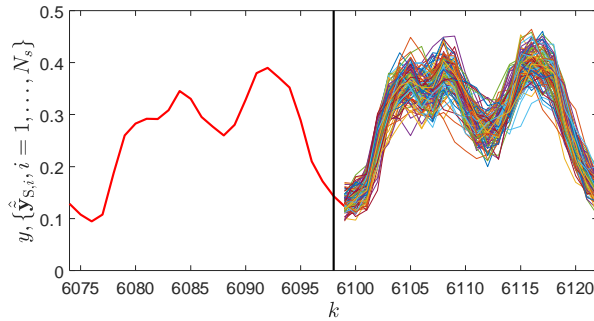


Figure 2.14: Scenario forecast example; the black vertical line represents the timestep in which the scenario forecasts is conducted, i.e. $k = 6099$ (adapted from [65])

The present method raises the next question:

Are the obtained scenario forecasts able to accurately describe the future values' uncertainty?

To offer answers to this question, Section 2.4.5 presents some experimental results.

2.3 Evaluation Values

The present thesis uses for most of the developed methods the same evaluation values, interval forecasts being the one exception. These values determine how well the methods estimate quantiles of the values being forecast, which reflects on the accuracy of their forecast uncertainty description. The quantile estimates used are for instance, the values of the quantile regressions based on the NNQF (cf. Section 2.2.1), the quantiles of the non-parametric distribution forecast (cf. Section 2.2.3), the quantiles of the parametric distribution forecasts (cf. Section 2.2.4), or the empirical quantiles of the values forming a scenario forecast at a specific point in time (cf. Section 2.2.5). These quantile estimates are then evaluated on a test set comprised of $N_T \in \mathbb{N}_{>0}$ input vectors and their corresponding desired outputs, i.e. \mathbf{x}_n and y_n with $n = 1, \dots, N_T$. Just as in Equation (1.21), y_n represents the time series values to be forecast.

As already mentioned, quantile estimates of the desired outputs, i.e. $\hat{y}_{(q),n}$, can be obtained with the methods developed as part of the present thesis⁵. Afterwards, the accuracy of these quantile estimates can be measured using several values. For instance, the average pinball-loss $Q_{\text{PL},(q)}$ is one of the values used in the present thesis. It is an error measure that describes how large is the deviation of the estimates from the true quantiles and is given as:

$$Q_{\text{PL},(q)} = \frac{1}{N_T} \sum_{n=1}^{N_T} \begin{cases} (q-1) \cdot (y_n - \hat{y}_{(q),n}) & , \text{ if } y_n \leq \hat{y}_{(q),n} \\ q \cdot (y_n - \hat{y}_{(q),n}) & , \text{ else .} \end{cases} \quad (2.22)$$

Another value that is used in the present work is the reliability deviation $Q_{\text{RD},(q)}$, which is based on the following: if an estimate of a quantile with a probability q of being greater or equal to a desired output is given, then

⁵ Please note that the tilde superscript is again used to denote the fact that all developed methods are based on quantile regressions obtained via the NNQF.

the difference between the percentage of desired outputs that are actually lower or equal to that estimate and q should be close to zero. The reliability deviation is defined as:

$$Q_{RD,(q)} = \frac{1}{N_T} \sum_{n=1}^{N_T} I(y_n \leq \hat{y}_{(q),n}) - q. \quad (2.23)$$

A great disadvantage of the reliability deviation is that it may consider a trivial quantile regression (i.e. a model that estimates the same value regardless of the input vector used) to be perfect, even though that may not be necessarily the case; this effect is later shown in Section 2.4.2 and Figure 2.17. Therefore, a modified version of the reliability deviation, $Q_{MRD,(q)}$, is also used in the present thesis. The value $Q_{MRD,(q)}$ is an average of the absolute values of reliability deviations calculated separately on $S_T \in \mathbb{N}_{>0}$ segments of the used test set, i.e.:

$$Q_{MRD,(q)} = \frac{1}{S_T} \sum_{i=1}^{S_T} |Q_{RD,(q)i}|, \text{ with}$$

$$Q_{RD,(q)i} = \frac{\sum_{n=1+\{(i-1) \cdot n_T\}}^{\max\{i \cdot n_T, N_T \cdot I(i=S_T)\}} I(y_n \leq \hat{y}_{(q),n})}{\max\{i \cdot n_T, N_T \cdot I(i=S_T)\} - (i-1) \cdot n_T} - q, \text{ and}$$

$$n_T = \text{floor}(N_T \cdot S_T^{-1}); \quad (2.24)$$

In the previous equation, $Q_{RD,(q)i}$ represents the reliability deviation of the i^{th} segment, n_T is the minimal number of values in a given segment, $\text{floor}(\cdot)$ is a function that rounds its input to its lowest closer integer, $I(\cdot)$ is an indicator function, and S_T is a free parameter representing the number of segments to be tested. In Section 2.4, $S_T = 10$ is used.

Evaluating a single quantile estimate is not enough to assess the possibility of obtaining quantile regressions with a given data mining technique nor to determine the quality of the forecast uncertainty described as a non-parametric CDF, a parametric CDF, or a scenario forecast. Therefore, the

averages of the values of $Q_{\text{PL},(q)}$, the absolute values of $Q_{\text{RD},(q)}$, and the values of $Q_{\text{MRD},(q)}$ obtained across $L \in \mathbb{N}_{>0}$ different quantile estimates are also calculated; in other words:

$$Q_{\text{PL}} = \frac{1}{L} \sum_{l=1}^L Q_{\text{PL},(q_l)}, \quad (2.25)$$

$$Q_{\text{RD}} = \frac{1}{L} \sum_{l=1}^L |Q_{\text{RD},(q_l)}|. \quad (2.26)$$

$$Q_{\text{MRD}} = \frac{1}{L} \sum_{l=1}^L Q_{\text{MRD},(q_l)}. \quad (2.27)$$

In the previous equations, q_l represents the probability corresponding to the l^{th} quantile estimate, while Q_{PL} , Q_{RD} , and Q_{MRD} are the average pinball-loss⁶, average reliability deviation, and average modified reliability deviation, respectively. Since all average values obtained represent a mean deviation from an optimum, the closer they are to zero the better the quantile regressions and the forecasts are.

In the case of evaluating interval forecasts other values have to be used; values that evaluate the intervals formed by pairs of quantile estimates (cf. Equation (2.5)) and not only the quantile estimates themselves. Therefore, the present thesis uses the following values to assess the quality of an interval forecast. The first value is the interval width Q_{IW} , which is given as follows:

$$Q_{\text{IW},(q_u, q_l)} = \frac{1}{N_{\text{T}}} \sum_{n=1}^{N_{\text{T}}} (\hat{y}_{(q_u),n} - \hat{y}_{(q_l),n}); \quad (2.28)$$

⁶ The non-parametric and parametric distribution forecasts are evaluated with Equation (2.25) and not with the more traditional continuous ranked probability score (CRPS) [28], since the former is related to the latter [29] and even approximates it if the results of several quantiles are used (as shown in Appendix A.4).

with $\hat{y}_{(q_u),n}$ and $\hat{y}_{(q_l),n}$ representing the quantile estimates forming the upper and lower interval bounds (cf. Equation (2.5)) of the given interval forecast for the n^{th} desired output. Since broad intervals are undesired, the lower $Q_{IW,(q_u,q_l)}$ is, the better the interval is considered. Using Equation (2.28) a value taken from a definition in [28] and referred to in the present work as the interval score $Q_{IS,(q_u,q_l)}$ can then be calculated. This value is given by the next equation:

$$Q_{IS,(q_u,q_l)} = \frac{2}{N_T \cdot (1 - (q_u - q_l))} \sum_{n=1}^{N_T} \begin{cases} (y_n - \hat{y}_{(q_u),n}) & , \text{ if } y_n > \hat{y}_{(q_u),n} \\ (\hat{y}_{(q_l),n} - y_n) & , \text{ if } y_n < \hat{y}_{(q_l),n} \\ 0 & , \text{ else} \end{cases} + Q_{IW,(q_u,q_l)} \cdot \quad (2.29)$$

As it is shown in Equation (2.29), $Q_{IS,(q_u,q_l)}$ considers not only the deviations outside the given interval, but also its width. Finally, the last value used is referred to as the modified interval reliability deviation $Q_{MIRD,(q_u,q_l)}$ and it is utilized to determine if an interval forecast with a desired probability $(q_u - q_l)$ of containing the future time series values actually fulfills that goal. Similarly as in Equation (2.24), the modified interval reliability deviation is given as the average of the absolute values of interval reliability deviations $Q_{IRD,(q_u,q_l),i}$; $i \in [1, S_T]$ obtained on S_T segments of the used test set, i.e.:

$$Q_{MIRD,(q_u,q_l)} = \frac{1}{S_T} \sum_{i=1}^{S_T} |Q_{IRD,(q_u,q_l),i}|, \text{ with} \\ Q_{IRD,(q_u,q_l),i} = \frac{\sum_{n=1+(i-1) \cdot n_T}^{\max\{i \cdot n_T, N_T \cdot I(i=S_T)\}} I(\hat{y}_{(q_l),n} < y_n \leq \hat{y}_{(q_u),n})}{\max\{i \cdot n_T, N_T \cdot I(i=S_T)\} - (i-1) \cdot n_T} - (q_u - q_l); \quad (2.30)$$

with n_T defined as in Equation (2.24).

The capability of a data mining technique for creating accurate interval forecasts cannot be determined by looking at a single interval. Therefore, the evaluation of a data mining technique should consist in averaging the $Q_{IW,(q_u,q_l)}$, $Q_{IS,(q_u,q_l)}$, and $Q_{MIRD,(q_u,q_l)}$ values for $L \in \mathbb{N}_{>0}$ different interval forecasts, i.e.:

$$Q_{IW} = \frac{1}{L} \sum_{j=1}^L Q_{IW,(q_{u,j},q_{l,j})}, \quad (2.31)$$

$$Q_{IS} = \frac{1}{L} \sum_{j=1}^L Q_{IS,(q_{u,j},q_{l,j})}, \quad (2.32)$$

$$Q_{MIRD} = \frac{1}{L} \sum_{j=1}^L Q_{MIRD,(q_{u,j},q_{l,j})}; \quad (2.33)$$

with $q_{u,j}$ and $q_{l,j}$ representing the probabilities of the upper and lower bounds of the j^{th} interval tested. Furthermore, Q_{IW} is the average interval width, Q_{IS} is the average interval score, and Q_{MIRD} is the average modified interval reliability deviation. Just as before, the closer these average values are to zero, the better the interval forecasts are considered.

2.4 Experiments

The current section presents a series of experimental results that outline the properties and capabilities of the probabilistic forecasting methods described in Section 2.2. The present section begins with a description of necessary preliminary information, e.g., the data used and the data mining techniques utilized. Interesting results for each of the previously described approaches are presented thereafter.

2.4.1 Preliminary Information

Data

Using the load benchmark generator [3, 172] described in Appendix A.1, five simulated load time series, $\{P_L[k]; k = 1, \dots, K\}$ with three years of hourly simulated values ($K = 26280$) are created. The difference between these time series is their noise percentage r_{prc} , i.e. 75, 60, 45, 30, or 15%. The time series are assumed to represent the load of a number of households with varying degrees of unpredictable effects. More information regarding the simulated time series, specifically the parameters used for their creation can be found in Appendix A.1. After their creation, the values of the simulated time series are normalized to values between zero and one and divided in half. The first half is used as a training set and the second is used as a test set. Furthermore, all time series used lack outliers and missing values and thus do not require further preprocessing. For the sake of illustration, Figure 2.15 depicts simulated time series with 75%, 45%, and 15% noise.

Data Mining

To obtain the presented results, five different data mining techniques are used. The first three are polynomial models (i.e. multiple linear regressions) with maximal allowed degrees of one, two, or three and thus are referred to as Poly1-3. Additionally, two non-linear regression techniques are also used. These are two artificial neural networks (traditional multi-layer perceptrons); one with a low amount of hidden neurons, i.e. four, and one with a larger amount, i.e. ten. Thereupon, the first is referred to as ANN4, while the second as ANN10. It is considered that using ANN4 and ANN10 is enough to observe the effects of having a neural network with a lower and a greater number of free parameters⁷.

⁷ Note that a prestudy was conducted; in it six and twenty hidden neurons were also tested. These numbers are not used further, as their results obtained in the prestudy did not vary greatly from those obtained when using four and ten hidden neurons.

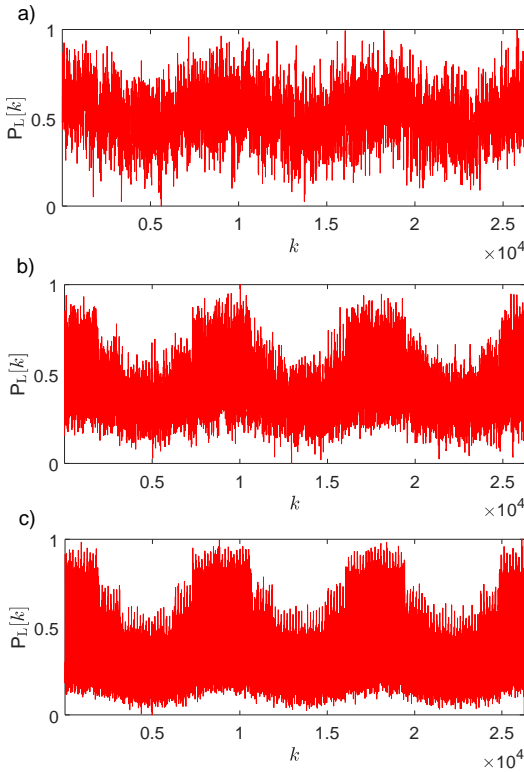


Figure 2.15: Simulated load time series with; a) 75% noise; b) 45% noise; c) 15% noise

Furthermore, the desired output and input vector, used to train quantile regressions based on the NNQF for the time series described above, are defined as:

$$\begin{aligned}
 y &:= P_L[k + H]; H \in \{1, 24, 48\} \\
 \mathbf{x} &:= [P_L[k], \dots, P_L[k - H_1]]^T; k \in [H_1 + 1, K - H], H_1 = 168.
 \end{aligned}
 \tag{2.34}$$

In other words, the regressions are trained to estimate quantiles for values 1, 24, or 48 hours into the future using the past autoregressive values of the last seven days, i.e. the most recent information available⁸. Different forecast horizons are used, since it is desired to determine their influence on the properties of the various probabilistic forecasts to be created. Moreover, the quantile regressions trained with each technique correspond to the probabilities $q \in \{0.01, 0.02, \dots, 0.99\}$. Likewise, the threshold value ϵ_{th} used in Equation (2.1) is set to infinity for all created quantile regressions. In addition, a feature selection based on a Wrapper method⁹ is applied, resulting in the selection of the four most relevant features [173]. In the present thesis the feature selection is conducted prior to the NNQF and the quantile regressions' training. Therefore, the regressions estimating different quantiles trained on a given training set using a given data mining technique and forecast horizon vary only in their estimated parameters (as in Example 2). The reason thereof is to assure that the quantiles being estimated stem from the same conditional distribution. An independent feature selection for regressions estimating different quantiles is possible, but is not studied further in the present work. This decision is based on a preliminary experiment that showed that an independent feature selection results in several issues, e.g., problems when training polynomials using a constrained least squares method (cf. Equation (2.8)), having the NNQF search for nearest neighbors in an extremely high dimensional feature space, etc. Still, future related works should try to solve the various issues, as an independent feature selection may help in improving the quantile regressions.

⁸ The reason behind the use of pure autoregressive models is that the time series are created without an exogenous component; meaning that they are not influenced by any external factors.

⁹ The selection starts with the training of regressions that only take one of the features as input. Thereafter the models are evaluated via the coefficient of determination [55], then the feature whose model performs best is selected as the first relevant feature. Afterwards, models using pairs formed by the first selected feature and all other ones as input are trained and evaluated. The feature forming the pair with the best result is then selected as the second relevant feature. Finally, the process is continued until the desired number of features has been chosen.

Furthermore, Equations (2.8) and (2.9) are used to assure the strictly increasing monotony of the non-parametric distribution forecasts (cf. Section 2.2.3). In the present section, the necessary global minimum is defined as $y_{\min} = 0$ (only positive power values are to be forecast) and the used regularization value is set to $\epsilon_r = 10^{-5}$.

Amount of Training Data and Number of Nearest Neighbors

Apart from the amount of noise in the time series, the data mining technique, and the forecast horizon, there are two other parameters that may have a major influence in the outcome of the forecasts; these are the number of nearest neighbors and the amount of training data. The former is relevant, since it defines the modified training set with which a quantile regression is trained, while the latter is important as the NNQF assumes a dense training set. The influence that these values have on the results is tested by using different amounts of nearest neighbors on different percentages, N_{prc} , of training data (note that the amount of training data N varies with the forecast horizon). Table 2.4 contains the number of nearest neighbors used in the different amounts of training data tested.

Using the previously described data mining techniques, simulated time series, amounts of training data, and number of nearest neighbors, a great number of quantile regressions based on the NNQF are obtained, with which a series of experiments are conducted. The results of these experiments are presented in the next sections. Notice that the training sets for $N_{\text{prc}} < 100\%$ contain the latest N_{prc} observations of the original training set.

2.4.2 Quantile Regressions based on the NNQF and Non-Parametric Distribution Forecast

The present section offers insight into certain properties of both quantile regressions based on the NNQF and the non-parametric distribution forecast created with them. Please note, that the reason for discussing the quantile re-

$N_{\text{prc}}[\%]$	H	N	N_{NN}
100	1	12970	{5, 10, 25, 50, 100, 250, 500, 1000, 2500, 5000, 7500, 10000, 12970}
	24	12947	{5, 10, 25, 50, 100, 250, 500, 1000, 2500, 5000, 7500, 10000, 12947}
	48	12923	{5, 10, 25, 50, 100, 250, 500, 1000, 2500, 5000, 7500, 10000, 12923}
75	1	9685	{5, 10, 25, 50, 100, 250, 500, 1000, 2500, 5000, 7500, 9685}
	24	9662	{5, 10, 25, 50, 100, 250, 500, 1000, 2500, 5000, 7500, 9662}
	48	9638	{5, 10, 25, 50, 100, 250, 500, 1000, 2500, 5000, 7500, 9638}
50	1	6401	{5, 10, 25, 50, 100, 250, 500, 1000, 2500, 5000, 6401}
	24	6378	{5, 10, 25, 50, 100, 250, 500, 1000, 2500, 5000, 6378}
	48	6354	{5, 10, 25, 50, 100, 250, 500, 1000, 2500, 5000, 6354}
25	1	3116	{5, 10, 25, 50, 100, 250, 500, 1000, 2500, 3116}
	24	3093	{5, 10, 25, 50, 100, 250, 500, 1000, 2500, 3093}
	48	3069	{5, 10, 25, 50, 100, 250, 500, 1000, 2500, 3069}

Table 2.4: Number of nearest neighbors used depending on the available training data; N_{prc} : percentage of training data; H : forecast horizon; N : amount of training data; N_{NN} : number of nearest neighbors

gressions together with the non-parametric distribution forecasts is that their evaluation results are identical to one another, since their quantile estimates used in Equations (2.22) to (2.27) are the same.

This section begins by studying the influence that the number of nearest neighbors have on the forecast quality. To do so, the quantile regressions trained using the time series with 45% noise and 100% of the training data are first evaluated. The results obtained on the corresponding test set are displayed in Figure 2.16¹⁰. Note that the results obtained for the time series with 75, 60, 30, and 15% of noise are found in Appendix A.5.

The results in Figure 2.16 show that trivial forecasts, i.e. the ones obtained using the maximal possible number of nearest neighbors, obtain better reliability deviation values than some non-trivial forecasts, regardless of forecast horizon and data mining technique used. This effect can be easily explained by the fact that the training set is extremely representative of the used test

¹⁰ The reason behind Poly1-3 delivering the same results, is that their selected features are the same.

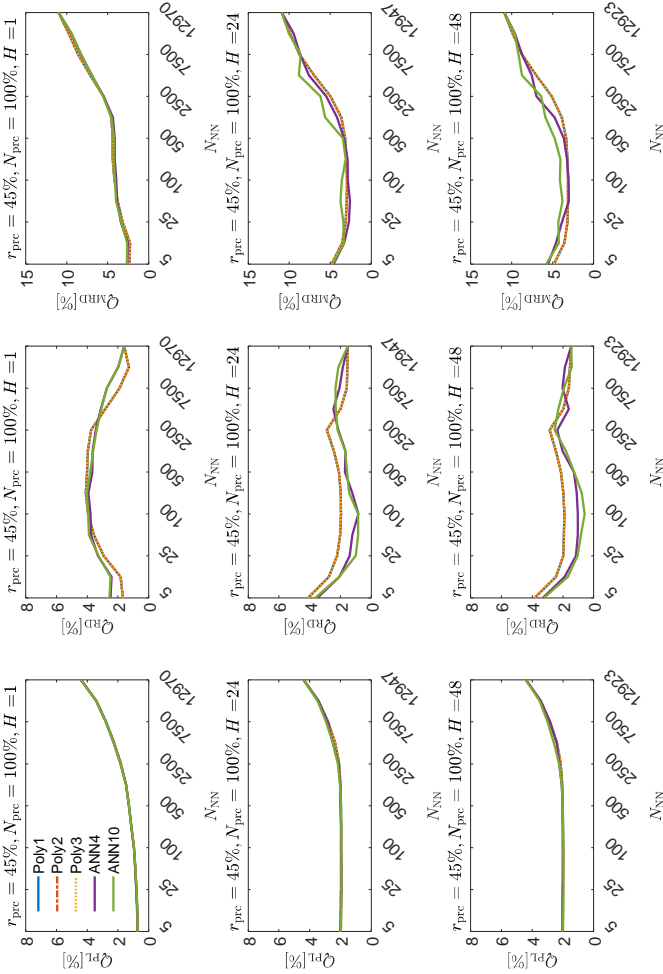


Figure 2.16: Average pinball-loss, reliability deviation, and modified reliability deviation values on the utilized test set for different amounts of nearest neighbors; r_{prc} : noise percentage; N_{prc} : percentage of training data; H : forecast horizon; N_{NN} : number of nearest neighbors; Q_{PL} : average pinball-loss; Q_{MIRD} : average modified reliability deviation; Q_{MIRD} : average modified reliability deviation

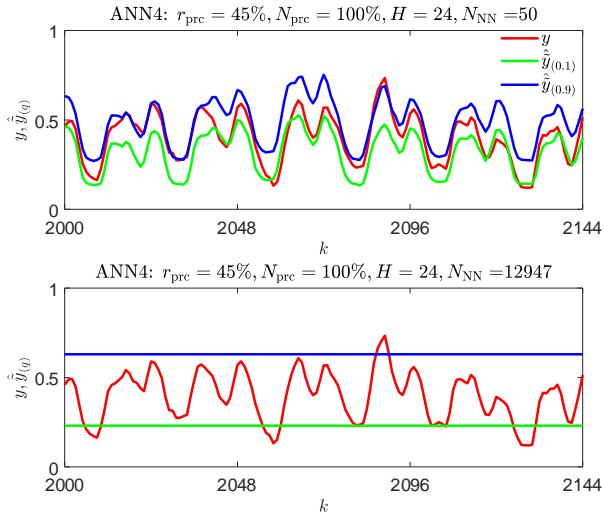


Figure 2.17: Example of non-trivial ($N_{\text{NN}} = 100$) and trivial forecasts ($N_{\text{NN}} = 12947$)

set. In other words, using the maximal number of possible nearest neighbors results in quantile regressions that estimate the unconditional quantile of all time series values in the training set. Therefore, in cases in which the time series values in the training set and the test set are similarly distributed, trivial forecasts are able to obtain low reliability deviations. Since it is undesired for an evaluation value to consider trivial results as accurate, it is concluded that the reliability deviation is not adequate to evaluate the obtained forecasts. For such reason, only the pinball-loss and the modified reliability deviation are used further. For the sake of illustration, Figure 2.17 shows examples of quantile regressions based on the NNQF trained for $N_{\text{NN}} = 100$ (i.e. non-trivial forecasts) and $N_{\text{NN}} = 12947$ (i.e. trivial forecasts).

In addition to the previous results, Figure 2.16 also shows that the number of nearest neighbors considered in the present thesis as optimal, i.e. the one for which the best combination of pinball-loss and modified reliability deviation values are obtained, is probably between the smallest and the

$r_{\text{pre}}[\%]$	H	Poly1			Poly2			Poly3			ANNA			ANN10		
		N_{NN}	$Q_{\text{PL}}[\%]$	$Q_{\text{MRD}}[\%]$	N_{NN}	$Q_{\text{PL}}[\%]$	$Q_{\text{MRD}}[\%]$	N_{NN}	$Q_{\text{PL}}[\%]$	$Q_{\text{MRD}}[\%]$	N_{NN}	$Q_{\text{PL}}[\%]$	$Q_{\text{MRD}}[\%]$	N_{NN}	$Q_{\text{PL}}[\%]$	$Q_{\text{MRD}}[\%]$
75	1	10	1.17	2.60	10	1.17	2.58	10	1.17	2.58	10	1.17	2.24	10	1.17	2.42
	24	25	3.60	4.16	25	3.60	4.15	25	3.60	4.15	50	3.61	4.17	50	3.62	4.28
	48	50	3.65	4.15	50	3.65	4.13	50	3.65	4.13	500	3.64	4.63	250	3.66	4.47
60	1	10	1.09	2.08	10	1.09	2.11	10	1.09	2.11	10	1.09	2.11	10	1.09	2.19
	24	50	3.03	2.84	50	3.03	2.84	50	3.03	2.84	50	3.04	2.75	100	3.04	2.84
	48	50	3.03	2.67	50	3.03	2.67	50	3.03	2.67	50	3.02	2.70	50	3.03	2.80
45	1	10	0.76	2.27	10	0.76	2.27	10	0.76	2.27	10	0.74	2.54	5	0.71	2.68
	24	100	1.96	2.96	100	1.96	2.96	100	1.96	2.96	50	1.94	2.59	250	1.95	3.09
	48	100	2.01	3.13	100	2.01	3.13	100	2.01	3.13	50	1.98	3.02	50	1.97	3.80
30	1	10	0.72	2.02	10	0.72	2.02	10	0.72	2.02	5	0.64	3.53	5	0.63	3.58
	24	25	1.64	2.40	25	1.64	2.40	25	1.64	2.40	50	1.61	2.40	100	1.61	2.55
	48	50	1.67	2.73	50	1.67	2.73	50	1.67	2.73	25	1.64	2.85	25	1.64	2.88
15	1	10	0.50	3.58	10	0.50	3.58	10	0.50	3.58	5	0.47	3.79	10	0.48	4.39
	24	25	1.25	3.66	100	1.26	3.37	100	1.26	3.37	50	1.21	2.97	50	1.19	3.31
	48	25	1.28	3.82	50	1.28	3.54	50	1.28	3.54	50	1.25	3.80	50	1.23	3.55

Table 2.5: Optimal number of nearest neighbors and evaluation results obtained for time series with different noise percentages; r_{pre} : noise percentage; H : forecast horizon; N_{NN} : number of nearest neighbors; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

largest possible number. To find the optimal number of nearest neighbors, the obtained average pinball-loss and modified reliability deviation values are first normalized to values between zero and one and then added together. Thereafter, the number of nearest neighbors delivering the lowest sum for a specific technique and forecast horizon is considered the optimal number. Table 2.5 contains for all time series, data mining techniques, and horizons the optimal number of nearest neighbors and test set evaluation values of the quantile regressions trained using 100% of the training data.

Table 2.5 shows that for $H = 24$ and $H = 48$ the number of optimal nearest neighbors is always between 25 and 500, regardless of noise amount and data mining technique used. In the case of $H = 1$ the optimal number is always chosen to be 5 or 10. Therefore, it can be suggested that while quantile regressions with short forecast horizons seem to benefit from using a low number of nearest neighbors, quantile regressions with larger forecast horizons may find – as previously assumed – an optimal number between the lowest and largest possibilities. Although it can be argued that this result is dependent on the used training data, it is still an interesting outcome that may offer a first clue when searching for the optimal number of nearest neighbors in other use cases. It can also be observed that the optimal numbers of nearest neighbors deliver models with pinball-loss values that improve with a reduction of the time series' noise. Nonetheless, there appears to be no clear relationship between the amount of noise and the modified reliability deviation. Furthermore, it also seems that increasing the forecast horizons from $H = 1$ to $H = 24$ or $H = 48$ generally worsens the pinball-loss and the modified reliability deviation values obtained.

The influence that modifying the amount of training data may have on the forecasts' accuracy can now be tested using the number of nearest neighbors considered to be optimal (cf. Table 2.5). Figure 2.18 shows the test sets' pinball-loss and modified reliability deviation values of models trained with various amounts of training data of the time series with 45% of noise.

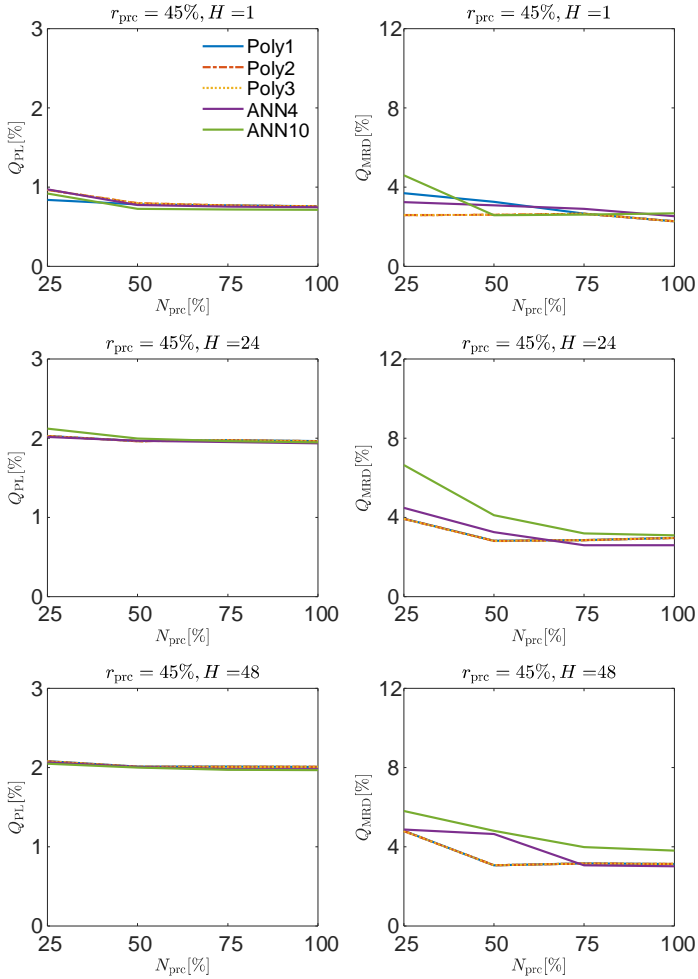


Figure 2.18: Test sets' average pinball-loss and modified reliability deviation values obtained for different amount of training data; r_{prec} : noise percentage; N_{prec} : percentage of training data; H : forecast horizon; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

As expected, Figure 2.18 shows that in most depicted cases the least amount of training data delivers the worst evaluation results (with Poly1-3 for $H = 1$ being the only exception). Furthermore when looking at the results, it is clear that increasing the amount of training data generally improves the accuracy of the models. Surprisingly, some modified reliability deviation values (e.g., those obtained by Poly1-3 for $H = 24$ and $H = 48$ and ANN10 for $H = 1$) appear to have a global minimum when using only 50% of the training data. This may be caused (i) by some random effects and/or (ii) by the selection of features that perform better on the test set than those selected when using all training data.

Another property that has to be investigated further, is the scalability of the NNQF-based regressions in the presence of an increasing amount of training data and how it compares to that of a more traditional k-nearest neighbors quantile regression (kNNQR). Figure 2.19 plots the computation time t for training and applying the 99 quantile regressions considered in the present section (i.e. those with probabilities $q \in \{0.01, \dots, 0.99\}$) against the percentage of training data used to train them (i.e. N_{pre}). It is important to mention, that the figure compares the results of kNNQRs based on [74] to those obtained by Poly1 and ANN10 (i.e. the most simple and the most complex technique used). Moreover, the number of nearest neighbors and forecast horizon chosen is 100 and 24 hours, respectively. In addition, the computer that is utilized has an Intel Core i7-4790 processor with 3.6 [GHz] and 16 [GB] of RAM.

At a first glance, Figure 2.19 depicts completely opposite behaviors for training and applying the regressions. During training, the kNNQR has a much better scalability than the NNQF-based regressions, as its training consists only on saving all available training data. In contrast, training a regression based on the NNQF requires first, finding the nearest neighbors and then computing the regression parameters. Hence, the more complex the data mining technique and the larger the amount of training data available are, the longer the training of a NNQF-based regression is going to

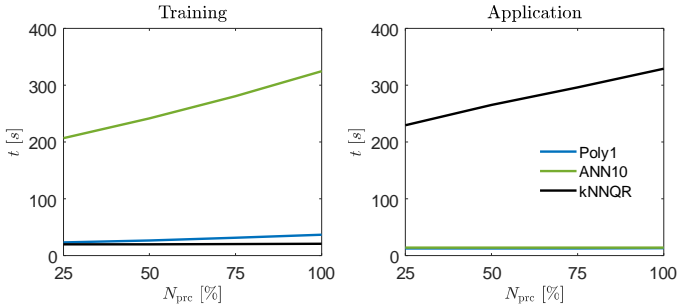


Figure 2.19: Scalability comparison for training and applying quantile regressions based on the NNQF and k-nearest neighbors quantile regressions (kNNQRs); note $N_{NN} = 100$ and $H = 24$ are used for the comparison; t : computation time; N_{pre} : percentage of training data (adapted from [165])

take. Nevertheless, the opposite can be said when looking at the application process. While the computation time for applying NNQF-based regressions remains more or less constant, the time for applying the kNNQR appears to be proportional to the amount of training data. This stems from the fact that the NNQF-based regressions are precomputed functions that only need to be evaluated. Conversely, the application of a kNNQR requires a computation of the nearest neighbors every time a forecast is needed; meaning that the larger the training set to search for the nearest neighbors is, the longer it is going to take. Even though the previous statement is also true for the NNQF (as shown in by the training times in Figure 2.19), the fact that the nearest neighbors have to be searched only once prior to the regressions training instead of every time a forecast is required, speaks in favor of using the NNQF over the kNNQR if scalability is of importance. Notice that the kNNQR is not tested further in the present thesis, since the possibility of using the developed methods in an online forecasting service in which the amount of training data will constantly increase (e.g., the Energy Lab 2.0 [10]) is

desired. Lastly it needs to be mentioned, that the kNNQR obtains similar evaluation values¹¹ as the techniques used for comparison (cf. Figure 2.16).

Finally, the present section offers a comparison between quantile regressions based on the NNQF and “true” quantile regressions trained by minimizing the sum of pinball-losses (cf. Equation (1.16)) [35]. The “true” quantile regression are three polynomial models with maximal allowed degrees of 1 to 3, which are further referred to as “True” Poly1-3. The training algorithm of these regressions is based on a MATLAB implementation found online¹², which has been modified to avoid quantile crossing. Figure 2.20 depicts the results obtained from the test set of the time series with 45% of noise. Please note, that the quantile regressions based on the NNQF use the optimal number of nearest neighbors contained in Table 2.5.

The comparison shows that the obtained pinball-loss values between both types of quantile regressions appear to be almost indistinguishable, however some differences start to appear when looking at the modified reliability deviations. For $H = 1$ is clear that the regressions based on the NNQF perform generally worse. Nonetheless, the polynomial models show similar modified reliability deviation values for the larger forecast horizons. Furthermore, the quantile regressions based on ANN4 show also similar results than the “true” quantile regressions for $H = 48$ and slightly better results for $H = 24$. The latter may be explained by the fact that the ANN4 models are able to describe possible non-linear effects that the polynomials cannot. On the contrary, quantile regressions based on ANN10 obtain the worse modified reliability deviation values. Please note, that the lack of “true” quantile regressions based on more complex data mining techniques stems from the fact, that a MATLAB implementation similar to the one of the polynomials has not been found at the moment of writing the present thesis. This

¹¹ kNNQR with $N_{NN} = 100$: $Q_{PL} = 1.85\%$ and $Q_{MRD} = 3.51\%$; Poly1 with $N_{NN} = 100$: $Q_{PL} = 1.96\%$ and $Q_{MRD} = 2.96\%$; ANN10 with $N_{NN} = 100$: $Q_{PL} = 1.95\%$ and $Q_{MRD} = 3.59\%$

¹² [de.mathworks.com/matlabcentral/fileexchange/32115-quantreg-x-y-tau-order-nboot-](https://de.mathworks.com/matlabcentral/fileexchange/32115-quantreg-x-y-tau-order-nboot)

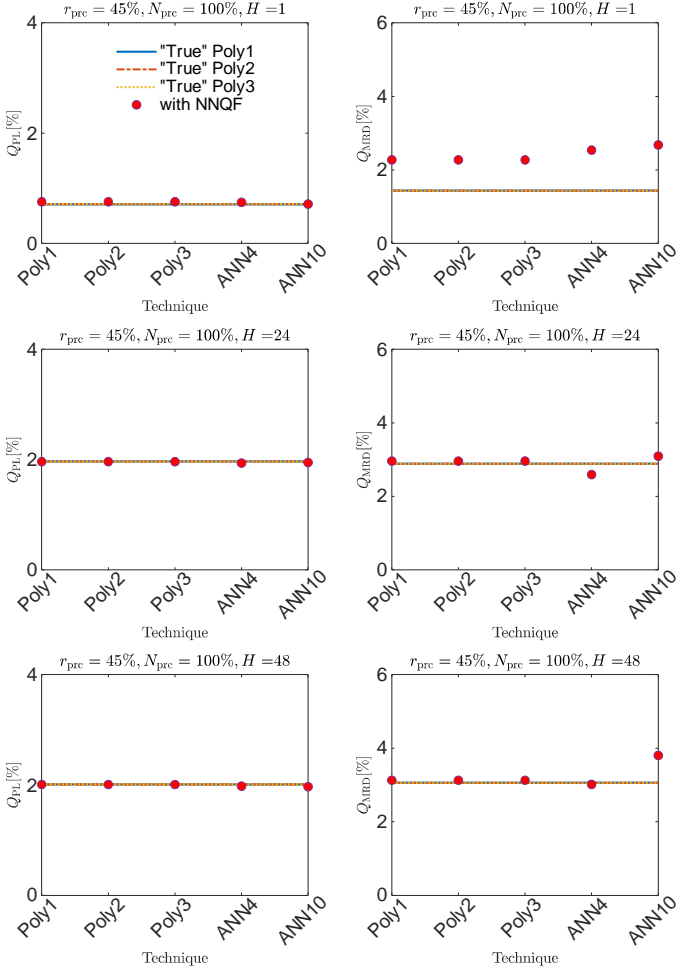


Figure 2.20: Comparison between quantile regressions based on the NNQF and “true” quantile regressions; r_{prc} : noise percentage; N_{prc} : percentage of training data; H : forecast horizon; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

may be attributed to the fact that training “true” quantile regressions with more complex techniques is – as previously mentioned – not a trivial task. Therefore, being able to easily train quantile regressions with complex data mining techniques (as e.g., ANN4) advocates in favor of using the NNQF, instead of directly minimizing the sum of pinball-losses. Additionally, using a given data mining technique to train quantile regressions based on the NNQF appears to generally take less time than directly minimizing the sum of pinball-losses. For instance, training the “true” polynomial quantile regressions takes approximately 64 seconds while training their polynomial NNQF-based counterparts takes only 37 seconds; this means that directly minimizing the sum of pinball-losses increases the computation time by approximately 79%.

2.4.3 Interval Forecast

Using the created quantile regressions based on the NNQF (cf. Section 2.4.1), 49 pairs centered on the regressions describing the median can be formed. These pairs represent interval forecasts with probabilities of containing the future time series values equal to $(q_u - q_l) \in \{0.02, 0.04, \dots, 0.98\}$.

Just as in the previous section, the forecasts obtained for the time series with 45% of noise using 100% of the training data are first evaluated. Figure 2.21 depicts the results obtained on the corresponding test set. Please refer to Appendix A.5 for the results obtained on the time series with 75, 60, 30, and 15% of noise.

Figure 2.21 shows that for all techniques the average interval width and average interval score values are almost identical, which is a similar outcome as the one observed from the single quantile regressions (cf. Figure 2.16). Moreover, it can be observed that the intervals’ average width increases with the number of nearest neighbors used. This constant increase for $N_{NN} > 2500$, which influences in part the increase in interval score, can be explained as an overestimation of the uncertainty caused by the large

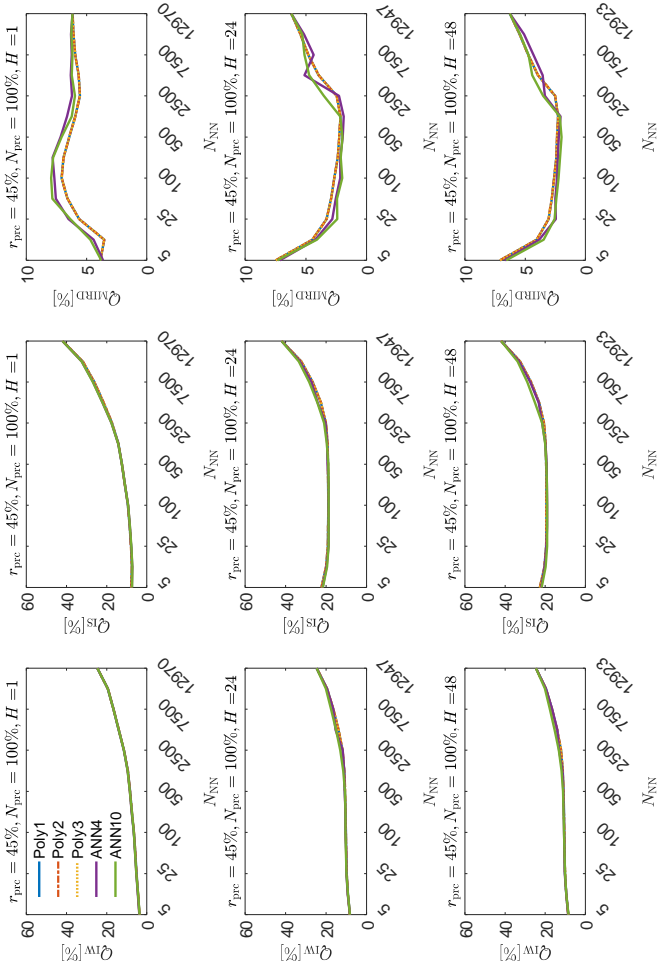


Figure 2.2.1: Average interval width, interval score, and modified interval reliability deviation values on the utilized test set for different amounts of nearest neighbors; r_{prec} : noise percentage; N_{prec} : percentage of training data; H : forecast horizon; N_{NN} : number of nearest neighbors; Q_{IW} : average interval width; Q_{IS} : average interval reliability score; Q_{MIRD} : average modified interval reliability deviation

numbers of nearest neighbors used. Furthermore, for $H = 24$ and $H = 48$ the intervals trained with $N_{\text{NN}} = 5$ not only obtain a slightly higher average interval score than those trained using $5 < N_{\text{NN}} < 2500$, but also the worse results in terms of modified interval reliability deviation. This can be explained by the fact that using $N_{\text{NN}} = 5$ results in narrow intervals that may underestimate the uncertainty and thus increase the number of deviations. Looking at the results for $H = 1$, it can also be seen that the Poly1-3 models deliver better results than the ANNs, which may be explained by the fact that in such case the relation between \mathbf{x} and y could be nearly linear. Surprisingly, the modified interval reliability deviation values obtained by the trivial forecasts for $H = 1$ (i.e. the ones obtained with the maximal number of possible nearest neighbors) are lower than for other non-trivial cases. Nonetheless, the trivial forecasts present as expected the worst interval width and interval score values.

Similarly as in Section 2.4.2, the number of nearest neighbors considered optimal in the present thesis is the one that delivers the best combination of interval score and modified interval reliability deviation values. Hence, the optimal number is determined by normalizing the interval score and modified interval reliability deviation, adding them together, and finding the N_{NN} value with the lowest sum. Table 2.6 contains the results obtained with the optimal number of nearest neighbors for the different noise percentages¹³.

Just as with the quantile regressions, the results in Table 2.6 may serve as a hint of what number of nearest neighbors to use, when creating interval forecast for other application cases. Moreover, when comparing the previous results to those of the quantile regressions (cf. Table 2.5), it can be observed (i) that for the shortest forecast horizon (i.e. $H = 1$) the lowest amounts of nearest neighbors (i.e. 5 and 10) are again the ones considered optimal and (ii) that in most cases the optimal number of nearest neighbors increased. Interestingly, there appears to be no clear relationship between

¹³ Note that the interval's width is not considered to determine the optimal number, as it is implicitly considered by the interval score.

$r_{\text{prec}}[\%]$	H	Poly1			Poly2			Poly3			ANN4			ANN10			
		N_{NS}	$Q_{\text{IW}}[\%]$	$Q_{\text{MIRD}}[\%]$	N_{NS}	$Q_{\text{IW}}[\%]$	$Q_{\text{MIRD}}[\%]$	N_{NS}	$Q_{\text{IW}}[\%]$	$Q_{\text{MIRD}}[\%]$	N_{NS}	$Q_{\text{IW}}[\%]$	$Q_{\text{MIRD}}[\%]$	N_{NS}	$Q_{\text{IW}}[\%]$	$Q_{\text{MIRD}}[\%]$	
75	1	10	6.63	11.59	3.05	10	6.58	11.57	2.94	10	6.58	11.57	2.94	10	6.67	11.48	2.90
	24	100	19.83	34.84	1.32	250	19.95	34.86	1.31	250	19.95	34.86	1.31	250	19.94	34.87	1.34
	48	500	19.93	35.27	1.43	500	19.94	35.29	1.43	500	19.94	35.29	1.43	500	20.03	35.21	1.34
60	1	5	5.51	11.15	3.76	5	5.48	11.12	3.84	5	5.48	11.12	3.84	10	6.44	10.66	3.83
	24	1000	16.74	29.51	1.58	1000	16.74	29.51	1.58	1000	16.74	29.51	1.58	500	16.58	29.39	1.54
	48	250	16.57	29.35	1.72	250	16.57	29.35	1.72	250	16.57	29.35	1.72	250	16.61	29.31	1.75
45	1	10	4.36	7.44	3.53	10	4.36	7.44	3.53	10	4.36	7.44	3.53	5	3.70	7.47	3.63
	24	1000	11.00	19.27	2.15	1000	11.00	19.27	2.15	1000	11.04	19.49	1.89	500	10.67	18.90	1.98
	48	1000	11.28	19.80	2.27	1000	11.28	19.80	2.27	1000	11.42	19.95	2.06	500	11.19	19.39	1.99
30	1	10	3.94	7.20	3.06	10	3.94	7.20	3.06	10	3.94	7.20	3.06	5	3.17	6.71	3.71
	24	100	9.00	16.12	1.69	100	9.00	16.12	1.69	100	9.00	16.12	1.69	25	8.78	16.14	1.25
	48	25	8.92	16.50	1.69	25	8.92	16.50	1.69	25	8.92	16.50	1.69	50	9.18	16.23	1.36
15	1	5	2.17	5.54	4.54	5	2.17	5.54	4.54	5	2.17	5.54	4.54	5	2.25	5.16	4.46
	24	1000	7.66	14.20	2.14	500	7.11	13.38	1.84	500	7.11	13.38	1.84	50	6.31	12.06	2.28
	48	1000	8.17	15.18	2.53	500	7.53	14.04	2.65	500	7.53	14.04	2.65	100	6.94	12.65	2.51

Table 2.6: Optimal number of nearest neighbors and interval evaluation results obtained for time series with different noise percentages; r_{prec} : noise percentage; H : forecast horizon; N_{NN} : number of nearest neighbors; Q_{IW} : average interval width; Q_{IS} : average interval score; Q_{MIRD} : average modified interval reliability deviation

the amount of noise and the optimal number of nearest neighbors. Finally, it can also be noticed that the interval width and interval score values seem to be inversely proportional to the amount of noise. In other words, the lower the inherent uncertainty of the time series being forecast, the narrower the obtained intervals are.

Using the number of nearest neighbors considered to be optimal (cf. Table 2.6), the influence that the amount of training data has on the interval forecasts' quality can be determined. Figure 2.22 depicts the results obtained using the various techniques and their optimal number of nearest neighbors for the time series with 45% of noise.

Figure 2.22 shows, as expected, that decreasing the amount of training data increases the intervals' width, which in turn increases the corresponding interval score. Similarly, the modified interval reliability deviation also seem to constantly increase with the decreasing amount of training data, with Poly2-3 for $H = 1$ and ANN4 for $H = 48$ being the only exceptions; these may be explained by the fact that the width of the created intervals may not necessarily affect the percentage of values laying within them. Regardless, it is clear that for most cases the quality of the interval forecasts decreases with the amount of available training data.

Finally, a comparison between interval forecasts created using quantile regressions based on the NNQF (using their optimal number of nearest neighbors) and those obtained with the same "true" quantile regressions used in Section 2.4.2 is presented. Figure 2.23 shows the results of this comparison.

The results depicted in Figure 2.23 show that the interval width and interval score values of interval forecasts with quantile regressions based on the NNQF and those created with the "true" quantile regressions are almost identical. In terms of modified interval reliability deviation, the "true" quantile regressions' intervals are for $H = 1$ always better. Nonetheless, for the larger forecast horizons (i.e $H = 24$ and $H = 48$) all quantile regressions based on the NNQF are able to deliver interval forecasts with a slightly better modified interval reliability deviations values than those of their "true"

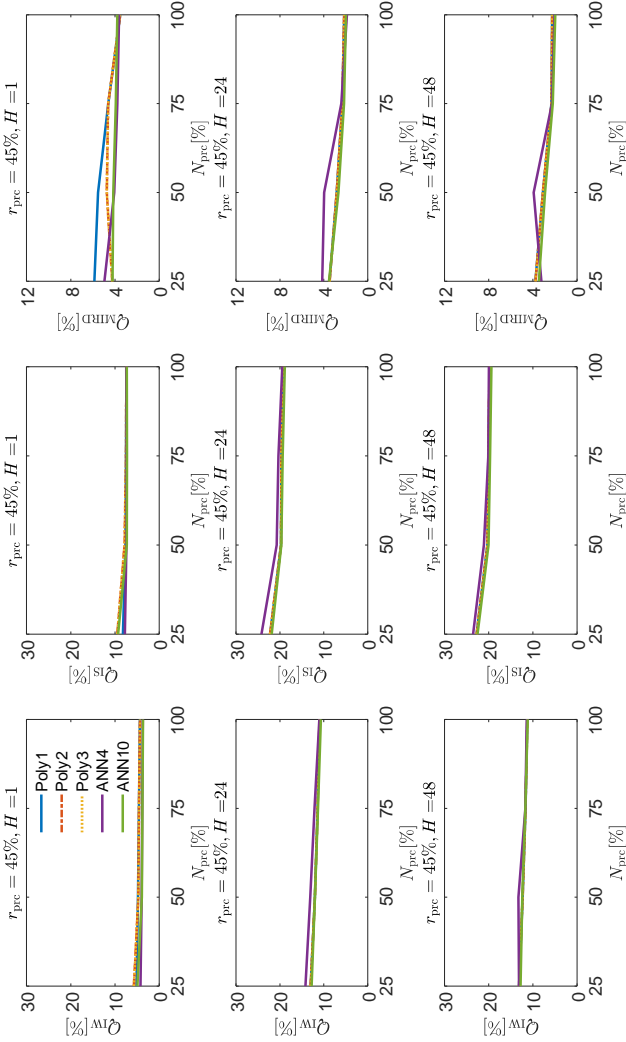


Figure 2.22: Average interval width, interval score, and modified interval reliability deviation values on the utilized test set for different amounts of training data; t_{prec} : noise percentage; H : forecast horizon; Q_{IW} : average interval width; Q_{IS} : average interval score; Q_{MIRD} : average modified interval reliability deviation

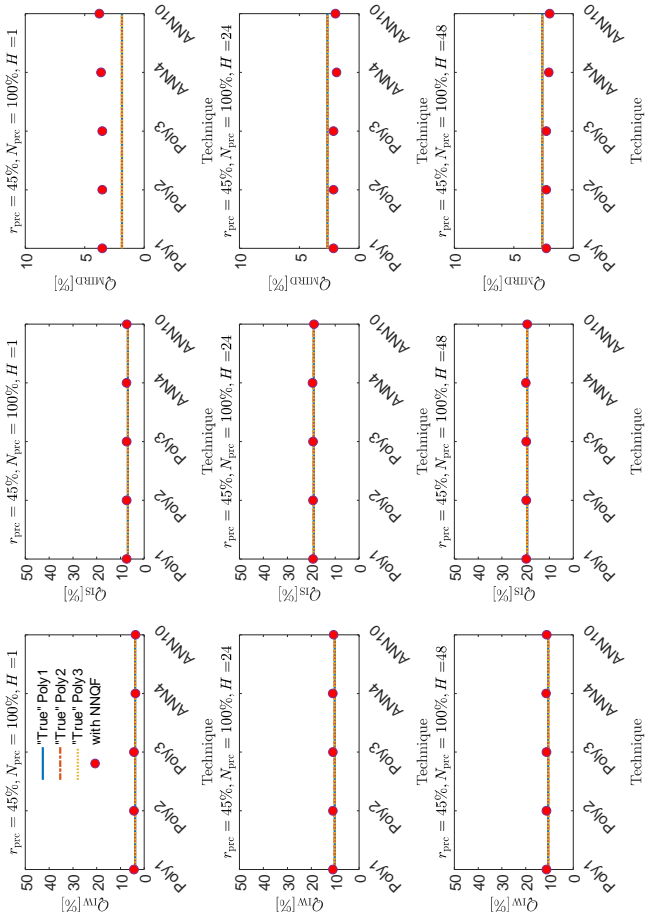


Figure 2.23: Comparison between interval forecasts created with quantile regressions based on the NNQF and those obtained with "true" quantile regressions; r_{prec} : noise percentage; N_{prec} : percentage of training data; H : forecast horizon; Q_{IR} : average interval width; Q_{IS} : average interval score; Q_{IRW} : average modified interval reliability deviation

counterparts. Moreover as mentioned in Section 2.4.2, using quantile regressions based on the NNQF allows the obtainment of interval forecasts using more complex data mining techniques. Likewise, the NNQF also reduces the computation time needed to train the interval forecasting models, as it avoids the minimization of the sum of pinball-losses.

2.4.4 Parametric Distribution Forecast

The present section answers the questions raised in Section 2.4.4 by using the quantile regressions based on the NNQF trained on 100% of the training data from the simulated time series with 45% of noise and the optimal number of nearest neighbors shown in Table 2.5. The use of the other time series is considered unnecessary, as the goal of the present section is just to compare the parametric distribution forecasts to the quantile regressions they are based on. Additionally, the quantile estimates used for evaluating the parametric distribution forecasts are those with corresponding probabilities $q \in \{0.01, \dots, 0.99\}$.

Since the method described in the present thesis tries to find a parametric CDF that best fits a non-parametric estimate, it is important to determine how much does the forecast accuracy change when compared to the non-parametric CDF (cf. Section 2.4.2). Therefore, Figures 2.24 to 2.26 plot the pinball-loss and modified reliability deviation – for every forecast horizon and data mining technique tested – of the parametric and non-parametric CDFs' quantile estimates against their corresponding probability q ¹⁴. Note that the parametric CDFs used are Beta, Gamma, Normal, and Uniform and that their necessary parameters are defined as shown in Table 2.2.

Figures 2.24 to 2.26 show that the pinball-losses given by the parametric distributions and their non-parametric counterparts are almost identical, regardless of forecast horizon and data mining technique used. Similarly the modified reliability deviation values are almost indistinguishable for

¹⁴ Poly1-3 are plotted together, since the feature selection process made them identical.

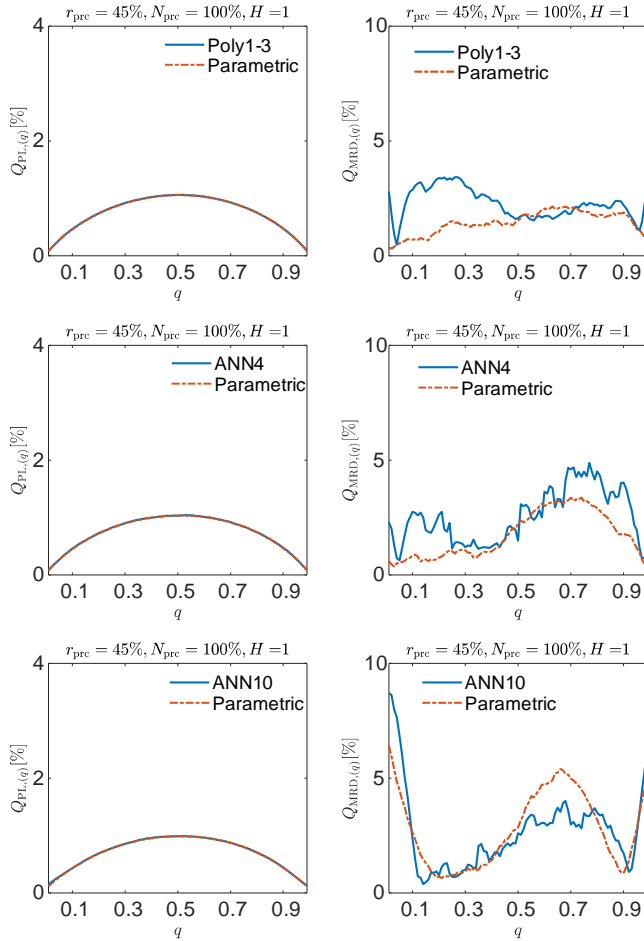


Figure 2.24: Pinball-loss and modified reliability deviation values obtained from the quantile estimates given by quantile regressions and their corresponding parametric distribution forecasts for $H = 1$; r_{pre} : noise percentage; N_{pre} : percentage of training data; H : forecast horizon; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation; q : probability value

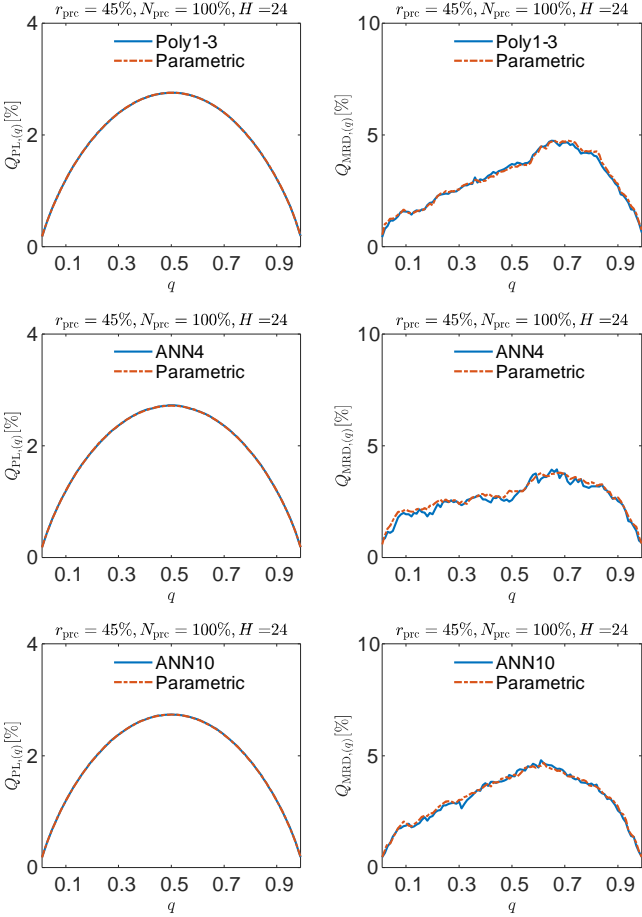


Figure 2.25: Pinball-loss and modified reliability deviation values obtained from the quantile estimates given by quantile regressions and their corresponding parametric distribution forecasts for $H = 24$; r_{prec} : noise percentage; N_{prec} : percentage of training data; H : forecast horizon; Q_{PPL} : average pinball-loss; Q_{MRD} : average modified reliability deviation; q : probability value

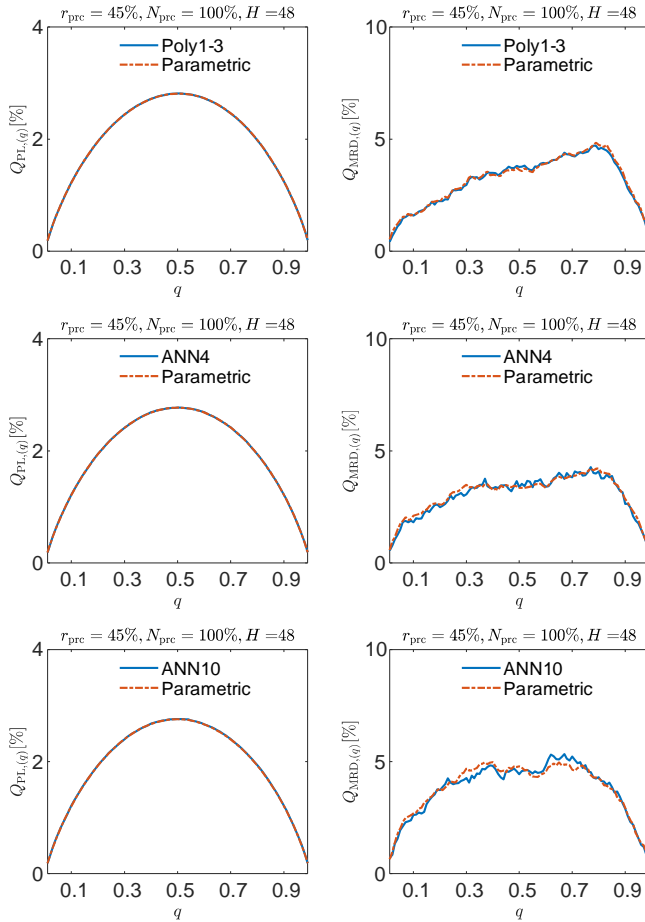


Figure 2.26: Pinball-loss and modified reliability deviation values obtained from the quantile estimates given by quantile regressions and their corresponding parametric distribution forecasts for $H = 48$; r_{prec} : noise percentage; N_{prec} : percentage of training data; H : forecast horizon; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation; q : probability value

$H = 24$ and $H = 48$, however for $H = 1$ major differences appear. For instance, the parametric CDFs based on Poly1-3 and ANN4 appear to obtain better modified reliability deviation values than the non-parametric estimates they are based on. A similar outcome is presented in Table 2.7 that contains the average pinball-loss and modified reliability deviation values for the non-parametric CDFs given by the various data mining techniques and those obtained by their corresponding parametric fits. Just as before, the pinball-loss values of the parametric and non-parametric CDFs are almost identical. Furthermore, the parametric fits on the estimates given by all regression models have worse modified reliability deviation values than their non-parametric counterparts for $H = 24$ and $H = 48$. In contrast for $H = 1$, the parametric CDFs obtain better average modified reliability deviation values when fitting the estimates given by Poly1-3 and ANN4. It can be then concluded, that the method described in Section 2.2.4 is able to find parametric CDFs that fit some given non-parametric CDFs (based on a number of quantile regressions) without decreasing and sometimes even improving the forecast accuracy.

Technique	H	Non-Parametric		Parametric	
		$Q_{PL}[\%]$	$Q_{MRD}[\%]$	$Q_{PL}[\%]$	$Q_{MRD}[\%]$
Poly1-3	1	0.76	2.27	0.76	1.44
	24	1.96	2.96	1.96	2.98
	48	2.01	3.13	2.01	3.16
ANN4	1	0.74	2.54	0.74	1.73
	24	1.94	2.59	1.94	2.70
	48	1.98	3.02	1.98	3.07
ANN10	1	0.71	2.68	0.71	2.80
	24	1.95	3.09	1.95	3.13
	48	1.97	3.80	1.97	3.83

Table 2.7: Average pinball-loss and modified reliability deviation values of non-parametric (i.e. quantile regressions) and parametric distribution forecasts; H : forecast horizon; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

Furthermore, Table 2.8 contains the percentage of times that a parametric CDF tested is selected as a best fit.

Technique	H	% of best fitting distributions chosen			
		Beta	Gamma	Normal	Uniform
Poly1-3	1	56.75	29.61	11.59	2.05
	24	42.58	26.06	31.36	0
	48	39.09	21.22	39.69	0
ANN4	1	29.35	40.49	10.64	19.53
	24	37.02	20.91	41.94	0.14
	48	30.01	19.31	50.34	0.34
ANN10	1	0.44	0.54	0.19	98.83
	24	30.92	28.52	40.34	0.22
	48	31.17	18.07	50.28	0.49

Table 2.8: Percentage of the chosen parametric distribution that best fits the non-parametric estimates; H : forecast horizon

Table 2.8 shows that for $H = 24$ and $H = 48$ the Uniform distribution is either not chosen or chosen less than 1% as the best fit, in contrast all other distributions are more or less evenly selected as the best fit, with the Gamma distribution being the one less chosen. For $H = 1$ the Uniform distribution stops being disregarded and is surprisingly chosen for ANN10 almost 100% of the time as the best fit.

In addition to the previous results, it is also necessary to take a look at the mean squared errors (MSEs) obtained by the selected parametric CDFs, since a parametric CDF is considered to best fit the non-parametric estimate when its MSE value is the lowest compared to the other parametric CDFs used (cf. Equation (2.17)). For the sake of interpretability, the results presented in the current section are the average of the square root of the MSE values obtained over the whole test set, i.e. the average root mean squared errors (RMSEs). Table 2.9 contains RMSEs obtained for every forecast horizon and data mining technique used.

H	RMSE [%]		
	Poly1-3	ANN4	ANN10
1	2.0	2.7	2.7
24	0.5	1.4	1.1
48	0.5	1.3	1.8

Table 2.9: Average root mean square error obtained by the parametric distributions considered the best fit for each data mining technique used; H : forecast horizon

The low RMSE values (all below 3%) contained in Table 2.9 demonstrate that the present thesis method is able to find parametric CDFs that accurately fit the non-parametric CDFs given by the used quantile regressions. It can also be seen that for the used data, the present contribution method results in parametric CDFs that are able to fit non-parametric CDFs based on polynomial models better than those based on artificial neural networks and that are also better at fitting forecasts for forecast horizons $H = 24$ and $H = 48$ than for $H = 1$. It is important to mention, that since the present thesis method does not restrict the use of other parametric distributions, the fit to the non-parametric CDFs can always be improved regardless of the time series being forecast.

The final experiment conducted is based on the following idea. If the parameters of a parametric CDF are fitted to the non-parametric estimate (using for instance a non-linear least squares approach), the result will have (i) a different expected value and variance than the non-parametric CDF and (ii) a RMSE value lower than or equal to that of the parametric CDF considered the best fit via the present thesis method. Therefore, the goal of this final experiment is to determine how well does the CDF, considered the best fit according to the present thesis method, fares in comparison to one who may be the optimal fit (e.g., one whose parameters have been fitted to minimize its RMSE). To achieve this goal, expected values and variances around the ones determined in Equations (2.14) and (2.15) are calculated. Thereafter, the best fitting parametric CDF from the four ones tested is determined

using the present thesis methodology for all combinations of the modified expected values and variances. Afterwards, the lowest obtained RMSE is determined and the procedure is repeated across the whole test set. Finally, the average of all the minimal RMSE values is calculated and compared to the values contained in Table 2.9. For the sake of illustration, Figure 2.27 shows how the RMSE values of a single forecast change when varying the expected value and the variance separately. Additionally, Table 2.10 contains the test set's average difference of the lowest obtained RMSE values and the ones contained in Table 2.9.

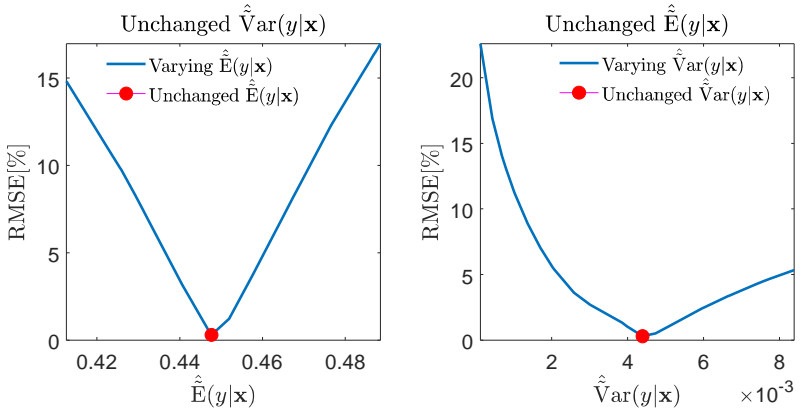


Figure 2.27: Root mean square error plotted against different expected values and variances

H	RMSE Difference [%]		
	Poly1-3	ANN4	ANN10
1	0.930	0.600	0.022
24	0.062	0.069	0.060
48	0.093	0.074	0.094

Table 2.10: Difference between the average root mean square error obtained with the present thesis method and the possible average root mean square error obtained by a parametric distribution considered an optimal fit; H : forecast horizon

As it can be seen the difference in RMSE values is always lower than 1%, hence supporting the fact that the presented method is truly able to find parametric CDFs that fit in an acceptable manner a given non-parametric estimate without requiring a computationally intensive fitting of parameters.

2.4.5 Scenario Forecast

The present section offers an experiment to answer the question raised in Section 2.4.5. This experiment consists in the calculation and evaluation of scenario forecasts using quantile regressions based on the NNQF trained on 100% of the training data coming from the time series with 45% of noise¹⁵. The scenario forecasts, which are created, are set to obtain forecasts in a rolling 24 hour period. Furthermore, the number of scenarios used in the creation of the different scenario forecasts are $N_s = \{10, 50, 100\}$. Moreover, the threshold values in Equation (2.20) are chosen as $y_{S_{\min}} = 0$ and $y_{S_{\max}} = 1$ and the quantile estimates calculated from the obtained scenarios are those with probabilities $q \in \{0.01, \dots, 0.99\}$. Note that the scenario forecasts presented are obtained only with the Poly1-3 techniques¹⁶. Scenario forecasts based on the ANN models are not created, because the current implementation of the present thesis method and the length of the test set used makes their obtainment extremely slow. For instance, the creation of the desired scenario forecasts using only 10 scenarios for the whole test set takes approximately 54 hours, which in turn will result in the calculation of scenario forecasts based on 100 scenarios taking approximately 22 days.

The pinball-loss and modified reliability deviation values obtained by each quantile estimate of the scenario forecasts are plotted in Figure 2.28 against their corresponding probability value q . For the sake of comparison, the results obtained by the quantile regressions based on the NNQF for $H =$

¹⁵ The use of the time series with 45% of noise is considered to be enough to fulfill the present section's goal, i.e. to compare the scenario forecasts to quantile regressions whose forecasting goal is the same.

¹⁶ Note that all polynomial models are the same, since they are trained using the same selected features. Therefore, they are referred in the present section together as Poly1-3.

24 and their optimal number of nearest neighbors (cf. Table 2.5) are also depicted.

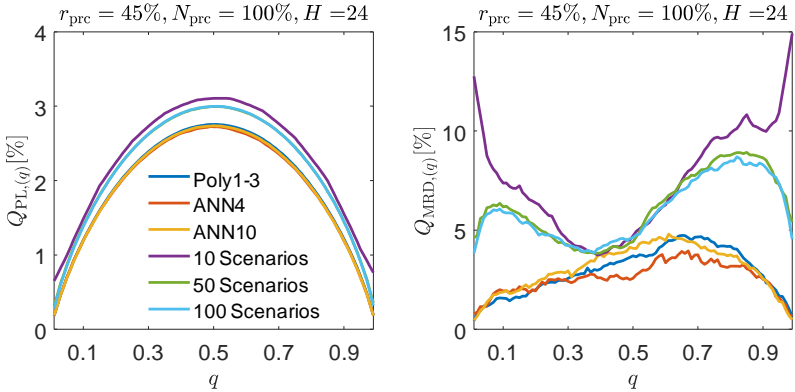


Figure 2.28: Comparison between the pinball-loss and modified reliability deviation values of the scenario forecast and NNQF-based quantile regressions; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

As it can be seen in Figure 2.28, the scenario forecasts are worse at describing the uncertainty of the next 24 hours than the normal quantile regressions. Nonetheless, the magnitude in the difference regarding the pinball-loss is quite different as that of the modified reliability deviation. First, it can clearly be seen, that the scenario forecasts obtain in general a higher pinball-loss value than the quantile regression used for comparison. However, the difference in pinball-loss never surpasses 1%. It is also clearly seen that the worse pinball-loss values are obtained by the scenario forecasts formed by only 10 scenarios. In contrast, the difference in pinball-loss between the other scenario forecasts is almost indistinguishable. The scenario forecasts also perform worse than the quantile regressions in terms of the modified reliability deviation. Just as before, the worst results are obtained by the scenario forecast formed by the lowest number of scenarios, with values sometimes surpassing 10%. All other scenario forecasts obtain

similar modified reliability deviations that never surpass 10%, but that are always higher than those obtained by the quantile regressions. Interestingly, the largest deviations are mostly found on the extremes of the distributions. This could be caused by the fact that scenario forecasts may underestimate the uncertainty. The reason thereof is that the quantile regressions used for the scenario forecasts are trained using measured values as input (cf. Equation (2.18)), hence they may be unable to correctly propagate the uncertainty when calculating the scenarios, i.e. when using forecast values as input (cf. Equation (2.20)). Evidence of this can be obtained by taking a look at the average interval width Q_{IW} (cf. Equation (2.31)) of intervals formed by the scenario forecasts and the quantile regressions compared to them¹⁷. The intervals of the best scenario forecast obtain a Q_{IW} that is 20%, 16%, and 22% smaller than that of the interval forecast trained using Poly1-3, ANN4, and ANN10, respectively. Therefore, it can be concluded that the scenario forecasts clearly underestimate the spread of the future values uncertainty.

As a final result, Table 2.11 contains the average values of the pinball-losses and modified reliability deviations shown in Figure 2.28.

	Technique			Number of Scenarios		
	Poly1-3	ANN4	ANN10	10	50	100
$Q_{PL}[\%]$	1.96	1.94	1.95	2.32	2.19	2.19
$Q_{MRD}[\%]$	2.96	2.59	3.09	7.41	6.18	6.00

Table 2.11: Comparison of average pinball-loss and modified reliability deviation values of scenario forecast and NNQF-based quantile regressions; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

Just as in Figure 2.28, Table 2.11 shows that the average pinball-loss of the scenario forecast is quite similar to that of the quantile regressions. In contrast, the modified reliability deviation values of the scenario forecasts are much higher, but are still not above 10%. In general it can be concluded,

¹⁷ The intervals are formed by pairs of quantile estimates centered on the estimated median and have probabilities $(q_u - q_l) \in \{0.02, 0.04, \dots, 0.98\}$ of containing the future values.

that the present thesis method is able to obtain – at least for the simulated time series used – more or less acceptable scenario forecasts. Regardless, the method still needs to be improved to reduce the underestimation of the forecast uncertainty. In the meantime, the other presented method should be preferred in cases in which an accurate description of the uncertainty is needed, but scenario forecasts are not strictly required.

Notice as a final remark, that the evaluation of the scenario forecasts using the pointwise evaluation metrics (i.e. the average pinball-loss and modified reliability deviation) is considered adequate for the experiment presented in the current section. Nonetheless, evaluation values for multivariate distributions (as e.g., the energy score [174]) are considered to be better at evaluating scenario forecasts, especially when comparing different scenario forecasting techniques. This is to be taken into consideration in future related works.

2.5 Novelty and Remaining Questions

The current chapter describes several methods whose novelty can be summarized as follows:

1. The possibility of training quantile regressions without the need of using Equation (1.16), thus opening the possibility of training quantile regressions with complex data mining techniques (i.e. ANN) without much effort, since their training algorithms remain unchanged (cf. Section 2.2.1)
2. The obtainment of interval forecasts based on the new type of quantile regressions (cf. Section 2.2.2)
3. The straightforward obtainment of non-parametric distribution forecasts (cf. Section 2.2.3)

4. The feasibility of obtaining parametric distribution forecasts (i.e. parametric probabilistic forecasts) without requiring a distribution assumption (cf. Section 2.2.4)
5. The creation of scenario forecasts without the need of explicitly describing the correlation structure between neighboring time series values (cf. Section 2.2.5)

Additionally, the acceptable results obtained in the experiments demonstrate that the developed methods offer possible solutions to the first three open questions in Section 1.4 and thus fulfill the first three objectives of the current thesis (cf. Section 1.5). Nonetheless, a series of new question can also be raised, for instance:

- How do different values of the parameter ϵ_{th} and different distance measures (cf. Equation (2.1)) influence the quality of the NNQF-based quantile regressions?
- How do quantile regressions based on the NNQF fare in comparison to “true” quantile regressions trained with more complex data mining techniques, as e.g., ANNs?
- Can the results of quantile regressions with different probabilities (trained with the same data mining technique) be improved by letting them choose their features independently of one another?
- Does the use of a non-linear interpolation (cf. Equation (2.10)) influences the quality of the non-parametric CDF forecasts and of all forecasts based on them?
- Does the selection of different $\hat{F}_s(y|\mathbf{x})$ and $\hat{F}_e(y|\mathbf{x})$ (cf. Equation (2.11)) functions influence the quality of the non-parametric CDF forecasts and of all forecasts based on them?

- What improvements can be done to the scenario forecast method to avoid underestimating the forecast uncertainty?
- How can the implementation of the scenario forecast approach be optimized, for it to allow the creation of scenario forecasts based on more complex data mining techniques (e.g., ANNs) in an acceptable amount of time?

Questions as the ones previously described are to be tackled in future related works.

3 Hierarchical Probabilistic Forecasting

3.1 Overview

The current chapter presents a concept for estimating the joint distribution of two correlated future time series values, as well as for obtaining coherent hierarchical probabilistic forecasts based only on several joint distribution forecasts. A coherent hierarchical forecast is a collection of forecasts for all time series in a hierarchy in which the forecasts of each time series actually represent the sum of those forming it. Since time series are normally forecast independently of one another, the coherency of the hierarchical forecasts is not assured. Furthermore, additional problems arise when trying to estimate coherent hierarchical probabilistic forecasts; since estimating the CDF of the sum of several correlated values requires the estimation of a multivariate CDF, which is not trivial to compute. Therefore, the current chapter offers an alternative based on the sequential estimation of joint distribution functions. An example of a time series hierarchy is given in Figure 3.1. The time series in the third level in Figure 3.1 represent, for example, loads of various households; with those in the second level representing loads at the substation level, and the time series at the top being the aggregation of all substations.

The present chapter is divided in three main sections. The first presents a straightforward method for obtaining joint distribution forecasts, the second describes an approach for obtaining coherent probabilistic hierarchical forecasts, and the last describes the approaches' novelty and some re-

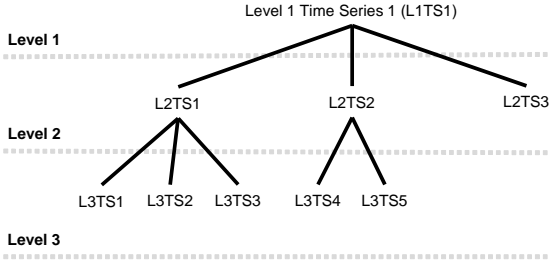


Figure 3.1: Example of a time series hierarchy

maining open questions. Furthermore, a series of examples are also presented. The data used for these examples stems from three correlated simulated load time series, $\{P_{Li}[k]; k = 1, \dots, K; i = 1, 2, 3\}$, and their sum $\{P_{LS}[k]; k = 1, \dots, K\}$. These time series represent three years of hourly measurements, i.e. $K = 26280$ and are obtained using the same benchmark generator as in [3, 156], as well as some additional modifications; more information thereof can be found in Appendix B.1. The used time series are assumed to represent the load of three really similar households, as well as the load measured at a substation they are all connected to. For the sake of illustration, Figure 3.2 shows a segment of the used time series.

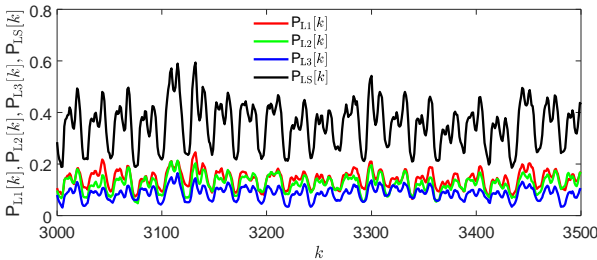


Figure 3.2: Simulated load time series used in the examples of the present chapter. Please refer to Appendix B.1 for more information about their obtainment.

3.2 Joint Distribution Forecast

The presented method begins with two correlated time series further referred to as $\{y_a[k]; k \in [1, K]\}$ and $\{y_b[k]; k \in [1, K]\}$. The goal is to estimate the joint distribution of the correlated future values $y_a[k + H]$ and $y_b[k + H]$, conditioned on some available information. For instance, to be consistent with Chapter 2 the desired outputs and the used input vector can be defined as follows:

$$\begin{aligned}
 y_a &:= y_a[k + H] \\
 y_b &:= y_b[k + H] \\
 \mathbf{x} &:= [y_a[k], \dots, y_a[k - H_1], y_b[k], \dots, y_b[k - H_1], \\
 &\quad \mathbf{u}^T[k], \dots, \mathbf{u}^T[k - H_1]]^T; k \in [H_1 + 1, K - H].
 \end{aligned} \tag{3.1}$$

Using the definitions given in Equation (3.1), $L \in \mathbb{N}_{>0}$ separate quantile regressions based on the NNQF are trained for each of the correlated values (cf. Section 2.2.3), i.e. $\{\hat{y}_{a,(q_l)}; l = 1, \dots, L \text{ and } q_1 < \dots < q_L\}$ and $\{\hat{y}_{b,(q_l)}; l = 1, \dots, L \text{ and } q_1 < \dots < q_L\}$. Afterwards, non-parametric estimates of the conditional CDFs of y_a and y_b , i.e. $\hat{F}(y_a|\mathbf{x})$ and $\hat{F}(y_b|\mathbf{x})$, are obtained (cf. Equation (2.12)). The estimated CDFs represent the marginal distributions of the joint CDF that is to be estimated. Note that the later used conditional Sklar theorem (cf. Equation (1.10)) requires the marginal distributions to be conditioned on the same value as the joint distribution, hence all regressions are trained with the same input vector.

After estimating the marginal distributions, L additional quantile regressions are trained, these regressions estimate quantiles of y_a conditioned on y_b and vice versa and are given further as:

$$\begin{aligned}
 \hat{y}_{a|b,(q_l)} &= f(y_b, \hat{\boldsymbol{\theta}}_{(q_l)}) \\
 \hat{y}_{b|a,(q_l)} &= f(y_a, \hat{\boldsymbol{\theta}}_{(q_l)});
 \end{aligned} \tag{3.2}$$

where $\hat{y}_{a|b,(q_l)}$ is an estimated quantile of y_a conditioned on y_b and $\hat{y}_{b|a,(q_l)}$ represent an estimated quantile of y_b conditioned on y_a .

By using all $\hat{y}_{a,(q_l)}$ and $\hat{y}_{b,(q_l)}$ as input in the regressions outlined in Equation (3.2), $2L^2$ estimates¹ of the quantiles of y_a and y_b conditioned on one another – and indirectly on \mathbf{x} – can be obtained, i.e.:

$$\begin{aligned}\hat{y}_{a|b,(q_l)j} &= f(\hat{y}_{b,(q_l)j}, \hat{\boldsymbol{\theta}}_{(q_l)}); l = 1, \dots, L, j = 1, \dots, L \\ \hat{y}_{b|a,(q_l)j} &= f(\hat{y}_{a,(q_l)j}, \hat{\boldsymbol{\theta}}_{(q_l)}); l = 1, \dots, L, j = 1, \dots, L.\end{aligned}\quad (3.3)$$

Afterwards, all realizations are converted into realizations of uniformly distributed random variables via the probability integral transform (i.e. using $\hat{F}(y_a|\mathbf{x})$ and $\hat{F}(y_b|\mathbf{x})$); since the joint distribution is later estimated using a copula (cf. Equation (1.10)). In addition, since bivariate samples are required for calculating the joint distribution of the future time series values, each transformed realization of y_a conditioned on y_b has to be paired with the q value of the quantile of y_b used for its obtainment. The same has to be done for each transformed realization of y_b conditioned on y_a . This results in a pair of vectors containing $2L^2$ values that are defined as:

$$\begin{aligned}\hat{\mathbf{u}}_{a|b} &= [\hat{u}_{a|b,1}, \dots, \hat{u}_{a|b,2L^2}]^T \\ &= [F(\hat{y}_{a|b,(q_1)1}|\mathbf{x}), \dots, F(\hat{y}_{a|b,(q_L)L}|\mathbf{x}), q_1 \cdot \mathbf{1}_{1 \times L}, \dots, q_L \cdot \mathbf{1}_{1 \times L}]^T \\ \hat{\mathbf{u}}_{b|a} &= [\hat{u}_{b|a,1}, \dots, \hat{u}_{b|a,2L^2}]^T \\ &= [q_1 \cdot \mathbf{1}_{1 \times L}, \dots, q_L \cdot \mathbf{1}_{1 \times L}, F(\hat{y}_{b|a,(q_1)1}|\mathbf{x}), \dots, F(\hat{y}_{b|a,(q_L)L}|\mathbf{x})]^T;\end{aligned}\quad (3.4)$$

Finally, the transformed samples can be used to estimate a copula that in turn will provide an estimate of the joint CDF using the conditional Sklar

¹ The number $2L^2$ stems from the fact that Equation (3.3) estimates L quantiles for each of the L quantile regressions of the 2 correlated time series.

theorem (Equation (1.10)). For instance the empirical copula [36]² can be used:

$$\begin{aligned}\hat{F}(y_a, y_b | \mathbf{x}) &= C(\hat{F}(y_a | \mathbf{x}), \hat{F}(y_b | \mathbf{x}) | \mathbf{x}) \\ &= \frac{1}{2L^2} \text{card}_{i=1, \dots, 2L^2} \{ \hat{u}_{a|b,i} \leq \hat{F}(y_a | \mathbf{x}) \wedge \hat{u}_{b|a,i} \leq \hat{F}(y_b | \mathbf{x}) \};\end{aligned}\tag{3.5}$$

where $\text{card}\{\cdot\}$ is the cardinality operator.

For the sake of illustration, Example 6 shows the obtainment of a joint distribution forecast based on the present approach.

Example 6: Joint Distribution Forecast

The current example shows the obtainment of bivariate forecasts using the previously described method. The experiment begins by reformulating Equation (3.1) as follows:

$$\begin{aligned}y_a &:= P_{L1}[k + H]; H = 24 \\ y_b &:= P_{L2}[k + H]; H = 24 \\ \mathbf{x} &:= [P_{L1}[k], \dots, P_{L1}[k - H_1], P_{L2}[k], \dots, P_{L2}[k - H_1], \\ &\quad P_{L3}[k], \dots, P_{L3}[k - H_1]]^T; k \in [H_1 + 1, K - H], H_1 = 168.\end{aligned}\tag{3.6}$$

Thereafter, several quantile regressions based on the NNQF with probabilities $q = \{0.01, 0.02, \dots, 0.99\}$ are trained using a polynomial with a maximal degree of 2. Similarly as in the examples of Chapter 2, the quantile regressions are trained to estimate quantiles of values 24 hours in advance using the previous week values of the three correlated time series. The reason for using the three time series as input, when only two are to be forecast, is the fact that the joint distribution forecast created is later used in the hierarchical forecast example in which the input vector has to be the same; as it is necessary for the conditional Sklar theorem. Figure 3.3 shows the non-parametric marginal

² Since the marginal distributions are estimated and thus assumed to be known, the empirical copula in Equation (3.5) is equal to the empirical probability measure; cf. [http://www.columbia.edu/rf2283/Conference/3Inference%20\(2\)%20Seagers.pdf](http://www.columbia.edu/rf2283/Conference/3Inference%20(2)%20Seagers.pdf).

distributions obtained with the method described in Section 2.2.3.

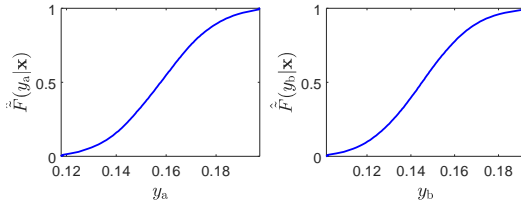


Figure 3.3: Example of the forecast marginal distributions

Afterwards, additional quantile regressions (cf. Equation (3.2)) with probabilities $q = \{0.01, 0.02, \dots, 0.99\}$ are trained. These are simple linear quantile regressions. Finally, using Equations (3.3) and (3.4), as well as the empirical copula shown in Equation (3.5), the desired joint distribution can be determined. This is depicted in Figure 3.4.

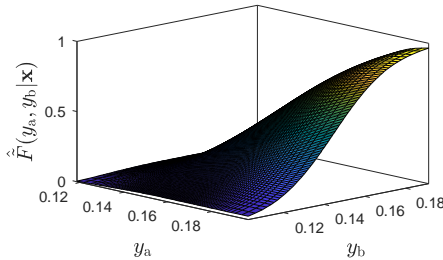


Figure 3.4: Example of the joint distribution forecast

3.3 Hierarchical Probabilistic Forecast

The presented approach is based on the possibility of estimating the CDF of the sum of two correlated values using their joint distribution function and on the following assumptions:

- *The values for which the joint distributions are obtained are all constrained by an upper and a lower threshold, i.e. $y_a \in [y_a, \bar{y}_a]$ and $y_b \in [y_b, \bar{y}_b]$*

- The estimated CDF of the sum is strictly monotonic increasing.

Under the previous assumptions the conditional CDF of y_{a+b} , i.e. the sum of y_a and y_b , can be estimated as follows:

$$\hat{F}(y_{a+b}|\mathbf{x}) = \int_{\underline{y}_b}^{y_{a+b}-\underline{y}_a} \int_{\underline{y}_a}^{y_{a+b}-y_b} \frac{\partial \hat{F}(y_a, y_b|\mathbf{x})}{\partial y_a \partial y_b} dy_a dy_b . \quad (3.7)$$

Using the methods described in Section 3.2 and Equation (3.7) a hierarchical forecast for hierarchies formed by two or more time series can be obtained. To do so, a given hierarchy has to be modified. For example, Figure 3.5 depicts the modification of the hierarchy shown in Figure 3.1. Once the hierarchy has been modified an iterative procedure can be used to calculate the CDF of the sum of each new pair, which results in the obtainment of coherent hierarchical probabilistic forecasts for each time series of the original hierarchy.

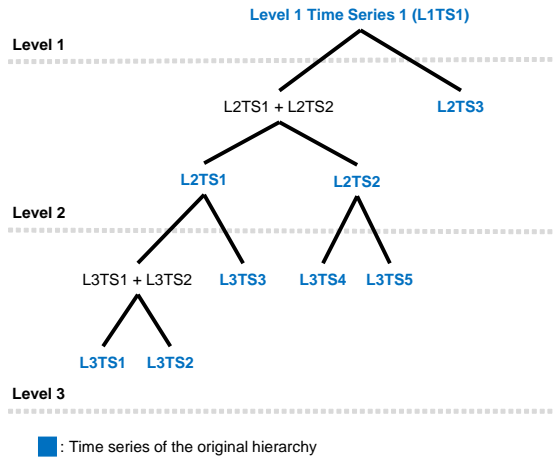


Figure 3.5: Example of a modified time series hierarchy

The main advantage of the present method over simply obtaining probabilistic forecasts for each time series independently is the fact that each probabilistic forecast obtained is coherent; if their joint distribution estimates are correct. This means that each probabilistic forecast in a specific level of the hierarchy actually represents the sum of the probabilistic forecasts at its lower levels. Note that the present method can be classified as a bottom up hierarchical forecasting method, since it bases the forecasts of the time series at higher levels completely on those at the lower levels³.

Example 7 shows results obtained by the present thesis' hierarchical forecasts.

Example 7: Hierarchical Forecast

The goal of the current experiment is to use the previously described method to create hierarchical forecasts for the sum of the three correlated time series shown in Figure 3.2, i.e. P_{LS} , and to compare the hierarchical forecast results to those of direct forecasts, i.e. quantile regressions that directly estimate the uncertainty of the aggregated time series future values. Note that the input vector \mathbf{x} and quantile regressions used for obtaining the necessary joint distribution forecasts are the same as in Example 6. Furthermore, the input vector and desired output for the direct forecasts are chosen as follows:

$$\begin{aligned} y &:= P_{LS}[k + H]; H = 24 \\ \mathbf{x} &:= [P_{LS}[k], \dots, P_{LS}[k - H_1]]^T; k \in [H_1 + 1, K - H], H_1 = 168. \end{aligned} \quad (3.8)$$

The direct forecasts are quantile regressions based on the NNQF with corresponding probabilities $q = \{0.01, 0.02, \dots, 0.99\}$ that are trained as polynomials with a maximal degree of two. As Equation (3.8) shows, the regressions are trained to estimate the quantiles of a value 24 hours in the future using only the previous week values of the aggregated time series as input. In addition, hierarchical forecasts obtained under the assumption of the time series

³ A more thorough description of the difference between bottom up and top down hierarchical forecasts can be found in [25].

future values being independent of one another are also obtained (i.e. the distribution of their future values' sum is obtained via convolution operations) and compared. Figure 3.6 plots the results obtained in terms of pinball-loss and modified reliability deviation against the probability of the quantiles being estimated. Note that the pinball-losses presented are obtained by dividing the obtained errors with the difference between the maximal and minimal value of the sum time series.

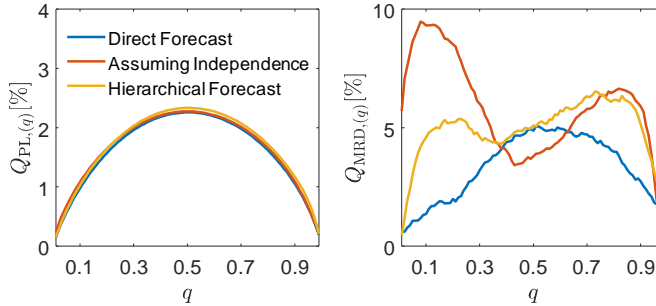


Figure 3.6: Results from the direct forecasts, the forecasts obtained assuming independence, and the forecasts created via the present thesis method; referred to in the figure as hierarchical forecast.

Figure 3.6 shows that the three types of forecast perform extremely similar in terms of pinball-loss. However, differences appear when looking at their modified reliability deviations. For instance, it is clear that the direct forecast delivers the best results in most cases. This is an expected outcome, as the direct forecast do not estimate the quantiles by aggregating some lower level forecasts (a calculation that may add additional forecasting errors), but rather estimate the values directly. Interestingly for q values close to 0.5, the forecast obtained under the assumption of independence performs better than the direct forecast, while those created with present thesis method obtain a similar result. Moreover, the advantage of using the present thesis method, instead of ignoring the dependence of the future time series values, is shown on the results obtained for the lower and higher quantiles. For instance, the present thesis method obtains much better estimates for $q < 0.4$ than the forecasts created assuming the independence of the future values. Likewise, a slight advantage of the present thesis approach can also be seen for $0.6 < q < 0.9$. Thereupon,

it can be argued that ignoring the – positive – correlation of the future values may result in an underestimation of their sum’s uncertainty. Figure 3.7 depicts a possible occurrence of this effect, as it shows that the intervals obtained with the forecasts that assume independence are generally smaller than those created via the present thesis’ method; thus pointing at the former possibly underestimating the sum’s variance.

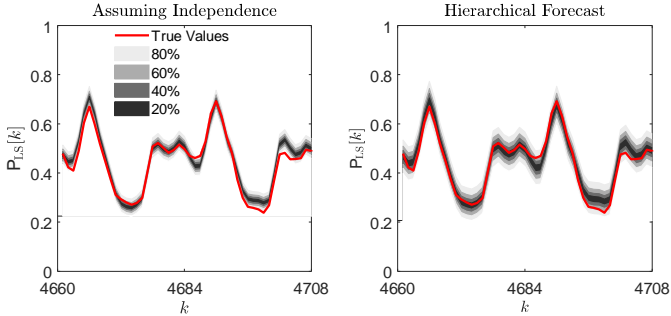


Figure 3.7: Interval forecasts obtained with the forecasts that assume independence of the summed values and those obtained by the present thesis approach, referred in the figure as hierarchical forecast.

An argument that can be made after looking at the results (especially those in Figure 3.6), is that direct forecasts should be preferred. However, this may only apply in cases in which information about the relationship of time series forming a hierarchy is irrelevant. The moment in which information of how the time series on different hierarchy levels influence each other becomes important, coherent hierarchical forecast should be used – as direct forecast do not provide that information in any way. Finally it needs to be mentioned, that future related works should try to improve the concept presented herein, for instance, using reconciliation approaches as the one shown in [85].

3.4 Novelty and Remaining Questions

The current chapter tries to answer the fourth open question in Section 1.4 by providing the concept of two methods that:

1. Obtain a non-parametric estimate of the joint probability distribution of two correlated future time series values based only on quantile regressions
2. Create a coherent hierarchical probabilistic forecast using only joint distributions and an iterative process

Even though the described approaches fulfill the thesis objective of offering straightforward methods for creating joint distribution and hierarchical coherent probabilistic forecasts, there are still issues that can be raised; for instance:

- Do the quality of the joint distribution forecasts is affected if a lower number of bivariate samples (cf. Equation (3.4)) are used?
- Do the data mining techniques used for the quantile regressions in Equation (3.2) and those used for the quantile regressions forming the marginal distributions have to be the same?
- Are the obtained hierarchical forecasts truly coherent?
- Can the accuracy of the coherent probabilistic forecasts be improved via reconciliation approaches, as e.g., in [85]?
- How does the accuracy of the obtained hierarchical probabilistic forecasts changes at different levels of the hierarchy?
- Are there easier and more comprehensive approaches that may result in similar forecasts?

Future related works should tackle some of the questions listed above.

4 Software Implementation

4.1 Overview

The current chapter discusses the Forecasting Extension of the open-source MATLAB toolbox SciXMiner [164], i.e. the implementation of the methods described in Chapter 2 and the concepts presented in Chapter 3. Currently, the SciXMiner toolbox already allows for the creation of time series point forecasting models. The new extension not only streamlines the previous functionality, but also widens it to allow the creation of probabilistic forecasts. Table 4.1 outlines the sections describing the theory and implementation of the various methods presented. The following sections offer a general description of the implementation of each method.

Method	Section	
	Theory	Implementation
Quantile Regressions based on the NNQF	2.2.1	4.2.1
Interval Forecast	2.2.2	4.2.2
Parametric Distribution Forecast	2.2.4	4.2.3
Scenario Forecast	2.2.5	4.2.4
Hierarchical Probabilistic Forecast	3.3	4.2.5

Table 4.1: Sections describing the theory and implementation of the present thesis methods

Additionally, the present chapter describes a procedure (cf. Section 4.3) to decouple the forecasting approaches from SciXMiner. This simplifies, for instance, their later implementation in the forecasting services being developed for the Energy Lab 2.0 [10].

4.2 SciXMiner Forecasting Extension

The extension implemented in SciXMiner allows the selection of various types of forecast. Figure 4.1 depicts the graphic user interface (GUI) of the programmed Forecasting Extension and shows the different types of forecasts that can be obtained. Note that the methods enumerated in the image are the ones more thoroughly described in the next sections. In addition, the extension also contains GUIs for creating and applying point forecasting models and polynomial based “true” quantile regressions¹, i.e. obtained by minimizing the sum of pinball-losses. These are not discussed further in the present work.

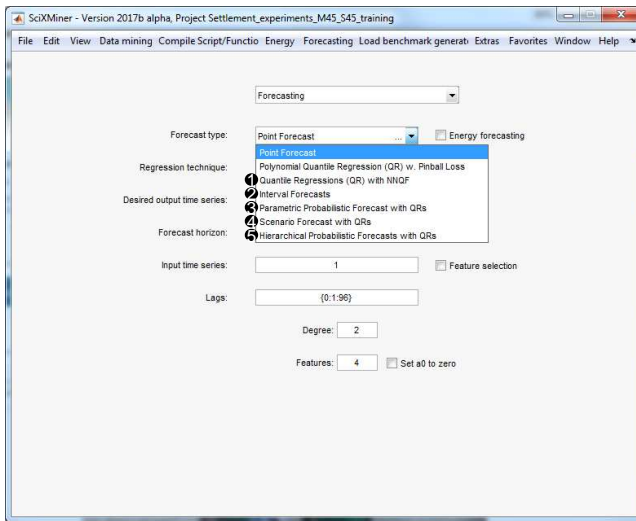


Figure 4.1: Different types of forecasts available in SciXMiner: 1: Quantile regression based on the NNQF; 2: Interval forecasts; 3: Parametric distribution forecasts; 4: Scenario forecasts; 5: Hierarchical probabilistic forecasts

¹ The implementation of the “true” quantile regressions is based on de.mathworks.com/matlabcentral/fileexchange/32115-quantreg-x-y-tau-order-nboot-

4.2.1 Quantile Regressions based on the NNQF

Figure 4.2 shows the GUI developed for creating quantile regressions based on the NNQF. As it can be seen, the GUI is divided in four fields which are described below².

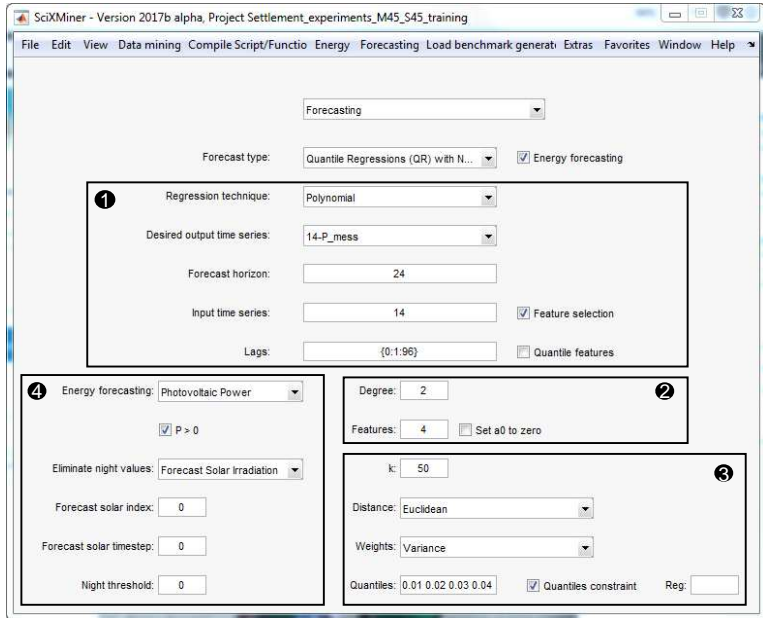


Figure 4.2: User interface for quantile regressions based in the NNQF;
1: General options; 2: Data mining technique parameters; 3:
NNQF parameters; 4: Energy forecasting specific options

1. General options: This field contains options which are not specific to quantile regressions based on the NNQF, as e.g., the time series to be forecast and the time series used as input. Table 4.2 contains information about each general option available.

² Options for normalizing the input matrix can be used when creating the quantile regressions, but cannot be defined directly in the Forecasting Extension.

GUI	Symbol	Equation	Additional Description
Regression technique	–	–	Data mining technique to be used (i.e. Polynomial, ANN or SVR)
Desired output time series	–	–	Time series to be forecast
Forecast horizon	H	(1.21)	Timesteps representing the forecast horizon
Input time series	–	–	Indexes of the time series whose values are used as input
Lags	H_1	(1.21)	Value representing the number of lags given as $\{0 : 1 : H_1\}$
Feature selection	–	–	Select to apply a feature selection
Quantile features	–	–	Select to apply an individual feature selection for each quantile regression

Table 4.2: General options of the quantile regressions based on the NNQF GUI

2. Regression technique parameters: The second field varies depending on the regression technique being used. For instance, in the case of polynomial regressions the maximal allowed degree and the option of setting the constant value to zero are shown, while in the case of ANNs (i.e. MLPs) the number of hidden layers and hidden neurons in each hidden layer are displayed. Furthermore, the number of features being selected – if a feature selection is to be applied – is also found in the present GUI field.

3. NNQF parameters: The third field contains all options necessary for the NNQF. Table 4.3 contains information about each available option.

GUI	Symbol	Equation	Additional Description
k	N_{NN}	(2.1)	Number of nearest neighbors
Distance	$d(\cdot, \cdot)$	(2.1)	Distance measure used for the NNQF
Weights	–	–	Weights applied to the features when calculating distances
Quantiles	q	(2.1)	Probabilities of the quantiles to be estimated by the regressions
Quantile constraint	–	(2.8), (2.9)	Apply the current thesis approaches to avoid quantile crossing
Reg	ϵ_r	(2.9)	Regularization value; if left empty $\epsilon_r = 10^{-5}$ is used

Table 4.3: NNQF options of the quantile regressions based on the NNQF GUI

4. Energy forecasting options: Last, the final field offers options available when forecasting energy time series. The field appears only if the energy forecasting checkbox in the GUI is selected. The field allows the specification of three different energy forecasting option fields for either PV, load, or wind power forecasting; these fields are selected via the energy forecasting dropdown menu. All three option fields contain a checkbox to assure that all forecasts are greater than zero, for instance by setting $y_{\min} = 0$ in Equations (2.8) and (2.9). When forecasting PV power, various options for removing night values from the training set and setting nighttime values during the models' application automatically to zero. If these options are used, an additional time series is created during the application procedure. The observations of the additional time series equal one if a value at a given timestep is considered a night value and zero otherwise.

Finally, once all desired options have been selected the quantile regressions can be created by going into the *Forecasting* menu element, then going to the *Forecasting Model (Regression)* field, and finally clicking on *Create*. All trained quantile regressions are saved in a MATLAB variable called *regr_single*. In addition, going again into the *Forecasting* menu element and into the *Forecasting Model (Regression)* field, but instead clicking on *Apply*, *Save*, or *Load* results in the following: (i) *Apply* adds to the SciXMiner project time series that contain the quantile regressions' outputs; (ii) *Save* saves the *regr_single* variable as a file with an *.fmodel* extension; (iii) *Load* enables the user to load regressions saved with an *.fmodel* extension back into SciXMiner. Notice that the previous approach is the only way for loading *.fmodel* files, since the regressions saved on those files may require functionalities that are exclusive of the SciXMiner Forecasting Extension.

4.2.2 Interval Forecast

As interval forecast are just concatenations of quantile regressions (cf. Equation (2.5)), the current interval forecast GUI is implemented to simply

plot interval forecasts formed by previously applied quantile regressions. The GUI used to create the plots is shown in Figure 4.3, while the description of each available option is contained in Table 4.4.

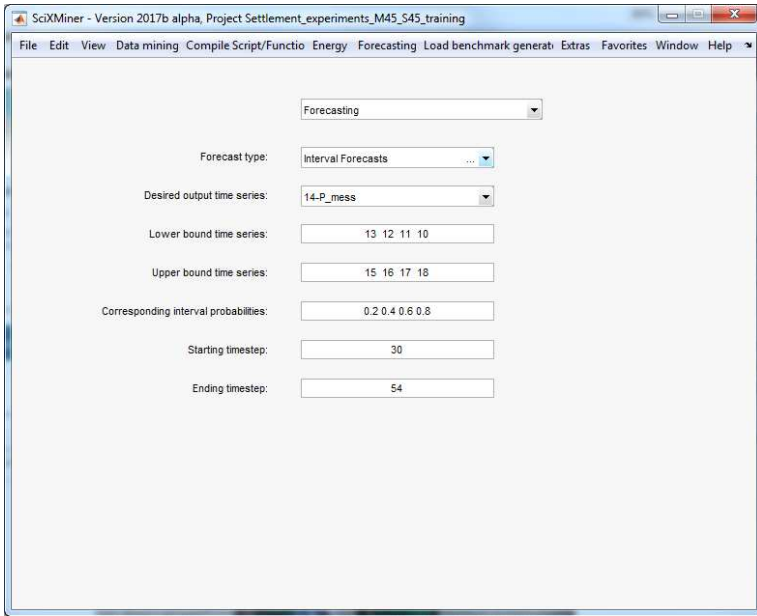


Figure 4.3: User interface for interval forecasts

GUI	Symbol	Equation	Additional Description
Desired output time series	–	–	Time series to be forecast
Lower bound time series	–	–	Index of the time series representing the lower interval bounds
Upper bound time series	–	–	Index of the time series representing the upper interval bounds
Corresponding interval probabilities	$q_u - q_l$	(2.5)	Probabilities of the intervals being created
Starting timestep	–	–	Timestep defining the start of the generated plot
Ending timestep	–	–	Timestep defining the end of the generated plot

Table 4.4: Options of the interval forecasts GUI

Once the options are selected, the desired plot is generated by first going into the *Forecasting* menu element, then going to the *Interval Forecast* field, and last, clicking on *Plot Interval Forecast*.

4.2.3 Parametric Distribution Forecast

The parametric distribution forecast GUI is implemented to obtain parametric distribution forecasts for each timestep of the time series being forecast using time series containing the quantile regressions' results. Figure 4.4 depicts the programmed GUI, which is divided in two main options fields further discussed in the current section.

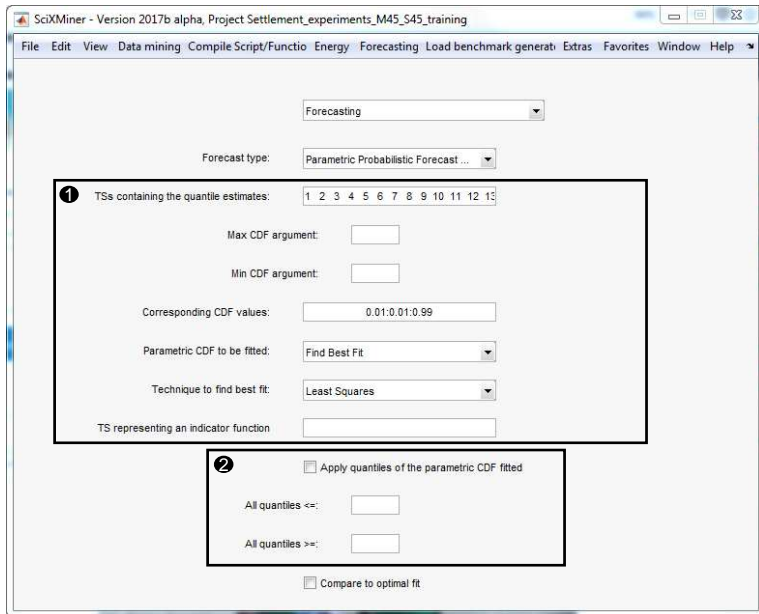


Figure 4.4: User interface for parametric forecasts; 1: Options for creating parametric distribution forecasts; 2: Options for applying the parametric distribution forecasts

1. Options for creating parametric distribution forecasts: Table 4.5 contains a brief description of the options that are necessary for creating parametric distribution forecasts.

GUI	Symbol	Equation	Additional Description
TSs containing the quantile estimates	\hat{y}_{q_l}	(2.10)	Indexes of the time series to be used for the non-parametric CDF
Max CDF argument	\bar{y}	(2.11)	The given value is used for all non-parametric CDFs estimated; if the field is left empty the value is defined as in Appendix A.3
Min CDF argument	\underline{y}	(2.11)	The given value is used for all non-parametric CDFs estimated; if the field is left empty the value is defined as in Appendix A.3
Corresponding CDF values	q_l	(2.10)	Probabilities that correspond to each quantile estimate (i.e. time series) being used
Parametric CDF to be fitted	$F_{\zeta_i}^{\zeta_i}(y \mathbf{x})$	(2.17)	Dropdown menu for selecting the parametric CDF to be fitted (i.e. Beta, Gamma, Normal, and Uniform); if <i>Find Best Fit</i> is selected the process for finding the best fit, described in Section 2.2.4, is used
Technique to find best fit	–	–	Dropdown menu for selecting the approach/metric to determine the best fit; in the present work (cf. Equation (2.17)) the <i>Least Squares</i> option is used
TS representing an indicator function	–	–	Index of a time series in SciXMiner that contains only values between 0 and 1 and can be used to define the timesteps where a parametric CDF is to be estimated (i.e. where the selected time series equals 0)

Table 4.5: Options for creating parametric distribution forecasts of the parametric distribution forecast GUI

2. Options for applying parametric distribution forecasts: The options used when applying the parametric distribution forecasts are contained in Table 4.6. The application consists in creating additional time series containing quantiles of the parametric CDFs estimated at each timestep.

After defining all necessary options, parametric distribution forecasts can be obtained by first going into the *Forecasting* menu element and then clicking on *Parametric Probabilistic Forecasts*. Afterwards, SciXMiner saves inside a variable called *cdf_bestfit* the parametric distribution estimated for each timestep of the forecast time series. Furthermore, the additional option *Compare to optimal fit* (cf. Figure 4.4) compares the best fit chosen via the

GUI	Symbol	Equation	Additional Description
Apply quantiles of parametric CDF fitted	–	–	Select if time series containing the quantile estimates are to be created and saved
All quantiles <=	–	–	Maximal value for a quantile estimate; if surpassed the estimate is replaced by this threshold; if the field is left empty no constraint is applied
All quantiles >=	–	–	Minimal value for a quantile estimate; if surpassed the estimate is replaced by this threshold; if the field is left empty no constraint is applied

Table 4.6: Options for applying parametric distribution forecasts of the parametric distribution forecast GUI

present thesis method to one which may be considered the optimal fit, as in Section 2.4.4.

4.2.4 Scenario Forecast

The scenario forecast GUI allows the obtainment of scenario forecasts using previously obtained quantile regressions (cf. Section 2.2.5). Therefore prior to creating the scenarios, the variable containing the quantile regressions, i.e. *regr_single*, has to be loaded into MATLAB. Figure 4.5 depicts the implemented GUI and how it can be divided in two main option fields.

1. Options for creating scenario forecasts: Table 4.7 contains the options for creating scenario forecast based on the present thesis method.

2. Options for creating and applying scenario forecasts: Options that can be used for creating and applying the scenario forecasts are contained in Table 4.8.

Once all options have been set, scenario forecasts can be created by first going into the *Forecasting* menu element, then going to the *Scenario Forecast* field, and finally clicking on *Create Scenario Forecasts with QR*. Thereafter, the scenario forecasts created are saved in a variable called *scenario-Forecast*. Furthermore, the scenario forecasts can be viewed by going into the *Forecasting* menu element, then to the *Scenario Forecast* field, and finally clicking on *View Scenario Forecasts*.

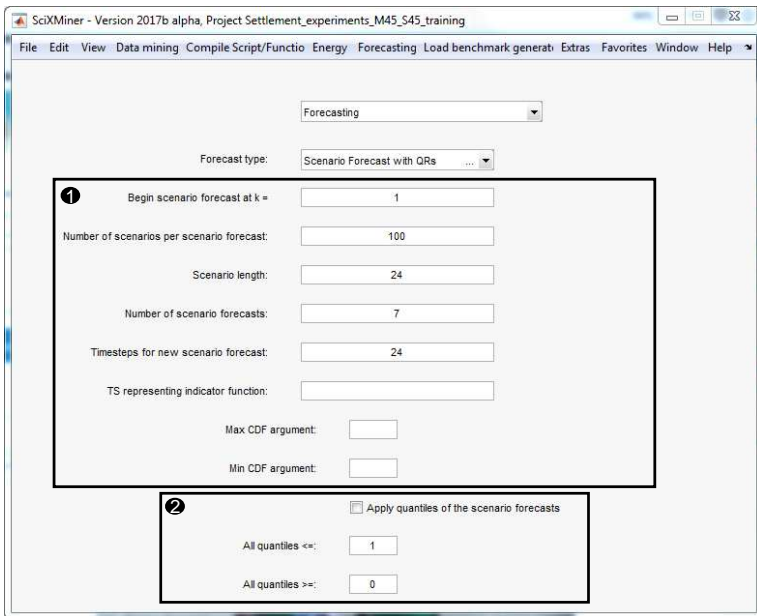


Figure 4.5: User interface for scenario forecasts; 1: Options for creating scenario forecasts; 2: Options for creating and applying scenario forecasts

GUI	Symbol	Equation	Additional Description
Begin scenario forecast at k =	–	–	Timestep to begin the first scenario forecast
Number of scenarios per scenario forecast	N_S	–	Number of scenarios per scenario forecasts (A scenario is a possible realization of a time series' future; A scenario forecast is formed by several possible scenarios)
Scenario length	H	(2.20)	Forecast horizon of the scenario forecasts to be created
Number of scenario forecasts	–	–	Number defining how many scenario forecasts are to be created
Timesteps for new scenario forecast	–	–	Timestep interval between each calculated scenario forecast
TS representing indicator function	–	–	Index of a time series in SciXMiner that contains only values between 0 and 1 and can be used to define the timesteps where a scenario forecast is set to zero (i.e. where the selected time series equals 1)
Max CDF argument	\bar{y}	(2.11)	The given value is used for all non-parametric CDFs estimated; if the field is left empty the value is defined as in Appendix A.3
Min CDF argument	\underline{y}	(2.11)	The given value is used for all non-parametric CDFs estimated; if the field is left empty the value is defined as in Appendix A.3

Table 4.7: Options for creating scenario forecasts of the scenario forecast GUI

GUI Field	Symbol	Equation	Additional Description
Apply quantiles of the scenario forecasts	–	–	Select if time series containing the quantile estimates of the scenario forecasts are to be created and saved (the quantiles are calculated on a timestep basis using Definition 5 of [166] and Method 10 in [167])
All quantiles <=	$y_{S_{\max}}$	(2.20)	Maximal allowed value for a scenario; if the field is left empty no constraint is applied
All quantiles >=	$y_{S_{\min}}$	(2.20)	Minimal allowed value for a scenario; if the field is left empty no constraint is applied

Table 4.8: Options for creating and applying scenario forecasts of the scenario forecast GUI

4.2.5 Hierarchical Probabilistic Forecast

Figure 4.6 depicts the GUI implemented for creating hierarchical probabilistic forecasts based on the present thesis' concept. Additionally, Table 4.9 contains the available options as well as a brief description about them.

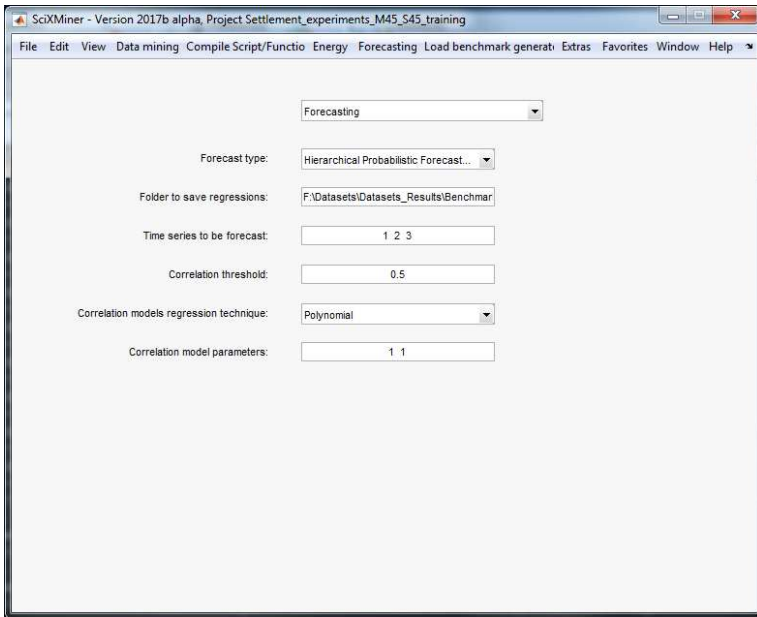


Figure 4.6: User interface for hierarchical forecasts

After the options have been defined, the hierarchical forecasts can be created by going to the *Forecasting* menu element, then going to the *Hierarchical Probabilistic Forecasts* field, afterwards going to the *Create* field, and lastly clicking on *Create QR Hierarchical Probabilistic Forecasts*. It is important to mention, that the quantile regressions (except for the ones in Equation (3.2)) are trained with the options defined in the quantile regressions' GUI (cf. Section 4.2.1). All quantile regressions are saved as files with an *.fmodel* extension in a folder called *models_hierarchicalForecast*

GUI Field	Symbol	Equation	Additional Description
Time series to be forecast	–	–	Index of time series for which the hierarchical forecast is to be created
Correlation threshold	–	–	Threshold defining if the correlation between two time series is enough to consider them correlated; if the threshold is not surpassed the joint distribution is obtained via a convolution operation
Correlation models regression technique	–	–	Dropdown menu defining the data mining technique used to create the quantile regressions in Equation (3.2)
Correlation model parameters	–	–	Parameters necessary for the data mining technique chosen previously; for instance, in the case of polynomials the first number is the maximal allowed degree and the second is the number of features to be selected

Table 4.9: Options of the hierarchical probabilistic forecasts GUI

(which in turn is saved in the folder defined in the GUI). If the *models_hierarchicalForecast* folder exists in the current MATLAB path, the hierarchical probabilistic forecasts can be created by going to the *Forecasting* menu element, then going to the *Hierarchical Probabilistic Forecasts* field, then to the *Apply* field, and finally clicking on *Apply QR Hierarchical Probabilistic Forecasts*. After the application, quantiles of the hierarchical probabilistic forecasts are added as time series.

4.3 Deployment without SciXMiner

The current section discusses the possibility of using the developed methods without the SciXMiner graphic user interface (GUI). The process for decoupling the implemented forecasting approaches from SciXMiner can be summarized in the following steps:

1. Use SciXMiner to train the models needed for creating the forecasts.
2. Develop a new “main” MATLAB script including the processes needed to obtain the desired forecasts: loading the necessary input data, pa-

rameters, and forecasting models, calling all necessary SciXMiner callbacks, etc.

3. Copy all SciXMiner scripts and functions required (i.e. all dependencies) to the folder where the “main” MATLAB script is saved.
4. Run the “main” MATLAB script (with SciXMiner closed) using an example input and compare the results to the corresponding example output.
5. If errors exist, check if they are caused by functions trying to update the SciXMiner GUI; these have to be commented out for the “main” MATLAB script to work without the SciXMiner GUI.
6. Once the “main” MATLAB script works error-free, the implemented forecasting approach can be deployed without SciXMiner

After following all previous steps, the MATLAB compiler can be used to compile the “main” MATLAB script and all its dependencies. This additional step allows for the application of the forecasting approach without the need of using MATLAB. Furthermore, an example (i.e. *example_script_2Compile.m*) and a template (i.e. *template_script_2Compile.m*) of the “main” MATLAB script can be found in the *application_specials* folder of the SciXMiner toolbox.

For the sake of illustration, Figure 4.7 depicts a pipeline implemented in the framework described in [175]; a framework that is planned to be used within the Energy Lab 2.0 [176]. The pipeline shown in Figure 4.7 starts by extracting data from a database, then conducting a forecast using the extracted data and previously trained models, and finally saving the results. Notice that the forecasting component (i.e. the block named FMForecastFunctionDocker) of the presented example was deployed using the procedure described in the present section.

It is important to mention, that other approaches for integrating forecasting methods on the Energy Lab 2.0 have also been investigated during the completion of the present thesis; e.g., using Apache Spark [177].

4.4 Novelty and Remaining Questions

The main contributions of the current chapter can be summarized as follows:

1. The development of the SciXMiner Forecasting Extension that not only streamlines the current forecasting functionality, but also includes all the methods described in Chapter 2 and the concepts presented in Chapter 3
2. The inclusion of some options specifically developed to forecast energy time series (cf. Section 4.2.1)
3. The description of a procedure that allows the decoupling of the forecasting approaches from SciXMiner

Furthermore, there is still some work to be done before the Forecasting Extension can be released together with a future version of SciXMiner, for instance:

- A more thorough documentation needs to be written
- The graphic user interfaces (GUIs) of the Forecasting Extension have to be made more user friendly.

5 Application

5.1 Overview

The current chapter presents the application of the methods described in Chapter 2 on real data – i.e. time series of real power measurements – and on control and scheduling tasks. The former is described in Section 5.2 and the latter in Section 5.3. Additional information about some of the described applications can be found in [165], [178], and [179].

5.2 Application on Real Data

The present section is divided in four parts. In Section 5.2.1 the datasets being used are described. Afterwards, in Section 5.2.2 the experiments conducted are shown and in Section 5.2.3 the data mining techniques utilized to create the forecasts are presented. Finally, the results, their analysis, and their discussion are reported through Sections 5.2.4, 5.2.5, and 5.2.6.

5.2.1 Data

In the present section two datasets are used: first, the dataset of the solar track of the Global Energy Forecasting Competition of 2014 (GEFCom14) [29] and second, a dataset stemming from the North Campus of the Karlsruhe Institute of Technology (KIT) [180]. To be more specific, the GEFCom14 dataset contains PV power time series¹ of three PV systems

¹ The PV power time series contained in the GEFCom14 dataset are not preprocessed further.

located in Australia, while the KIT dataset describes the energy consumption of a large research campus² [10]. Please notice that all the time series contained in both datasets have an hourly resolution and that both the PV power and load have been normalized to values between zero and one. In addition, both forecast time series (i.e. PV power and load) are referred to in the present section as $\{P[k]; k = 1, \dots, K\}$. Table 5.1 offers more information about the used datasets.

Dataset	Forecast Time Series	Measurement Period	K	Available Exogenous Time Series	
				Used as Input	Not Used as Input
GEFCom14	PV Power	01.04.12 - 01.07.14	19704	Forecast surface solar radiation, forecast surface thermal radiation, and forecast top net solar radiation	Forecast total column liquid and ice water, forecast surface pressure, forecast relative humidity at 1000 [mbar], forecast total cloud cover, forecast wind velocity (U and V components), forecast 2 [m] temperature, and forecast total precipitation
KIT	Load	26.01.16 - 07.04.18	19248	Time series indicating if a day is a workday and time series indicating if a day is a weekend or a holiday	-

Table 5.1: Information of the real data used in the current thesis

In addition to the PV power time series, the GEFCom14 dataset contains corresponding forecast weather data from which only the time series describing the forecast surface solar radiation $\{\hat{G}_s[k]; k = 1, \dots, K\}$, surface thermal radiation $\{\hat{G}_{th}[k]; k = 1, \dots, K\}$, and top net solar radiation $\{\hat{G}_t[k]; k = 1, \dots, K\}$ are used³. The KIT dataset also contains two additional time series; one defining if a day is a workday $\{D_w[k]; k = 1, \dots, K\}$ and another determining if the day is a weekend or a holiday $\{D_f[k]; k = 1, \dots, K\}$.

Lastly, since it is assumed that knowledge of the seasonality of the forecast time series (i.e. the daily seasonality of PV and the weekly seasonality of load)⁴ may benefit the forecasting models, three additional time series are

² The preprocessing applied to the KIT load time series is briefly outlined in Appendix C.1.

³ All radiation values used are first order differences of the ones contained in the dataset, as they are originally provided as accumulated values.

⁴ Notice that while the time series used may have other seasons only the ones previously mentioned are considered in the current chapter.

calculated and used later on. These time series contain the mean, maximal, and minimal values of several seasons of the time series of interest and are obtained as follows:

$$\begin{aligned}
 P_{\text{mean}}[k] &= \text{mean}\{P[k], P[k - H_s], \dots, P[k - n_s H_s]\} \\
 P_{\text{max}}[k] &= \text{max}\{P[k], P[k - H_s], \dots, P[k - n_s H_s]\} \\
 P_{\text{min}}[k] &= \text{min}\{P[k], P[k - H_s], \dots, P[k - n_s H_s]\}; k \in [n_s H_s + 1, K],
 \end{aligned}
 \tag{5.1}$$

where H_s represents the number of time steps forming a time series' season and n_s is the number of seasons considered. Since all time series have an hourly resolution, the values are set to: $H_s = 24$ and $n_s = 7$ for PV power and $H_s = 168$ and $n_s = 3$ for load. To be more specific, in the case of PV power the new time series represent mean, maximal, and minimal values of the power measured at the same hour for the past week. In the case of load, in contrast, the new time series contain mean, maximal, and minimal values of the load measured on the same weekday at the same hour for the past month.

5.2.2 Experiments

Two specific experiments are conducted in the next section; both deal with the creation of forecast for a forecast horizon equal to 24 hours, i.e. $H = 24$ (since all time series used have an hourly resolution). Notice that some parts of the present section are more thoroughly described in [165]. Additionally as some computation times are later used for comparison, it is important to mention that all calculations are conducted on a computer with an Intel Core i7-4790 processor with 3.6 [GHz] and 16 [GB] of RAM.

Experiment 1

The first experiment consists in solving the tasks of GEFCom14 using the quantile regressions based on the nearest neighbors quantile filter (NNQF)⁵ (cf. Section 2.2.1) and comparing the results to those of various benchmarks, including the competition's winner. GEFCom14 was divided in 15 tasks with a training and a test period. During the competition, only Task 4 to 15 were relevant for the final result. The goal during the training periods was to obtain models able to provide probabilistic forecasts during the test periods in a 24 hours basis.

In the present thesis, the data stemming from the training periods is used to train NNQF-based quantile regressions for $q = 0.01, 0.02, \dots, 0.99$ that are later evaluated on the test period's data using the average pinball-loss (cf. Equation (2.25)). Specifics about the data used on the training and test period of each relevant task are available in Appendix C.2. In addition, the comparison also takes into consideration the computation time needed for training and applying/testing the probabilistic forecasts in Task⁶ 15. However, since the competition's winner used a much more powerful computer than the one used for the current thesis, a direct comparison of the computation times would not be appropriate. Thereupon, a value referred to as the computational effort is used. This value takes into consideration the computation time, the number of processing cores, and the clock rate. It is defined as:

$$C = t \cdot N_{\text{cores}} \cdot f_{\text{clock}}; \quad (5.2)$$

where C is the computational effort, t represents the computation time, N_{cores} is the number of processing cores, and f_{clock} is the clock rate of the processor. In other words, the computational effort is a dimensionless value that, under the assumption of a processor being used at their maximum po-

⁵ Notice that the threshold value ϵ_{th} used in the NNQF (cf. Equation (2.1)) is set to infinity.

⁶ Only Task 15 is considered, as this is the only one for which the competition's winner reports its computation times.

tential ⁷, represents the amount of clock cycles needed to train and/or apply the forecasting models. Furthermore, the values of the computer used in the current thesis are $N_{\text{cores}} = 8$ and $f_{\text{clock}} = 3.6$ [GHz], while the values reported by the competition's winner are $t = 662$ [s] for training, $t = 96$ [s] for testing, $N_{\text{cores}} = 256$, and $f_{\text{clock}} = 2.6$ [GHz] [142]. Note that information about other benchmarks used is available in the next section.

Experiment 2

The goal of the second experiment is to obtain and compare (for both the GEFCom14 and the KIT datasets) the results of all techniques described in Chapter 2. To be more specific, the results shown later stem from quantile regressions based on the NNQF⁸ (cf. Section 2.2.1), interval forecasts (cf. Section 2.2.2), parametric probabilistic forecasts (cf. Section 2.2.4), and scenario forecasts (cf. Section 2.2.5). In the current experiment, 70% of the time series is used for training the models and the remaining 30% is used for testing them. Furthermore, to obtain more representative results, the test set is divided in ten segments in which all models are evaluated. The obtained results can then be used to form box plots that better represent the capabilities of each model⁹. Lastly, note that only one of the PV power time series available on the GEFCom14 dataset is used in the current experiment.

5.2.3 Data Mining Techniques

Experiment 1

For the first experiment, six different data mining techniques are used. The first four are polynomial models with a maximal allowed degree of one to four (i.e. Poly1-4). The last two techniques are ANNs (i.e. MLPs) with

⁷ Notice that the computational effort does not consider parallelization effects

⁸ In the current experiment, the NNQF's threshold value ϵ_{th} (cf. Equation (2.1)) is set to infinity.

⁹ Evaluating the models on ten different segments of the test set eliminates the risk of considering a trivial forecast to be perfect. Therefore the value S_{T} used in Equations (2.24) and (2.30) is set equal to one in the present experiment.

the first having six hidden neurons (i.e. ANN6) and the second ten (i.e. ANN10). In addition, the number of nearest neighbors used for training the quantile regressions based on the NNQF are $N_{\text{NN}} = 50, 100, 150, 200$ and the desired output and input vector are defined as:

$$\begin{aligned} y &:= P[k + H]; H = 24 \\ \mathbf{x} &:= [\mathbf{u}[k], \dots, \mathbf{u}[k - H_1]]^T; k \in [H_1 + 1, K - H]; H_1 = 24 \quad (5.3) \\ \mathbf{u}[k] &:= [\hat{G}_s[k + H], \hat{G}_{\text{th}}[k + H], \hat{G}_t[k + H]]^T. \end{aligned}$$

The reason for only using exogenous values as input is that PV power measurements were not available during the test periods of the competition. Also, the power values with a corresponding $\hat{G}_s[k + H]$ lower than or equal to $100000 \text{ [Jh}^{-1}\text{m}^{-2}]$ are assumed to be night values; meaning that they are removed from the training set and automatically set to zero during the models' application. More information about this procedure can be found in Appendix C.3. Additionally, all models are trained using four features (normalized to values between zero and one) that are selected with the same procedure used in Section 2.4. For the sake of illustration, Appendix C.4 shows examples of some features selected on Task 15 of the competition.

It must be stated, that the trained regressions are not only compared to the results of the competition's winner, but also to those obtained by polynomial models with maximal allowed degrees of one to four that are trained by minimizing the sum of pinball-losses (cf. Equation (1.16)), i.e. "True" Poly1-4.

Experiment 2

As stated previously, the goal of the current experiment is to obtain forecasts using the present thesis techniques. To achieve this goal, quantile regressions must be obtained, as they are required by all of the present thesis' methods. For such reason, quantile regressions based on the nearest neighbors quantile filter (NNQF) are trained using the same data mining

techniques as in Section 2.4. These techniques are polynomial models with maximal allowed degrees of one, two, and three (Poly1-3) and ANNs with four (ANN4) and ten (ANN10) hidden neurons. Moreover, the desired output and the input vector used are defined as:

$$\begin{aligned}
 y &:= P[k + H]; H = 24 \\
 \mathbf{x} &:= [P[k], \dots, P[k - H_1], P_{\text{mean}}[k], \dots, P_{\text{mean}}[k - H_1], \\
 &\quad P_{\text{max}}[k], \dots, P_{\text{max}}[k - H_1], P_{\text{min}}[k], \dots, P_{\text{min}}[k - H_1], \\
 &\quad \mathbf{u}[k], \dots, \mathbf{u}[k - H_1]]^T; k \in [H_1 + 1, K - H].
 \end{aligned} \tag{5.4}$$

Please notice that depending on the dataset used the vector $\mathbf{u}[k]$ and the number of lags H_1 in Equation (5.4) are defined differently. Table 5.2 contains the values used for both the GEFCom14 and the KIT datasets.

Dataset	H_1	$\mathbf{u}[k] :=$
GEFCom14	24	$[\hat{G}_s[k + H], \hat{G}_{\text{th}}[k + H], \hat{G}_t[k + H]]^T$
KIT	168	$[D_w[k + H], D_f[k + H]]^T$

Table 5.2: Number of lags and exogenous inputs used for each dataset

The difference in H_1 comes from the difference in seasonality between PV and load time series considered in the present thesis; i.e. daily for PV and weekly for load. In other words, the values of H_1 are simply defined as the number of hours representing the time series considered season. Furthermore, all necessary quantile regressions are trained using the NNQF with 100 nearest neighbors (cf. Section 2.2.1) and four features that are normalized to values between zero and one and selected with the forward feature selection used in Section 2.4. Readers interested in the features that are selected for each technique are referred to Appendix C.5.

In the case of GEFCom14, two types of quantile regressions are trained. The first one uses the exogenous input defined in Table 5.2, while the second one does not. The goal is to gain insight on the influence that forecast

weather data usage may have on the PV power probabilistic forecasts. Additionally, all future PV power values with corresponding $P[k]$ and $P[k - H_1]$ values lower than or equal to 0.05 are considered night values. In other words, they are removed from the training set and set to zero during the models' application. Readers are referred to Appendix C.3 for more information on the previously described approach.

Finally it needs to be mentioned, that in the current experiment "true" quantile regressions are used as benchmark. To be more specific, these regressions are polynomials with a maximum degree of three (i.e. "true" Poly3) that are trained by minimizing the sum of pinball-losses (Equation (1.16)).

5.2.4 Results Experiment 1

Table 5.3 contains the average pinball-loss – averaged across all relevant tasks – of all tested approaches.

The results contained in Table 5.3 show, at a first glance, that none of the NNQF-based regressions obtained an average pinball-loss higher than 2%. Additionally, the results show that while the polynomial NNQF-based regressions have their best performance when using the largest amount of nearest neighbors (i.e. 200), the ANNs perform best when using the lowest amount (i.e. 50). Also, the differences between the pinball-loss values of quantile regressions based on the NNQF and that of the competition's winner never surpass 1%; instead it ranges between 0.73% (Poly2, $N_{NN} = 50$) and 0.30% (ANN10, $N_{NN} = 50$). Furthermore, Table 5.3 demonstrate that the "true" polynomial quantile regressions have lower Q_{PL} values than the polynomial NNQF-based regressions. This is an expected result, as the former are trained to actually minimize the sum of pinball-losses. In any case, the ANNs obtain better results than every polynomial model used, thus supporting the fact that the NNQF allows the training of quantile regressions with more complex techniques without the need of modifying their original

N_{NN}	50	100	150	200
	Q_{PL} [%]			
with NNQF				
Poly1	1.93	1.92	1.90	1.90
Poly2	1.94	1.93	1.93	1.92
Poly3	1.92	1.92	1.91	1.91
Poly4	1.94	1.93	1.93	1.92
ANN6	1.55	1.57	1.58	1.61
ANN10	1.51	1.56	1.57	1.59
Benchmarks				
“True” Poly1				1.81
“True” Poly2				1.76
“True” Poly3				1.76
“True” Poly4				1.76
GEFCom14 Winner				1.21

Table 5.3: GEFCom14 average pinball-loss; the techniques with only one result do not use nearest neighbors; N_{NN} : number of nearest neighbors; Q_{PL} : average pinball-loss (adapted from [165])

cost function nor their training algorithm. Regardless, the results also revealed that the NNQF-based quantile regressions are unable to obtain better results than those of the winner, which in part may be attributed to the fact that rather simple data mining techniques are being used. Still, the advantage of using the NNQF can be shown by considering the computational effort. For the sake of illustration, Figure 5.1 plots the computational effort C needed for (a) training and (b) applying the quantile regressions in Task 15 against the pinball-loss obtained on that task. Note that the number of nearest neighbors used by the NNQF-based regressions depicted is the one with which they obtained their best result in Table 5.3. Readers interested in the computation times in seconds are referred to Appendix C.4 for more information.

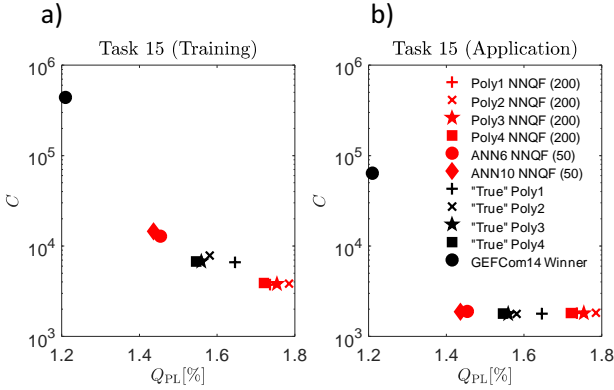


Figure 5.1: Pinball-loss and computational effort for either training or applying the quantile regressions; **Red**: with NNQF; **Black**: Benchmarks; Q_{PL} : average pinball-loss; C : computational effort (adapted from [165])

Figure 5.1 demonstrates that even though the differences in accuracy between the quantile regressions based on the NNQF and those of the competition’s winner are minimal, the differences in computational effort are not. For example, the approach used by the competition’s winner requires approximately 30 times more computational effort than ANN10 in both training and application.

Looking at Figure 5.1a, it can be observed that the “true” quantile regressions have lower pinball-loss values than the NNQF-based polynomials, but require a higher effort for their training. This is explained by the fact that minimizing the “true” quantile regressions’ sum of pinball-losses is more difficult than minimizing the sum of squared errors used by the NNQF-based polynomials. Therefore, using the NNQF can be viewed as a trade-off between training speed and accuracy in terms of pinball-loss. At the same time, the figure also shows that the NNQF allows the training of ANN quantile regressions without significantly increasing the necessary computational effort.

In turn, Figure 5.1b demonstrates that the computational effort for applying the “true” and the NNQF-based quantile regressions is much lower than that of GEFCom14 winner’s method. This outcome can be traced back to the fact that the former are pretrained functions that just need to be evaluated, while the latter is a type of nearest neighbors regression that calculates the nearest neighbors every time a new forecast is conducted. As shown in Section 2.4.2 Figure 2.19, calculating the nearest neighbors every time a forecast is required not only increases the application C value, but also makes the computational effort during application proportional to the amount of training data; an undesired property in cases in which the amount of training data constantly increases, e.g., in an online forecasting service. In contrast, by calculating the nearest neighbors only once prior to the models training, the NNQF-based regressions retain low application C values and avoid the scalability issue mentioned previously.

5.2.5 Results Experiment 2 (GEFCom14)

Figure 5.2 depicts box plots of the average pinball-loss (Q_{PL} , cf. Equation (2.25)) and modified reliability deviation (Q_{MRD} , cf. Equation (2.27)) values of the NNQF-based quantile regressions and “true” Poly3 benchmarks. Notice that the use of weather forecasts as input is characterized by the ARX acronym, while the AR acronym is used to reflect the lack of weather information as input. It must be mentioned, that the Q_{MRD} is calculated using day values, as the trivial night values skew the results. To be more specific, the Q_{MRD} values are obtained on a subset defined by the true values or the median estimates being greater than 0.05 (as in [165]).

Figure 5.2 shows that the median Q_{PL} of both ARX and AR models is lower than 2% and that the use of forecast weather data improves the overall performance in terms of Q_{PL} . Furthermore, while the Q_{PL} values of all AR models (including the benchmark’s) are similar, the values obtained using weather forecast do vary. For instance, the “true” Poly3 ARX regressions

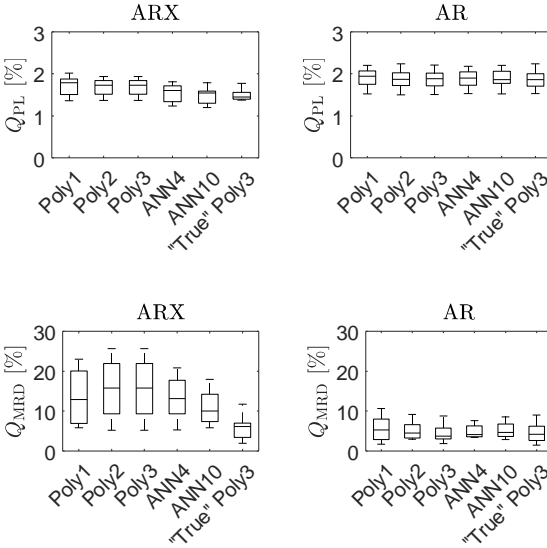


Figure 5.2: Results obtained by quantile regressions on the GEFCom14 dataset; ARX: Models using forecast weather data as input; AR: pure autoregressive models; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

perform slightly better than their NNQF counterparts, however their training procedure is much slower. To be more specific, the NNQF-based ARX Poly3 quantile regressions are trained in approximately 37 [s], while the training of their “true” variant takes around 112 [s], i.e. an increase in computation time of around 203%. This indicates again the trade-off between pinball-loss accuracy and computation time, which is more thoroughly discussed in Section 5.2.4. Nevertheless, the trade-off does not seem to apply in the present example to the ARX ANNs, since even though they are slower to train (between 246 and 272 [s]) and more complex than the ARX “true” Poly3 benchmark, they do not perform drastically better in terms of Q_{PL} .

Surprisingly, the presence of forecast weather seems to have a negative effect on the average modified reliability deviation, as the values obtained

by the AR models are better than those of the ARX regressions. Likewise, Figure 5.2 shows that while the ARX benchmark performs much better than the ARX NNQF-based regressions in terms of Q_{MRD} , it still obtains a larger median Q_{MRD} than its AR variant. Henceforth it can be argued that, the used weather forecasts improve the pinball-loss by reducing the distance between the quantile regressions and the true values, yet generate at the same time small deviations that greatly affect the regressions' reliability (especially that of the NNQF-based regressions). A possible explanation may be (i) that the weather forecast used over-/underestimates their true value and/or (ii) that the quantile regressions are unable to properly propagate the weather forecast "uncertainty". To investigate this effect further, Figure 5.3 shows the mean of the Q_{MRD} values obtained by the ANN10 NNQF-based quantile regressions when manually adding a bias between minus one and one to them.

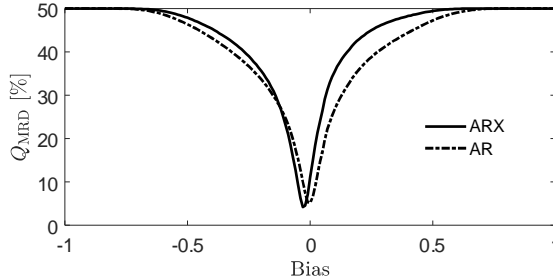


Figure 5.3: Reliability deviation obtained by adding a bias to the quantile regressions; ARX: Models using forecast weather data as input; AR: pure autoregressive models; Q_{MRD} : average modified reliability deviation

As Figure 5.3 demonstrates, the AR ANN10 regressions have their optimal value when the added bias equals zero, in contrast the ARX variant has its optimal result for a bias equal to -0.03 . This suggests that the used forecast weather data add in average a small bias that greatly affects the Q_{MRD} results, as such value only cares for the percentage of values under

the quantile regressions and not for the magnitude of the deviations. Therefore, future related works should investigate some approaches to mitigate this modified reliability deviation increase (e.g., ways of propagating the forecast weather uncertainty through the models).

The results of the intervals formed by the trained quantile regressions (cf. Section 2.2.2) are shown in Figure 5.4. The evaluation values depicted are the average interval width (Q_{IW} , cf. Equation (2.31)), the average interval score (Q_{IS} , cf. Equation (2.32)), and the average modified interval reliability deviation (Q_{MIRD} , cf. Equation (2.33)). Just as in Section 2.4.3, these intervals have probabilities ranging from 0.02 to 0.98 and are formed by 48 pairs of quantile regressions that are centered on the median estimate.

As expected, Figure 5.4 shows that the interval forecasts that use information about the future weather are not only narrower, but also perform overall better than the pure autoregressive intervals in terms of their interval score. Moreover, the Q_{IW} values of the ARX NNQF-based regressions show a smaller variance than those of the “true” Poly3 benchmark. Additionally, all pure autoregressive quantile regressions (including the benchmark) formed intervals with similar interval score values. In contrast, the ARX benchmark shows better median Q_{IS} values than the NNQF-based polynomials, but similar median Q_{IS} than the ARX ANN10 intervals.

Figure 5.4 shows also, that the use of weather forecasts increases the models’ median Q_{MIRD} values, with ANN4 being the only exception. In the case of the ARX models, the benchmark seems to perform generally better than the intervals formed by NNQF-based polynomial regressions, but worse than the intervals created with the ANNs. In turn when looking at the AR results, the NNQF-based Poly1, Poly3, ANN4, and ANN10 regressions are the ones that seem to perform better than the benchmark.

For the next evaluation, parametric distribution (CDF) forecasts are created using the method described in Section 2.2.4 and the non-parametric distribution forecasts formed by the trained NNQF-based quantile regressions (cf. Section 2.2.3). The results obtained are all depicted in Figure 5.5.

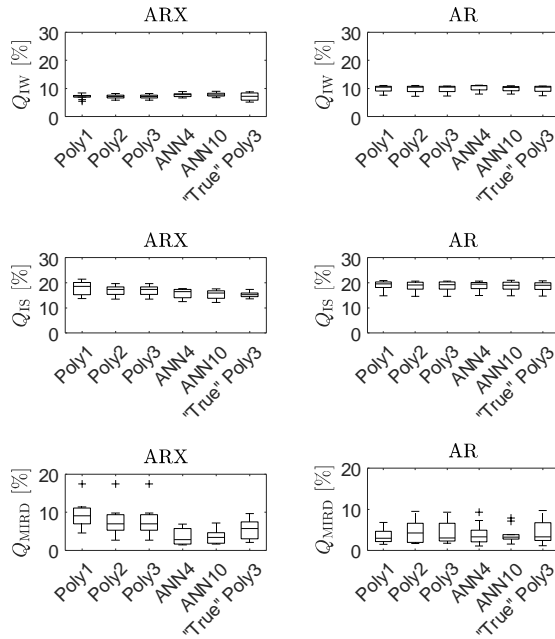


Figure 5.4: Results obtained by interval forecasts on the GEFCom14 dataset. The intervals are formed using the trained quantile regressions and Equation (2.5); ARX: Models using forecast weather data as input; AR: pure autoregressive models; Q_{IW} : average interval width; Q_{IS} : average interval score; Q_{MIRD} : average modified interval reliability deviation

As it can be seen in Figure 5.5, the parametric CDF forecasts obtained similar average pinball-loss values than the non-parametric forecasts they are based on. Additionally, even though the parametric CDF forecasts based on the ARX models seem to have better modified reliability deviation values than the non-parametric distribution forecasts, they still perform worse than the ones not using weather information. In general, the box plots of the parametric and non-parametric CDF forecasts lay really close to one another. Therefore, the results demonstrate that the present thesis method is

able to obtain parametric probabilistic forecasts with a similar accuracy to that of the non-parametric variant they are based on.

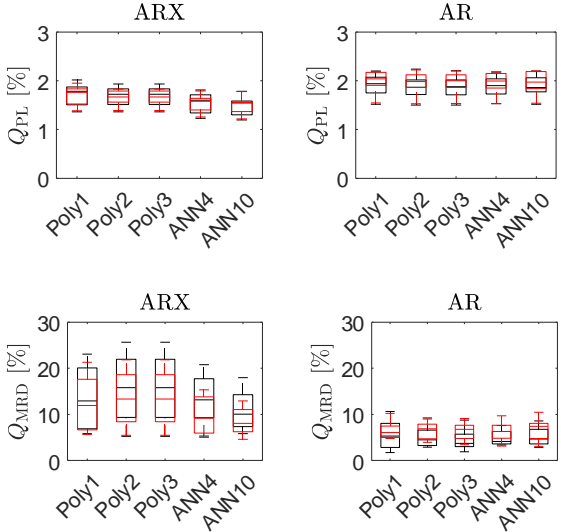


Figure 5.5: Results obtained by parametric and non-parametric distribution forecasts on the GEFCom14 dataset. The forecasts are obtained with the trained quantile regressions and with the methods described in Sections 2.2.3 and 2.2.4; **Black:** Non-parametric; **Red:** Parametric; ARX: Models using forecast weather data as input; AR: pure autoregressive models; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

As a last point, scenario forecasts that estimate the uncertainty of the future values on a rolling 24 hour period are created using the method described in Section 2.2.5¹⁰. Figure 5.6 shows a comparison between the obtained results and those of the previously described NNQF-based quantile

¹⁰ The scenario forecasts consists of 50 scenarios that are based on Poly3 NNQF-based quantile regressions that use the input vectors defined in Section 5.2.3 and a forecast horizon of $H = 1$. Furthermore, y_{Smax} and y_{Smin} (cf. Equation (2.20)) are defined as one and zero respectively.

regressions (i.e. the non-parametric probabilistic forecasts), which also forecast 24 hours in advance.

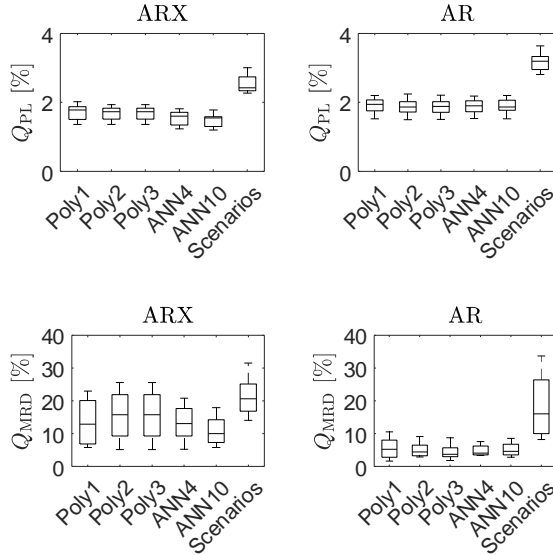


Figure 5.6: Results obtained by (Poly3) scenario forecasts and quantile regressions used for comparison on the GEFCom14 dataset. The scenario forecasts are obtained using Poly3 quantile regressions and the method described in Section 2.2.5; ARX: Models using forecast weather data as input; AR: pure autoregressive models; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

Figure 5.6 shows that just as in Chapter 2, the scenario forecasts deliver overall worse results than the quantile regressions used to solve the same task. In terms of Q_{PL} , the median values obtained by the scenario forecasts are around the 3% mark, while the median Q_{MRD} values are all higher than 15%. Interestingly, using the forecast weather as input seems to reduce the variance of the the scenario forecasts' Q_{MRD} values (an opposite effect to the one observed on the NNQF-based regressions). Still the median Q_{MRD} value obtained by the AR scenario forecasts is lower than that of its ARX

counterpart. As already mentioned in Chapter 2, future related works should study methods for improving the quality of the scenario forecasts, as well as for reducing the computation time needed to create them. The main reason for the latter is that applying the ARX scenario forecast on the whole test set takes approximately 2.6 [h], while the application of the quantile regressions takes less than a minute.

Readers interested in the numeric values of the presented results and some additional information are referred to Appendix C.5.1. Furthermore, Appendix C.5.1 also contains each technique’s selected features; showing that values of the additional time series created in Equation (5.1) are actually used by the regressions (especially those describing maximal values).

5.2.6 Results Experiment 2 (KIT)

Considering the KIT dataset, Figure 5.7 depicts the box plots of the pinball-loss (Q_{PL} , cf. Equation (2.25)) and average modified reliability deviation (Q_{MRD} , cf. Equation (2.27)) values obtained by both the NNQF-based quantile regressions and the “true” Poly3 benchmark.

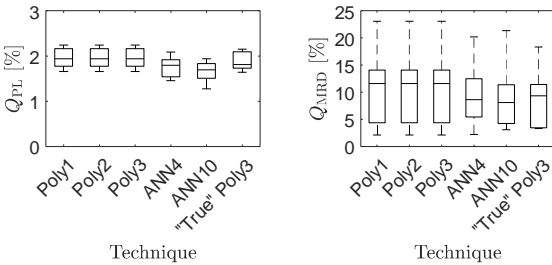


Figure 5.7: Results obtained by quantile regressions on the KIT dataset; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

The results in Figure 5.7 show that all data mining techniques are able to obtain Q_{PL} values with a low amount of variance and with median values that never surpass 2%. Likewise, it can be observed that the ANNs

deliver slightly better results than the polynomials. Moreover, the median Q_{PL} of the used benchmark is slightly better than that of the NNQF based polynomials; an expected result considering that the “true” Poly3 models are actually trained to minimize the sum of pinball-losses. Still it must be considered, that training all “true” Poly3 regressions takes around 107 [s], while their NNQF counterparts are trained in only 50 [s]. In other words, training the former takes approximately 114% more time. This offers more evidence to the trade-off between computation time and pinball-loss accuracy described in Section 5.2.4. Furthermore, the ANNs obtain in general the best Q_{PL} values, thus supporting the fact that the NNQF allows quantile regressions to be trained with more complex techniques using their traditional cost functions and training algorithms. In terms of Q_{MRD} , the variance of all results appears to be much larger than that of the corresponding pinball-losses, with that of the NNQF-based polynomials being the largest. Moreover, both the ANN10 and the “true” Poly3 benchmark obtain similar Q_{MRD} box plots.

After training the quantile regressions, 49 intervals can be obtained (cf. Section 2.2.2); these are all centered on the regression describing the median and have probabilities ranging from 0.02 to 0.98. Figure 5.8 depicts box plots of the intervals’ average interval width (Q_{IW} , cf. Equation (2.31)), average interval score (Q_{IS} , cf. Equation (2.32)), and average modified interval reliability deviation (Q_{MIRD} , cf. Equation (2.33)).

As the results show, all intervals created with quantile regressions based on the NNQF have a similar average width and are narrower than those of the benchmark. Moreover, the benchmark appears to perform better in terms of Q_{IS} and Q_{MIRD} than the NNQF-based polynomials; especially considering the fact that the variance of the NNQF-based polynomials’ Q_{MIRD} values seems to be relatively large. Nevertheless, the best results in terms of Q_{IS} and Q_{MIRD} are generally obtained by the intervals stemming from the ANN quantile regressions. This supports once again, the possibility of

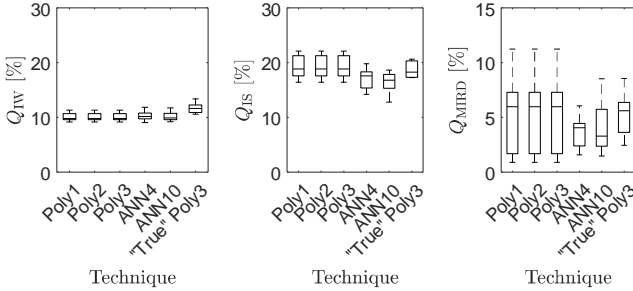


Figure 5.8: Results obtained by interval forecasts on the KIT dataset. The intervals are formed using the trained quantile regressions and Equation (2.5); Q_{IW} : average interval width; Q_{IS} : average interval score; Q_{MIRD} : average modified interval reliability deviation

training accurate probabilistic forecasts using the combination of a complex data mining technique and the NNQF.

Now, using the method described in Section 2.2.4 and the trained NNQF-based quantile regressions, parametric distribution (CDF) forecasts can be calculated. Figure 5.9 depicts the results obtained by the parametric CDF forecasts and their non-parametric counterparts.

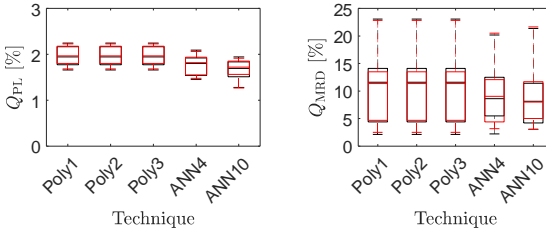


Figure 5.9: Results obtained by parametric and non-parametric distribution forecasts on the KIT dataset. The forecasts are obtained with the trained quantile regressions and with the methods described in Sections 2.2.3 and 2.2.4; **Black**: Non-parametric; **Red**: Parametric; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

As Figure 5.9 shows, both types of forecasts obtained almost indistinguishable Q_{PL} and Q_{MRD} values. Therefore, the results support that the present thesis approach is able to obtain parametric distribution forecasts that retain the accuracy of the non-parametric forecasts they are based on.

Finally, Figure 5.10 depicts the results of scenario forecasts¹¹ created with the present thesis method (cf. Section 2.2.5), as well as those stemming from the NNQF-based quantile regressions (i.e. the non-parametric probabilistic forecasts) presented previously. Just as in Chapter 2, the scenario forecasts are set to estimate everyday the next 24 hours.

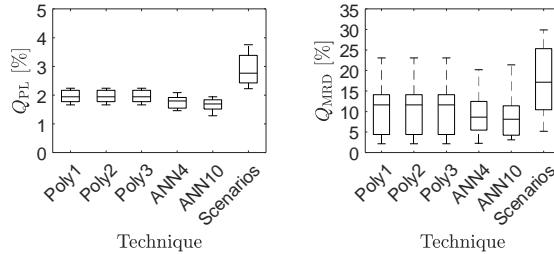


Figure 5.10: Results obtained by (Poly3) scenario forecasts and quantile regressions used for comparison on the KIT dataset. The scenario forecasts are obtained using Poly3 quantile regressions and the method described in Section 2.2.5; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

As expected, the scenario forecasts deliver the worse results. For instance, their resulting Q_{PL} values surpass the 2% mark and have a greater variance than those of the NNQF-based quantile regressions, while their Q_{MRD} values show not only a great amount of variance, but also a median value greater than 15%. For the previous reasons, methods for improving the quality of the scenario forecasts should be investigated in future related works.

¹¹ Notice that the scenario forecasts are formed by 50 scenarios based on Poly3 NNQF-based quantile regression created with the input vector described in Section 5.2.3 and a forecast horizon of $H = 1$. Additionally, the values y_{Smax} and y_{Smin} (cf. Equation (2.20)) are set equal to one and zero, respectively.

Moreover, ways of speeding up the creation of the scenario forecasts have also to be researched further. This is of relevance considering that obtaining the scenario forecast for the present experiment takes around 4.8 [h], while simply applying the regressions takes less than a minute.

Please note, that interested readers are referred to Appendix C.5.2 for more information about the results presented in the current section. Additionally, the features selected as most relevant for each technique are also shown in Appendix C.5.2. The selected features show that the time series containing maximal values in Equation (5.1) is the additional time series that is preferred by all techniques.

5.3 Application in Control and Scheduling

The current section describes an example on how the present thesis' probabilistic forecasts can be used in control and scheduling tasks. Notice that a more thorough description of the application case presented can be found in [178] and [179].

To begin with the description, the concept of a dispatchable feeder (DF) has to be introduced; this is a grid connected cluster of controllable and uncontrollable distributed energy resources and loads that is operated in a way in which the power exchange with the grid is regulated in accordance to a pre-computed dispatch schedule (DiS) [181]. In the current example, a DF formed by an uncontrollable load, an uncontrollable renewable generator, and a controllable energy storage system (ESS) is used. For the sake of illustration, Figure 5.11 depicts the system in question; as the figure shows the load and the renewable generated power (considered a negative load) are aggregated into the so-called inflexible active power.

The application case considered is the participation of the DF in the day-ahead market, which is summarized in the following steps: first, the dispatchable feeder communicates everyday (at midday) the day ahead dispatch schedule (DiS) of its aggregated active power (i.e. the aggregation

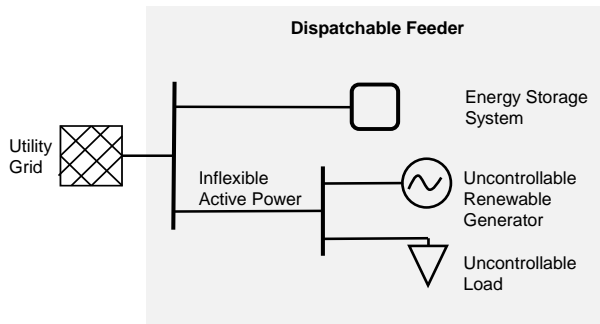


Figure 5.11: Dispatchable Feeder (adapted from [178])

of the ESS power and the inflexible active power), then during operation the ESS is used to assure that the previously communicated DiS is actually tracked. While the second step uses only point power forecast, the first step requires a description of the forecast uncertainty. Even though scenario forecasts are the ones most commonly used in this context, probabilistic forecasts of the inflexible active power and energy are the ones found most useful in the present application case. Additionally, using probabilistic forecasts allows for the calculation of a DiS that can be tracked in operation with at least a given probability; this probability is referred to as the security level of the computed DiS in [178] and in the remainder of the current section.

Unfortunately, obtaining the necessary energy probabilistic forecasts is not a simple task. As the power values are correlated over time, the energy forecasts cannot be calculated via the convolution of their corresponding power probabilistic forecasts; instead the joint distribution (CDF) of the correlated values has to be used. In the present example however, probabilistic energy forecasts for the next 48 hours are needed, meaning that a 48 dimensional CDF would be required for their straightforward calculation. Therefore to avoid estimating the CDF, an alternative has been developed. This is divided in the following steps:

1. Train quantile regressions based on the NNQF (cf. Section 2.2.1) for both renewable power and load separately. These can then be used to estimate probabilistic power forecasts for the next 48 hours of both uncontrollable values.
2. Integrate some of the estimated quantiles (e.g., the median, the 0.9 quantile, and/or the 0.1 quantile) over periods of 48 hours.
3. Train quantile regressions to forecast the energy uncertainty with the previously integrated values as input.
4. Obtain (assuming the independence of both uncontrollable values) the probabilistic forecasts for the next 48 hours of the inflexible active power and energy with a timestep-wise convolution operation.

As an example, Figure 5.12 depicts power and energy probabilistic forecasts obtained with the previously described procedure. These forecasts are used in [178] and are created using data provided by the Australian electricity distribution company Ausgrid [182]¹².

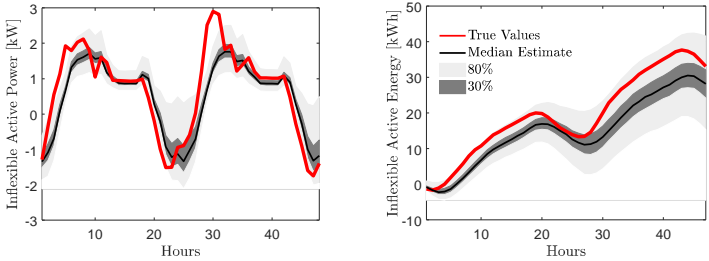


Figure 5.12: Probabilistic forecasts for the inflexible active power and energy (adapted from [178])

The advantage of calculating a DiS based on probabilistic forecasts is clearly shown in [178] and [179]. The study considers a cost policy in which

¹² www.ausgrid.com.au/

the cost of imbalances (i.e. the cost for deviating from the DiS during operation) is twice the cost for the DiS. Table 5.4 contains the costs (i.e. the average cost obtained over several days) in € for computing and tracking a DiS based on probabilistic forecast (i.e. probabilistic forecast scheduling, PFS), point forecasts (i.e. deterministic forecast scheduling, DFS), or scenario forecasts (i.e. scenario forecast scheduling, SFS). While the use of point forecast is to show the influence that disregarding the uncertainty has on the results, the use of scenario forecasts is to offer a comparison with an approach commonly used in similar scheduling and control schemes.

Lastly, notice that the PFS results are divided in two groups, since some are obtained by the scheme used in [178] and some by its improvement described in [179].

	DFS	SFS	PFS [178]						PFS [179]					
Security Level [%]	NA	NA	18	24	30	36	42	48	42	48	54	60	66	72
Cost DiS [€]	4.86	5.40	5.78	5.93	6.08	6.23	6.40	6.73	5.48	5.87	6.29	6.68	6.84	6.96
Cost Imbalances [€]	4.06	3.41	3.06	2.90	2.78	2.66	2.57	2.49	3.07	2.51	2.19	2.02	1.98	1.90
Total Cost [€]	8.92	8.81	8.84	8.83	8.87	8.90	8.97	9.22	8.55	8.38	8.48	8.70	8.83	8.86

Table 5.4: Costs for using direct forecast scheduling (DFS), probabilistic forecast scheduling (PFS), or scenario forecast scheduling (SFS) in the simulation described in [178] and [179]; DiS: dispatch schedule

As Table 5.4 shows, the DFS is able to estimate the cheapest DiS by ignoring the forecast uncertainty. At the same time this also results in the DFS having the highest cost of imbalances. In comparison, the description of the uncertainty allows the PFS to obtain a DiS with fewer imbalances during operation – independently of the security level used. It can also be observed that increasing the security level increases the corresponding DiS cost, but decreases the cost of imbalances. This behavior results in a security level of 48% being the one for which the PFS used in [179] obtains the lowest total cost¹³. In other words, the use of probabilistic forecasts allows the de-

¹³ The results also show, that the PFS used in [179] obtains lower costs of imbalances at higher security levels than that of [178]. This is an expected outcome as the former is an improvement of the latter.

termination of a schedule that limits the number of imbalances, as well as the total costs. In addition, the PFS (with a security level of 48%) performs better than the commonly used SFS. This is due to the cost of imbalances of the PFS being considerably lower. Interested readers are referred to [178] and [179] for more in detail information about the presented results and for an example in which a different cost policy is used.

As a final remark, note that probabilistic forecasts based on the methods described in the present thesis have also been used in a work studying the dispatchability of a population of electric vehicles [183].

5.4 Novelty and Remaining Questions

The novelty of the current chapter can be summarized in the following points:

1. The application of the methods described in Chapter 2 on real energy data, i.e. load and PV power time series
2. The comparison between NNQF-based regressions and the method used by the winner of the Global Energy Forecasting Competition of 2014
3. The presentation of how the present thesis' probabilistic forecasts have been used as part of a real-world energy control and scheduling task

By providing results of the application of the herein developed probabilistic forecasts on both real-world data and real-world control and scheduling applications, the current chapter offers a possible solution to the fifth and final open question in Section 1.4 and thus fulfills the final objective of the current thesis (cf. Section 1.5). Nevertheless, there are still remaining questions, for example:

- Can the modified reliability deviation of the ARX models in Section 5.2.5 be improved by propagating through them the uncertainty of the used weather forecasts?
- What is the best approach for using probabilistic values (e.g., probabilistic weather forecasts) as input?
- Can the results presented in the current chapter be improved by using other values as input (e.g., forecast temperature in the case of the KIT load)?
- Can the overall results of the scenario forecasts be improved?
- Are there better methods for creating the energy probabilistic forecasts used in Section 5.3?

Future related works should try to find answers to the previously described questions.

6 Summary and Outlook

The increasing installation of volatile renewable power systems has made the balancing of energy demand and supply more difficult. Forecasting models help in reducing this difficulty by providing some information about the future. Nevertheless, most forecasting models do not quantify in any way the forecast uncertainty, which may be required for making the control and scheduling of an energy system more robust. Probabilistic forecasts offer a solution, since they are able to describe the uncertainty in various ways: for instance, as intervals with a certain probability of containing a future value or a future value's probability distribution function. Unfortunately, the creation of probabilistic forecasts is not trivial and poses its own additional challenges (cf. Section 1.4). Therefore, the present thesis tackles some of those challenges by developing new data-driven probabilistic forecasting methods within an energy-related context.

To begin, Chapter 2 describes several methods able to obtain different types of probabilistic forecasts all based on quantile regressions that have been trained using a newly developed nearest neighbors quantile filter (NNQF). The first two methods use the NNQF-based quantile regressions to obtain either interval or non-parametric distribution forecasts. The latter can be used thereafter, in two other new methods: one to obtain parametric distribution forecasts and another to create scenario forecasts. Afterwards, Chapter 3 presents a concept for obtaining joint distribution forecasts of two correlated values, as well as coherent hierarchical probabilistic forecasts. Then, Chapter 4 presents first, the MATLAB/SciXMiner implementation of the methods described previously and second, a procedure that allows the

use of the implemented methods outside of SciXMiner. Finally, Chapter 5 shows the application of the developed forecasting approaches on real-world data and on a real-world control and scheduling task.

To be more specific, the major contributions of the current thesis can be summarized as follows:

1. The development of the new nearest neighbors quantile filter (NNQF) that allows quantile regressions to be trained using the traditional cost function of the data mining technique used (e.g., the sum of squared errors), instead of the non-differentiable sum of pinball losses. Using the NNQF not only avoids the problems that may arise from the non-differentiability of the sum of pinball-losses (e.g., higher computation times for training and problems when using gradient based training algorithms), but also allows for the use of training algorithms that have already been programmed and can be found in traditional machine learning and statistical libraries (cf. Section 2.2.1)
2. The obtainment of interval forecasts based on the new type of quantile regressions (cf. Section 2.2.2)
3. A new method for creating non-parametric distribution forecasts based only on previously computed quantile regressions and some known general properties of a cumulative distribution function (cf. Section 2.2.3)
4. A novel approach for obtaining parametric distribution forecasts (i.e. parametric probabilistic forecasts) without requiring a distribution assumption, but based only on corresponding non-parametric distribution forecasts (cf. Section 2.2.4)
5. A new heuristic for creating scenario forecasts that does not require an explicit description of the correlation structure between neighboring time series values (cf. Section 2.2.5)

6. A thorough study of all the methods presented in Chapter 2 using various simulated load time series (cf. Section 2.4)
7. A new concept for estimating a non-parametric joint probability distribution of two correlated future time series values using only quantile regressions (cf. Section 3.2)
8. A concept for obtaining coherent hierarchical probabilistic forecast based on an iterative method that only uses bivariate joint distribution forecasts (cf. Section 3.3)
9. The development of the new SciXMiner Forecasting Extension that not only streamlines the previous forecasting capabilities, but also adds to the toolbox all the methods described in Chapter 2, as well as the concepts presented in Chapter 3 (cf. Section 4.2)
10. The possibility of decoupling the implemented forecasting approaches from SciXMiner (cf. Section 4.3)
11. The study of the methods described in Chapter 2 using real-world data stemming from the solar track of the Global Energy Forecasting Competition of 2014 and from power consumption measurements of the North Campus of the Karlsruhe Institute of Technology (Sections 5.2.4, 5.2.5, and 5.2.6)
12. The successful application of probabilistic forecasts created with the present thesis methods on a real-world energy control and scheduling task (cf. Section 5.3)

Though the objectives of the present thesis have been fulfilled, there are still several open questions to be addressed in future related works; some of which have been listed in Sections 2.5, 3.4, 4.4, and 5.4. Still, there are several aspects that the author considers to be of major importance and thus are discussed further in the present section.

As the complexity of the energy system continues to grow, the interest towards methods able to describe the correlation between several future time series values may also increase. Therefore, future related works should try to expand, improve, and test the concepts, described in Chapter 3, for both the estimation of joint probability distributions and coherent hierarchical forecasts. A first step to be taken, could be the implementation of reconciliation approaches that may help in improving the accuracy of the hierarchical forecasts. Additionally, a comparison between the concepts presented and the alternative used in Section 5.3 – which avoids the estimation of the joint distribution – should also be studied further.

Likewise, future research should experiment more with the developed probabilistic forecasting approaches in real-world settings. A possibility thereof is to utilize the Energy Lab 2.0 of the Karlsruhe Institute of Technology, as it will not only provide the necessary real-world data, but will also allow the coupling of the present thesis approaches with other researchers' control and scheduling algorithms. This coupling could be done, for instance, using the framework described in Section 4.3. Furthermore, in cases in which forecast weather data is used as input, methods for propagating their uncertainty through the forecasting models have to be developed and tested. Such approaches could help in mitigating some negative effects that using uncertain forecast weather data may cause (cf. Section 5.2.5).

Finally, as deep learning approaches have started to gain the interest of the forecasting community, future related works should study probabilistic forecasts based on them. To be more specific, a comparative study of the methods developed herein and probabilistic forecasts coming from deep neural networks is to be done. This study should not only concentrate on the overall forecasting accuracy, but also in finding out the tipping point in which a complex deep neural network should be preferred over more simple approaches. To find this out, aspects like forecast accuracy, computation time, model complexity, amount of available data, complexity of the forecasting task, etc. should be taken into consideration.

A Probabilistic Forecasting

A.1 Load Time Series Benchmark Generator

The time series obtained via the benchmark generator are formed by five distinct components. For instance, a value of a simulated load time series $\{P_L[k]; k = 1, \dots, K\}$ can be written as:

$$\begin{aligned} P_L[k] &= \alpha_1 \cdot P_{L,T}[k] + \alpha_2 \cdot P_{L,P}[k] + \alpha_3 \cdot P_{L,Pa}[k] \\ &\quad + \alpha_4 \cdot P_{L,S}[k] + \alpha_5 \cdot P_{L,X}[k] \\ \text{s.t. } \sum_{i=1}^5 \alpha_i &= 1, \alpha_i \in [0, 1] \forall i; \end{aligned} \tag{A.1}$$

with $P_{L,T}[k]$ representing a trend component, $P_{L,P}[k]$ a periodic component, $P_{L,Pa}[k]$ a pattern component, $P_{L,S}[k]$ a stochastic component, and $P_{L,X}[k]$ an exogenous component. The factors α_i define the influence of each component on the resulting time series and hence are the parameters to be defined by the user. Notice that in the present thesis, the periodic component represents a daily, weekly, and yearly seasonality, the pattern component is a household standard pattern defined by EWE “Netz GmbH”¹, and the exogenous component represents the influence that temperature has on load. In addition to the previous factors, the user of the benchmark gener-

¹ ewe-netz.de/strom/1988.php

ator can also define the length of the simulated time series K and the total energy content E_L , which is defined as:

$$E_L = \sum_{k=1}^K P_L[k]. \tag{A.2}$$

For the sake of illustration, Figures A.1 to A.6 depict a simulated load time series and each of its components for $\alpha_1 = 0.05$, $\alpha_2 = 0.05$, $\alpha_3 = 0.65$, $\alpha_4 = 0.20$, and $\alpha_5 = 0.05$.

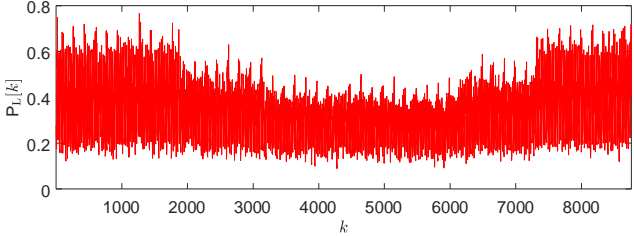


Figure A.1: Simulated time series

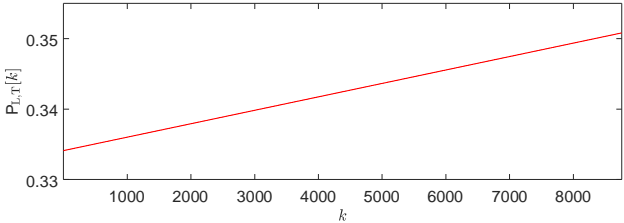


Figure A.2: Simulated time series trend component

Please refer to [3] for further information regarding other user defined parameters, as well as more in detail information about the load benchmark generator.

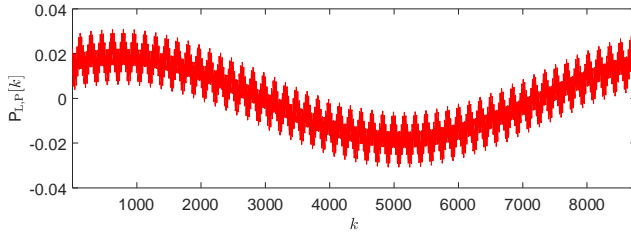


Figure A.3: Simulated time series periodic component

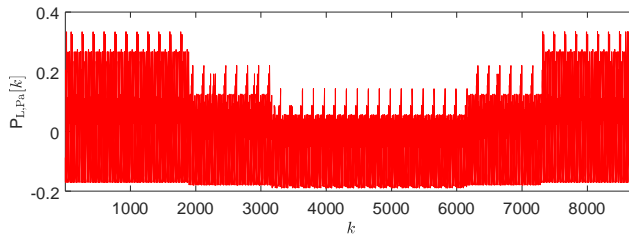


Figure A.4: Simulated time series pattern component

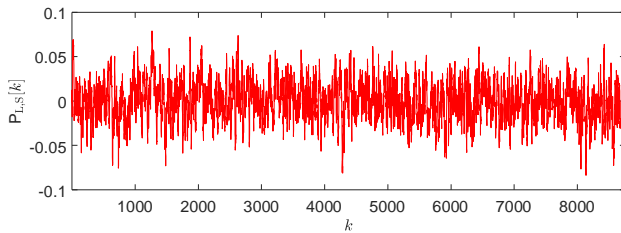


Figure A.5: Simulated time series stochastic component

With the described load benchmark generator several simulated load time series used on Chapter 2 are created as follows. The parameters used to obtain load time series in Section 2.2 are contained in Table A.1.

Similarly, the parameters used to create the time series with varying amounts of noise in Section 2.4 are contained in Table A.2

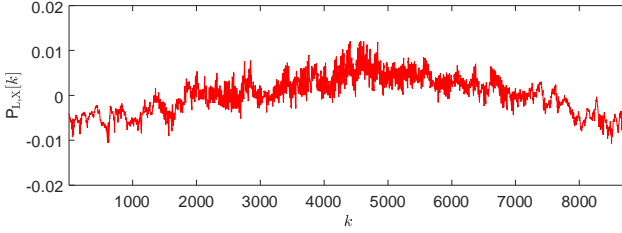


Figure A.6: Simulated time series exogenous component

K	E_L	α_1	α_2	α_3	α_4	α_5
26280	3	0.05	0.05	0.70	0.20	0.00

Table A.1: Load benchmark generator parameters for the time series used in Section 2.2

$r_{\text{prc}}[\%]$	K	E_L	α_1	α_2	α_3	α_4	α_5
75	26280	3	0.00	0.10	0.15	0.75	0.00
60	26280	3	0.00	0.10	0.30	0.60	0.00
45	26280	3	0.00	0.10	0.45	0.45	0.00
30	26280	3	0.00	0.10	0.60	0.30	0.00
15	26280	3	0.00	0.10	0.75	0.15	0.00

Table A.2: Load benchmark generator parameters for the time series used in Section 2.4

A.2 Calculation of an Empirical Quantile

The value of $\tilde{y}_{(q),n}$ – i.e. the empirical quantile of the values inside of $\mathbf{y}_{\text{NN},n}$ – is determined using Definition 5 of [166] and Method 10 in [167]. In the case of finding N_{NN} nearest neighbors, the calculation starts by defining the values

$$\mathbf{y}_{\text{NN},n} = [y_{\text{NN},n1}, \dots, y_{\text{NN},ni}, \dots, y_{\text{NN},nN_{\text{NN}}}]^T, \quad (\text{A.3})$$

as quantiles with probabilities

$$[q_{\text{NN},n1}, \dots, q_{\text{NN},nN_{\text{NN}}}]^T = \left[\frac{0.5}{N_{\text{NN}}}, \frac{1.5}{N_{\text{NN}}}, \dots, \frac{N_{\text{NN}} - 0.5}{N_{\text{NN}}} \right]^T. \quad (\text{A.4})$$

Thereafter, $\tilde{y}_{(q),n}$ can be obtained using a linear interpolation, i.e.:

$$\begin{aligned} \tilde{y}_{(q),n} = & \sum_{i=2}^{N_{NN}} \left(\left[\frac{y_{NN,ni} - y_{NN,n(i-1)}}{q_{NN,ni} - q_{NN,n(i-1)}} (q - q_{NN,n(i-1)}) + y_{NN,n(i-1)} \right] \right. \\ & \cdot I(q_{NN,n(i-1)} \leq q < q_{NN,ni}) \left. \right) + y_{NN,n1} \cdot I(q < q_{NN,n1}) \\ & + y_{NN,nN_{NN}} \cdot I(q_{NN,nN_{NN}} \leq q). \end{aligned} \quad (A.5)$$

In the previous equation, $I(\cdot)$ is an indicator function that equals one if its condition is fulfilled and zero if it is not.

A.3 Calculation of the Maximal and Minimal Non-Parametric Distribution Values

In the present thesis, the maximal and minimal values of the non-parametric distribution values (cf. Section 2.2.3) are determined as follows:

$$\underline{y} = \hat{y}_{q_1} - \frac{\hat{y}_{q_L} - \hat{y}_{q_1}}{q_L - q_1} \cdot q_1 \text{ and} \quad (A.6)$$

$$\bar{y} = \hat{y}_{q_L} + \frac{\hat{y}_{q_L} - \hat{y}_{q_1}}{q_L - q_1} \cdot (1 - q_L). \quad (A.7)$$

Please note, the developed method allows for other definitions of \underline{y} and \bar{y} , depending on the corresponding application.

A.4 Approximation of the Continuous Ranked Probability Score

The continuous ranked probability score (CRPS) is a value typically used for evaluating the performance of a distribution forecast. The CRPS on a single realization y_n (i.e. $Q_{C,n}$) can be defined as²:

$$Q_{C,n} = 2 \int_0^1 \begin{cases} (u-1) \cdot (y_n - \hat{F}^{-1}(u|\mathbf{x}_n)) & , \text{ if } y_n \leq \hat{F}^{-1}(u|\mathbf{x}_n) \\ u \cdot (y_n - \hat{F}^{-1}(u|\mathbf{x}_n)) & , \text{ else} \end{cases} du; \quad (\text{A.8})$$

with $\hat{F}^{-1}(u|\mathbf{x})$ representing the inverse of an estimated CDF. Averaging the values obtained with Equation (A.8) on all realizations of the test set, results in a value Q_C . Even though the CRPS is commonly utilized, it is not used in the present thesis, as it is already related to the easier to compute average pinball-loss Q_{PL} (cf. Equation (2.25)). An additional reason for using Q_{PL} instead of Q_C is that the former can approximate the latter if Q_{PL} is calculated using several quantile estimates. For instance in the present thesis, an estimated CDF is evaluated by calculating and averaging the pinball-loss of $L = 99$ quantiles $\{\hat{y}_{(q_l)}; l = 1, \dots, L \text{ and } q_1 < \dots < q_L\}$ with probabilities $q_1 = 0.01$ to $q_L = 0.99$. Using these quantile estimates, Equation (A.8) can be approximated as:

$$Q_{C,n} \approx \sum_{l=1}^L \frac{2}{L} \begin{cases} (q_l - 1) (y_n - \hat{y}_{(q_l),n}) & , \text{ if } y_n \leq \hat{y}_{(q_l),n} \\ q_l (y_n - \hat{y}_{(q_l),n}) & , \text{ else} \end{cases} \quad (\text{A.9})$$

Equation (A.9) shows that in the present thesis Q_C can be approximated by Q_{PL} , since averaging the approximation of $Q_{C,n}$ over the whole test set results in $Q_C \approx 2 \cdot Q_{PL}$. Thus, it is considered that in the present thesis the average pinball-loss is enough to evaluate the performance of both the parametric and non-parametric distribution forecasts.

² cf. robjhyndman.com/files/1-ForecastEvaluation.pdf

A.5 Experimental Results

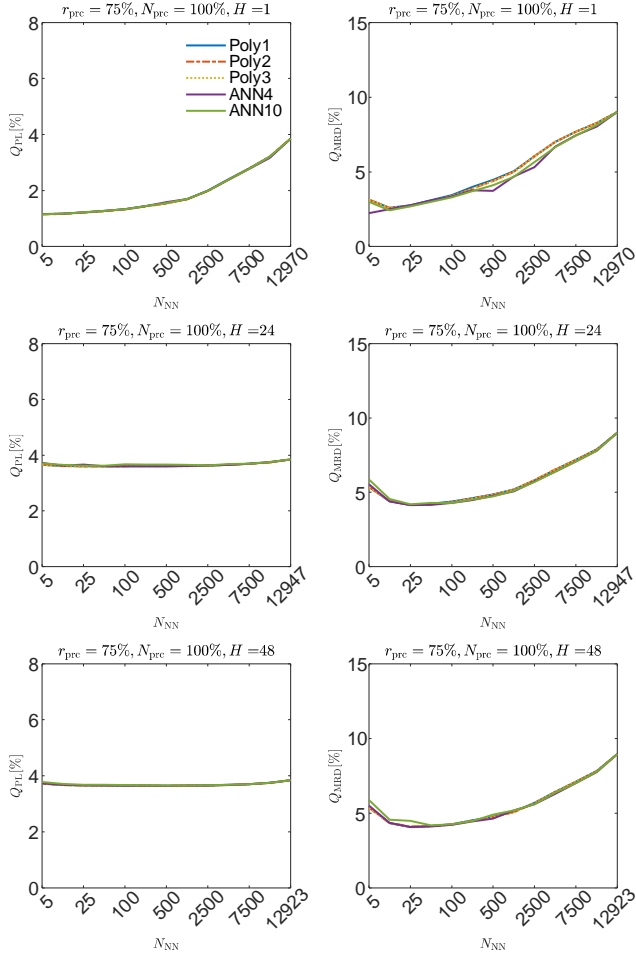


Figure A.7: Results of quantile regressions based on the NNQF for the time series with 75% of noise

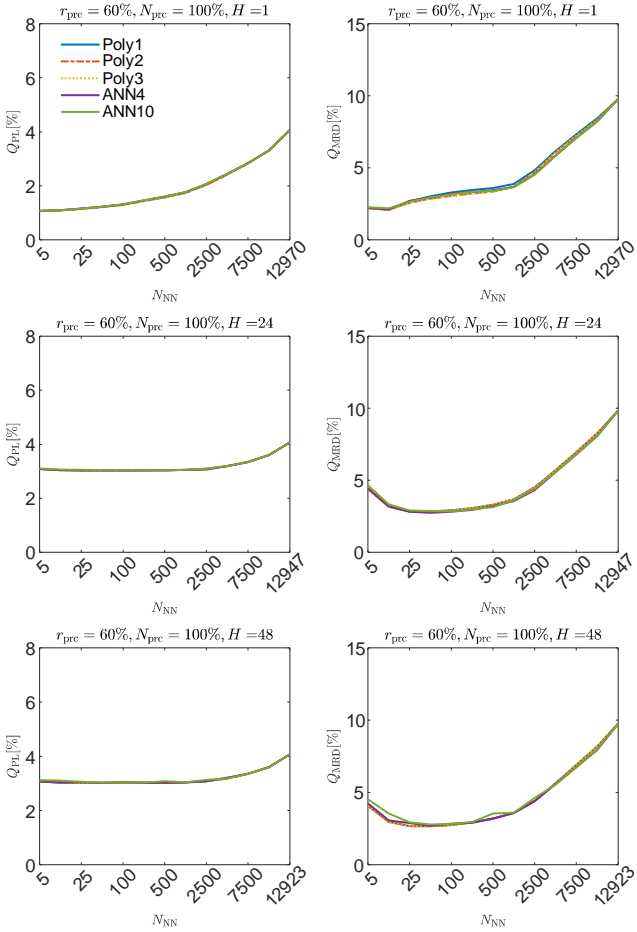


Figure A.8: Results of quantile regressions based on the NNQF for the time series with 60% of noise

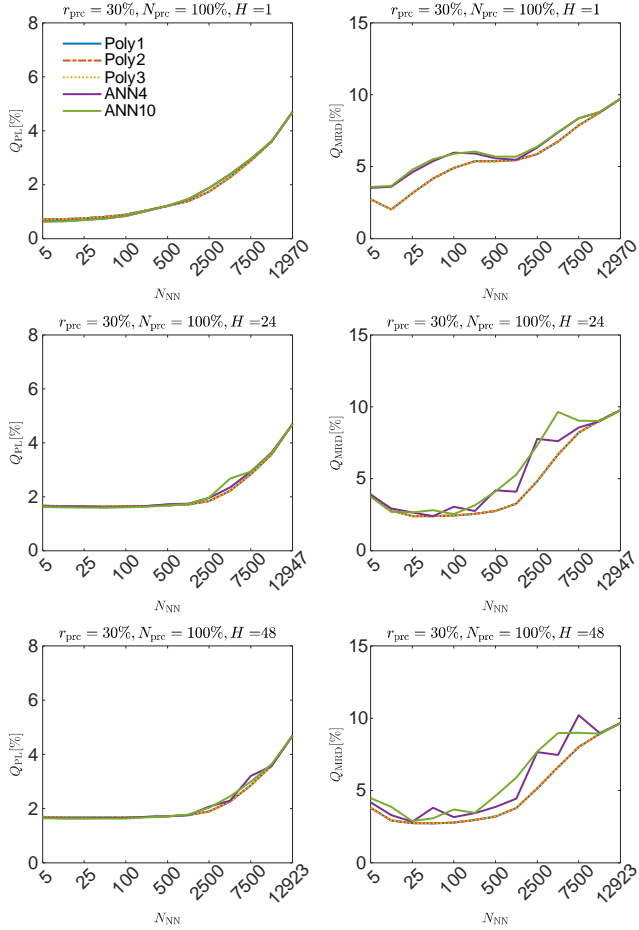


Figure A.9: Results of quantile regressions based on the NNQF for the time series with 30% of noise

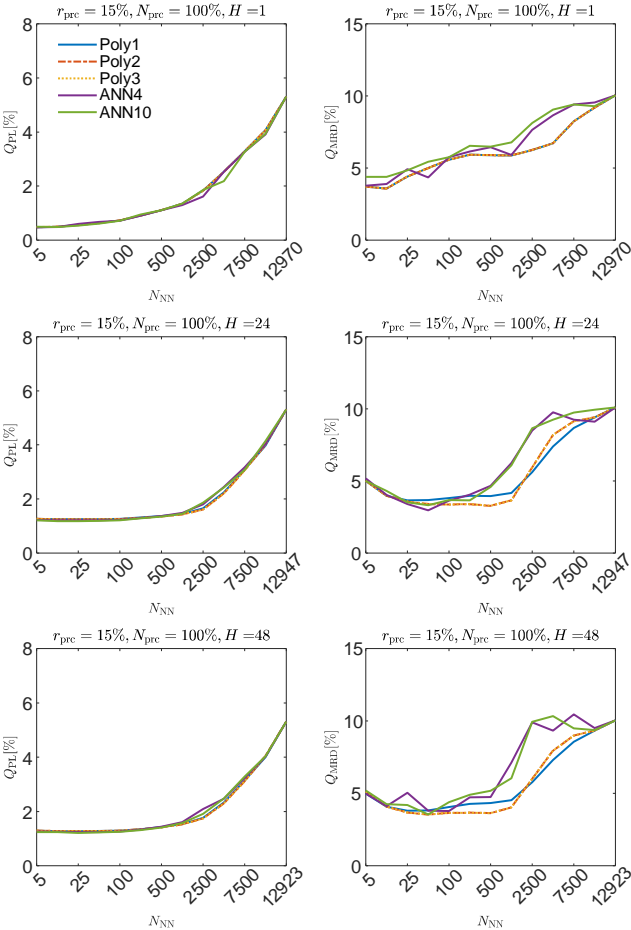


Figure A.10: Results of quantile regressions based on the NNQF for the time series with 15% of noise

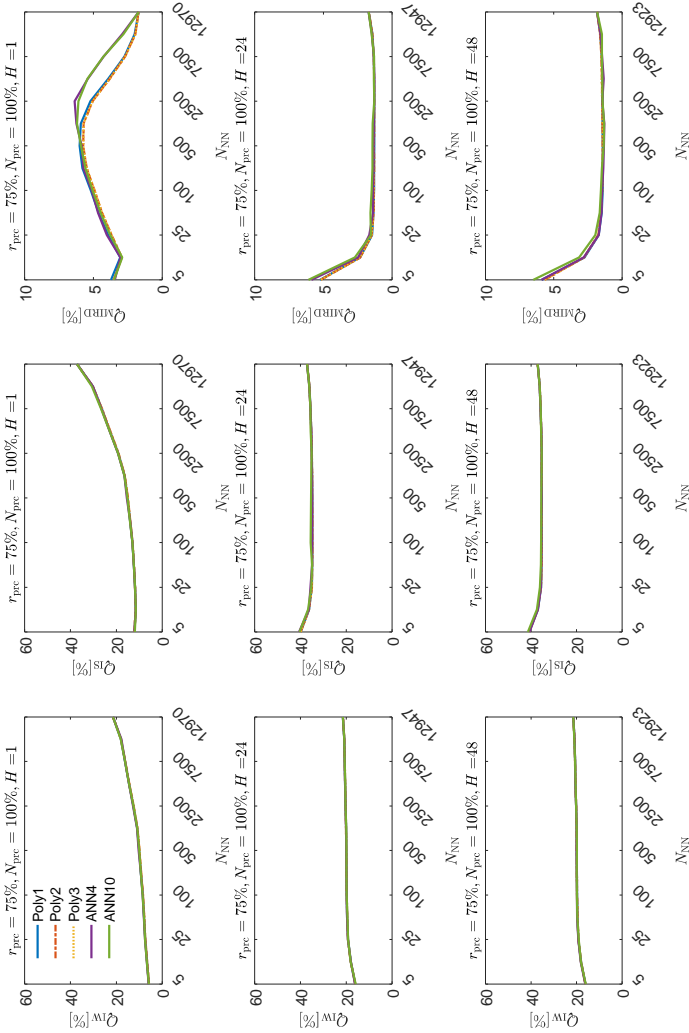


Figure A.11: Results of interval forecasts for the time series with 75% of noise

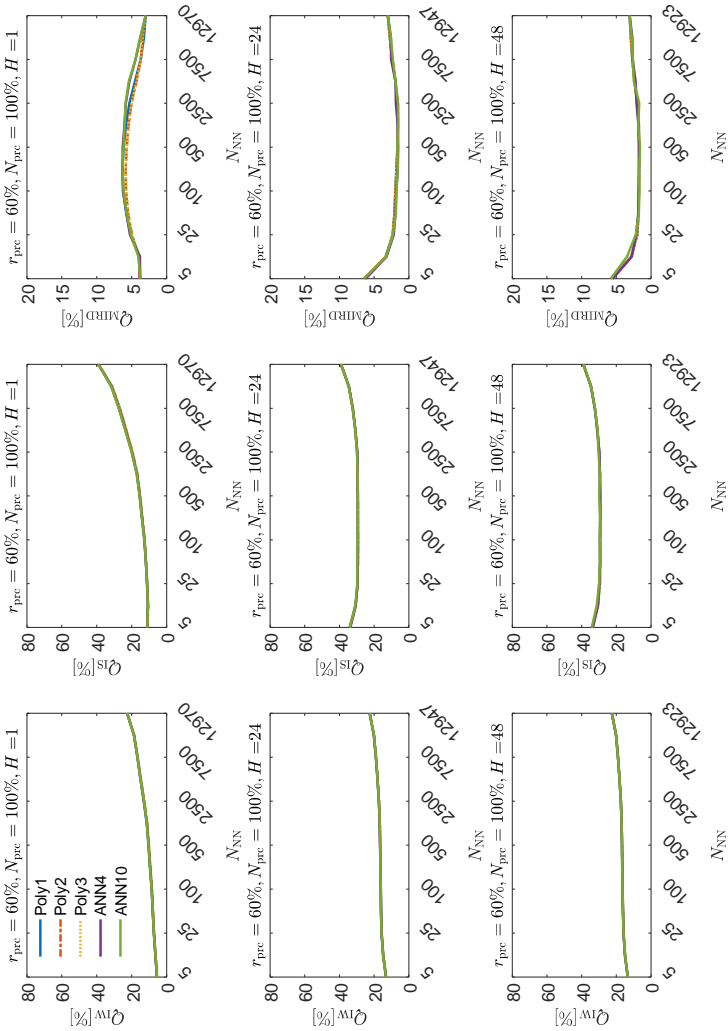


Figure A.1.2: Results of interval forecasts for the time series with 60% of noise

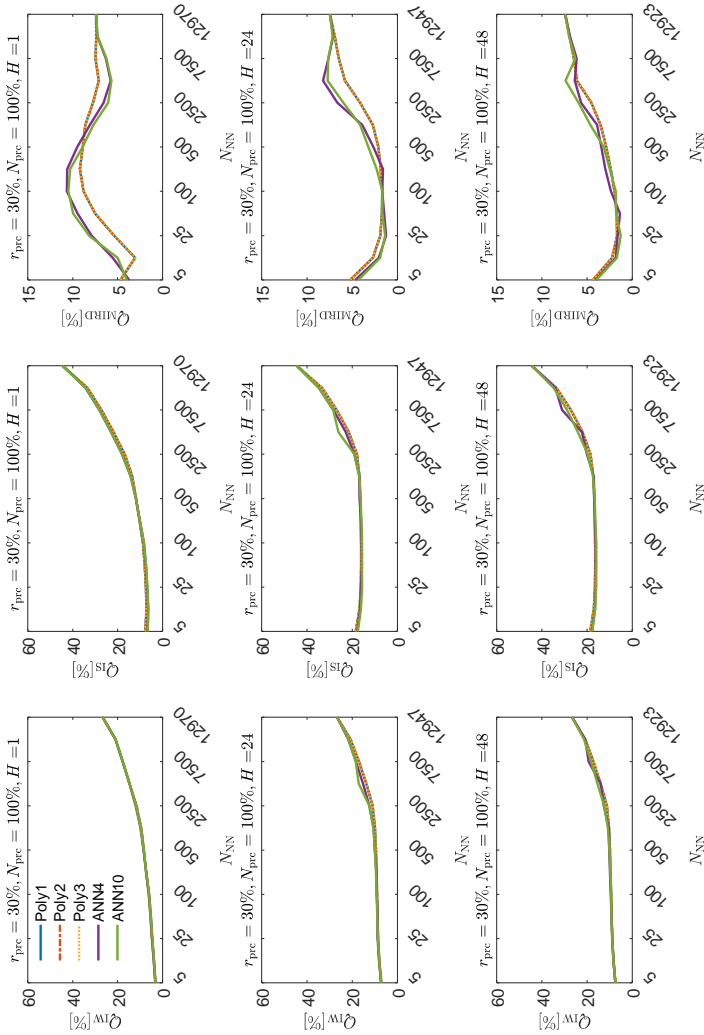


Figure A.13: Results of interval forecasts for the time series with 30% of noise

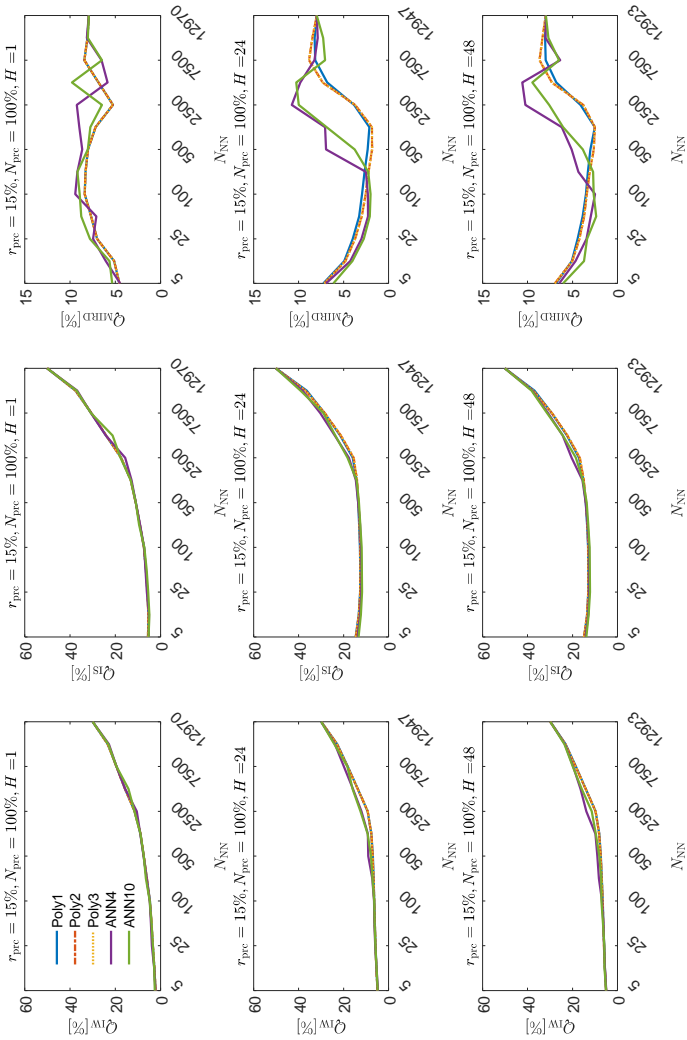


Figure A.14: Results of interval forecasts for the time series with 15% of noise

B Hierarchical Probabilistic Forecasting

B.1 Load Time Series Benchmark Generator

The load benchmark generator parameters used to obtain the time series for the examples in Chapter 3 are contained in Table B.1.

$r_{\text{prc}}[\%]$	K	E_L	α_1	α_2	α_3	α_4	α_5
$P_{L1}[k]$	26280	4	0.00	0.10	0.45	0.45	0.00
$P_{L2}[k]$	26280	3	0.00	0.10	0.45	0.45	0.00
$P_{L3}[k]$	26280	2	0.00	0.10	0.45	0.45	0.00

Table B.1: Load benchmark generator parameters for the time series used in Chapter 3

After the time series have been created, the second and third simulated load time series are modified as follows:

$$P_{L2}^*[k] = P_{L2}[k] + (P_{L1}[k])^2; k = 1, \dots, K, \quad (\text{B.1})$$

$$P_{L3}^*[k] = P_{L3}[k] + (P_{L2}^*[k])^2; k = 1, \dots, K; \quad (\text{B.2})$$

with $P_{L2}^*[k]$ and $P_{L3}^*[k]$ representing observations of the modified time series. The reason behind this modifications is to assure that the time series are dependent of one another and that their dependency is more complex than a simple linear relationship. For the sake of simplicity, the observations coming from the modified time series are referred in Chapter 3 as $P_{L2}[k]$ and $P_{L3}[k]$.

C Application

C.1 KIT Dataset Preprocessing

The KIT dataset has been preprocessed as follows:

1. The power time series is differentiated once.
2. All power values for which their difference surpasses a threshold are set to zero.
3. All power values lower than a given threshold are set as not a number (i.e. NaNs).
4. The NaN values are replaced by interpolating between them.
5. A Hampel filter is applied to remove the remaining outliers.
6. The time series is normalized to values between zero and one.

Please note that the lack of details regarding the parameters used, stems from the desire of keeping the information of the KIT's power consumption as confidential as possible.

C.2 GEFCom14 Training and Test Data

Information regarding the amount data used on each of GEFCom14's relevant tasks is given in Table C.1.

Task	Training			Test		
	K	# Days	Dates	K	# Days	Dates
4	10944	456	01.04.12 - 30.06.13	744	31	01.07.13 - 31.07.13
5	11688	487	01.04.12 - 31.07.13	744	31	01.08.13 - 31.08.13
6	12432	518	01.04.12 - 31.08.13	720	30	01.09.13 - 30.09.13
7	13152	548	01.04.12 - 30.09.13	744	31	01.10.13 - 31.10.13
8	13896	579	01.04.12 - 31.10.13	720	30	01.11.13 - 30.11.13
9	14616	609	01.04.12 - 30.11.13	744	31	01.12.13 - 31.12.13
10	15360	640	01.04.12 - 31.12.13	744	31	01.01.14 - 31.01.14
11	16104	671	01.04.12 - 31.01.14	672	28	01.02.14 - 28.02.14
12	16776	699	01.04.12 - 28.02.14	744	31	01.03.14 - 31.03.14
13	17520	730	01.04.12 - 31.03.14	720	30	01.04.14 - 30.04.14
14	18240	760	01.04.12 - 30.04.14	744	31	01.05.14 - 31.05.14
15	18984	791	01.04.12 - 31.05.14	720	30	01.06.14 - 30.06.14

Table C.1: Training and test data of the relevant tasks of GEFCom14 (adapted from [165])

C.3 Methods for Identifying Night Values

When forecasting PV power time series it is important to consider that approximately half of the values are equal to zero, since they are measured at night. In the current thesis night values are removed from the training set prior to the models training and set automatically to zero during the models application. This is done by defining different criteria to decide if a future value is to be consider a night value or not. It is important to mention that the criteria use only information stemming from input data used and that it is the same for both training and application of a model. Since the input of

the PV power forecasting models trained for the first and second experiment in Section 5 vary, their night value identification criteria are also different.

Experiment 1

In the first experiment, the models receive only exogenous values as input, from which the forecast surface solar radiation $\{\hat{G}_s[k]; k = 1, \dots, K\}$ is the one selected to define if a value is a day or a night value. To be more specific, if an input vector \mathbf{x} (cf. Equation (5.3)) inside the training set has a $\hat{G}_s[k + H]$ value lower than or equal to $100000 \text{ [Jh}^{-1}\text{m}^{-2}\text{]}$, then it is removed from the training set together with its corresponding desired output. During application, the same condition is used to define which values are set to zero; this is defined for a given quantile regression trained with the nearest neighbors quantile filter (NNQF) as follows:

$$\hat{y}_q = \begin{cases} 0 & , \text{ if } \hat{G}_s[k + H] \leq 100000 \text{ [Jh}^{-1}\text{m}^{-2}\text{]} \\ f(\mathbf{x}, \hat{\boldsymbol{\theta}}_q) & , \text{ else} \end{cases} . \quad (\text{C.1})$$

Experiment 2

In the second experiment, autoregressive values are also used as input, thus the method for identifying and removing night values is different. In this case, if an input vector \mathbf{x} (cf. Equation (5.4)) contains values of $P[k]$ and $P[k - H_1]; H_1 = 24$ lower than or equal to 0.05, it is removed from the training set together with its corresponding desired output. Similarly,

when applying an NNQF-based regression, this condition is defined with the following equation:

$$\hat{y}_q = \begin{cases} 0 & , \text{ if } (P[k] \wedge P[k - H_1]) \leq 0.05 \\ f(\mathbf{x}, \hat{\theta}_q) & , \text{ else} \end{cases} . \quad (\text{C.2})$$

Note that the previously described method is used under the consideration that the PV power time series has an hourly resolution and a 24 hours seasonality.

C.4 Results Experiment 1

Table C.2 contains examples of features selected for the data mining techniques used to solve GEFCom14's Task 15.

Technique	Selected Features			
	1	2	3	4
	NNQF			
Poly1	$\hat{G}_s[k + 24]$	$\hat{G}_t[k + 18]$	$\hat{G}_t[k + 6]$	$\hat{G}_t[k + 20]$
Poly2	$\hat{G}_s[k + 24]$	$(\hat{G}_t[k + 21])^2$	$\hat{G}_t[k + 5]$	$(\hat{G}_t[k + 24])^2$
Poly3	$\hat{G}_s[k + 24]$	$(\hat{G}_t[k + 21])^3$	$\hat{G}_t[k + 5]$	$(\hat{G}_t[k + 24])^2$
Poly4	$\hat{G}_s[k + 24]$	$(\hat{G}_t[k + 21])^4$	$\hat{G}_t[k + 6]$	$(\hat{G}_t[k + 24])^2$
ANN6	$\hat{G}_s[k + 24]$	$\hat{G}_t[k + 22]$	$\hat{G}_t[k + 24]$	$\hat{G}_t[k + 20]$
ANN10	$\hat{G}_s[k + 24]$	$\hat{G}_t[k + 22]$	$\hat{G}_{th}[k + 20]$	$\hat{G}_t[k + 2]$
	Benchmarks			
"True" Poly1	$\hat{G}_s[k + 24]$	$\hat{G}_t[k + 18]$	$\hat{G}_t[k + 6]$	$\hat{G}_t[k + 20]$
"True" Poly2	$\hat{G}_s[k + 24]$	$(\hat{G}_t[k + 21])^2$	$\hat{G}_t[k + 5]$	$(\hat{G}_t[k + 24])^2$
"True" Poly3	$\hat{G}_s[k + 24]$	$(\hat{G}_t[k + 21])^3$	$\hat{G}_t[k + 5]$	$(\hat{G}_t[k + 24])^2$
"True" Poly4	$\hat{G}_s[k + 24]$	$(\hat{G}_t[k + 21])^4$	$\hat{G}_t[k + 6]$	$(\hat{G}_t[k + 24])^2$

Table C.2: Examples of selected features used by different techniques for solving Task 15 of GEFCom14

The computation times for training and applying the quantile regressions used for Task15 (cf. Section 5.2.4) are contained in Table C.3. Notice that the computer used has an Intel Core i7-4790 processor with 3.6 [GHz] and 16 [GB] of RAM.

N_{NN}	50	100	150	200	50	100	150	200
	t [s] (Training)				t [s] (Application)			
with NNQF								
Poly1	121	124	126	129	62	63	63	63
Poly2	124	128	131	133	63	63	63	63
Poly3	124	126	130	132	63	63	63	63
Poly4	125	129	132	135	63	62	63	63
ANN6	445	449	461	453	66	65	65	65
ANN10	505	505	511	518	65	66	69	65
Benchmarks								
Poly1 TQR	229				62			
Poly2 TQR	272				62			
Poly3 TQR	235				62			
Poly4 TQR	234				62			

Table C.3: Time for training and applying the quantile regressions used in Task 15 of GEF-Com14 (adapted from [165])

C.5 Results Experiment 2

C.5.1 Results Experiment 2 (GEFCom14)

The current section provides the results used for the box plots in Section 5.2.5, as well as some additional information.

To begin with, the features selected by the data mining techniques used via the forward feature selection are given in Table C.4. Afterwards, the evaluation values obtained by the different quantile regressions are shown in Table C.5. In turn, Table C.6 contains the results of the different interval

forecasts created. Furthermore, the results obtained by parametric distributions forecasts are contained in Table C.7. Additionally, Table C.8 contains the percentage of times that each of the parametric distributions tested (i.e. Beta, Gamma, Normal, and Uniform) is considered the best fit. Lastly, Table C.9 shows the evaluation values coming from the scenario forecasts.

Technique	Selected Features			
	1	2	3	4
	ARX			
Poly1	$\hat{G}_s[k + 24]$	$P_{\max}[k]$	$\hat{G}_t[k + 24]$	$\hat{G}_{th}[k + 22]$
Poly2	$\hat{G}_s[k + 24]$	$P_{\max}[k]$	$\hat{G}_t[k + 24]$	$(\hat{G}_s[k + 24])^2$
Poly3	$\hat{G}_s[k + 24]$	$P_{\max}[k]$	$\hat{G}_t[k + 24]$	$(\hat{G}_s[k + 24])^2$
ANN4	$\hat{G}_s[k + 24]$	$P_{\max}[k]$	$\hat{G}_t[k + 24]$	$P_{\max}[k - 21]$
ANN10	$\hat{G}_s[k + 24]$	$P_{\max}[k - 24]$	$\hat{G}_t[k + 24]$	$P_{\max}[k - 20]$
"True" Poly3	$\hat{G}_s[k + 24]$	$P_{\max}[k]$	$\hat{G}_t[k + 24]$	$(\hat{G}_s[k + 24])^2$
	AR			
Poly1	$P_{\max}[k]$	$P[k]$	$P[k - 2]$	$P[k - 22]$
Poly2	$P_{\max}[k]$	$P[k]$	$(P_{\max}[k - 1])^2$	$P[k - 19]$
Poly3	$P_{\max}[k]$	$P[k]$	$(P_{\max}[k - 1])^3$	$(P[k - 22])^3$
ANN4	$P_{\max}[k]$	$P[k]$	$P[k - 3]$	$P_{\min}[k]$
ANN10	$P_{\max}[k]$	$P[k - 1]$	$P[k]$	$P_{\text{mean}}[k - 3]$
"True" Poly3	$P_{\max}[k]$	$P[k]$	$(P_{\max}[k - 1])^3$	$(P[k - 22])^3$

Table C.4: Selected features used by the different data mining techniques on the GEFCom14 dataset

ARX	Test Set Segment																			
	1	2	3	4	5	6	7	8	9	10										
Poly1	8.55	7.88	7.51	7.28	7.24	7.49	6.92	6.92	5.98	5.43	$Q_{IS} [\%]$									
Poly2	7.78	6.96	6.85	6.48	7.18	8.16	7.50	7.58	6.61	5.80	20.12	21.38	18.77	20.96	19.77	18.36	15.25	15.81	13.72	14.55
Poly3	7.78	6.96	6.85	6.48	7.18	8.16	7.50	7.58	6.61	5.80	17.48	18.95	17.31	19.72	18.29	18.10	16.04	15.37	13.52	13.92
ANNA	8.00	7.47	7.83	7.15	8.10	8.93	8.51	7.75	7.12	6.61	16.28	17.40	16.55	17.34	17.67	17.64	15.26	14.04	12.57	12.50
ANN10	7.94	7.55	7.89	7.27	8.27	9.02	8.27	7.84	7.02	6.76	15.76	16.55	16.08	16.79	17.56	17.51	15.51	13.79	12.37	12.13
"True" Poly3	7.24	5.85	5.83	5.15	6.72	8.43	8.63	8.82	7.85	7.12	15.00	15.30	14.45	15.35	15.93	17.41	16.70	15.28	13.59	13.72
ARX	$Q_{std} [\%]$										$Q_{IS} [\%]$									
Poly1	4.61	8.92	5.18	7.45	7.02	11.39	11.06	9.27	11.15	17.43										
Poly2	2.72	8.20	5.35	9.79	7.65	5.34	9.42	2.86	6.37	17.42										
Poly3	2.72	8.20	5.35	9.79	7.65	5.34	9.42	2.86	6.37	17.42										
ANNA	5.11	1.64	2.47	3.32	1.92	1.45	1.67	6.10	6.90	5.75										
ANN10	5.42	1.70	4.05	2.86	4.69	1.86	1.92	4.17	7.23	2.19										
"True" Poly3	2.11	7.19	7.47	8.53	6.55	4.91	5.00	3.10	2.49	9.71										
AR	$Q_{IW} [\%]$										$Q_{IS} [\%]$									
Poly1	11.11	10.80	10.74	10.36	10.55	10.87	10.29	9.10	8.07	7.57	20.77	19.57	19.65	18.31	20.22	20.84	20.37	18.09	14.84	16.18
Poly2	11.10	10.88	10.81	10.54	10.48	10.66	9.82	8.91	7.87	7.23	20.27	19.07	19.17	18.12	20.06	20.66	20.72	17.52	14.64	16.12
Poly3	10.92	10.79	10.79	10.52	10.45	10.64	9.83	8.92	7.96	7.31	20.32	19.10	19.26	18.05	20.00	20.61	20.47	17.51	14.65	15.91
ANNA	11.02	11.17	11.13	11.01	10.90	11.08	10.21	9.49	8.66	7.98	20.70	19.47	19.39	18.24	19.77	20.36	20.13	17.74	14.98	16.03
ANN10	10.38	10.63	10.77	10.31	10.49	11.00	10.41	9.23	8.44	8.01	21.04	19.00	19.05	17.76	19.43	20.63	20.18	17.50	14.85	16.53
"True" Poly3	10.86	10.75	10.79	10.49	10.49	10.72	9.98	8.99	8.08	7.45	20.04	18.92	19.14	17.85	19.89	20.45	20.78	17.41	14.74	15.88
AR	$Q_{std} [\%]$										$Q_{IS} [\%]$									
Poly1	1.50	1.69	2.18	4.70	3.43	2.50	6.82	4.37	2.58	5.84										
Poly2	5.99	4.81	3.72	9.04	2.02	1.72	6.60	1.71	3.48	9.52										
Poly3	3.00	3.37	1.75	7.28	2.35	2.04	6.62	3.06	2.60	9.34										
ANNA	1.87	3.42	3.20	9.33	2.13	2.17	7.32	1.14	4.98	4.52										
ANN10	2.81	3.47	3.09	7.93	1.57	3.13	7.14	1.60	3.43	3.84										
"True" Poly3	4.18	3.21	2.23	9.72	2.43	1.15	6.79	2.55	3.51	8.13										

Table C.6: Results obtained by interval forecasts on the GEFCom14 dataset; ARX: Models using forecast weather data as input; AR: pure autoregressive models; Q_{IW} : average interval width; Q_{IS} : average interval score; Q_{MIRD} : average modified interval reliability deviation

	Test Set Segment																			
	1	2	3	4	5	6	7	8	9	10										
ARX	Q_{PL} [%]																			
Poly1	1.84	1.95	1.72	1.84	1.81	1.82	1.52	1.57	1.38	1.48	Q_{MRD} [%]									
Poly2	1.67	1.79	1.65	1.82	1.79	1.83	1.66	1.56	1.39	1.48	18.36	21.28	16.59	17.63	13.53	7.49	6.47	5.64	6.65	10.27
Poly3	1.67	1.79	1.65	1.82	1.79	1.83	1.66	1.56	1.39	1.48	16.48	20.56	18.63	21.83	16.93	10.17	8.43	5.49	6.18	8.95
ANN4	1.57	1.64	1.59	1.59	1.75	1.79	1.54	1.40	1.26	1.29	16.48	20.56	18.63	21.83	16.93	10.17	8.43	5.49	6.18	8.95
ANN10	1.53	1.57	1.56	1.54	1.74	1.78	1.56	1.37	1.22	1.23	8.87	13.78	14.07	15.33	13.33	9.50	8.15	4.98	5.97	5.75
AR	Q_{PL} [%]										Q_{MRD} [%]									
Poly1	2.18	2.02	2.07	1.94	2.16	2.19	2.19	1.91	1.55	1.72	6.56	4.77	5.61	7.49	5.02	4.80	10.22	5.58	6.84	7.87
Poly2	2.11	1.96	2.01	1.91	2.13	2.17	2.22	1.86	1.53	1.72	7.88	7.61	7.36	9.02	4.62	3.94	8.64	5.79	4.72	6.34
Poly3	2.13	1.97	2.03	1.91	2.12	2.17	2.20	1.86	1.53	1.70	7.61	7.58	7.20	9.08	4.19	3.40	8.50	5.64	4.78	6.32
ANN4	2.17	1.98	2.02	1.90	2.08	2.15	2.18	1.85	1.54	1.71	8.19	6.99	7.34	9.69	4.54	3.15	7.61	5.09	4.81	5.57
ANN10	2.19	1.95	2.00	1.87	2.05	2.19	2.20	1.84	1.54	1.77	8.58	8.01	7.85	10.43	4.70	3.05	8.02	4.97	4.79	6.98

Table C.7: Results obtained by parametric distribution forecasts on the GEFCom14 dataset; ARX: Models using forecast weather data as input; AR: pure autoregressive models; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

Technique	% of best fitting distributions chosen			
	Beta	Gamma	Normal	Uniform
	ARX			
Poly1	44.02	13.68	42.30	0.00
Poly2	43.84	7.95	48.21	0.00
Poly3	43.78	6.05	50.17	0.00
ANN4	45.90	5.44	43.65	5.00
ANN10	47.54	6.07	39.51	6.89
	AR			
Poly1	41.59	0.36	32.22	25.84
Poly2	42.68	1.57	29.07	26.67
Poly3	42.48	2.11	27.91	27.50
ANN4	43.59	3.71	27.18	25.52
ANN10	44.57	4.40	26.19	24.84

Table C.8: Percentage of times a tested distribution is chosen on the GEFCom14 dataset

	Test Set Segment									
	1	2	3	4	5	6	7	8	9	10
ARX	Q_{PL} [%]									
Scenarios (Poly3)	2.73	2.74	2.46	2.27	2.32	2.38	2.36	2.34	2.64	3.00
	Q_{MRD} [%]									
Scenarios (Poly3)	14.11	16.89	15.01	18.62	21.73	19.96	25.15	21.37	26.86	31.61
AR	Q_{PL} [%]									
Scenarios (Poly3)	3.64	3.45	3.33	2.96	3.00	3.16	3.32	2.81	2.87	3.22
	Q_{MRD} [%]									
Scenarios (Poly3)	10.89	10.08	9.81	8.26	13.19	18.91	26.42	22.59	29.57	33.76

Table C.9: Results obtained by scenario forecasts on the GEFCom14 dataset; ARX: Models using forecast weather data as input; AR: pure autoregressive models; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

C.5.2 Results Experiment 2 (KIT)

The present section presents the results that are depicted in the figures shown in Section 5.2.6.

First, Table C.10 shows the features that were selected by the forward feature selection and that are used by the different data mining technique. Table C.11 contains the results obtained by the different quantile regressions. In addition, Table C.12 shows the results of the interval forecasts used. Moreover, the results of the parametric distribution forecasts are contained in Table C.13. Table C.14 contains the percentage of times one of the tested distributions (i.e. Beta, Gamma, Normal, and Uniform) is chosen on the test set. To conclude, Table C.15 shows the evaluation values stemming from the scenario forecasts.

Technique	Selected Features			
	1	2	3	4
Poly1	$P_{\max}[k - 144]$	$P[k]$	$P_{\max}[k - 168]$	$D_f[k - 2]$
Poly2	$P_{\max}[k - 144]$	$P[k]$	$P_{\max}[k - 168]$	$D_f[k - 2]$
Poly3	$P_{\max}[k - 144]$	$P[k]$	$P_{\max}[k - 168]$	$D_f[k - 2]$
ANN4	$P_{\max}[k - 144]$	$P[k]$	$D_w[k - 2]$	$D_w[k - 35]$
ANN10	$P_{\max}[k - 144]$	$P[k]$	$D_f[k]$	$P_{\max}[k - 168]$
"True" Poly3	$P_{\max}[k - 144]$	$P[k]$	$P_{\max}[k - 168]$	$D_f[k - 2]$

Table C.10: Selected features used by the different data mining techniques on the KIT dataset

	Test Set Segment									
	1	2	3	4	5	6	7	8	9	10
	Q_{PL} [%]									
Poly1	2.01	2.24	1.66	2.16	1.76	1.97	1.77	1.91	1.81	2.24
Poly2	2.01	2.24	1.66	2.16	1.76	1.97	1.77	1.91	1.81	2.24
Poly3	2.01	2.24	1.66	2.16	1.76	1.97	1.77	1.91	1.81	2.24
ANN4	1.92	1.78	1.46	1.82	1.64	1.54	1.51	1.89	1.94	2.09
ANN10	1.94	1.67	1.43	1.78	1.51	1.28	1.56	1.83	1.72	1.86
“True” Poly3	2.09	2.15	1.71	2.06	1.64	1.86	1.73	1.76	1.75	2.13
	Q_{MRD} [%]									
Poly1	4.28	7.42	2.13	11.84	15.76	14.09	12.48	23.07	4.39	11.38
Poly2	4.28	7.42	2.13	11.84	15.76	14.09	12.48	23.07	4.39	11.38
Poly3	4.28	7.42	2.13	11.84	15.76	14.09	12.48	23.07	4.39	11.38
ANN4	5.46	6.72	3.35	5.90	11.00	12.48	10.49	20.17	2.22	16.43
ANN10	4.23	7.77	3.09	7.78	11.93	8.46	11.39	21.35	3.97	11.23
“True” Poly3	3.46	5.55	3.32	9.25	11.94	11.41	9.46	18.30	3.31	10.89

Table C.11: Results obtained by quantile regressions on the KIT dataset; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

	Test Set Segment									
	1	2	3	4	5	6	7	8	9	10
	Q_{IW} [%]									
Poly1	9.61	9.56	9.69	9.45	10.78	9.19	10.17	10.71	11.32	9.87
Poly2	9.61	9.56	9.69	9.45	10.78	9.19	10.17	10.71	11.32	9.87
Poly3	9.61	9.56	9.69	9.45	10.78	9.19	10.17	10.71	11.32	9.87
ANN4	9.73	9.74	9.83	10.02	10.81	9.05	10.30	10.87	11.82	10.60
ANN10	9.57	9.80	9.55	9.84	10.79	9.21	10.03	10.76	11.73	10.46
"True" Poly3	10.75	11.12	10.93	10.59	12.51	11.97	11.32	12.29	13.36	12.25
	Q_{IS} [%]									
Poly1	19.33	22.09	16.44	21.24	17.59	21.26	17.30	18.32	17.99	22.11
Poly2	19.33	22.09	16.44	21.24	17.59	21.26	17.30	18.32	17.99	22.11
Poly3	19.33	22.09	16.44	21.24	17.59	21.26	17.30	18.32	17.99	22.11
ANN4	18.31	17.47	14.24	17.68	16.30	14.84	15.35	18.27	18.80	19.78
ANN10	18.63	16.47	14.18	17.36	15.33	12.78	15.51	17.85	17.07	17.85
"True" Poly3	20.30	20.63	17.24	19.54	17.24	18.26	17.34	17.73	18.31	20.55
	Q_{MIRD} [%]									
Poly1	5.96	11.24	0.90	9.72	1.70	5.99	1.91	6.52	1.29	7.29
Poly2	5.96	11.24	0.90	9.72	1.70	5.99	1.91	6.52	1.29	7.29
Poly3	5.96	11.24	0.90	9.72	1.70	5.99	1.91	6.52	1.29	7.29
ANN4	5.04	2.06	4.15	1.60	3.91	4.03	4.45	4.12	2.39	6.06
ANN10	6.14	2.36	5.67	1.49	5.74	8.53	2.80	2.38	3.79	2.63
"True" Poly3	5.57	5.67	4.09	7.09	8.55	5.79	2.46	3.64	6.37	2.70

Table C.12: Results obtained by interval forecasts on the KIT dataset; Q_{IW} : average interval width; Q_{IS} : average interval score; Q_{MIRD} : average modified interval reliability deviation

	Test Set Segment									
	1	2	3	4	5	6	7	8	9	10
	Q_{PL} [%]									
Poly1	2.00	2.23	1.68	2.17	1.80	1.97	1.79	1.96	1.85	2.22
Poly2	2.00	2.23	1.68	2.17	1.80	1.97	1.79	1.96	1.85	2.22
Poly3	2.00	2.23	1.68	2.17	1.80	1.97	1.79	1.96	1.85	2.22
ANN4	1.88	1.80	1.47	1.85	1.67	1.53	1.55	1.94	1.97	2.06
ANN10	1.92	1.68	1.45	1.81	1.56	1.27	1.59	1.89	1.76	1.85
	Q_{MRD} [%]									
Poly1	3.52	6.62	2.50	11.89	16.06	13.51	12.72	22.82	4.68	10.87
Poly2	3.52	6.62	2.50	11.89	16.06	13.51	12.72	22.82	4.68	10.87
Poly3	3.52	6.62	2.50	11.89	16.06	13.51	12.72	22.82	4.68	10.87
ANN4	4.40	6.70	3.53	6.49	12.09	11.92	11.35	20.52	3.17	15.06
ANN10	3.03	7.48	4.34	7.82	12.99	8.15	11.71	21.64	5.00	10.23

Table C.13: Results obtained by parametric distribution forecasts on the KIT dataset; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

Technique	% of best fitting distributions chosen			
	Beta	Gamma	Normal	Uniform
Poly1	18.25	31.63	50.12	0.00
Poly2	18.25	31.63	50.12	0.00
Poly3	18.25	31.63	50.12	0.00
ANN6	21.58	27.06	50.87	0.49
ANN10	22.31	26.40	50.43	0.86

Table C.14: Percentage of times a tested distribution is chosen on the KIT dataset

	Test Set Segment									
	1	2	3	4	5	6	7	8	9	10
	Q_{PL} [%]									
Scenarios (Poly3)	2.22	2.44	2.65	3.40	2.41	3.35	3.38	2.42	2.87	3.76
	Q_{MRD} [%]									
Scenarios (Poly3)	5.19	8.24	10.43	17.30	27.64	17.03	11.43	25.33	25.14	29.87

Table C.15: Results obtained by scenario forecasts on the KIT dataset; Q_{PL} : average pinball-loss; Q_{MRD} : average modified reliability deviation

List of Figures

1.1	Historical and future planned percentage of renewables in the German gross electricity consumption, heating sector, and transport sector	2
1.2	Pinball-loss function	9
1.3	Examples of probabilistic forecasting methods	15
1.4	Maturity of point and probabilistic energy forecasting research fields	17
1.5	Heat maps depicting normalized average daily household electrical loads across several months and different weekdays	18
1.6	Examples of energy time series forecast applications based on their forecast horizon	21
2.1	Techniques described in the present section and their relation to one another	32
2.2	Simulated load time series used as an example in Section 2.2	33
2.3	Estimation of a future value's quantile using a quantile regression based on the NNQF	35
2.4	Example training set coming from the time series shown in Figure 2.2	37
2.5	Example training set after its filtering with the NNQF	37
2.6	Quantile regressions based on the NNQF trained on the example	38
2.7	Time series obtained with the quantile regressions shown in Figure 2.6	38
2.8	Obtainment of interval forecasts using quantile regressions based on the NNQF	40

2.9	Example of interval forecasts	41
2.10	Principles behind the non-parametric distribution forecast	44
2.11	Examples of non-parametric distribution forecasts	46
2.12	Examples of parametric distribution forecasts	50
2.13	Steps for creating a scenario forecast	54
2.14	Scenario forecast example	55
2.15	Simulated load time series with 75%, 45%, and 15% of noise	62
2.16	Average pinball-loss, reliability deviation, and modified reliability deviation values on the utilized test set for different amounts of nearest neighbors	66
2.17	Example of non-trivial and trivial forecasts	67
2.18	Test sets' average pinball-loss and modified reliability deviation values obtained for different amount of training data	70
2.19	Scalability comparison for training and applying quantile regressions based on the nearest neighbors quantile filter and k-nearest neighbors quantile regressions	72
2.20	Comparison between quantile regressions based on the NNQF and "true" quantile regressions	74
2.21	Average interval width, interval score, and modified interval reliability deviation values on the utilized test set for different amounts of nearest neighbors	76
2.22	Average interval width, interval score, and modified interval reliability deviation values on the utilized test set for different amounts of nearest neighbors	80
2.23	Comparison between interval forecasts created with quantile regressions based on the NNQF and those obtained with "true" quantile regressions	81
2.24	Pinball-loss and modified reliability deviation values obtained from the quantile estimates given by quantile regressions and their corresponding parametric distribution forecasts for a forecast horizon equal to one hour	83

2.25	Pinball-loss and modified reliability deviation values obtained from the quantile estimates given by quantile regressions and their corresponding parametric distribution forecasts for a forecast horizon equal to 24 hours	84
2.26	Pinball-loss and modified reliability deviation values obtained from the quantile estimates given by quantile regressions and their corresponding parametric distribution forecasts for a forecast horizon equal to 48 hours	85
2.27	Root mean square error plotted against different expected values and variances	89
2.28	Comparison between the pinball-loss and modified reliability deviation values of the scenario forecast and quantile regressions based on the NNQF	91
3.1	Example of a time series hierarchy	98
3.2	Simulated load time series used in the examples of the present chapter	98
3.3	Example of the forecast marginal distributions	102
3.4	Example of the joint distribution forecast	102
3.5	Example of a modified time series hierarchy	103
3.6	Results from the direct forecasts, the forecasts obtained assuming independence, and the forecasts created via the present thesis method	105
3.7	Interval forecasts obtained with the forecasts that assume independence of the summed values and those obtained by the present thesis approach	106
4.1	Different types of forecasts available in SciXMiner	110
4.2	User interface for quantile regressions based in the NNQF . . .	111
4.3	User interface for interval forecasts	114
4.4	User interface for parametric forecasts	115

4.5	User interface for scenario forecasts	118
4.6	User interface for hierarchical forecasts	120
4.7	Example of a forecasting model implemented in the framework described in [175] using the steps presented in the current section	123
5.1	Pinball-loss and computational effort for either training or ap- plying the quantile regressions	134
5.2	Results obtained by quantile regressions on the GEFCom14 dataset	136
5.3	Modified reliability deviation obtained by adding a bias to the quantile regressions	137
5.4	Results obtained by interval forecasts on the GEFCom14 dataset	139
5.5	Results obtained by parametric and non-parametric distribution forecasts on the GEFCom14 dataset	140
5.6	Results obtained by scenario forecasts and quantile regressions used for comparison on the GEFCom14 dataset	141
5.7	Results obtained by quantile regressions on the KIT dataset . . .	142
5.8	Results obtained by interval forecasts on the KIT dataset	144
5.9	Results obtained by parametric and non-parametric distribution forecasts on the KIT dataset	144
5.10	Results obtained by scenario forecasts and quantile regressions used for comparison on the KIT dataset	145
5.11	Dispatchable Feeder	147
5.12	Probabilistic forecasts for the inflexible active power and energy	148
A.1	Simulated time series	158
A.2	Simulated time series trend component	158
A.3	Simulated time series periodic component	159
A.4	Simulated time series pattern component	159
A.5	Simulated time series stochastic component	159
A.6	Simulated time series exogenous component	160

A.7	Results of quantile regressions based on the NNQF for the time series with 75% of noise	163
A.8	Results of quantile regressions based on the NNQF for the time series with 60% of noise	164
A.9	Results of quantile regressions based on the NNQF for the time series with 30% of noise	165
A.10	Results of quantile regressions based on the NNQF for the time series with 15% of noise	166
A.11	Results of interval forecasts for the time series with 75% of noise	167
A.12	Results of interval forecasts for the time series with 60% of noise	168
A.13	Results of interval forecasts for the time series with 30% of noise	169
A.14	Results of interval forecasts for the time series with 15% of noise	170

List of Tables

1.1	Categorization of energy-related forecasting approaches based on their forecast horizon	20
1.2	Examples of point forecasting approaches found in the literature	24
1.3	Examples of probabilistic forecasting approaches found in the literature	26
2.1	Examples of starting and ending parts of the non-parametric CDF estimate	45
2.2	Examples of how the parameters of four different parametric distributions are obtained.	49
2.3	Type and parameters of the parametric CDFs shown in Figure 2.12	50
2.4	Number of nearest neighbors used depending on the available training data	65
2.5	Optimal number of nearest neighbors and evaluation results obtained for time series with different noise percentages	68
2.6	Optimal number of nearest neighbors and interval evaluation results obtained for time series with different noise percentages .	78
2.7	Average pinball-loss and modified reliability deviation values of non-parametric (i.e. quantile regressions) and parametric distribution forecasts	86
2.8	Percentage of the chosen parametric distribution that best fits the non-parametric estimates	87
2.9	Average root mean square error obtained by the parametric distributions considered the best fit for each data mining technique used	88

2.10	Difference between the average root mean square error obtained with the present thesis method and the possible average root mean square error obtained by a parametric distribution considered an optimal fit	89
2.11	Comparison of average pinball-loss and modified reliability deviation values of scenario forecast and quantile regressions based on the NNQF	92
4.1	Sections describing the theory and implementation of the present thesis methods	109
4.2	General options of the quantile regressions based on the NNQF GUI	112
4.3	NNQF options of the quantile regressions based on the NNQF GUI	112
4.4	Options of the interval forecasts GUI	114
4.5	Options for creating parametric distribution forecasts of the parametric distribution forecast GUI	116
4.6	Options for applying parametric distribution forecasts of the parametric distribution forecast GUI	117
4.7	Options for creating scenario forecasts of the scenario forecast GUI	119
4.8	Options for creating and applying scenario forecasts of the scenario forecast GUI	119
4.9	Options of the hierarchical probabilistic forecasts GUI	121
5.1	Information of the real data used in the current thesis	126
5.2	Number of lags and exogenous inputs used for each dataset . . .	131
5.3	GEFCom14 average pinball-loss	133
5.4	Costs for using direct forecast scheduling (DFS), probabilistic forecast scheduling (PFS), or scenario forecast scheduling (SFS) in the simulation described in [178] and [179]	149

A.1	Load benchmark generator parameters for the time series used in Section 2.2	160
A.2	Load benchmark generator parameters for the time series used in Section 2.4	160
B.1	Load benchmark generator parameters for the time series used in Chapter 3	171
C.1	Training and test data of the relevant tasks of GEFCom14	174
C.2	Examples of selected features used by different techniques for solving Task 15 of GEFCom14	176
C.3	Time for training and applying the quantile regressions used in Task 15 of GEFCom14	177
C.4	Selected features used by the different data mining techniques on the GEFCom14 dataset	178
C.5	Results obtained by quantile regressions on the GEFCom14 dataset	179
C.6	Results obtained by interval forecasts on the GEFCom14 dataset	180
C.7	Results obtained by parametric distribution forecasts on the GEFCom14 dataset	181
C.8	Percentage of times a tested distribution is chosen on the GEFCom14 dataset	182
C.9	Results obtained by scenario forecasts on the GEFCom14 dataset	182
C.10	Selected features used by the different data mining techniques on the KIT dataset	183
C.11	Results obtained by quantile regressions on the KIT dataset	184
C.12	Results obtained by interval forecasts on the KIT dataset	185
C.13	Results obtained by parametric distribution forecasts on the KIT dataset	186
C.14	Percentage of times a tested distribution is chosen on the KIT dataset	186

C.15 Results obtained by scenario forecasts on the KIT dataset 186

Bibliography

- [1] NAJAFI, M. R.; ZWIERS, F. W.; GILLETT, N. P.: Attribution of Arctic Temperature Change to Greenhouse-Gas and Aerosol Influences. *Nature Climate Change* 5(3) (2015), pp. 246–249.
- [2] BMWI: Die Energie der Zukunft – Fünfter Monitoring-Bericht zur Energiewende, Berichtsjahr 2015. Tech. Rep., Bundesministerium für Wirtschaft und Energie (BMWi), 2016.
- [3] WACZOWICZ, S.: *Konzept zur datengetriebenen Analyse und Modellierung des preisbeeinflussten Verbrauchsverhaltens*. Dissertation, Karlsruher Institut für Technologie (KIT), Fakultät für Maschinenbau, submitted on December 22, 2017, 2018.
- [4] AGEE-STAT: Zeitreihen zur Entwicklung der erneuerbaren Energien in Deutschland. Tech. Rep., Umweltbundesamt, Fachgebiet I 2.5 Energieversorgung und -daten, Geschäftsstelle der Arbeitsgruppe Erneuerbare Energien-Statistik (AGEE-Stat), 2017.
- [5] BMU; BMWI: Das Energiekonzept der Bundesregierung 2010 und die Energiewende 2011. Tech. Rep., Bundesministerium für Wirtschaft und Technologie (BMWi), Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit (BMU), 2011.
- [6] ROBINIUS, M.; OTTO, A.; HEUSER, P.; WELDER, L.; SYRANIDIS, K.; RYBERG, S. D.; GRUBE, T.; MARKEWITZ, P.; PETERS, R.; STOLTEN, D.: Linking the Power and Transport Sectors - Part 1: The Principle of Sector Coupling. *Energies* 10(7) (2017), p. 956.

- [7] ROBINIUS, M.; OTTO, A.; SYRANIDIS, K.; RYBERG, S. D.; HEUSER, P.; WELDER, L.; GRUBE, T.; MARKEWITZ, P.; TIETZE, V.; STOLTEN, D.: Linking the Power and Transport Sectors - Part 2: Modelling a Sector Coupling Scenario for Germany. *Energies* 10(7) (2017), p. 957.
- [8] ROHJANS, S.; LEHNHOFF, S.; SCHÜTTE, S.; ANDRÉN, F.; STRASSER, T.: Requirements for Smart Grid Simulation Tools. In: *Proc., 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE), Istanbul, Turkey*, pp. 1730–1736, IEEE, Piscataway, NJ, USA, 2014.
- [9] FANG, X.; MISRA, S.; XUE, G.; YANG, D.: Smart Grid - the New and Improved Power Grid: A Survey. *IEEE Communications Surveys Tutorials* 14(4) (2012), pp. 944–980.
- [10] HAGENMEYER, V.; ÇAKMAK, H. K.; DÜPMEIER, C.; FAULWASSER, T.; ISELE, J.; KELLER, H. B.; KOHLHEPP, P.; KÜHNAPFEL, U.; STUCKY, U.; WACZOWICZ, S.; MIKUT, R.: Information and Communication Technology in Energy Lab 2.0: Smart Energies System Simulation and Control Center with an Open-Street-Map-based Power Flow Simulation Example. *Energy Technology* 4 (2016), pp. 145–162.
- [11] JAKOB, W.; GONZÁLEZ ORDIANO, J. Á.; LUDWIG, N.; MIKUT, R.; HAGENMEYER, V.: Towards Coding Strategies for Forecasting-Based Scheduling in Smart Grids and the Energy Lab 2.0. In: *Proc., Genetic and Evolutionary Computation Conference Companion, Berlin, Germany*, pp. 1271–1278, ACM, New York, NY, USA, 2017.
- [12] GOTTWALT, S.; GÄRTTNER, J.; SCHMECK, H.; WEINHARDT, C.: Modeling and Valuation of Residential Demand Flexibility for Renewable Energy Integration. *IEEE Transactions on Smart Grid* 8(6) (2017), pp. 2565–2574.

-
- [13] LUDWIG, N.; WACZOWICZ, S.; MIKUT, R.; HAGENMEYER, V.: Mining Flexibility Patterns in Energy Time Series from Industrial Processes. In: *Proc., 27. Workshop Computational Intelligence, Dortmund, Germany*, pp. 13–32, KIT Scientific Publishing, Karlsruhe, Germany, 2017.
- [14] DANNECKER, L.: *Energy Time Series Forecasting: Efficient and Accurate Forecasting of Evolving Time Series from the Energy Domain*. Springer Vieweg, Wiesbaden, Germany, 2015.
- [15] KREUTZ, S.; BELITZ, H.-J.; REHTANZ, C.: The Impact of Demand Side Management on the Residual Load. In: *Proc., 2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe), Gothenburg, Sweden*, pp. 1–5, IEEE, Piscataway, NJ, USA, 2010.
- [16] MÜLLER, D.; MONTI, A.; STINNER, S.; SCHLÖSSER, T.; SCHÜTZ, T.; MATTHES, P.; WOLISZ, H.; MOLITOR, C.; HARB, H.; STREBLOW, R.: Demand Side Management for City Districts. *Building and Environment* 91 (2015), pp. 283–293.
- [17] DOTZAUER, E.: Simple Model for Prediction of Loads in District-Heating Systems. *Applied Energy* 73(3) (2002), pp. 277–284.
- [18] YU, F.; XU, X.: A Short-Term Load Forecasting Model of Natural Gas Based on Optimized Genetic Algorithm and Improved BP Neural Network. *Applied Energy* 134 (2014), pp. 102–113.
- [19] SCHIEBAHN, S.; GRUBE, T.; ROBINIUS, M.; TIETZE, V.; KUMAR, B.; STOLTEN, D.: Power to Gas: Technological Overview, Systems Analysis and Economic Assessment for a Case Study in Germany. *International Journal of Hydrogen Energy* 40(12) (2015), pp. 4285–4294.

- [20] BÖTTGER, D.; GÖTZ, M.; LEHR, N.; KONDZIELLA, H.; BRUCKNER, T.: Potential of the Power-to-Heat Technology in District Heating Grids in Germany. *Energy Procedia* 46 (2014), pp. 246–253.
- [21] KOHLHEPP, P.; HAGENMEYER, V.: Technical Potential of Buildings in Germany as Flexible Power-to-Heat Storages for Smart Grid Operation. *Energy Technology* 5(7) (2017), pp. 1084–1104.
- [22] HIPPERT, H. S.; PEDREIRA, C. E.; SOUZA, R. C.: Neural Networks for Short-Term Load Forecasting: A Review and Evaluation. *IEEE Transactions on Power Systems* 16(1) (2001), pp. 44–55.
- [23] FEINBERG, E. A.; GENETHLIOU, D.: Load Forecasting. In: *Applied Mathematics for Restructured Electric Power Systems*, pp. 269–285, Springer, Boston, MA, USA, 2005.
- [24] TAKIYAR, S.; SINGH, V.: Trend Analysis and Evolution of Short Term Load Forecasting Techniques. In: *Proc., 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions)*, Noida, India, pp. 1–6, IEEE, Piscataway, NJ, USA, 2015.
- [25] HYNDMAN, R. J.; ATHANASOPOULOS, G.: *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 2014.
- [26] MAIMON, O.; ROKACH, L.: *Data Mining and Knowledge Discovery Handbook*, Vol. 2. Springer, New York, NY, USA, 2005.
- [27] GNEITING, T.: Quantiles as Optimal Point Forecasts. *International Journal of Forecasting* 27(2) (2011), pp. 197–207.
- [28] GNEITING, T.; KATZFUSS, M.: Probabilistic Forecasting. *Annual Review of Statistics and its Application* 1 (2014), pp. 125–151.
- [29] HONG, T.; PINSON, P.; FAN, S.; ZAREIPOUR, H.; TROCCOLI, A.; HYNDMAN, R. J.: Probabilistic Energy Forecasting: Global Energy

- Forecasting Competition 2014 and Beyond. *International Journal of Forecasting* 32(3) (2016), pp. 896 – 913.
- [30] GONZÁLEZ ORDIANO, J. Á.; WACZOWICZ, S.; HAGENMEYER, V.; MIKUT, R.: Energy Forecasting Tools and Services. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8(2) (2018), p. e1235.
- [31] CARLTON, M. A.; DEVORE, J. L.: *Probability with Applications in Engineering, Science, and Technology*. Springer, Cham, Switzerland, 2017.
- [32] SCHAY, G.: *Introduction to Probability with Statistical Applications*. Birkhäuser, Cham, Switzerland, 2016.
- [33] ZÖRNIG, P.: *Probability Theory and Statistical Applications: A Profound Treatise for Self-Study*. Walter de Gruyter GmbH & Co KG, Berlin, Germany, 2016.
- [34] MURPHY, K. P.: *Machine Learning: A Probabilistic Perspective*. MIT press, Cambridge, MA, USA, 2012.
- [35] FAHRMEIR, L.; KNEIB, T.; LANG, S.; MARX, B.: *Regression: Models, Methods and Applications*. Springer, Berlin, Germany, 2013.
- [36] JAWORSKI, P.; DURANTE, F.; HARDLE, W. K.; RYCHLIK, T.: *Copula Theory and its Applications*. Springer Berlin Heidelberg, Berlin, Heidelberg, Germany, 2010.
- [37] PATTON, A. J.: Copula-Based Models for Financial Time Series. In: *Handbook of Financial Time Series*, pp. 767–785, Springer Berlin Heidelberg, Berlin, Heidelberg, Germany, 2009.
- [38] SMITH, M. S.: Copula Modelling of Dependence in Multivariate Time Series. *International Journal of Forecasting* 31(3) (2015), pp. 815–833.

- [39] PATTON, A. J.: Modelling Asymmetric Exchange Rate Dependence. *International Economic Review* 47(2) (2006), pp. 527–556.
- [40] BARTSCHAT, A.; REISCHL, M.; MIKUT, R.: Data Mining Tools. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (2019), p. e1309.
- [41] FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P.: From Data Mining to Knowledge Discovery in Databases. *AI Magazine* 17 (1996), pp. 37–54.
- [42] HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer New York, New York, NY, USA, 2008.
- [43] STULP, F.; SIGAUD, O.: Many Regression Algorithms, One Unified Model: A Review. *Neural Networks* 69 (2015), pp. 60–79.
- [44] JUBAN, R.; OHLSSON, H.; MAASOUMY, M.; POIRIER, L.; KOLTER, J. Z.: A Multiple Quantile Regression Approach to the Wind, Solar, and Price Tracks of GEFCom2014. *International Journal of Forecasting* 32(3) (2016), pp. 1094–1102.
- [45] GAGLIANONE, W. P.; LIMA, L. R.: Constructing Density Forecasts from Quantile Regressions. *Journal of Money, Credit and Banking* 44(8) (2012), pp. 1589–1607.
- [46] DONEIT, W. V.: *Eine neue Methode zum robusten Entwurf von Regressionsmodellen bei beschränkter Rohdatenqualität*. Dissertation, Karlsruher Institut für Technologie (KIT), Karlsruhe, 2018.
- [47] BROCKWELL, P. J.; DAVIS, R. A.: *Introduction to Time Series and Forecasting*. Springer International Publishing, Cham, Switzerland, 2016.
- [48] GARDNER JR., E. S.: Exponential Smoothing: The State of the Art Part II. *International Journal of Forecasting* 22(4) (2006), pp. 637 – 666.

-
- [49] DE GOOIJER, J. G.; HYNDMAN, R. J.: 25 Years of Time Series Forecasting. *International Journal of Forecasting* 22(3) (2006), pp. 443–473.
- [50] LJUNG, L.: *System Identification - Theory for the User*. Prentice Hall, Upper Saddle River, NJ, USA, 1999.
- [51] SHUMWAY, R. H.; STOFFER, D. S.: *Time Series Analysis and its Applications*. Springer, Cham, Switzerland, 2017.
- [52] SMOLA, A.; SCHÖLKOPF, B.: A Tutorial on Support Vector Regression. *Statistics and Computing* 14(3) (2004), pp. 199–222.
- [53] HONG, W.-C.: Electric Load Forecasting by Support Vector Model. *Applied Mathematical Modelling* 33(5) (2009), pp. 2444–2454.
- [54] RAZA, M. Q.; KHOSRAVI, A.: A Review on Artificial Intelligence Based Load Demand Forecasting Techniques for Smart Grid and Buildings. *Renewable and Sustainable Energy Reviews* 50 (2015), pp. 1352–1372.
- [55] MIKUT, R.: *Data Mining in der Medizin und Medizintechnik*. Universitätsverlag Karlsruhe, Karlsruhe, Germany, 2008.
- [56] SUN, A. Y.; WANG, D.; XU, X.: Monthly Streamflow Forecasting using Gaussian Process Regression. *Journal of Hydrology* 511 (2014), pp. 72 – 81.
- [57] TASICARA OGLU, A.; SANANDAJI, B. M.: Short-Term Residential Electric Load Forecasting: A Compressive Spatio-Temporal Approach. *Energy and Buildings* 111 (2016), pp. 380–392.
- [58] YANG, D.; GU, C.; DONG, Z.; JIRUTITIJAROEN, P.; CHEN, N.; WALSH, W. M.: Solar Irradiance Forecasting using Spatial-Temporal Covariance Structures and Time-Forward Kriging. *Renewable Energy* 60 (2013), pp. 235–245.

- [59] ZHAO, Y.; LI, J.; YU, L.: A Deep Learning Ensemble Approach for Crude Oil Price Forecasting. *Energy Economics* 66 (2017), pp. 9 – 16.
- [60] ATHANASOPOULOS, G.; AHMED, R. A.; HYNDMAN, R. J.: Hierarchical Forecasts for Australian Domestic Tourism. *International Journal of Forecasting* 25(1) (2009), pp. 146–166.
- [61] FLIESS, M.; JOIN, C.; VOYANT, C.: Prediction Bands for Solar Energy: New Short-Term Time Series Forecasting Techniques. *Solar Energy* 166 (2018), pp. 519–528.
- [62] ZHANG, Y.; WANG, J.; WANG, X.: Review on Probabilistic Forecasting of Wind Power Generation. *Renewable and Sustainable Energy Reviews* 32 (2014), pp. 255–270.
- [63] CHATFIELD, C.: Calculating Interval Forecasts. *Journal of Business & Economic Statistics* 11(2) (1993), pp. 121–135.
- [64] TORGO, L.; OHASHI, O.: 2D-Interval Predictions for Time Series. In: *Proc., 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, California, USA*, pp. 787–794, ACM, New York, NY, USA, 2011.
- [65] GONZÁLEZ ORDIANO, J. Á.; APPINO, R. R.; FAULWASSER, T.; HAGENMEYER, V.; MIKUT, R.: On the Creation of Quantile Regression based Scenario and Parametric Probabilistic Forecasts in the Context of Energy Forecasting. In: *Proc., 38th International Symposium on Forecasting, Boulder, CO, USA*, p. 74, International Institute of Forecasters (IIF), Medford, MA, USA, 2018.
- [66] BRAHIM-BELHOUARI, S.; BERMAK, A.: Gaussian Process for Nonstationary Time Series Prediction. *Computational Statistics & Data Analysis* 47(4) (2004), pp. 705–712.

-
- [67] GIRARD, A.; RASMUSSEN, C. E.; CANDELA, J. Q.; MURRAY-SMITH, R.: Gaussian Process Priors with Uncertain Inputs Application to Multiple-Step Ahead Time Series Forecasting. In: *Proc., Advances in Neural Information Processing Systems 15 (NIPS 2002)*, pp. 545–552, MIT press, Cambridge, MA, USA, 2002.
- [68] RASMUSSEN, C. E.; WILLIAMS, C. K. I.: *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, USA, 2006.
- [69] ROBERTS, S.; OSBORNE, M.; EBDEN, M.; REECE, S.; GIBSON, N.; AIGRAIN, S.: Gaussian Processes for Time-Series Modelling. *Phil. Trans. R. Soc. A* 371(1984) (2013), p. 20110550.
- [70] KOU, P.; LIANG, D.; GAO, L.; LOU, J.: Probabilistic Electricity Price Forecasting with Variational Heteroscedastic Gaussian Process and Active Learning. *Energy Conversion and Management* 89 (2015), pp. 298 – 308.
- [71] MORI, H.; KURATA, E.: Application of Gaussian Process to Wind Speed Forecasting for Wind Power Generation. In: *Proc., 2008 IEEE International Conference on Sustainable Energy Technologies, Singapore*, pp. 956–959, IEEE, Piscataway, NJ, USA, 2008.
- [72] BREMNES, J. B.: Probabilistic Wind Power Forecasts Using Local Quantile Regression. *Wind Energy* 7(1) (2004), pp. 47–54.
- [73] GAILLARD, P.; GOUDE, Y.; NEDELLEC, R.: Additive Models and Robust Aggregation for GEFCom2014 Probabilistic Electric Load and Electricity Price Forecasting. *International Journal of Forecasting* 32(3) (2016), pp. 1038–1050.
- [74] MA, X.; HE, X.; SHI, X.: A Variant of K Nearest Neighbor Quantile Regression. *Journal of Applied Statistics* 43(3) (2016), pp. 526–537.
- [75] MEINSHAUSEN, N.: Quantile Regression Forests. *Journal of Machine Learning Research* 7 (2006) Jun, pp. 983–999.

- [76] NIELSEN, H. A.; MADSEN, H.; NIELSEN, T. S.: Using Quantile Regression to Extend an Existing Wind Power Forecasting System with Probabilistic Forecasts. *Wind Energy* 9(1-2) (2006), pp. 95–108.
- [77] PINSON, P.; NIELSEN, H. A.; MØLLER, J. K.; MADSEN, H.; KARINIOTAKIS, G. N.: Non-Parametric Probabilistic Forecasts of Wind Power: Required Properties and Evaluation. *Wind Energy* 10(6) (2007), pp. 497–516.
- [78] BESSA, R. J.; MIRANDA, V.; BOTTERUD, A.; WANG, J.; CONSTANTINESCU, E. M.: Time Adaptive Conditional Kernel Density Estimation for Wind Power Forecasting. *IEEE Transactions on Sustainable Energy* 3(4) (2012), pp. 660–669.
- [79] JUBAN, J.; SIEBERT, N.; KARINIOTAKIS, G. N.: Probabilistic Short-Term Wind Power Forecasting for the Optimal Management of Wind Generation. In: *Proc., 2007 IEEE Lausanne Power Tech, Lausanne, Switzerland*, pp. 683–688, IEEE, Piscataway, NJ, USA, 2007.
- [80] ZHANG, Y.; WANG, J.: GEFCom2014 Probabilistic Solar Power Forecasting based on k-Nearest Neighbor and Kernel Density Estimator. In: *Proc., 2015 IEEE Power & Energy Society General Meeting, Denver, CO, USA*, pp. 1–5, IEEE, Piscataway, NJ, USA, 2015.
- [81] PINSON, P.; MADSEN, H.: Ensemble-Based Probabilistic Forecasting at Horns Rev. *Wind Energy* 12(2) (2009), pp. 137–155.
- [82] GNEITING, T.; RAFTERY, A. E.: Weather Forecasting with Ensemble Methods. *Science* 310(5746) (2005), pp. 248–249.
- [83] ZHU, Y.: Ensemble Forecast: A New Approach to Uncertainty and Predictability. *Advances in Atmospheric Sciences* 22(6) (2005), pp. 781–788.
- [84] BRÖCKER, J.; SMITH, L. A.: From Ensemble Forecasts to Predictive Distribution Functions. *Tellus A* 60(4) (2008), pp. 663–678.

-
- [85] TAIEB, S. B.; TAYLOR, J. W.; HYNDMAN, R. J.: Coherent Probabilistic Forecasts for Hierarchical Time Series. In: *Proc., 34th International Conference on Machine Learning, Sydney, Australia*, pp. 3348–3357, PMLR, 2017.
- [86] COCOCCIONI, M.; D’ANDREA, E.; LAZZERINI, B.: One Day-Ahead Forecasting of Energy Production in Solar Photovoltaic Installations: An Empirical Study. *Intelligent Decision Technologies* 6(3) (2012), pp. 197–210.
- [87] LEI, M.; SHIYAN, L.; CHUANWEN, J.; HONGLING, L.; YAN, Z.: A Review on the Forecasting of Wind Speed and Generated Power. *Renewable and Sustainable Energy Reviews* 13(4) (2009), pp. 915–920.
- [88] ALMONACID, F.; RUS, C.; PÉREZ-HIGUERAS, P.; HONTORIA, L.: Calculation of the Energy Provided by a PV Generator. Comparative Study: Conventional Methods vs. Artificial Neural Networks. *Energy* 36(1) (2011), pp. 375–384.
- [89] FUENTES, M.; NOFUENTES, G.; AGUILERA, J.; TALAVERA, D.; CASTRO, M.: Application and Validation of Algebraic Methods to Predict the Behaviour of Crystalline Silicon PV Modules in Mediterranean Climates. *Solar Energy* 81(11) (2007), pp. 1396–1408.
- [90] REKIOUA, D.: *Wind Power Electric Systems*. Springer, London, UK, 2014.
- [91] IDOWU, S.; SAGUNA, S.; ÅHLUND, C.; SCHELÉN, O.: Forecasting Heat Load for Smart District Heating Systems: A Machine Learning Approach. In: *Proc., 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), Venice, Italy*, pp. 554–559, IEEE, Piscataway, NJ, USA, 2014.
- [92] HAMMERSTROM, D. J.; AMBROSIO, R.; BROUS, J.; CARLON, T. A.; CHASSIN, D. P.; DESTEESE, J. G.; GUTTROMSON, R. T.;

- HORST, G. R.; JÄRVEGREN, O. M.; KAJFASZ, R.; ET AL.: Pacific Northwest GridWise Testbed Demonstration Projects, Part I. Olympic Peninsula Project. Tech. Rep., Pacific Northwest National Laboratory (PNNL), Richland, Washington, 2007.
- [93] GONZÁLEZ ORDIANO, J. Á.; WACZOWICZ, S.; REISCHL, M.; MIKUT, R.; HAGENMEYER, V.: Photovoltaic Power Forecasting using Simple Data-driven Models without Weather Data. *Computer Science - Research and Development* 32(1-2) (2017), pp. 237–246.
- [94] PEDRO, H. T.; COIMBRA, C. F.: Assessment of Forecasting Techniques for Solar Power Production with No-Exogenous Inputs. *Solar Energy* 86(7) (2012), pp. 2017–2028.
- [95] GONZÁLEZ ORDIANO, J. Á.; DONEIT, W.; WACZOWICZ, S.; GRÖLL, L.; MIKUT, R.; HAGENMEYER, V.: Nearest-Neighbor Based Non-Parametric Probabilistic Forecasting with Applications in Photovoltaic Systems. In: *Proc., 26. Workshop Computational Intelligence, Dortmund, Germany*, pp. 9–30, KIT Scientific Publishing, Karlsruhe, Germany, 2016.
- [96] CHANDOLA, V.; BANERJEE, A.; KUMAR, V.: Anomaly detection: A survey. *ACM Computing Surveys (CSUR)* 41(3) (2009), p. 15.
- [97] WANG, Y.; CHEN, Q.; HONG, T.; KANG, C.: Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges. *IEEE Transactions on Smart Grid* (2018), pp. 1–1.
- [98] AHMAD, A.; HASSAN, M.; ABDULLAH, M.; RAHMAN, H.; HUSSIN, F.; ABDULLAH, H.; SAIDUR, R.: A Review on Applications of ANN and SVM for Building Electrical Energy Consumption Forecasting. *Renewable and Sustainable Energy Reviews* 33 (2014), pp. 102–109.

-
- [99] HAHN, H.; MEYER-NIEBERG, S.; PICKL, S.: Electric Load Forecasting Methods: Tools for Decision Making. *European Journal of Operational Research* 199(3) (2009), pp. 902–907.
- [100] MCSHARRY, P. E.; BOUWMAN, S.; BLOEMHOF, G.: Probabilistic Forecasts of the Magnitude and Timing of Peak Electricity Demand. *IEEE Transactions on Power Systems* 20(2) (2005), pp. 1166–1172.
- [101] GHIASSI, M.; ZIMBRA, D. K.; SAIDANE, H.: Medium Term System Load Forecasting with a Dynamic Artificial Neural Network Model. *Electric Power Systems Research* 76(5) (2006), pp. 302–316.
- [102] ALFARES, H. K.; NAZEERUDDIN, M.: Electric Load Forecasting: Literature Survey and Classification of Methods. *International Journal of Systems Science* 33(1) (2002), pp. 23–34.
- [103] ALMESHAI EI, E.; SOLTAN, H.: A Methodology for Electric Power Load Forecasting. *Alexandria Engineering Journal* 50(2) (2011), pp. 137–144.
- [104] METAXIOTIS, K.; KAGIANNAS, A.; ASKOUNIS, D.; PSARRAS, J.: Artificial Intelligence in Short Term Electric Load Forecasting: A State-of-the-Art Survey for the Researcher. *Energy Conversion and Management* 44(9) (2003), pp. 1525–1534.
- [105] FOLEY, A. M.; LEAHY, P. G.; MARVUGLIA, A.; MCKEOGH, E. J.: Current Methods and Advances in Forecasting of Wind Power Generation. *Renewable Energy* 37(1) (2012), pp. 1–8.
- [106] MONTEIRO, C.; BESSA, R.; MIRANDA, V.; BOTTERUD, A.; WANG, J.; CONZELMANN, G.: Wind Power Forecasting: State of the Art 2009. Tech. Rep., Argonne National Laboratory (ANL), 2009.
- [107] WAN, C.; ZHAO, J.; SONG, Y.; XU, Z.; LIN, J.; HU, Z.: Photovoltaic and Solar Power Forecasting for Smart Grid Energy Management. *Power and Energy Systems* 1(4) (2015), pp. 38–46.

- [108] SOMAN, S. S.; ZAREIPOUR, H.; MALIK, O.; MANDAL, P.: A Review of Wind Power and Wind Speed Forecasting Methods with Different Time Horizons. In: *Proc., North American Power Symposium (NAPS), Arlington, TX, USA*, pp. 1–8, IEEE, Piscataway, NJ, USA, 2010.
- [109] WANG, X.; GUO, P.; HUANG, X.: A Review of Wind Power Forecasting Models. *Energy Procedia* 12 (2011), pp. 770–778.
- [110] HUANG, Y.; LU, J.; LIU, C.; XU, X.; WANG, W.; ZHOU, X.: Comparative Study of Power Forecasting Methods for PV Stations. In: *Proc., 2010 International Conference on Power System Technology (POWERCON), Hangzhou, China*, pp. 1–6, IEEE, Piscataway, NJ, USA, 2010.
- [111] OGLIARI, E.; GRIMACCIA, F.; LEVA, S.; MUSSETTA, M.: Hybrid Predictive Models for Accurate Forecasting in PV Systems. *Energies* 6(4) (2013), pp. 1918–1929.
- [112] KOSTYLEV, V.; PAVLOVSKI, A.: Solar Power Forecasting Performance—Towards Industry Standards. In: *Proc., 1st International Workshop on the Integration of Solar Power into Power Systems, Aarhus, Denmark*, Energynautics GmbH, Langen, Germany, 2011.
- [113] ASARE-BEDIAKO, B.; KLING, W.; RIBEIRO, P.: Future Residential Load Profiles: Scenario-Based Analysis of High Penetration of Heavy Loads and Distributed Generation. *Energy and Buildings* 75 (2014), pp. 228 – 238.
- [114] SUGANTHI, L.; SAMUEL, A. A.: Energy Models for Demand Forecasting - A Review. *Renewable and Sustainable Energy Reviews* 16(2) (2012), pp. 1223–1240.

-
- [115] ESPINOZA, M.; SUYKENS, J. A.; BELMANS, R.; DE MOOR, B.: Electric Load Forecasting. *IEEE Control Systems* 27(5) (2007), pp. 43–57.
- [116] HYNDMAN, R. J.; FAN, S.: Density Forecasting for Long-Term Peak Electricity Demand. *IEEE Transactions on Power Systems* 25(2) (2010), pp. 1142–1153.
- [117] ANTONANZAS, J.; OSORIO, N.; ESCOBAR, R.; URRACA, R.; MARTINEZ-DE PISON, F.; ANTONANZAS-TORRES, F.: Review of Photovoltaic Power Forecasting. *Solar Energy* 136 (2016), pp. 78–111.
- [118] JURSA, R.; ROHRIG, K.: Short-Term Wind Power Forecasting using Evolutionary Algorithms for the Automated Specification of Artificial Intelligence Models. *International Journal of Forecasting* 24(4) (2008), pp. 694–709.
- [119] PELLAND, S.; REMUND, J.; KLEISSL, J.; OOZEKI, T.; DE BRABANDERE, K.: Photovoltaic and Solar Forecasting: State of the Art. *IEA PVPS, Task 14* (2013), pp. 1–36.
- [120] DURAN, M. J.; CROS, D.; RIQUELME, J.: Short-Term Wind Power Forecast Based on ARX Models. *Journal of Energy Engineering* 133(3) (2007), pp. 172–180.
- [121] LANGE, M.; FOCKEN, U.: New Developments in Wind Energy Forecasting. In: *Proc., 2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, Pittsburgh, PA, USA*, pp. 1–8, IEEE, Piscataway, NJ, USA, 2008.
- [122] SHARMA, V.; CALI, U.; HAGENMEYER, V.; MIKUT, R.; GONZÁLEZ ORDIANO, J. Á.: Numerical Weather Prediction Data Free Solar Power Forecasting with Neural Networks. In: *Proc., Ninth*

- International Conference on Future Energy Systems, Karlsruhe, Germany*, pp. 604–609, ACM, New York, NY, USA, 2018.
- [123] CATALÃO, J.; POUSINHO, H.; MENDES, V.: Short-Term Wind Power Forecasting in Portugal by Neural Networks and Wavelet Transform. *Renewable Energy* 36(4) (2011), pp. 1245–1251.
- [124] BACHER, P.; MADSEN, H.; NIELSEN, H. A.; PERERS, B.: Short-Term Heat Load Forecasting for Single Family Houses. *Energy and Buildings* 65 (2013), pp. 101–112.
- [125] BOUZERDOUM, M.; MELLIT, A.; PAVAN, A. M.: A Hybrid Model (SARIMA–SVM) for Short-Term Power Forecasting of a Small-Scale Grid-Connected Photovoltaic Plant. *Solar Energy* 98 (2013), pp. 226–235.
- [126] CHAOUACHI, A.; KAMEL, R. M.; NAGASAKA, K.: Neural Network Ensemble-Based Solar Power Generation Short-Term Forecasting. *JACIII* 14(1) (2010), pp. 69–75.
- [127] DING, Y.; NEUMANN, M. A.; STAMM, E.; BEIGL, M.; INOUE, S.; PAN, X.: A Personalized Load Forecasting Enhanced by Activity Information. In: *Proc., 2015 IEEE First International Smart Cities Conference (ISC2), Guadalajara, Mexico*, pp. 1–6, IEEE, Piscataway, NJ, USA, 2015.
- [128] FAN, S.; HYNDMAN, R.: Short-Term Load Forecasting based on a Semi-Parametric Additive Model. *IEEE Transactions on Power Systems* 27(1) (2012), pp. 134–141.
- [129] KANDIL, M.; EL-DEBEIKY, S. M.; HASANIEN, N.: Long-Term Load Forecasting for Fast Developing Utility Using a Knowledge-Based Expert System. *IEEE Transactions on Power Systems* 17(2) (2002), pp. 491–496.

-
- [130] LI, Y.; SU, Y.; SHU, L.: An ARMAX Model for Forecasting the Power Output of a Grid Connected Photovoltaic System. *Renewable Energy* 66 (2014), pp. 78–89.
- [131] LIN, K.-P.; PAI, P.-F.: Solar Power Output Forecasting using Evolutionary Seasonal Decomposition Least-Square Support Vector Regression. *Journal of Cleaner Production* 134 (2016), pp. 456–462.
- [132] LIU, H.; LIU, D.; LIANG, Y.-M.; ZHENG, G.: Research on Natural Gas Load Forecasting based on Least Squares Support Vector Machine. In: *Proc., 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, Shanghai, China, Vol. 5, pp. 3124–3128, IEEE, Piscataway, NJ, USA, 2004.
- [133] MELLIT, A.; PAVAN, A. M.; LUGHI, V.: Short-Term Forecasting of Power Production in a Large-Scale Photovoltaic Plant. *Solar Energy* 105 (2014), pp. 401–413.
- [134] SENJYU, T.; TAKARA, H.; UEZATO, K.; FUNABASHI, T.: One-Hour-Ahead Load Forecasting using Neural Network. *IEEE Transactions on Power Systems* 17(1) (2002), pp. 113–118.
- [135] SIDERATOS, G.; HATZIARGYRIOU, N. D.: An Advanced Statistical Method for Wind Power Forecasting. *IEEE Transactions on Power Systems* 22(1) (2007), pp. 258–265.
- [136] TAO, C.; SHANXU, D.; CHANGSONG, C.: Forecasting Power Output for Grid-Connected Photovoltaic Power System without using Solar Radiation Measurement. In: *Proc., 2nd IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, Hefei, China, pp. 773–777, IEEE, Piscataway, NJ, USA, 2010.
- [137] TAYLOR, J. W.; BUIZZA, R.: Neural Network Load Forecasting with Weather Ensemble Predictions. *IEEE Transactions on Power Systems* 17(3) (2002), pp. 626–632.

- [138] YANG, H.-T.; HUANG, C.-M.; HUANG, Y.-C.; PAI, Y.-S.: A Weather-Based Hybrid Method for 1 Day Ahead Hourly Forecasting of PV Power Output. *IEEE Transactions on Sustainable Energy* 5(3) (2014), pp. 917–926.
- [139] BESSA, R. J.; MIRANDA, V.; BOTTERUD, A.; ZHOU, Z.; WANG, J.: Time-Adaptive Quantile-Copula for Wind Power Probabilistic Forecasting. *Renewable Energy* 40(1) (2012), pp. 29–39.
- [140] HABEN, S.; GIASEMIDIS, G.: A Hybrid Model of Kernel Density Estimation and Quantile Regression for GEFCom2014 Probabilistic Load Forecasting. *International Journal of Forecasting* 32(3) (2016), pp. 1017–1022.
- [141] HONG, T.; WILSON, J.; XIE, J.: Long Term Probabilistic Load Forecasting and Normalization with Hourly Information. *IEEE Transactions on Smart Grid* 5(1) (2014), pp. 456–462.
- [142] HUANG, J.; PERRY, M.: A Semi-Empirical Approach using Gradient Boosting and K-Nearest Neighbors Regression for GEFCom2014 Probabilistic Solar Power Forecasting. *International Journal of Forecasting* 32(3) (2016), pp. 1081–1086.
- [143] KOU, P.; LIANG, D.; GAO, F.; GAO, L.: Probabilistic Wind Power Forecasting with Online Model Selection and Warped Gaussian Process. *Energy Conversion and Management* 84 (2014), pp. 649–663.
- [144] LIU, B.; NOWOTARSKI, J.; HONG, T.; WERON, R.: Probabilistic Load Forecasting via Quantile Regression Averaging on Sister Forecasts. *IEEE Transactions on Smart Grid* 8(2) (2017), pp. 730–737.
- [145] NAGY, G. I.; BARTA, G.; KAZI, S.; BORBÉLY, G.; SIMON, G.: GEFCom2014: Probabilistic Solar and Wind Power Forecasting using a Generalized Additive Tree Ensemble Approach. *International Journal of Forecasting* 32(3) (2016), pp. 1087–1093.

- [146] PINSON, P.: Very-Short-Term Probabilistic Forecasting of Wind Power with Generalized Logit–Normal Distributions. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 61(4) (2012), pp. 555–576.
- [147] SANJARI, M. J.; GOOI, H.: Probabilistic Forecast of PV Power Generation Based on Higher Order Markov Chain. *IEEE Transactions on Power Systems* 32(4) (2017), pp. 2942–2952.
- [148] XIE, J.; HONG, T.: GEFCom2014 Probabilistic Electric Load Forecasting: An Integrated Solution with Forecast Combination and Residual Simulation. *International Journal of Forecasting* 32(3) (2016), pp. 1012–1016.
- [149] HONG, T.; FAN, S.: Probabilistic Electric Load Forecasting: A Tutorial Review. *International Journal of Forecasting* 32(3) (2016), pp. 914–938.
- [150] JUNG, J.; BROADWATER, R. P.: Current Status and Future Advances for Wind Speed and Power Forecasting. *Renewable and Sustainable Energy Reviews* 31 (2014), pp. 762–777.
- [151] VAN DER MEER, D.; WIDÁN, J.; MUNKHAMMAR, J.: Review on Probabilistic Forecasting of Photovoltaic Power Production and Electricity Consumption. *Renewable and Sustainable Energy Reviews* 81 (2018), pp. 1484 – 1512.
- [152] KELES, D.; SCELLE, J.; PARASCHIV, F.; FICHTNER, W.: Extended Forecast Methods for Day-Ahead Electricity Spot Prices Applying Artificial Neural Networks. *Applied Energy* 162 (2016), pp. 218–230.
- [153] AGGARWAL, S. K.; SAINI, L. M.; KUMAR, A.: Electricity Price Forecasting in Deregulated Markets: A Review and Evaluation. *International Journal of Electrical Power & Energy Systems* 31(1) (2009), pp. 13–22.

- [154] WERON, R.: Electricity Price Forecasting: A Review of the State-of-the-Art with a Look into the Future. *International Journal of Forecasting* 30(4) (2014), pp. 1030–1081.
- [155] KLAIBER, S.; WACZOWICZ, S.; MIKUT, R.; KONOTOP, I.; WESTERMANN, D.; BRETSCHNEIDER, P.: A Contribution to the Load Forecast of Price Elastic Consumption Behaviour. In: *Proc., 2015 IEEE Eindhoven PowerTech, Eindhoven, Netherlands*, pp. 1–6, IEEE, Piscataway, NJ, USA, 2015.
- [156] KLAIBER, S.; WACZOWICZ, S.; KONOTOP, I.; WESTERMANN, D.; MIKUT, R.; BRETSCHNEIDER, P.: Prognose für preisbeeinflusstes Verbrauchsverhalten. *at-Automatisierungstechnik* 65(3) (2017), pp. 179–188.
- [157] WACZOWICZ, S.; REISCHL, M.; KLAIBER, S.; BRETSCHNEIDER, P.; KONOTOP, I.; WESTERMANN, D.; HAGENMEYER, V.; MIKUT, R.: Virtual Storages as Theoretically Motivated Demand Response Models for Enhanced Smart Grid Operations. *Energy Technology* 4 (2016), pp. 163–176.
- [158] KUMMEROW, A.; KLAIBER, S.; NICOLAI, S.; BRETSCHNEIDER, P.: Recursive Analysis and Forecast of Superimposed Generation and Load Time Series. In: *Proc., International ETG Congress 2015; Die Energiewende-Blueprints for the New Energy Age, Bonn, Germany*, pp. 1–6, VDE, Frankfurt am Main, Germany, 2015.
- [159] CANNON, A. J.: Quantile Regression Neural Networks: Implementation in R and Application to Precipitation Downscaling. *Computers & Geosciences* 37(9) (2011), pp. 1277–1284.
- [160] SHIM, J.; HWANG, C.; SEOK, K.: Support Vector Quantile Regression with Varying Coefficients. *Computational Statistics* 31(3) (2016), pp. 1015–1030.

-
- [161] TAYLOR, J. W.: A Quantile Regression Neural Network Approach to Estimating the Conditional Density of Multiperiod Returns. *Journal of Forecasting* 19(4) (2000), pp. 299–311.
- [162] MÜHLPFORDT, T.; PAULSON, J. A.; BRAATZ, R. D.; FINDEISEN, R.: Output Feedback Model Predictive Control with Probabilistic Uncertainties for Linear Systems. In: *Proc., 2016 American Control Conference (ACC), Boston, MA, USA*, pp. 2035–2040, IEEE, Piscataway, NJ, USA, 2016.
- [163] VRAKOPOULOU, M.; HISKENS, I. A.: Optimal Policy-Based Control of Generation and HVDC Lines in Power Systems under Uncertainty. In: *Proc., 2017 IEEE Manchester PowerTech, Manchester, UK*, pp. 1–6, IEEE, Piscataway, NJ, USA, 2017.
- [164] MIKUT, R.; BARTSCHAT, A.; DONEIT, W.; GONZÁLEZ ORDIANO, J. Á.; SCHOTT, B.; STEGMAIER, J.; WACZOWICZ, S.; REISCHL, M.: The MATLAB Toolbox SciXMiner: User’s Manual and Programmer’s Guide. Tech. Rep., arXiv:1704.03298, 2017.
- [165] GONZÁLEZ ORDIANO, J. Á.; GRÖLL, L.; MIKUT, R.; HAGENMEYER, V.: Probabilistic Energy Forecasting using Quantile Regressions based on a new Nearest Neighbors Quantile Filter. *arXiv preprint arXiv:1903.07390* (2019).
- [166] HYNDMAN, R. J.; FAN, Y.: Sample Quantiles in Statistical Packages. *The American Statistician* 50(4) (1996), pp. 361–365.
- [167] LANGFORD, E.: Quartiles in Elementary Statistics. *Journal of Statistics Education* 14(3) (2006).
- [168] BOWMAN, K.; SHENTON, L.: Estimation: Method of Moments. In: *Encyclopedia of Statistical Sciences*, American Cancer Society, Atlanta, GA, USA, 2006.

- [169] STEPHENS, M. A.: Use of the Kolmogorov-Smirnov, Cramér-Von Mises and Related Statistics without Extensive Tables. *Journal of the Royal Statistical Society. Series B (Methodological)* (1970), pp. 115–122.
- [170] STEPHENS, M. A.: EDF Statistics for Goodness of Fit and Some Comparisons. *Journal of the American Statistical Association* 69(347) (1974), pp. 730–737.
- [171] THAS, O.: *Comparing Distributions*. Springer, New York, NY, USA, 2010.
- [172] KLAIBER, S.: *Prognoseverfahren für beeinflusstes Verbrauchsverhalten in Energiesystemen*. Dissertation, (eingereicht 2019 an der TU Ilmenau), 2019.
- [173] REISCHL, M.; GRÖLL, L.; MIKUT, R.: Optimized Classification of Multiclass Problems Applied to EMG-Control of Hand Prostheses. In: *Proc., 2004 IEEE International Joint Conference on Neural Networks, Budapest, Hungary*, pp. 1473–1478, IEEE, Piscataway, NJ, USA, 2004.
- [174] GNEITING, T.; STANBERRY, L. I.; GRIMIT, E. P.; HELD, L.; JOHNSON, N. A.: Assessing Probabilistic Forecasts of Multivariate Quantities, with an Application to Ensemble Predictions of Surface Winds. *Test* 17(2) (2008), p. 211.
- [175] LIU, J.; BRAUN, E.; DÜPMEIER, C.; KUCKERTZ, P.; RYBERG, D. S.; ROBINIUS, M.; STOLTEN, D.; HAGENMEYER, V.: A Generic and Highly Scalable Framework for the Automation and Execution of Scientific Data Processing and Simulation Workflows. In: *Proc., 2018 IEEE International Conference on Software Architecture (ICSA), Seattle, WA, USA*, pp. 145–155, IEEE, Piscataway, NJ, USA, 2018.

- [176] DÜPMEIER, C.; STUCKY, K.-U.; MIKUT, R.; HAGENMEYER, V.: A Concept for the Control, Monitoring and Visualization Center in Energy Lab 2.0. In: *Proc., 4th D-A-CH Energy Informatics Conference, Karlsruhe, Germany*, pp. 83–94, Springer International Publishing, Cham, Switzerland, 2015.
- [177] GONZÁLEZ ORDIANO, J. Á.; BARTSCHAT, A.; LUDWIG, N.; BRAUN, E.; WACZOWICZ, S.; RENKAMP, N.; PETER, N.; DÜPMEIER, C.; MIKUT, R.; HAGENMEYER, V.: Concept and Benchmark Results for Big Data Energy Forecasting Based on Apache Spark. *Journal of Big Data* 5(1) (2018), p. 11.
- [178] APPINO, R. R.; GONZÁLEZ ORDIANO, J. Á.; MIKUT, R.; FAULWASSER, T.; HAGENMEYER, V.: On the Use of Probabilistic Forecasts in Scheduling of Renewable Energy Sources Coupled to Storages. *Applied Energy* 210 (2018), pp. 1207 – 1218.
- [179] APPINO, R. R.; GONZÁLEZ ORDIANO, J. Á.; MIKUT, R.; HAGENMEYER, V.; FAULWASSER, T.: Storage Scheduling with Stochastic Uncertainties: Feasibility and Cost of Imbalances. In: *Proc., 2018 Power Systems Computation Conference (PSCC), Dublin, Ireland*, pp. 1–7, IEEE, Piscataway, NJ, USA, 2018.
- [180] SCHLEGEL, T.: *Analyse der Strom- und Wärmelastgänge für zukünftige Bedarfsprognosen am Standort Campus Nord des Karlsruher Instituts für Technologie*. Master’s Thesis, Universität Koblenz-Landau, 2016.
- [181] SOSSAN, F.; NAMOR, E.; CHERKAOUI, R.; PAOLONE, M.: Achieving the Dispatchability of Distribution Feeders Through Prosumers Data Driven Forecasting and Model Predictive Control of Electrochemical Storage. *IEEE Transactions on Sustainable Energy* 7(4) (2016), pp. 1762–1777.

- [182] RATNAM, E. L.; WELLER, S. R.; KELLETT, C. M.; MURRAY, A. T.: Residential Load and Rooftop PV Generation: an Australian Distribution Network Dataset. *International Journal of Sustainable Energy* 36(8) (2017), pp. 787–806.
- [183] APPINO, R. R.; MUÑOZ-ORTIZ, M.; GONZÁLEZ ORDIANO, J. Á.; MIKUT, R.; HAGENMEYER, V.; FAULWASSER, T.: Reliable Dispatch of Renewable Generation via Charging of Time-varying PEV Populations. *IEEE Transactions on Power Systems* 34(2) (2019), pp. 1558–1568.