UNIVERSITÄT AUGSBURG

Uncertainty Estimation of Solar Power Forecasts in the Context of Decentralized Energy Systems

Dissertation
zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

an der Fakultät für angewandte Informatik der Universität Augsburg

---

vorgelegt von:	Oliver Dölle
geboren am:	22. 03. 1990 in Halle (Saale)
Anfertigung am Lehrstuhl:	Regelungstechnik Fakultät für Angewandte Informatik
1. Gutachter:	Prof. Dr.-Ing. habil. Christoph Ament
2. Gutachter:	Prof. Dr.-Ing. Lars Mikelsons
3. Gutachter:	Prof. Dr.-Ing. Stefan Niessen, MBA
Tag der mündlichen Prüfung:	22. 11. 2024

Acknowledgements

This dissertation began during my time as a doctoral student at Siemens Technology in Erlangen and has accompanied me beyond that period. Over the course of the past six years, the continuous support and encouragement of my colleagues, friends, family, and mentors have helped me navigate the ups and downs to ultimately finish this thesis. I would like to take this opportunity to express my appreciation to them.

Firstly, I wish to convey my deepest gratitude to my university supervisor, Prof. Christoph Ament. His steadfast support and invaluable feedback have been instrumental in refining the topic of my dissertation and significantly enhancing its quality. Furthermore, I extend my thanks to Prof. Lars Mikelsons and Stefan Niessen for their co-supervision. Their insightful remarks and comments have especially enriched the analysis and method sections.

Moreover, I wish to express my gratitude to all the members of my old research group at Siemens Technology. Thereby, I would like to particularly acknowledge my mentor, Dr. Arvid Amthor, for his guidance and backing, which provided me with the necessary freedom to explore diverse solutions and research challenges. Special thanks also go to the colleagues involved in my research projects Stefan, Thomas S., Thomas B., Lisa, Peter, Martin as well as my previous managers Dr. Jochen Schäfer and Dr. Sebastian Thiem.

I would also like to thank my office colleagues and co-PhD colleagues Sebastian, Lukas, and Thanh. Our brainstorming sessions on the whiteboard, in the beer garden or in our doctoral rounds have always led to open discussions and yielded insightful suggestions. You have been more than just colleagues to me, proving the saying “a sorrow shared is a sorrow halved” true on more than one occasion. Additionally, I had the privilege of supervising various interns, bachelor’s, and master’s students during this time. Thank you, Nico, Alina, Ileskhan, Florian, Syed, Krishna and Albertine for your invaluable contributions, which in some instances extended even beyond the supervision period.

Finally, I would like to thank my family and friends. Above all, my parents, for their love and for nurturing my curiosity, perseverance and critical thinking. Last but most important, thank you Miri, for always supporting and encouraging me, as well as keeping up with me during the sometimes challenging times. As for Matthias, your laughter and the joy you have in discovering a little more of the world every day fill me with incredible happiness and energy.

Abstract

Energy systems worldwide are changing considerably with the ongoing expansion of renewable energy sources. Forecasts are imperative, to ensure efficient operation and trade with external entities in these complex decentralized energy systems. However, their inherent uncertainty can lead to forecasting errors and consequently to suboptimal operational plans and bidding behavior. One potential solution is to not only predict a single value, but to also estimate the existing uncertainty using probabilistic forecasts. In this context, probabilistic forecasts of generated photovoltaics (PV) power are particularly important, as its installed capacity is the fastest growing of all renewable energies.

This thesis focuses on questions that still need to be addressed for a transition of probabilistic PV power forecasts to an applied industrial use in decentralized energy systems. To this end, the work of the author's corresponding publications is extended and supplemented, while establishing an overarching context. Four approaches (e.g., mixture density network (MDN), generalized autoregressive model with conditional heteroscedasticity (GARCH)) that have yielded good results in solar irradiation forecasts or other forecasting fields are adopted and investigated in depth for PV power and compared to established methods.

Additionally, a simulation with 24 different initializations and different amounts of training data is carried out in this thesis. Beforehand, there were no studies regarding the probabilistic prediction quality of PV power forecasts with limited amount of data, although this is indispensable for commissioning in practice. During the generation of the forecasts in this thesis, no manual intervention is applied, as this would also not be feasible in practice. Instead, several regularization methods are used. Furthermore, an automated time decomposition approach is developed for the autoregressive models with exogenous input (ARX), followed by a higher-level greedy search algorithm to determine the model order automatically. To represent the influence of possibly suboptimal model structures, extensions for modeling the epistemic uncertainty are implemented and analyzed for each approach.

The simulations are conducted on the basis of PV power measurements from three sites in Central Europe spanning a period of around two years. The results indicate that even with seven days of training data, nearly all the methods show better forecast accuracies than the reference case of the complete history persistence ensemble. For all uncertainty representation forms the ARX-based probabilistic predictions are outperforming their respective neural network counterparts. Nevertheless, after six months of available days of training data, the behavior reverses and neural network approaches perform better on average. In general, the approaches with a continuous distribution have the best forecasting quality. Hence, the GARCH model in combination with the ARX model is recommended over the entire commissioning period, as it achieves excellent results both with a small (skill score: 31.4 %) and large (skill score: 34.3 %) amount of available training data in comparison. However, when provided with enough data, the MDN model surpasses the other methods in terms of overall forecasting accuracy with an improvement over the benchmark of 39.8 %.

Contents

Nomenclature **vii**

1 Introduction **1**
1.1 Motivation 1
1.2 Objective and structure of this thesis 4

2 Background & analysis **7**
2.1 Fundamentals of time series forecasting 8
2.1.1 Process of time series forecasting 8
2.1.2 Mathematical concept of time series forecasting 9
2.1.3 Bias-variance tradeoff 11
2.2 Physical principle of a PV system 13
2.3 Applications of PV power forecasts in decentralized energy systems . . . 16
2.4 Scope refinement – generating PV power forecasts for decentralized energy systems 19
2.4.1 Forecast type selection 20
2.4.2 Use case specification 21
2.4.3 Algorithm specification 24
2.5 From a deterministic to a probabilistic forecast 30
2.5.1 Sources and types of uncertainty 30
2.5.2 Difference between a prediction and a confidence interval 32
2.5.3 Types of probabilistic forecasts 34
2.6 Selected probabilistic approaches for this thesis 37
2.7 Summary of the specified scope and derived research objectives 38

3 Test and evaluation framework **41**
3.1 Data sets used and preprocessing applied 41
3.2 Simulation setup 43
3.3 Forecast evaluation 44
3.3.1 What constitutes a good forecast? 44
3.3.2 Evaluating forecast quality 45
3.3.3 How does the forecast quality affect potential use cases? 50

4 Methodology **55**
4.1 Underlying forecasting frameworks 56
4.1.1 ARX model 56
4.1.2 MLP approach 62

4.2 Probabilistic representation using ensemble members 68
  4.2.1 Bootstrapping approaches for the ARX model 68
  4.2.2 Monte-Carlo dropout with output calibration 74
4.3 Probabilistic representation using quantiles 79
  4.3.1 Pinball loss function 79
  4.3.2 Quantile regression with the ARX model 80
  4.3.3 Quantile neural network 81
4.4 Probabilistic representation using a continuous probability distribution . . 83
  4.4.1 GARCH model in combination with the ARX model 83
  4.4.2 Mixture density network 86

5 Results 91
5.1 Bootstrapping approaches for the ARX model 92
5.2 Monte-Carlo dropout with output calibration 95
5.3 Quantile regression with the ARX model 98
5.4 Quantile neural network 100
5.5 GARCH model in combination with the ARX model 102
5.6 Mixture density network 104
5.7 Overall Comparison 107

6 Summary and outlook 111
6.1 Summary and conclusion 111
6.2 Outlook 114

Bibliography xv

A Appendix A-1
A.1 List of contributions A-1
A.2 List of supervised students' theses A-2
A.3 Data analysis of remaining sites A-3
A.4 Additional result diagramms A-4
  A.4.1 Bootstrapping approaches for the ARX model A-4
  A.4.2 Monte-Carlo dropout with output calibration A-5
  A.4.3 GARCH model in combination with the ARX model A-6
  A.4.4 Mixture density network A-7

Index xiii

Nomenclature

Remarks on nomenclature and notation

In this section all global variables and the mathematical notation are specified. For better readability, variables are occasionally assigned multiple or different meanings, if confusion can be ruled out by the context of use (e.g., in algorithms scopes). However, in these cases the variables are introduced each time locally directly at the respective scope. Equations may be repeated between chapters to avoid disrupting the reading flow.

If variables have multiple indices, they are separated by commas for a clearer legibility. Furthermore, running indices are displayed in italics and indices with a fixed meaning in roman. Accordingly, the symbol index does not include the variable notations of all the italic indices that occur, but only their base form.

If the dimension of the used vectors and matrices is not significant for the understanding of the equation (e.g., in generic axioms), no additional variable for the dimension is introduced to improve readability. For instance, in some cases $\mathbf{x} \in \mathbb{R}$ is used, although the dimension of $\mathbf{x}$ could be greater than one.

The sets used in this work (e.g., bootstrapped training data set) are multi-sets and may therefore contain duplicate elements.

While citations that refer to an entire paragraph are placed after the punctuation mark, citations within a sentence refer directly to it.

Mathematical notation

Notation	Meaning
x	Scalar variable
$\boldsymbol{x}$	Vector
$\boldsymbol{X}$	Matrix
$\boldsymbol{x}^\mathsf{T}$, $\boldsymbol{X}^\mathsf{T}$	Transpose of a vector or matrix
$\boldsymbol{x} \in \mathbb{R}^N$	Vector with the dimension N, whereby all elements are real numbers
$\overline{\boldsymbol{x}}$	Mean value of $\boldsymbol{x}$
$\hat{x}$	Estimation of x
$\mathring{x}$	Variable of interest for signal x
$\tilde{x}$	Preprocessed form of x (e.g., stationarized, standardized)

(To be continued)

Notation	**Meaning**
x^*	(Bootstrapped) sample of x
$f(x)$	Function f of x
$f(x;\theta)$	Function f of x with fixed parameter θ
$\arg\min_\theta f(x;\theta)$	Value θ for which $f(x)$ is minimal
$\max_i \theta_i$	Maximum value of θ_i for all given i
$\max(x_1,\dots,x_N)$	Maximum value x among $x_1,\dots,x_N$
$x[t]$	Value of the signal x at time t
$x_\upsilon[t]$	Value of the quantile of the probability υ of x at time t
$\mathbb{E}[\boldsymbol{x}]$	Expected value of $\boldsymbol{x}$
$\mathrm{Var}(\boldsymbol{x})$	Variance of $\boldsymbol{x}$; $\mathrm{Var}(\boldsymbol{x}) = \mathbb{E}\Big[(\boldsymbol{x}-\mathbb{E}[\boldsymbol{x}])^2\Big]$
$\mathrm{p}(x)$	Probability density function (PDF) of x
$\mathrm{P}(x)$	Cumulative distribution function (CDF) of x
$\mathrm{P}^{-1}(x)$	Percentile function of x, inverse CDF of x
$\Pr(x)$	Probability of the event x
$\mathrm{p}(y \mid x)$	Conditional probability distribution of y given x
$\mathrm{p}(x)^{\rceil}$	Upper bound of $\mathrm{p}(x)$
$\mathrm{p}(x)_{\rfloor}$	Lower bound of $\mathrm{p}(x)$
$x \sim p$	x is sampled from or distributed according to the distribution p
$\mathcal{N}\big(\mu,\sigma^2\big)$	Normal distribution with mean μ and standard deviation σ
$\{1,\dots,N\}$	A finite multi-set of the elements $1,2,\dots,N$
$\#(\mathcal{S})$	Cardinality of the multi-set $\mathcal{S}$
$\{\boldsymbol{x}_n,\boldsymbol{y}_n\}_{n=1}^N$	Abbreviation for $\{(\boldsymbol{x}_n,\boldsymbol{y}_n) : n \in \{1,2,\dots,N\}\}$
$\cup$	Union of sets
$\sum_{i\in\mathcal{S}}$	Sum over all elements of the set $\mathcal{S}$
$\wedge$	Logical and
$[i]_{i\in\mathcal{S}}^\top$	Abbreviation for a vector where all elements of $\mathcal{S}$ are separated entries
$\exp(x)$	Exponential function, alternative notation for e^x
$\log(x)$	Abbreviation for the natural logarithm $\log_e(x)$
$\propto$	Proportional to
$\approx$	Approximately equal to
$\in$	Is an element of
$\boldsymbol{X} \in \mathbb{R}_{>0}^{D_1\times D_2}$	The dimension of $\boldsymbol{X}$ is $D_1 \times D_2$ and its entries are element of positive $\mathbb{R}$
Δ	Difference
$\lVert\boldsymbol{x}\rVert_2$	Euclidean / ℓ_2 norm; $\lVert\boldsymbol{x}\rVert_2 = \sqrt{\sum_{n=1}^N x_n^2}, \boldsymbol{x} \in \mathbb{R}^N$
$\mathbb{1}(x)$	Heaviside function (unit step function) of x
$[a,b]$	Closed interval $[a,b] = \{x \mid a \le x \le b\}$
$h_j^{(l)}$	For MLPs: variable h refers to the layer l and the perceptron j

Symbols

Symbol	Meaning
$b_j^{(l)}$	Bias at layer l for perceptron j
C	Number of used lead time steps
C_{albedo}	Albedo value
$C_{\text{Max-norm}}$	Hyperparameter denoting the scaling value of the ℓ_2 norm
$C_{\text{per},1}$	Circumsolar brightening coefficient
$C_{\text{per},2}$	Horizon brightening coefficient
C_{ϑ}	Temperature coefficient
D	Number of used lags / time steps
D_{ar}	Number of used autoregressive time lags
D_{GHI}	Number of used lags from the GHI signal
D_{res}	Number of used lags of the model residuals
D_{Tamb}	Number of used lags the ambient temperature
D_{v}	Number of used lags of the conditional variance
$\mathcal{D}$	Data set
$\mathcal{D}_{\text{cal}}$	Data set used for calibration
$\mathcal{D}_{\text{cal excl}}$	Data set used exclusively for calibration
$\mathcal{D}_{\text{poss cal}}$	Data set from which the calibration data is sampled from
$\mathcal{D}_{\text{poss val}}$	Data set from which the validation data is sampled from
$\mathcal{D}_{\text{train}}$	Data set used for training
$\mathcal{D}_{\text{val}}$	Data set used for validation
E	Energy
F	Number of used signals / features
$G_{\text{hor,dir}}$	Direct irradiation on the horizontal plane
$G_{\text{hor,diff}}$	Diffuse irradiation on the horizontal plane
$G_{\text{POA,dir}}$	Direct irradiation on the plane of array
$G_{\text{POA,diff}}$	Diffuse irradiation on the plane of array
$G_{\text{POA,glob}}$	Global (total) irradiation on the plane of array
$G_{\text{POA,ref}}$	Reflected irradiation on the plane of array
h	Denomination the hour of the day
$h_j^{(l)}$	The output of the perceptron at layer l for perceptron j
i	Running index
K	Number components in a mixture model
k	Denomination of a component in a mixture model
$\mathcal{L}$	Loss function
ℓ	Likelihood
ℓ_2	Euclidean norm

(To be continued)

Symbol	Meaning
M	Number of ensemble members
m	Denomination of an ensemble member
N	Number of data points / perceptrons
N_p	Number of parameters
$\mathrm{NCRPS}_\mathrm{forecast}$	NCRPS of the forecast
$\mathrm{NCRPS}_\mathrm{ref}$	NCRPS of the reference forecast / benchmark
n	A specific data point, running index for the number of data points
$\mathbb{N}$	The set of natural numbers, excluding zero
P	Power
$\overline{P}_\text{peak, daily}$	Mean maximum daily produced power of the PV panel
P_PV	PV power
$P_\mathrm{PV,csp}$	PV power under clear sky conditions
$\hat{P}_\text{PV,day ahead}[t]$	Estimated PV power of the day ahead at times t
P_rated	Rated power of the PV panel
$\mathrm{Pr}_\mathrm{drop}$	Dropout probability
Pr_mt	Marginal probability threshold
PL	Pinball loss
$\mathcal{P}$	Population
Q	Number of used time steps
R	Number of residual ensemble members
$\mathbb{R}$	Real number
$S(\cdot)$	Scoring function
$\mathcal{S}$	Set
SS	Skill score
T	Sample Time
t	Time / point in time
$\mathcal{T}_\mathrm{ar}$	Set of used autoregressive time lags
$\mathcal{T}_\mathrm{Tamb}$	Set of used time lags of the ambient temperature
$\mathcal{T}_\mathrm{GHI}$	Set of used time lags of the GHI signal
u	Additional (exogenous) input signal
v	Parameter defining the kurtosis of the skewed-t distribution
$\boldsymbol{X}_\mathrm{f}$	Input matrix to generate the forecast
x	Model or function input signal
x_GHI	GHI signal
x_Tamb	Ambient temperature signal
y	Model or function output signal (e.g., PV power)
y_cal	Model or function output of the calibration data set
y_test	Model or function output of the test data set

(To be continued)

Symbol	**Meaning**
y_{υ}	Model or function output for the quantile υ
Z_{cal}	Z-score of the calibration data set
$z_j^{(l)}$	Dropout variable at layer l for perceptron j
$\mathbb{Z}$	Integer number
α	Specified confidence level for an interval
α_{S}	Azimuth of the sun
α_{PAO}	Azimuth of the plane of array
β_{m}	Exponential decay rate of δ_{m}
β_{s}	Step size
β_{v}	Exponential decay rate of δ_{v}
Γ	Gamma function
γ_{S}	Elevation of the sun
γ_{tilt}	Incident angle of the sunlight on the tilted plane
γ_{PAO}	Elevation of the plane of array
δ_{m}	Estimate of the 1st moment (mean) of the gradient
δ_{v}	Estimate of the 2nd moment (uncentered variance) of the gradient
ε	Model residual
ζ	Auxiliary variable in equations (e.g., for better readability)
$\eta_{\mathrm{DC}\rightarrow\mathrm{AC}}$	Efficiency factor of the inverter
η_{loss}	Efficiency factor due to losses
Θ	Model order
θ	Parameter
θ_{ar}	Parameter of an autoregressive lag
θ_{c}	Parameter representing the intercept
θ_{GHI}	Parameter of the GHI signal
θ_{res}	Parameter of past model residuals
θ_{Tamb}	Parameter of the ambient temperature signal
θ_{v}	Parameter of past conditional variance
$\vartheta_{\mathrm{panel}}$	Temperature of the panel
κ	Clear sky index
λ	Parameter defining the skewness of the skewed-t distribution
μ	Mean value, mean forecast
Ξ	Distribution of a calculated statistic (e.g., mean, median)
ξ_{BP}	Cost per kWh from the backup provider
$\xi_{\mathrm{LEM,ic}}$	Internal price premium or discount for non-compliant bids of the local energy market after consolidation
$\xi_{\mathrm{LEM,s}}$	Benefit per kWh for selling to the local energy market
$\xi_{\mathrm{LEM,s}\rfloor}$	Necessary minimum price per kWh to prevent losses over long periods

(To be continued)

Symbol	Meaning
ξ_{p}	Penalty costs per kWh for non-compliance to a previous bid
ξ_{rel}	Additional costs due to forecast inaccuracies in the reliability
$\xi_{\mathrm{WEM,b}}$	Cost per kWh for buying from the wholesale energy market
$\xi_{\mathrm{WEM,s}}$	Benefit per kWh for selling to the wholesale energy market
ϱ	Patience counter
ϱ_{MAX}	Maximum value of the patience counter
σ	Standard deviation
σ_{cal}	Standard deviation of the calibration data set
σ_{test}	Standard deviation of the test data set
$\sigma\Delta\kappa t$	Solar variability
τ	Time lag denoting the forecast horizon
$\tau_{\mathrm{lag,endo}}$	Lag time of an endogenous signal
$\tau_{\mathrm{lead,endo}}$	Lead time of an endogenous signal
τ_{exo}	Lead or lag time of an exogenous signal
υ	Quantile and percentile
$\phi_j^{(l)}$	Activation function at layer l for perceptron j
φ	Weighting factor of a mixing component
χ	(Random) noise
Ψ	Kernel function
ψ	Lag of an ARX model
$\omega_{i,j}^{(l)}$	Model weight from perceptron i at layer $(l-1)$ to perceptron j at layer l

Acronyms and abbreviations

Abbreviation	Meaning
AR	Autoregressive (model)
ARIMA	Autoregressive integrated moving average (model)
ARMA	Autoregressive moving average (model)
ARMAX	Autoregressive moving average (model) with exogenous input
ARX	Autoregressive (model) with exogenous input
a. u.	Arbitrary unit
CCF	Cross correlation function
CDF	Cumulative distribution function
CH-PeEn	Complete-history persistence ensemble
CRPS	Continuous ranked probability score

(To be continued)

Abbreviation	**Meaning**
CRUDE	Calibrating regression uncertainty distributions empirically
DES	Decentralized energy system
EU	European Union
GARCH	Generalized autoregressive conditional heteroscedasticity
GMM	Gaussian mixture model
GHI	Global horizontal irradiance
LEM	Local energy market
LSTM	Long short-term memory
NN	Neural network
MA	Moving average
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MC	Monte Carlo
MDN	Mixture density network
MLP	Multi-layer perceptron
MPC	Model predictive control
MSE	Mean squared error
NCRPS	Normalized continuous ranked probability score
NWP	Numerical weather predictions
OLS	Ordinary least square
PACF	Partial autocorrelation function
PDF	Probability density function
PIAW	Prediction interval average width
PICP	Prediction interval coverage probability
POA	Plane of array
PV	Photo voltaic
RMSE	Root mean squared error
Ref.	Reference
SARIMA	Seasonal autoregressive integrated moving average model
SS	Skill score
VRE	Variable renewable energy
WEM	Wholesale energy market

"Solar PV's installed power capacity is poised to surpass that of coal by 2027, becoming the largest in the world."

International Energy Agency [109]

1

Introduction

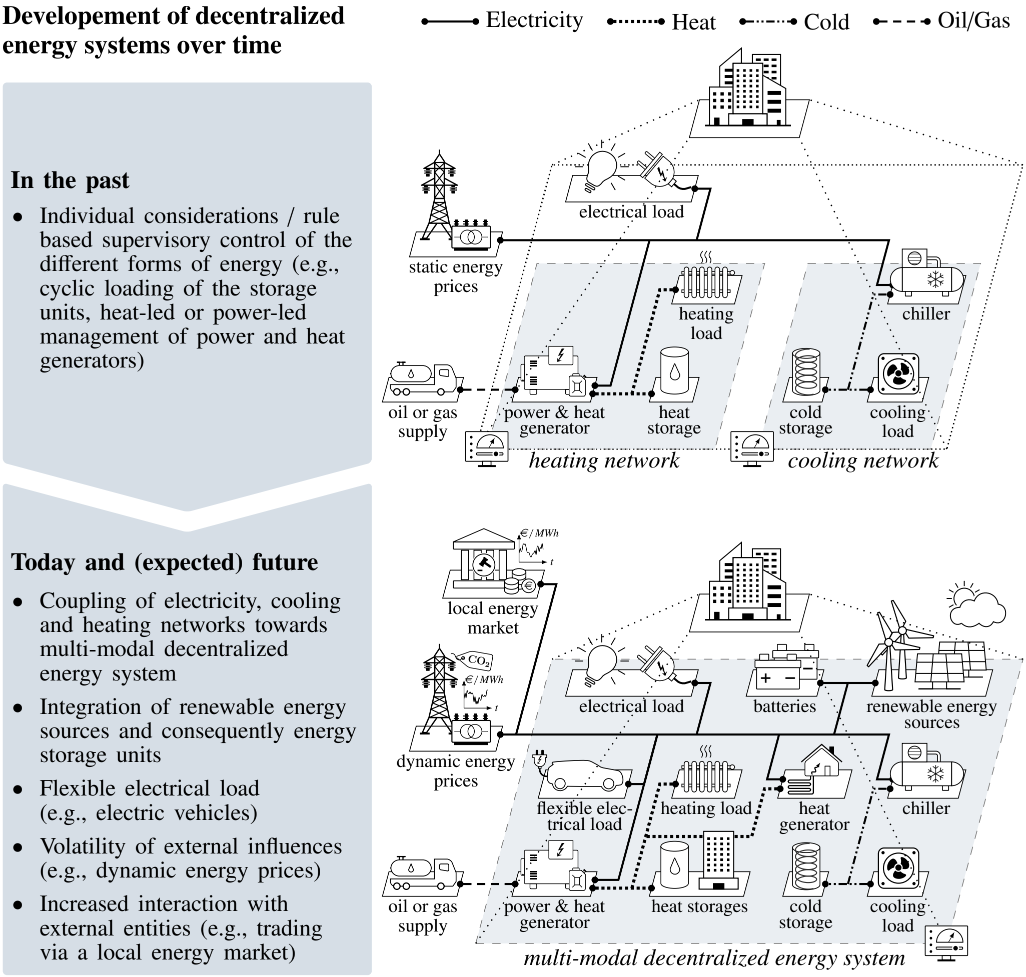
1.1 Motivation

To achieve the global pursuit of carbon neutrality, energy systems worldwide are subject to a rapid transformation. The share of variable renewable energy (VRE) of gross electricity consumption has grown, for instance in Germany, from 10.3 % in 2005 to 46.2 % in 2022 [74]. Moreover, according to the Renewable Energy Sources Act of 2022, it is expected to reach at least 80 % by 2030, which indicates an even faster acceleration of expansion [29]. Globally, simulations by the International Renewable Energy Agency estimate that the VRE share must be around 86 % by 2050 to meet the set two-degree target [181].

However, VREs are dependent on the weather condition and therefore, as the name suggests, highly volatile. They are a considerable burden for the power system, which requires a balance between consumption and generation to maintain stable 50 Hz grid operation. To cope with the volatile generation, regulatory frameworks and economic incentives are being established to stimulate a more grid-friendly consumption behavior. For example, according to EU Directive 2019/944 Article 11, end customers are able to demand time-based electricity prices [70]. Local energy markets, where so-called prosumers trade energy locally with each other, is another concept currently being explored [95].

In addition, as a technical compensation strategy, the power grid is also being expanded and coupled with networks of other forms of energy (e.g., heating and cooling networks) along with the installation of buffer capacities. This leads in combination with additional assets installed for CO_2 reduction to complex multi-modal decentralized energy systems (DESs) (see Figure 1.1).

An efficient coordination of energy conversion, storage, and use in these complex systems, as well as coupling with external entities while considering their possible volatility, is no longer feasible with a traditional heuristic operational control. Instead, higher-level model-based control strategies, which incorporate forecasts of the volatile influences, are necessary for an efficient operation.



Developement of decentralized energy systems over time

In the past

- Individual considerations / rule based supervisory control of the different forms of energy (e.g., cyclic loading of the storage units, heat-led or power-led management of power and heat generators)

Today and (expected) future

- Coupling of electricity, cooling and heating networks towards multi-modal decentralized energy system
- Integration of renewable energy sources and consequently energy storage units
- Flexible electrical load (e.g., electric vehicles)
- Volatility of external influences (e.g., dynamic energy prices)
- Increased interaction with external entities (e.g., trading via a local energy market)

Figure 1.1: Development of decentralized energy systems (e.g., building complexes, airports, production sites) over time. The increasing complexity leads to higher demands on the operational management, as e.g., forecasts of the volatile influences (such as generated photovoltaic (PV) power) have to be created and incorporated.

Photovoltaic (PV) power forecasts, in particular, are receiving increasing attention, after being declared the most immature area of energy forecasting by world-renowned energy forecasters back in 2016 [100]. A particularly strong increase in grid penetration is expected for PV systems, since they have a higher acceptance compared to other VREs [73]. Furthermore, their costs are steadily decreasing[1], and they can be installed decentrally[2] close to the consumers. In 2021, for instance, PV systems accounted for the largest share of investment in renewable energy systems in Germany with more than one-third [11]. This is also necessary, as the installed PV capacity in Germany is set to increase 4.5-fold to 345.4 GW till 2037, based on its current grid expansion plan [30].

However, the inherent uncertainty of forecasts can lead to forecasting errors and consequently suboptimal operation schedules. This is particularly likely to occur at the local level due to the lack of spatial aggregation effects. A potential solution is to estimate the existing uncertainty using probabilistic forecasts. These provide not only a single forecasted value, but also e.g., a probability distribution, which enables superior operation planning[3] [15] and enhanced market bidding strategies [51]. Thereby, the systems exploit that the existing uncertainty is not constant. For instance, a forecast of the expected PV power on a cloudy and windy day is very likely associated with a higher uncertainty than on a cloudless day. In both cases, however, classical deterministic forecasts only estimate the expected arithmetic mean. This neglects valuable information for risk assessment which could be used to reduce ensuing opportunity costs of (conservative) operating strategies. An example of probabilistic forecasting of PV power, together with a comparison of the estimated uncertainty of a day with higher and lower volatilities, is illustrated in Figure 1.2.

Given these advantages, the organizer of the recurring global energy forecasting competition has stated that “the transition from a deterministic to a probabilistic view [...] [is] probably the most important step in the recent history of energy forecasting” [101]. In addition, the authors in Ref. [221] concluded that “probabilistic modeling of solar power and probabilistic power system operations are expected to become the norm in the future”.

However, although quantifying uncertainty is critical for the future, there is comparatively little research in this area, as most PV power forecast research still focuses on the deterministic prediction accuracy [2].

---

[1] The cost per installed kW capacity of solar energy has decreased by about 15% yearly over the last ten years [79], which is significantly more than for wind power [110].

[2] In Germany, for example, wind offshore plants require a significantly larger expansion of the transmission grid infrastructure due to the high concentration of distribution in the north, which in turn is associated with increased costs and time expenditure [73].

[3] A majority of grid operators e.g., are already exploring probabilistic load flow simulations which in turn requires the probabilistic distributions of the solar and load forecasting errors [221].

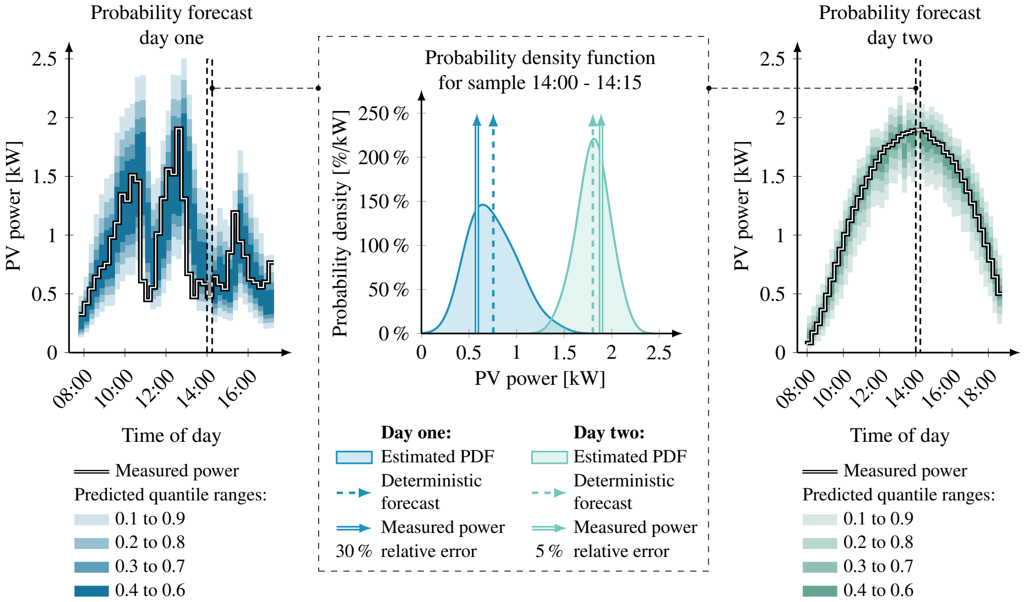


Figure 1.2: Comparison between probabilistic forecast for a day with high and low volatility. The two outer images show the probabilistic forecast for one of the two days, while the centre graph illustrates the respective estimated probability density functions for the period from 14:00 to 14:15. The measured value and the deterministic prediction in the probability density function (PDF) have thereby the form of a Dirac delta distribution with a width close to zero and a height close to infinity. While the deterministic forecast does not provide any information about the uncertainties, the probabilistic forecast clearly exhibits a wider prediction interval with a smaller peak on the more volatile day. Moreover, it is apparent that the relative error of the deterministic forecast is larger for the time step of the day with higher uncertainty.

1.2 Objective and structure of this thesis

Probabilistic PV power forecasts, enable more efficient management of distributed energy systems and lower-risk trading with external markets. Although considerable research has been done in the recent years, there are still some gaps remaining for the transition to an applied industrial use. As the current leading author in the field of solar forecasting notes in Ref. [219]: “solar forecasters in academia tend to overlook the importance of operational solar forecasting”.

The objective of this work is to advance the field of probabilistic PV power forecasting, by focusing on questions that still need to be addressed for a practical use in multi-modal DESs. For this purpose, the structure summarized in Figure 1.3 is used.

First, Chapter 2 starts with an introduction to the necessary fundamentals for time series prediction as well as the physical background of PV power generation. Subsequently, on the basis of possible applications and characteristic qualities of PV forecasts, their practical requirements for on-site energy systems are determined to further refine the scope of this thesis. Following these requirements, the specific scientific gaps considered in this work are defined.

Afterwards, Chapter 3 presents an overview of the analysis framework used in this thesis. Therefore, the data preparation and the creation of individual simulated forecast initializations are presented first, followed by the introduction of methods and metrics for the evaluation of the probabilistic forecasting algorithms. In this context, a distinction is also made between the value and accuracy of a forecast, which should be kept in mind when considering forecast results. Finally, the benefits of increasing the accuracy of probabilistic prediction are discussed with the example use case of local energy markets.

Chapter 4 subsequently introduces the underlying deterministic forecasting methods and elaborates on how they are tailored for a (semi) automated commissioning for PV power forecasting in this thesis. Afterwards, the adapted and newly developed probabilistic methods are explained in detail.

An analysis of the forecasting performance of the different algorithms is provided in Chapter 5. Given the diversity of probabilistic methods used in this work, as well as the first-time

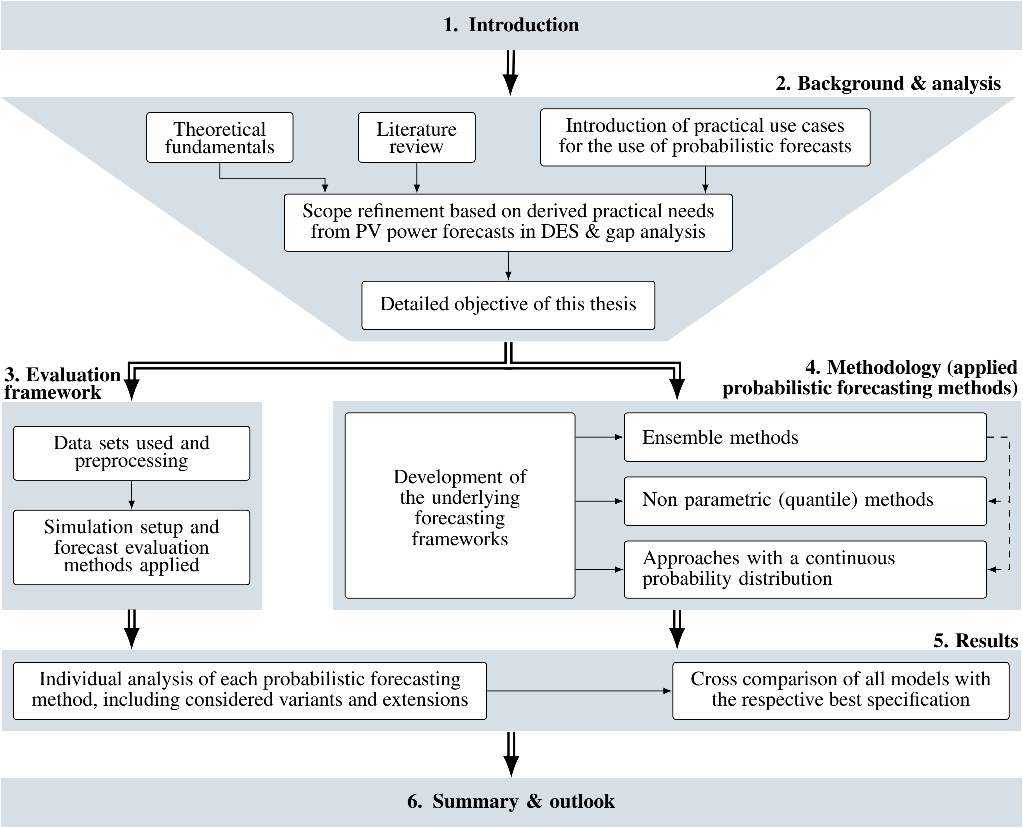

1. Introduction
2. Background & analysis
Theoretical fundamentals
Literature review
Introduction of practical use cases for the use of probabilistic forecasts
Scope refinement based on derived practical needs from PV power forecasts in DES & gap analysis
Detailed objective of this thesis
3. Evaluation framework
Data sets used and preprocessing
Simulation setup and forecast evaluation methods applied
4. Methodology (applied probabilistic forecasting methods)
Development of the underlying forecasting frameworks
Ensemble methods
Non parametric (quantile) methods
Approaches with a continuous probability distribution
5. Results
Individual analysis of each probabilistic forecasting method, including considered variants and extensions
Cross comparison of all models with the respective best specification
6. Summary & outlook

Figure 1.3: Overall structure of this thesis.

application of certain methods for PV power forecasting, they are first analyzed in detail. Then, an overall evaluation with the respective best model specifications for each approach is performed.

Finally, Chapter 6 summarizes the results of this work and provides an overview of remaining questions and possible further research opportunities.

A complete list of publications generated as part of this work, including related research topics, can be found in Appendix A.1.

“Probably the most important step in the recent history of energy forecasting is the transition from a deterministic to a probabilistic view.”

Tao Hong [101]
(Founder of the Global Energy Forecasting Competition and the IEEE working group on energy forecasting)

2

Background & analysis

This chapter analyzes the state of the art of probabilistic PV power forecasting to further focus the scope of this thesis and to identify related research gaps.

Therefore, Section 2.1 first provides a general overview of time series forecasting. Afterwards, Section 2.2 introduces the underlying physical principles of PV systems to provide the domain knowledge necessary for the development of the forecasts. The different use cases for PV power forecasts described in Section 2.3 are then used to derive the specific technical requirements for applications in practice.

Section 2.4 provides an overview of the state of the art of deterministic PV predictions, which are often used as a starting point for probabilistic forecasts. Based on the information and the collected requirements, two deterministic approaches, the first based on a statistical time series model and the second based on a neural network, are selected as underlying forecasting structures for this thesis.

The following Section 2.5 summarizes the state of the art of probabilistic forecasts and outlines how they can be generated. A brief introduction and justification of the selected probabilistic methods for this thesis can be found in Section 2.6.

Finally, Section 2.7 summarizes the identified gaps in the scientific literature and presents the resulting specified scope of this thesis.

2.1 Fundamentals of time series forecasting

2.1.1 Process of time series forecasting

Time series forecasting is build on the assumption that, the future of a time series can be estimated based on knowledge about the past and present by identifying systematic patterns [168]. Generally, the steps shown in the Figure 2.1 are recommended for the creation of a time series forecast [107, 150]. These steps are briefly[1] explained below along with a reference to the respective sections of this thesis in brackets:

- **1. Defining the use case and need** – Depending on the specific application, the detailed requirements (e.g., forecast horizon, sample time, computational resources) must be defined first (Section 2.4.2). Based on these, the necessary data and the methods to be used can often be derived. In addition, it should be established what the user regards as a good or sufficient forecast (Section 3.3.1 and Section 3.3.3).

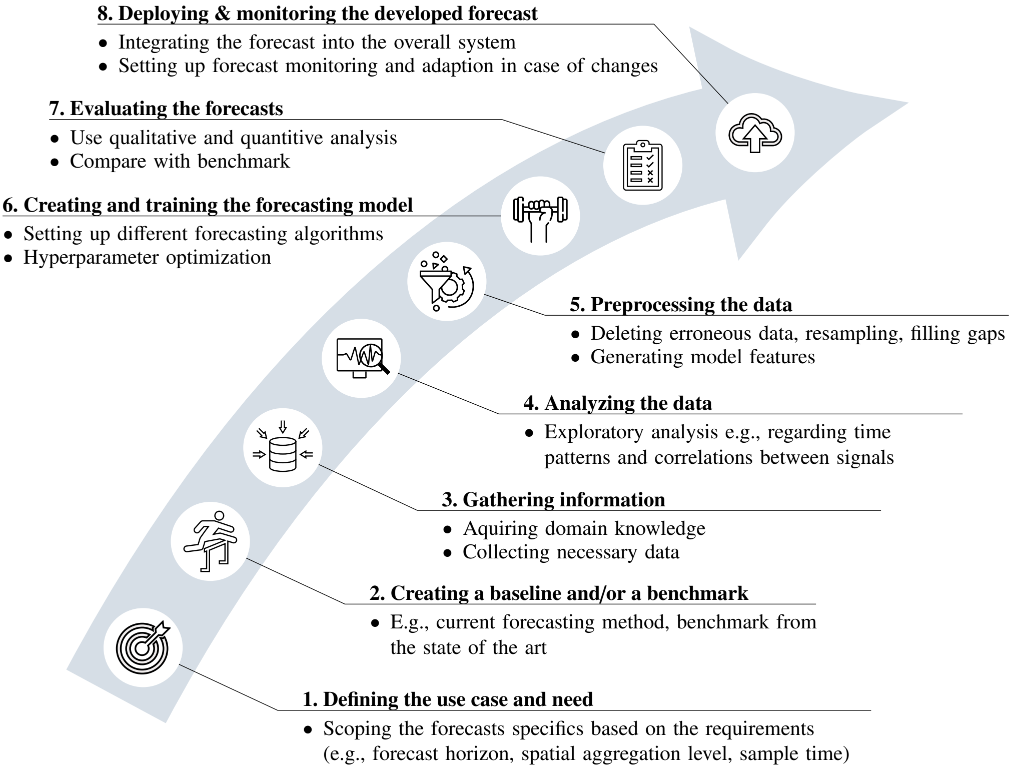

8. Deploying & monitoring the developed forecast
• Integrating the forecast into the overall system
• Setting up forecast monitoring and adaption in case of changes
7. Evaluating the forecasts
• Use qualitative and quantitive analysis
• Compare with benchmark
6. Creating and training the forecasting model
• Setting up different forecasting algorithms
• Hyperparameter optimization
5. Preprocessing the data
• Deleting erroneous data, resampling, filling gaps
• Generating model features
4. Analyzing the data
• Exploratory analysis e.g., regarding time patterns and correlations between signals
3. Gathering information
• Aquiring domain knowledge
• Collecting necessary data
2. Creating a baseline and/or a benchmark
• E.g., current forecasting method, benchmark from the state of the art
1. Defining the use case and need
• Scoping the forecasts specifics based on the requirements (e.g., forecast horizon, spatial aggregation level, sample time)

Figure 2.1: Common steps of a forecasting process.

---

[1] For a deeper introduction into the general theory of time series forecasting, the book [107] by Hyndman and the review paper [168] are recommended.

- **2. Creating a baseline or a benchmark** – For the evaluation of the forecast, in addition to the sole quantification via metrics, a comparison with other methods is often helpful in order to be able to assess the quality of the forecast. For this purpose, it is useful to select a rudimentary forecasting algorithm as benchmark (Algorithm 1 on p. 50) or the currently implemented approach.
- **3. Gathering information** – Subsequently, the relevant data have to be collected, and domain knowledge has to be acquired in order to understand the causal relationships of the underlying system (Section 2.2). The latter enables a more efficient selection of methods and input data for forecasting.
- **4. Analyzing the data** – In this step, time series plots should be created and analyzed, e.g., in terms of trends or cyclic patterns, as these are important for the preprocessing and modeling.
- **5. Preprocessing the data** – For successful forecasting, the data must be cleaned beforehand (e.g., deleting incorrect data, resampling, filling of gaps) (Section 3.1). Furthermore, depending on the algorithm, features may have to be developed, or the time series may have to be stationarized (Section 4.1).
- **6. Creating and training the forecasting model** – In this step, the optimal model structure and the corresponding model parameters are determined usually for several forecasting algorithms (Chapter 4).
- **7. Evaluating the forecasts** – Subsequently, the individual algorithms are compared according to the previously defined quality criteria (Chapter 5).
- **8. Deploying & monitoring the developed forecast** – Finally, the forecast will be deployed on the target system and integrated into the overall concept (e.g., energy management system). In addition, the quality of the forecast should be continuously assessed in order to ensure that it is still accurate during operation. A direct deployment into a productive environment is not part of this work. However, especially the commissioning and the change during the ongoing operation of the forecast is one of the focus points of this thesis.

It should be emphasized that the process is usually very iterative in an academic setting or when forecasts are generated manually. Especially for the determination of the optimal forecasting algorithm and its model structure, the fifth to seventh step are usually repeated with continuous adjustments. At the same time, however, this leads to a time-consuming commissioning process which is often infeasible in industrial applications.

2.1.2 Mathematical concept of time series forecasting

Mathematically, time series forecasting can be viewed as a regression problem whereby one or more dependent variables, in this context the signal to be forecasted, are estimated by means of a set of predictor variables.

Regression problems are based on the following principle: for a given data set $\mathcal{D} = \{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^{N} = (\boldsymbol{X}, \boldsymbol{Y})$ consistent of $N \in \mathbb{Z}_{\geq 0}$ observations pairs of the model output $\boldsymbol{y} \in \mathbb{R}^C$ and the model input[2] $\boldsymbol{x} \in \mathbb{R}^D$, there is an unknown function $\boldsymbol{y} = f(\boldsymbol{x})$,[3] which can be sufficiently estimated by $\hat{f}$ [153]. In the context of time series forecasting the model output is denoted as:

$$\boldsymbol{y} \in \left\{y\left[t + \tau_{\text{lead,endo},c}\right]\right\}_{c=0}^{C} \tag{2.1}$$

with $y[t] \in \mathbb{R}$ being the to be forecasted signal y at time $t \in \mathbb{R}$ and $C \in \mathbb{Z}_{\geq 0}$ defining the number of lead times $\tau_{\text{lead,endo}} \in \mathbb{Z}_{\geq 0}$ in the future a forecast is desired for. The model inputs are either univariate or multivariate. For the former, an autoregressive, also often called endogenous, approach is followed by attempting to predict the time series only with past data of the time series itself, resulting in

$$\boldsymbol{x} \in \left\{y\left[t - \tau_{\text{lag,endo},d}\right]\right\}_{d=1}^{D} \tag{2.2}$$

with $D \in \mathbb{N}$ defining the number of used time steps $\tau_{\text{lag,endo}} \in \mathbb{N}$ in the past. In the multivariate case, an additional number $F \in \mathbb{N}$ of exogenous input signals $u_f \in \mathbb{R}$ often called features are used, leading to

$$\boldsymbol{x} \in \left\{\left\{y\left[t - \tau_{\text{lag,endo},d}\right]\right\}_{d=1}^{D} \cup \left\{u_f\left[t - \tau_{\text{exo},f,q}\right]\right\}_{f=1,q=1}^{F,Q}\right\} \tag{2.3}$$

with $Q \in \mathbb{N}$ defining the used time steps $\tau_{\text{exo},f} \in \mathbb{N}$ of the exogenous signal in the past. Not to be neglected in the formulation of $\mathcal{D}$, is the necessity that the training data set reflects the overall behavior, or at least the expected future operating range behavior, of the system being predicted. Otherwise, even if $\hat{f}$ coincides with f, the forecast accuracy may still be low.

The precise formulation and the estimation process of $\hat{f}$ depend in particular on the used forecasting algorithm. However, the common approach for deterministic forecasts with parametric algorithms is to determine the model parameters $\boldsymbol{\theta} \in \mathbb{R}$ by minimizing a loss function of the model residuals [133]

$$\boldsymbol{\varepsilon} = \boldsymbol{y} - \hat{f}(\boldsymbol{x}; \boldsymbol{\theta}). \tag{2.4}$$

The quadratic loss is thereby usually used as minimization objective, which penalizes large values more. Averaged over all residuals of the data set, this leads to determining the parameters by minimizing the mean squared error (MSE) [133]:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \left(\boldsymbol{y}_n - \hat{f}(\boldsymbol{x}_n; \boldsymbol{\theta})\right)^2. \tag{2.5}$$

---

[2] Often the synonyms features, covariates and predictors are also used for the model input.

[3] For a better understanding, a probabilistic formulation is not used at this point and will rather be introduced later in Section 2.5.

2.1.3 Bias-variance tradeoff

The central objective of the forecast generation is that the estimated model does not only depict the training data set, but more importantly the underlying true behavior of the system. The deviation of the forecast from this ground truth is often called the generalization error. As the ground truth is in practice not known, the generalizing error is instead estimated by applying the forecasting model to a previously not observed data set, referred to as test data. Afterwards a metric e.g., the MSE is used to determine the resulting error terms. [135, 168]

During the generation of the forecast, there are two primary contributing factors that can be optimized in order to keep the generalization error as low as possible. In addition to the previously mentioned representative selection of the training data set, the selection of the model architecture and its expressive capacity[4] is decisive [154]. For a closer look at the influence of the model capacity on the generalization error, it is useful to analyze the error components of the MSE. The MSE of a test set can be decomposed into the three terms, bias, variance, and noise [111, 135]:

$$\begin{aligned}\mathrm{MSE} &= \mathbb{E}(\boldsymbol{Y}-\hat{\boldsymbol{Y}})^2 \\ &= \overbrace{\underbrace{\left(\mathbb{E}[\hat{\boldsymbol{Y}}]-\boldsymbol{Y}\right)^2}_{\text{Bias}^2} + \underbrace{\mathbb{E}\left[(\hat{\boldsymbol{Y}}-\mathbb{E}[\hat{\boldsymbol{Y}}])^2\right]}_{\text{Variance}}}^{\text{Reducible error}} + \overbrace{\underbrace{\mathrm{Var}(\boldsymbol{\varepsilon})}_{\text{Noise}}}^{\text{Irreducible error}}.\end{aligned} \tag{2.6}$$

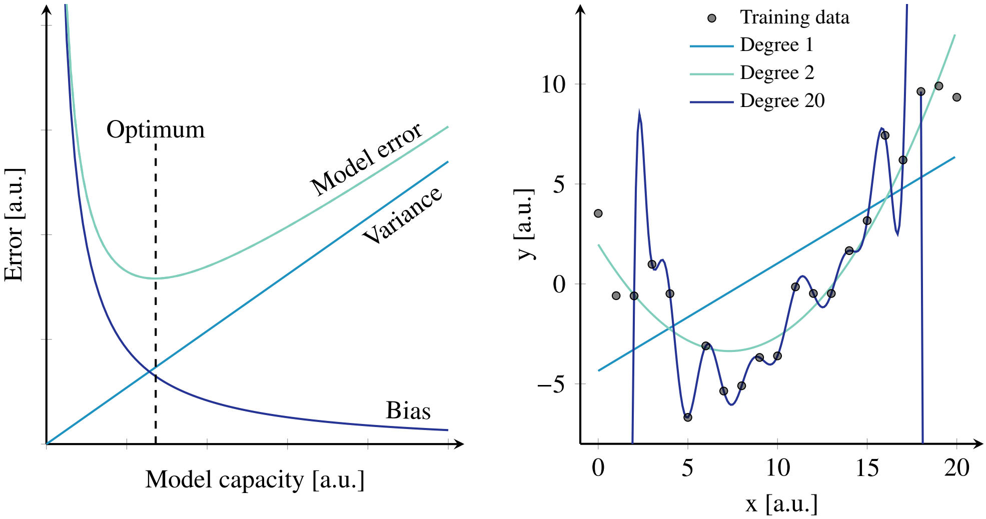
Thereby $\mathrm{Var}(\boldsymbol{\varepsilon})$ represents the irreducible error, caused e.g., by random noise $\varepsilon \in \mathbb{R}$ in the system behaviour. It is the absolute lower error bound one can achieve and can not be minimized or reduced by different model structures [111]. The bias and variance, on the other hand, can be influenced by the model structure. However, their dependence on the model capacity is contradictory which leads to the so-called bias-variance tradeoff.

The bias describes the systematic deviation of the forecast and is caused by a misspecified model, i.e. a model that does not match the true behaviour of the system. This behaviour is also often labeled underfitting in the machine learning context. With increasing model capacity, the systematic bias decreases, as the model is able to represent a larger and more complex variety of mathematical functions. Hence, one objective in the modeling process involves granting the model sufficient flexibility. e.g., the necessary amount of parameters[5] and model structure. However, the declining behavior is not linear, as the influence of additional model capacity on the bias error decreases (see Figure 2.2a).

---

[4] In the literature, the synonyms model expressiveness, flexibility, capacity, and complexity are also frequently used in this context.

[5] The number of parameters is a very significant measure of model complexity, but not the only one. For instance, although $y = \theta_1 \cdot x + \theta_2$ and $y = \theta_1 \cdot e^{\theta_2 \cdot x}$ have the same number of parameters the latter possesses a higher complexity due to the differences in functional form. One approach to quantify complexity is e.g., the geometric model complexity, which quantifies the number of distinguishable probability distributions a model can account for. [112, 154]



(a) Decomposition elements of the MSE vs. model capacity. Adapted and modified from Ref. [89].

(b) Example of over- (degree 20) and underfitting (degree 1). Adapted and modified from Ref. [153].

Figure 2.2: The dependence of the individual error components of the MSE on the model capacity differs.

The variance, on the other hand, characterises the deviation from the ground truth caused by the specific sampling of the observations the model has been trained on [89]. For instance, if two different data sets are used to train a model, the resulting predictions may also differ, even though they are supposed to represent the same data gernerating process. This effect amplifies with increasing model complexity, as the noise and small fluctuations of the training data set are also modeled. This results in the so-called overfitting (see Figure 2.2b) [153]. Thereby the variance error is proportional to the system noise and the ratio of the number of parameters $N_\mathrm{p} \in \mathbb{N}$ to the number of training data $N \in \mathbb{N}$ [190]:

$$\mathbb{E}\left[(\hat{\boldsymbol{Y}} - \mathbb{E}[\hat{\boldsymbol{Y}}])^2\right] \propto \mathrm{Var}(\epsilon)\frac{N_\mathrm{p}}{N}. \tag{2.7}$$

Hence, the variance error can be decreased by reducing the model capacity. Additional, different regularisation methods should be used [153]. Nevertheless, the model capacity should be adjusted so that the sum of the two types of error is minimal (see Figure 2.2a) [89, 153]. In practice, however, an additional aspect must be considered during commissioning. First, the optimal model capacity shifts depending on the available amount of data (see (2.7)). As can be seen in Figure 2.3, the accuracy of the complex function for the test data is similar to the accuracy of the underlying function with a sufficient amount of data. Second, a manual adjustment for each forecast is not always feasible. This challenge will be discussed in more detail in Section 2.4.2, since this aspect is often neglected in the current literature on PV power forecasting.

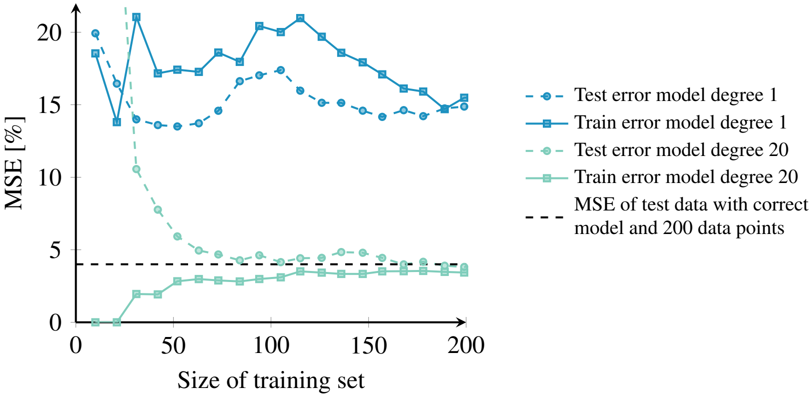


Figure 2.3: Error of estimated models for the test and training data set vs. amount of training data for different model orders. To generate the data sets a model order of degree two was used. Depending on the number of available training data, either the lower order model or the higher order model has a smaller MSE for the test data.
For instance, the accuracy of the complex function for the test data is similar to the accuracy of the underlying function with a sufficient amount of data. Therefore, only the irreducible part of the error due to noise is present at this point. In addition, the number of data points is sufficient to significantly reduce the variance error component. The behavior of the first-order model is exactly the opposite. The MSE does not decrease significantly as the number of training data increases because the variance component is already small. Instead, the bias error component is high due to the insufficient order of the model. Adapted and modified from Ref. [153].

2.2 Physical principle of a PV system

To successfully develop a forecast for a system, it is generally beneficial to build up domain expertise by e.g., comprehending its underlying physical principles. Therefore, the physical modeling chain from the global horizontal irradiance (GHI) to the generated PV power, summarized in Figure 2.4, is elaborated briefly in the following.

The GHI denotes the total amount of hemispheric irradiance received from the sun by a surface horizontal to the ground. It can be either simply measured locally on the ground by a pyranometer or estimated using geostationary weather satellites together with clear sky and cloud models. For the latter, several environmental factors such as aerosols (e.g., dust, salt), water content in the air, and solar geometry are considered [10]. To convert the GHI into the irradiance on the (tilted) plane of array (POA) multiple steps are necessary.

Initially, the GHI is decomposed by a separation model[6] into its elements the direct irradiance $G_{\mathrm{hor,dir}} \in \mathbb{R}_{\geq 0}$ and the diffuse irradiance $G_{\mathrm{hor,diff}} \in \mathbb{R}_{\geq 0}$, because their conversion to the POA differs (see also Figure 2.5a, p. 15) [177]. Subsequently, the conversion for these individual components is performed using transposition models. For the direct irradiance, this is often

---

[6] The interested reader is guided to [217], which provides an overview and comparison of several separation models.

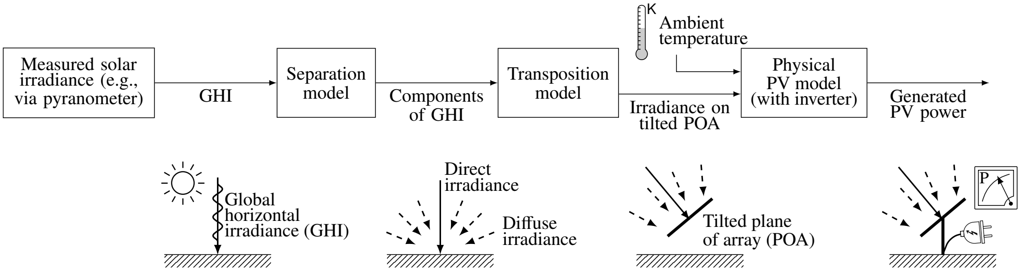


Figure 2.4: Physical modeling chain from GHI to the generated PV power.

done in a straightforward manner based on the geometric orientation between the sun and the POA as follows:

$$G_{\text{POA,dir}} = G_{\text{hor,dir}} \cdot \max\left(0, \frac{\cos \gamma_{\text{tilt}}}{\sin \gamma_{\text{S}}}\right), \tag{2.8a}$$

$$\text{with} \quad \gamma_{\text{tilt}} = \arccos(-\cos \gamma_{\text{S}} \cdot \sin \gamma_{\text{POA}} \cdot \cos(\alpha_{\text{S}} - \alpha_{\text{POA}}) + \sin \gamma_{\text{S}} \cdot \cos \gamma_{\text{POA}}), \tag{2.8b}$$

whereby $\gamma_{\text{S}} \in [0°, 180°]$ and $\alpha_{\text{S}} \in [0°, 360°]$ are the elevation and azimuth of the sun, $\gamma_{\text{POA}} \in [0°, 180°]$ and $\alpha_{\text{POA}} \in [0°, 360°]$ the elevation and azimuth of the POA and $\gamma_{\text{tilt}} \in [0°, 180°]$ the incident angle of the sunlight on the tilted plane (see also Figure 2.5b) [64, 177].

In contrast, the conversion of the diffusive irradiance is rather complex and not yet mature [214]. However, in multiple reviews the Perez translation model outperformed others and is therefore considered state of the art [214]. It is defined as:

$$G_{\text{POA,diff}} = G_{\text{hor,diff}} \left(\frac{(1 + \cos \gamma_{\text{POA}})\big(1 - C_{\text{per,1}}\big)}{2} + \frac{C_{\text{per,1}} \cdot \max(0, \cos \gamma_{\text{tilt}})}{\max(0.087, \sin \gamma_{\text{S}})} + C_{\text{per,2}} \cdot \sin \gamma_{\text{POA}} \right), \tag{2.9}$$

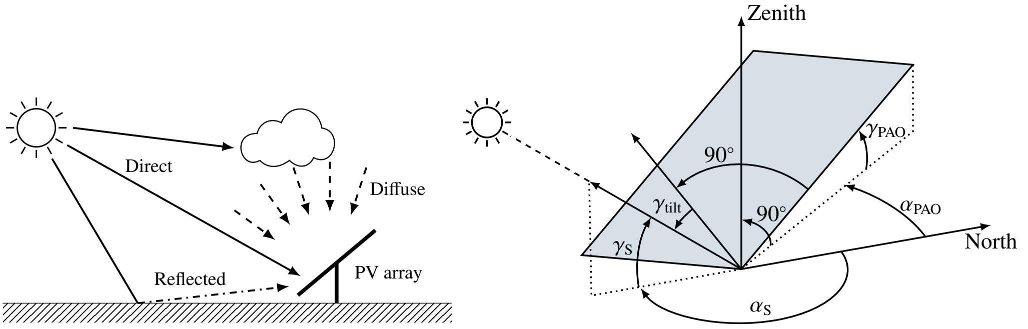
where the circumsolar brightening coefficient $C_{\text{per,1}} \in \mathbb{R}$ and the horizon brightening coefficient $C_{\text{per,2}} \in \mathbb{R}$ are functions depending on the location, solar position and atmospheric clearness class [166, 177]. In addition to the direct and diffuse irradiance, the reflected irradiance from the ground is also part of the POA's global irradiance:

$$G_{\text{POA,ref}} = \frac{\text{GHI} \cdot C_{\text{albedo}} \cdot (1 - \cos \gamma_{\text{POA}})}{2}, \tag{2.10}$$

where the Albedo value $C_{\text{albedo}} \in \mathbb{R}_{\geq 0}$ depends on the reflective characteristics of the ground [177]. Hence, the total irradiance on the PV panel is

$$G_{\text{POA,glob}} = G_{\text{POA,dir}} + G_{\text{POA,diff}} + G_{\text{POA,ref}}. \tag{2.11}$$

The PV panel converts this irradation based on the photoelectric effect into electrical power. As can be seen from the characteristic curve field in Figure 2.6, additional factors such as the



(a) Components of solar irradiance, adopted and modified from Ref. [1].

(b) Geometric orientation between the sun and the POA, adopted and modified from Ref. [176].

Figure 2.5: To estimate the irradiance on POA, the irradiance must be decomposed into its components and afterwards converted using the present trigonometric relations.

cell temperature must be also taken into account for the modeling. A simplified possibility to calculate the output power is:

$$P = P_{\text{rated}} \cdot \frac{G_{\text{POA,glob}}}{1000\text{W}/\text{m}^2} \cdot \eta_{\text{DC}\to\text{AC}} \cdot \eta_{\text{loss}} \cdot \Big(1 + C_\vartheta \cdot \big(\vartheta_{\text{panel}} - 25\text{K}\big)\Big), \tag{2.12a}$$

whereby $P_{\text{rated}} \in \mathbb{R}_{\geq 0}$ is the rated power of the PV panel, $\eta_{\text{loss}} \in \mathbb{R}_{\geq 0}$ the losses of the panel caused by e.g., dirt, shading and aging, $\eta_{\text{DC}\to\text{AC}} \in \mathbb{R}_{\geq 0}$ the inverter loss, $C_\vartheta \in \mathbb{R}_{\leq 0}$ the temperature coefficient and $\vartheta_{\text{panel}} \in \mathbb{R}$ the temperature of the panel [162, 184]. The latter can be e.g., modeled by a polynomial depending on the windspeed, ambient temperature and cell specific parameters [6, 162]. An overview and comparison of multiple physical PV models can be found in Refs. [80, 141, 162].

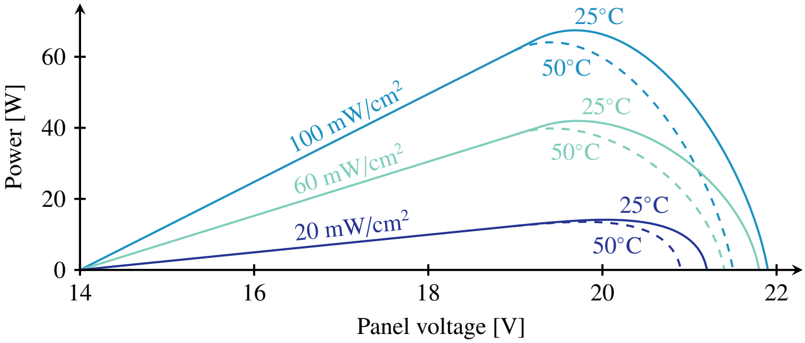


Figure 2.6: Exemplary characteristic power-voltage curve of a solar cell. The colors represent different input light irradiance levels and the line types illustrate the divergence of the characteristic curve at different panel temperatures. Adopted and modified from Ref. [66].

2.3 Applications of PV power forecasts in decentralized energy systems

To manage the energy system at the level of a whole building complex (e.g., aiport, university campus, industrial site) in general, a supervisory control is required, which is used to optimize the set-points of the controllers of individual assets (e.g., PV power inverter, compression chiller, battery) at the process level. For a better understanding of the architecture at hand, Figure 2.7 depicts the automation pyramid known from industrial manufacturing in the context of decentralized energy systems.

In the past, heuristic rule-based controllers were often used for predefined scenarios. These can also be combined with predictions (e.g., of the generated PV power) to better compensate occurring volatility or utilize them in case of electricity prices. However, these so-called "model free" approaches have their limits due to the sheer complexity of the system and the possible use cases. As an alternative, in particular model predictive control (MPC) approaches are being studied and used, as they have already proven themselves in other application areas (e.g., process automation) for the higher-level operational control of base-automated sub-processes [53].

Taking into account forecasts and specified boundary conditions, an MPC determines the optimal dispatch scheduling of the individual assets of the energy system with respect to a specified cost function. For example, a minimization of the absolute energy costs and the CO_2 emission can be the objective (see Figure 2.8).

Figure 2.9 (p. 18) illustrates a selection of three possible business cases for the operation of DESs. These can be implemented or supported with a MPC and are explained briefly in the following:

- **Peak shaving (demand side management)** – From a purchased capacity of 100 MWh per year, a distinction is usually drawn between an energy rate and a capacity charge for

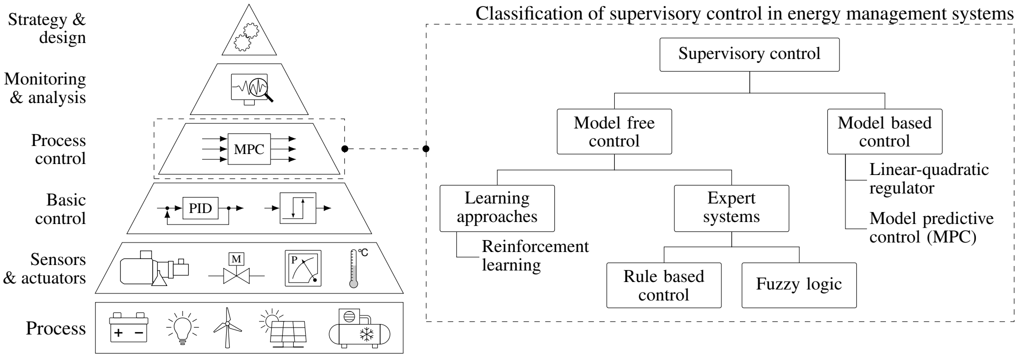
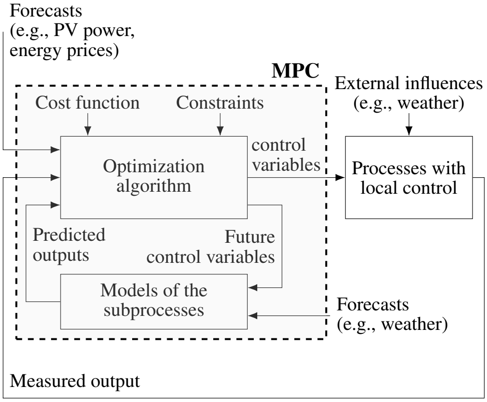


Figure 2.7: Diagram of an automation pyramid and the classification of supervisory control in the context of multi-modal decentralized energy systems.



Concept of an MPC

Step 1: Mathematical models of the underlying subprocesses (e.g., energy conversion and energy storage units) and forecasts are used to calculate future system states/outputs for possible process sequences.

Step 2: With regard to a defined cost function criterion the performance of the different possible process sequences is evaluated for a set prediction horizon.

Step 3: The process sequence (future manipulated inputs) which provides the best performance subject to fixed constraints (e.g., maximum activation times of machines) is selected as the optimal solution.

Step 4: The manipulated variables of the “optimal” solution for the next time step are passed on to the lower local controllers.

Step 5: After a sample time, the system status and the predictions are updated based on current measurements. Subsequently, the process starts again from Step 1.

Figure 2.8: Block diagram and concept of an MPC in the context of multi-modal distributed energy systems. Steps 1-3 are generally carried out by solving an optimization problem.

the purchase of electrical power [28]. This means that not only the total purchased power (energy rate) is considered, but also the highest peak (capacity charge) which occurred during the billing period. Accordingly, demand side management attempts to maintain a constant level of purchased power to minimize the total costs. However, incorrect forecasts and planning mistakes resulting therefrom can lead to an one-time peak that nullifies most economic savings and greatly increases the overall costs.

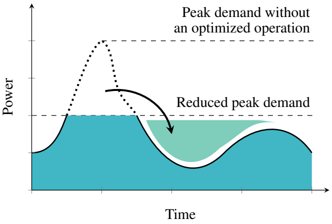
- **Participating in local electricity markets** – A currently studied concept is the trading of DESs at local energy market (LEM), where they can provide surplus power. However, both simulations [185] and practical investigations with real participants in the field [95] have shown that the possible profit depends heavily on the accuracy of the predicted generated energy as well as the consumption. For instance, losses may occur if not enough energy is provided and thus the difference has to be purchased from a backup provider e.g., the wholesale energy market. A more detailed description of the use case is also provided in Section 3.3.3, when the benefits of increasing the accuracy of probabilistic prediction and a necessary accuracy baseline are discussed.
- **Providing flexibility via tertiary control reserve market** – At the tertiary control reserve, short-term negative and positive energy reserves are provided at the request of the transmission system operator in order to compensate for power fluctuations in the grid. Participation in the tertiary control reserve market is possible for DES after a prequalification procedure. However, if there is an actual need and the DES is not able to provide its offered energy, there could be high fines and possibly additional contractual penalties [52].

Incorrect forecasts and resulting inefficient operational management can lead to economic losses in all of these use cases, though this can be particularly high when interacting with external markets. Therefore, in practice, additional buffers are usually included in some form within the planning, if possible [15]. In order to keep the costs of opportunity as low as

Demand side management

- Load shifting to reduce the capacity charged costs

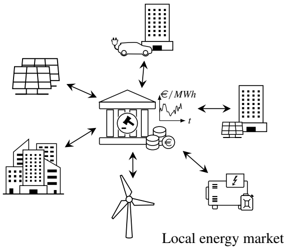
  Examplary prices of Stadtwerke Augsburg (2024):
  - ~ 0.43 €/kWh
  - Per commenced kW: ~ 138 €



Risks:
Single (faulty) incidents can nullify the economical savings, as e.g., the highest peak in the year is used for the capacity charge.

Market participation

- Participation in an electricity market via retailer or via a local energy market

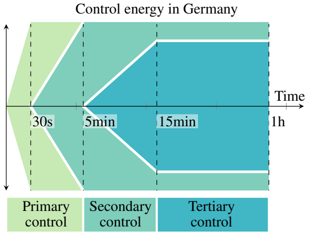


Risks:
Implicit and explicit opportunity costs due to suboptimal bidding behavior. Penalties for non-adherence to bids.

Providing flexibility

- Participation in the market for control energy (minutes reserve) e.g., with a virtual power plant

  Average positive minute reserve prices in 2022 in Germany:
  - In case of demand: 0.81 €/kWh
  - Capacity price for held reserve: 0.0048 €/kWh



Risks:
Fine of 0.1 €/kWh - 1 €/kWh and further contractual penalties.

Figure 2.9: Illustrations of exemplary potential business cases in decentralized energy systems. In particular, the trading with external parties leads to higher profits, but is also associated with higher penalties if the promised energy is not provided. Data Refs.: [31, 52, 194], illustrations adopted and modified from Refs. [12, 32, 95].

possible, probabilistic forecasts can support the risk assessment. Consequently various studies (see e.g. review paper [14] for an overview) have shown, that the use of probabilistic forecasts is more efficient and economically beneficial.

The most straightforward way to incorporate probabilistic forecasts into energy management systems is to use a quantile forecast[7] instead of the deterministic average forecast. This is especially interesting for PV systems, as the economic penalty of the forecast error is not symmetric. Surplus power can generally be dissipated, while a lack of energy cannot be easily compensated. The advantage in this case is that deterministic energy management structure can be kept. However, not all probabilistic information is taken into account which in turn results in opportunity costs. Further integration possibilities for probabilistic predictions are the consideration of different probabilistic scenarios or the integration in stochastic optimizations. The latter considers the entire probabilistic distribution and therefore commonly requires a continuous representation of the cumulative distribution function (CDF) [128].

An overview of implemented use cases as well as methodological approaches for the integration of probabilistic solar forecasting methods can be found in the review paper Ref. [128].

---

[7] An example quantile forecast would be a point forecasts denoting with a predicted 60% probability that at least this amount of power will be generated. In Section 2.5.3 and Section 4.3 this quantile representation form is discussed in detail.

2.4 Scope refinement – generating PV power forecasts for decentralized energy systems

There are several forecasting algorithms which are used for the prediction of PV power. Their selection and specification is significantly influenced by the specific requirements and constraints of the respective use case. For a better overview, the essential characteristics of PV forecasts are summarized in Figure 2.10. In the following, these characteristics are explained in more detail and a short review of the state of the art is given. In addition, the scope is further refined based on the deduced requirements for the practical applications in DES .

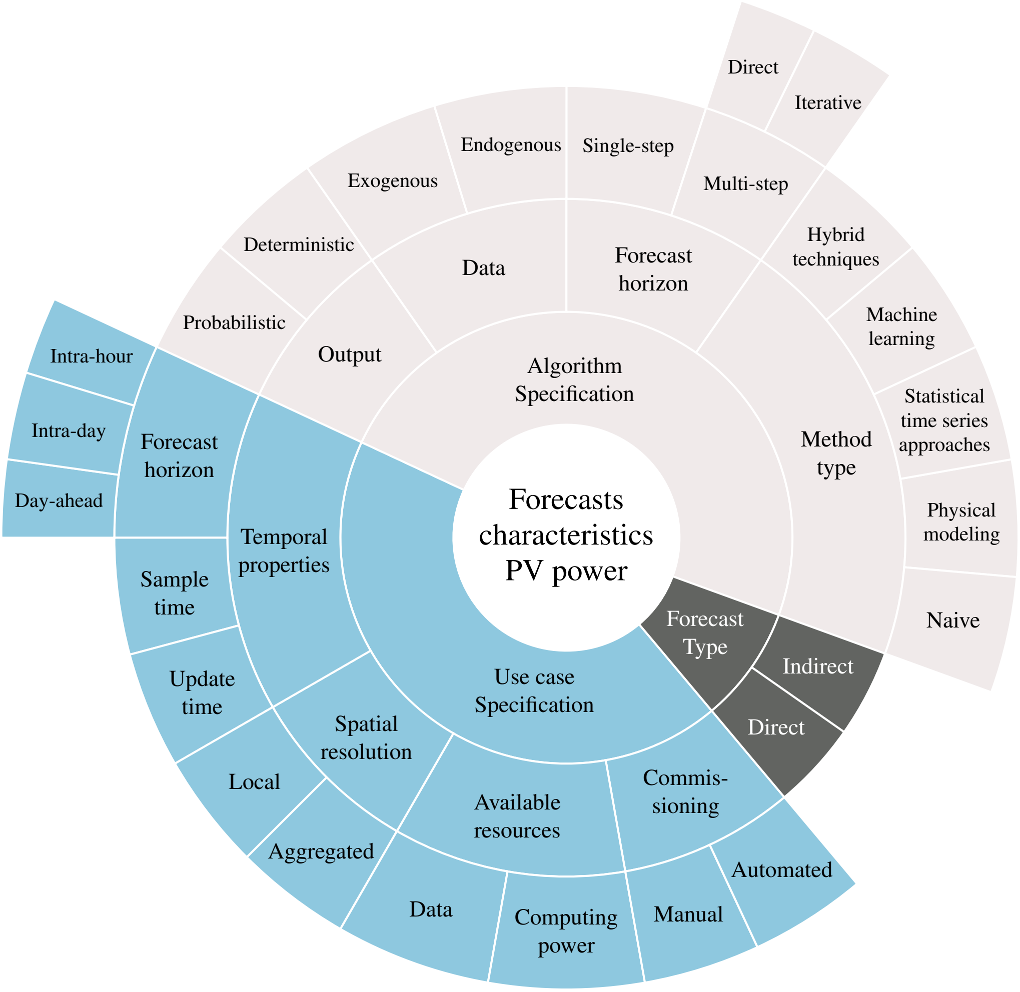


Figure 2.10: Sunburst diagram summarizing a selection of the various characteristics and decisions that need to be considered when developing a PV power forecast.

2.4.1 Forecast type selection

For PV power prediction, a distinction is generally made between two different types or approaches: A direct approach and an indirect one. While in the direct approach, as the name suggests, the PV power is predicted directly with the help of a dynamic model, in the indirect approach the irradiance at the respective location is predicted first. Then, either with the physical modeling chain described in Section 2.2 or with the help of a static black box model, the conversion into PV power is done [220]. In the scientific literature, the indirect approach is predominantly used. To quote the review paper [220, p.13] written by 33 scientists of the solar forecasting community: "most researchers would take [the indirect] two-step procedure to forecast solar power output". Presumably, this is partly related to the historical context of this domain, as many experts come from the field of weather forecasting. Furthermore due to the spatial low pass filtering of widespread free-field PV systems the PV power is easier to predict than the irradiance value at one specific point [220].[8] However, as already mentioned, forecasters in academia occasionally show a tendency of not paying enough consideration to practical operational issues and requirements [219]. Especially for relatively small building installions, the indirect PV power forecasting approach offers several disadvantages, which are discussed in the following:

- **Difficulty to parametrize the physical model chain** – The necessary technical PV parameters for the installation (e.g., orientation, angle) are not always known to the commissioning engineer. For instance, subcontractors are often entrusted with this task, while the documentation is not always well maintained and at hand. This is especially the case for roof top and building installations if differently oriented PV systems are connected to the same power inverter. At the same time, the installation parameters are essential for the conversion of the irradiance to the inclined PV system, as illustrated in Section 2.2.
- **Challenges in the calibration and creation of data driven models** – Measuring devices for local irradiance (e.g., pyranometers) are seldom installed at rooftop systems, which inhibits the calibration and parameter identification of the respective on-site irradiance forecast and the ensuing submodels.
- **Increased model complexity** – Shading aspects occur more frequently with rooftop installation than with open space solar power plants. Consequently they have to be considered and modeled as well.
- **Incorporation of the additional sources of uncertainty** – For a probabilistic representation of the forecast uncertainty, the additional uncertainty caused by the performed conversion would also have to be incorporated. More research is needed for this transformation, as for instance, the first publication on probabilistic transposition models did not occur until 2020 [175].

---

[8] This effect, nevertheless, is only valid for larger PV plants. For smaller systems (e.g., rooftop installations), the spatial aggregation is still comparatively low. As a consequence, the volatility is higher on the inclined plane than on the horizontal plane due to the impact of the installation angle [220].

Given the mentioned drawbacks and the fact that potential users of the probabilistic forecasts are grid operators and plant owners, this thesis focuses on the direct prediction of PV power and its probability distribution. For this purpose, several methods and approaches from the state of the art of solar irradiance forecasting will also be modified in this work to adapt them for the direct forecasting of PV power.

2.4.2 Use case specification

Temporal properties

For the temporal specification of PV power forecasts, three parameters are of particular importance: The forecast horizon, the forecast resolution and the forecast interval. In this, the forecast horizon describes the time span between forecast generation and the forecast value that lies furthest in the future. Accordingly, with increasing forecast horizon, the difficulty of the forecast and consequently the forecast error [156] increases. The PV forecasting community generally distinguishes between three primary categories of forecast horizons [2]:

- **Intra-hour** – This includes forecasts from a few seconds in the future, up to an hour, which are used in particular for peak load management and grid stability (e.g., monitoring for real-time electricity dispatch) [46].
- **Intra-day** – The forecast horizon spans several hours of the day and is primarily used for the control of energy system e.g., with regard to unit commitment and economic dispatch [221]
- **Day-ahead** – This forecast extends over the next day and is especially important for the long term planning of energy systems as well as the participation in external markets [2].

The choice of the forecast horizon always influences the selection of the signals to be considered for the forecast. For very short time horizons, the correlation between proximate time points of PV power dominates, which can be depicted by an autoregressive model approach. At longer horizons (e.g., greater than four hours), however, this influence decreases, while the present physical characteristics and thus the importance of external meteorological input signals and models increases. [180]

This thesis investigates the forcast quality for the intra-day use case with focus on the next six hours, as this time span is needed to also include the scheduling of the thermal side (e.g., thermal storage, heat pumps) in the optimal dispatch calculation of multimodal onsite energy systems [15].

The time resolution describes the sample time of the forecast. With decreasing sample time, the difficulty of the forecast increases, as volatile moments are no longer compensated by the temporal aggregation [156]. A large number of current numerical weather predictions (NWP) models and consequently also of studies on PV power generation consider an hourly resolution [219]. However, since January 1st, 2021 (with some granted derogations until 2024) the harmonized imbalance settlement period and therefore predominantly billing resolution in the European Union is 15 minutes [71]. Consequently, this temporal resolution is also used in this thesis. The last temporal parameter, the forecast update rate, describes the time between two consecutive forecast generations. Analogous to the sample time, 15 minutes is also assumed for this parameter in this thesis.

Spatial resolution

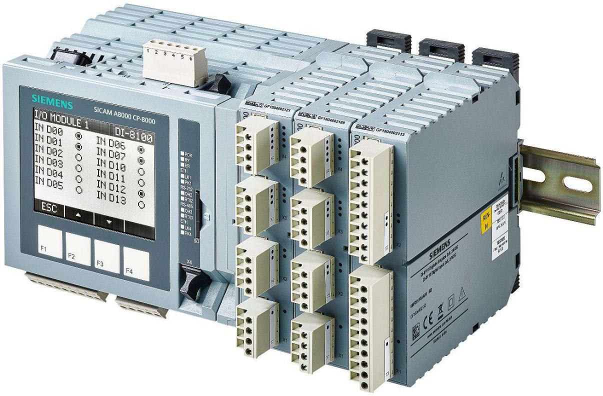
Analogous to the temporal component, the spatial resolution of the forecasts has an influence on the prediction quality, as local deviations can compensate each other by aggregation [156]. Hence, smaller local rooftop systems are used in this work to cover the boundary cases with respect to volatility. This is especially important since according to the EU-wide European Solar Roof Initiative, all new residential, public and commercial buildings will be required to install PV roof systems by 2029, leading to a significant increase of smaller local PV systems [72].

Available computing power

The available computing power for the generation of the forecasts could range from local edge devices to the use of cloud hyperscaler architectures. The latter scales the available resources based on current demand, preventing computing power from being a restricting factor. However, the required cloud infrastructure and communication technologies are comparatively expensive and a redundant local solution is often required in practice anyway to ensure reliable operation (e.g., in the event of network communication issues). In contrast, purely local solutions have the cost advantage that they can be used both for managing the energy system and for generating forecasts. Given their significant restrictions in terms of computing power, local edge devices often do not allow the training of forecasting models that are very complex (see, for example, the technical specification of a SICAM A8000 unit in Figure 2.11). To allow a comparison of the solutions for these two situations with low and high available computing power, both cases are considered in the selection of the prediction algorithms in the next chapter.

Available data

The consideration of the available amount of (limited) data is an important aspect for practical applications, which is often neglected in scientific studies. A majority of publications often use a relatively large amount of data, e.g., more than half a year, when generating and evaluating their forecasts. To the best of the author's knowledge, a comparison with different numbers of training days is made only in [127], where also not less than 60 days of training data were studied. Especially for newly installed plants, however, only limited data is available. Accordingly, model approaches must be chosen that can cope with little data during initialization.



(a) Siemens SICAM A8000 CP-8000 [189]

Processor	Dual-core ARM Cortex-A9 MPCore
Clock speed	800 Mhz
Memory	DDR3 RAM 512 MB

(b) Technical specification [188]

Figure 2.11: Example edge device (a) including its main technical specifications (b). When considering the computing capacity, it should be noted that other applications often run parallel on the device, so that in practice only a part can be used.

One objective of this work is to identify such approaches. For this purpose, the algorithms in this thesis are investigated for both an initialization operation period with little data (7 days) and for a regular operation period (six months of data / 182 days).

In addition to local measurements on site, external weather forecasts should also be included in the forecasting process, as they tend to improve the forecast quality considerably [180]. For this purpose, there are a number of services (e.g., Meteoblue [144], Solcast [191]) that provide the necessary data the day before, e.g., via an application programming interface (API).

Commissioning process

As explained in Section 2.1.3, there is a bias-variance trade off when building forecast models. This dilemma is further amplified by the varying amount of training data during the commissioning process. On one hand, with limited data, the model complexity should not be too high in order to avoid overfitting and consequently poor forecasts. On the other hand, after a longer period of time and thus a sufficient number of training data, a suitable model complexity should be available to ensure forecasting quality as high as possible.

It should also be noted that, contrary to the scientific context, where the model structures and hyperparameter settings are often optimized manually [119], the commissioning effort in practice must be as low as possible and if possible without manual intervention. This is, firstly, due to the effort to keep the costs as low as possible and, secondly, caused by the fact that the average commissioning engineer does not have the necessary expertise to make individual adjustments [160]. For instance, the relatively time-consuming process of modeling and identification (approx. 50% of commissioning time [45]) is often cited as one of the main reasons why model-based control in the context of multimodal distributed energy systems has so far been predominantly applied only in feasibility studies and special solutions [196]. Hence, this thesis investigates (semi) automated forecast generation as well as the feasibility of updating the forecasting algorithms with as little manual intervention as possible throughout the entire commissioning process.

2.4.3 Algorithm specification

General overview of deterministic PV power forecasting methods

There are a variety of forecasting algorithms for PV power, ranging from physical and statistical approaches to more complex neural networks. Figure 2.12 provides an overview of established methods. In the following, they are briefly introduced and references are provided to related solar forecasting studies.

The so-called naïve forecasting methods are, as the name suggests, rudimentary approaches, which are commonly used as benchmarks for other methods. Their forecasting accuracy is often used to quantify the overall predictability of a time series, which can subsequently be incorporated, for instance, in the form of a skill score. The simplest approach is the persistence model, where the last measured value is taken as the prediction [9]. Given a sample time $T \in \mathbb{R}_{>0}$, this results in:

$$y[t] = y[t - T]. \tag{2.13}$$

An benchmark extension of this is the smart persistence [75, 149], where the average over N values of the last days at the same time is taken:

$$y[t] = \frac{1}{N} \sum_{i=1}^{N} y\left[t - i\frac{24\text{h}}{T}\right]. \tag{2.14}$$

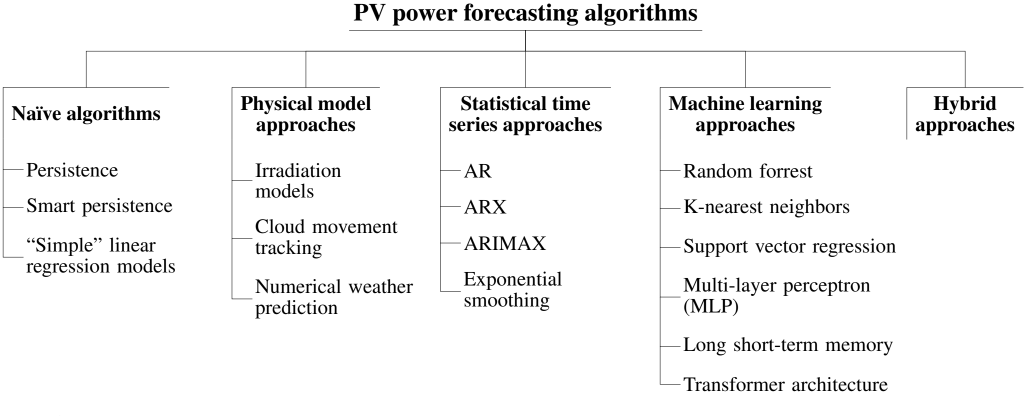

PV power forecasting algorithms
Naïve algorithms
Persistence
Smart persistence
"Simple" linear regression models
Physical model approaches
Irradiation models
Cloud movement tracking
Numerical weather prediction
Statistical time series approaches
AR
ARX
ARIMAX
Exponential smoothing
Machine learning approaches
Random forrest
K-nearest neighbors
Support vector regression
Multi-layer perceptron (MLP)
Long short-term memory
Transformer architecture
Hybrid approaches

Figure 2.12: A classification of deterministic PV power forecasting methods with selected examples.

Other naïve forecasting approaches include simple linear regression model approaches.

The physical approaches use gray or white box models [80, 162] to describe the underlying process behaviour. On the one hand, this can be done by converting irradiance into the generated electrical power as described in Section 2.2. In this case, the actual forecast is performed by NWP , which are mostly based on the physical laws of motion and thermodynamics [108]. The disadvantages of these indirect approaches for practical applications were described in Section 2.4.1. Another physical approach to improve the forecasts quality is to incorporate and model cloud motions. For this purpose, satellite images [69, 197] or local sky imagers [165] (e.g., digital cameras) are used.

The statistical time series approaches include in particular the autoregressive integrated moving average (ARIMA) [179] model family, which was proposed by Box and Jenkins [21] as early as 1970 in their seminar textbook and has since become the most widely used time series forecasting method [218]. The model family comprises a large number of submodels or model components which can be combined with each other. The respective algorithm acronym is then composed of the acronyms of the individual components. One of the basic forms is the autoregressive (AR) [13] model approach where the future is described as a linear combination of lagged values of the same signal. Combining the latter for instance with a moving average (MA) [130] component of the model errors results in ARMA [42, 137, 138, 151] models. The additional linear incorporation of lags of an eXogenous input (X), for instance from NWP, in turn lead to ARX [7, 13] or ARMAX [130] models. Both have shown significantly improved prediction quality compared to their counterparts without additional signals [13, 16, 130, 202]. Other possible extensions are, for example, the differentiation of the time series for stationarization using a so-called integrated (I) part, leading to ARIMA [163] models and the linear consideration of a seasonal (S) component leading to SARIMA [20, 202] models.

Other commonly used statistical time series forecasting methods are exponential smoothing models [62], where exponentially decreasing weights are applied to past observations for the estimation of different decomposed model componensts (e.g., trend) [107].

Machine learning approaches were introduced in the mid-20th century and constitute the majority of new publications by now [218]. Furthermore, they have shown the greatest relative progress in forecast performance in recent years [156], due to methodological improvements and advances in computational power. Nevertheless, several comparative studies have not found superior forecast quality from machine learning models compared to the classical statistical time series forecasting models [48, 156].

The most common machine learning approach is based on artificial neural networks [9]. These include, for example, classic multi-layer perceptrons (MLPs), which have a feedforward structure and estimate their parameters using backpropagation. In addition, classic recurrent neural networks (RNN) and long short-term memory (LSTM) models are also adopted, which possess internal states (memory) and therefore dynamical behaviour due to an internal feedback.

Other machine learning approaches used for PV power forecasts are, for instance, k-Nearest Neighbors [163], where based on e.g., the Euclidian distance of several features, similar past observations are found for the prediction. Another appraoch are support vector regression machines [44, 178], where nonlinear model formulation are achieved by mapping the predictors with kernel functions into a higher-dimensional feature space [9]. In addition, random forest models are also applied, which average the prediction of several regression tree models.

New methods for the general forecasting of time series are continuously being developed and adapted. For example, Neural Basis Expansion Analysis for Time Series (NBEATS) has outperformed previous forecasting competition winners by 3 % [159].

Besides the presented forecasting models, hybrid approaches are often adopted, where methods are used either in series or in parallel (also called model stacking) to foster their individual strengths. For the latter e.g., a weighting based on the forecasting accuracy of the test set is carried out. In many studies, the use of a broad set of forecasting techniques has increased the robustness and, consequently, the forecast quality over a longer period of time [218].

The selection of the algorithm used should be made according to the use case. To quote Hong et. al: "It is very important for researchers and practitioners to understand that a universally best technique simply does not exist" [99].

Comparison difficulties between studies and algorithms

Although there are a large number of publications on forecasting methods of PV power, a comparison between different papers and the methods used in them is almost unfeasible [9, 215, 220]. This can be attributed to the following issues:

- **Different data sets and use cases** – There are a large number of different use cases and thus different temporal and spatial characteristics of PV forecasts and the data sets used. These characteristics in turn have a considerable influence on the predictability of the time series. Depending on the local climatic settings, the weather conditions may be predominantly fluctuating or predominantly steady. Furthermore, a reduced sample rate and a longer forecast horizon complicate the forecast accuracy. As a result, direct comparison of methods from papers with different data sets is difficult [15, 220]. At the same time, both data and used code are rarely shared [215]. Moreover, the prediction quality of each algorithm also depends on the use case, as Ref. [213] concluded after analyzing data sets with seven different climate zones in the United States. In the study, different machine learning approaches were preferable depending on the weather conditions.
- **No standardized error metric** – Unfortunately, no standard metric for quantifying forecasting quality has been established in the scientific literature yet. As a result, a number of different metrics such as root mean squared error (RMSE), mean absolute percentage error (MAPE), mean absolute error (MAE) or Pearson's coefficients are e.g., used for deterministic forecasts. Moreover, the respective normalization of the metrics varies, as papers use the range, the maximum value, the installed capacity, or the mean value of the measured values as demoninator [220]. Thus, without numerical specification of the

normalization quantity – which is often missing – even a comparison between papers with the same data set may be difficult. In addition, none of the metrics mentioned above take into account the variability and uncertainty of the data, which again supports the previous bullet point.

- **Different pre- and postprocessing methods** – The preprocessing method also influences the prediction quality. For example, it can be decisive how data gaps are filled and whether night time values and times with low solar irradiance are also taken into account during training or for the calculation of the metrics. The latter entail smaller errors caused by the small measurements and therefore distort the overall reported error metric value. [9]
- **Conscious or unconscious manipulation by researchers** – A relative improvement in the forecast accuracy by a new proposed method is nearly always expected by journal reviewers and editors. Occasionally, this can lead to a conscious or unconscious manipulation of data sets and results [213]. For instance, test data may be deliberately selected where the methods under investigation show good results while comparatively poor performance benchmarks are chosen [101]. In a cross comparison of papers, the authors in Ref. [156] found, for instance, that studies with smaller test sets also often proclaim smaller forecast errors on average. Furthermore, several papers – also due to the mentioned aspect – shy away from a direct comparison with classically established models or state of the art approaches [101]. As a consequence, many papers show only the superiority of single methods and not its weaknesses. This makes it difficult to draw a generalization, as cross paper comparisons are necessary, which as mentioned above is challenging.

  A common subconscious error is that several models or hyperparameter sets are tested and only the best method is published afterwards [98]. This compromises the "out of sample testing", as the chosen solution is not tested again on a neutral test data set.

  Furthermore, some data are used as input or for preprocessing (e.g., clear sky recorded by satellites [48]) which are not available in real time and therefore distort the prediction quality for practical applications.

Several points can be concluded from the points listed above. On the one hand, the establishment of and compliance with a standard is necessary, especially with respect to preprocessing and evaluation. In a joined publication [220] leading experts in this field have proposed scientific best practices for deterministic predictions and extended them to probabilistic predictions in several responses [143]. These proposed best practices are adopted in this work (see Chapter 3).

On the other hand, the prediction quality is use case dependent and therefore studies with more extensive comparisons are necessary. Accordingly, one of the objectives of this thesis is to compare different approaches in detail and to address both their strengths and weaknesses, for the use case of intraday forecasting for onsite multi-modal DESs.

Selected algorithms for this thesis

Based on the specifications of the use case described in Section 2.4.2 and the derived forecast requirements, two different methodological approaches are studied in more detail in this thesis. First, time series models, explicitly autoregressive exogenous (ARX) models, are analyzed more closely. As the acronym suggests, the forecast is a linear combination of past time steps of the to be forecasted signal and exogenous signals:

$$y[t] = \sum_{i=1}^{D_{\mathrm{ar}}} \theta_{\mathrm{ar},i} \cdot y[t - i \cdot T] + \sum_{f=1}^{F} \sum_{j=0}^{D_f} \theta_{\mathrm{exo},f,j} \cdot x_f[t - j \cdot T], \tag{2.15}$$

where $F \in \mathbb{N}$ defines the number of additional features $x_f \in \mathbb{R}$, $D_f \in \mathbb{N}$ the model order of the respective feature, and $D_{\mathrm{ar}} \in \mathbb{N}$ the autoregressive model order. The parameters of the respective lags are in turn denoted by $\theta_{\mathrm{ar},i} \in \mathbb{R}$ as well as $\theta_{\mathrm{exo},f,i} \in \mathbb{R}$.

Due to their parameter linear mathematical structure, the parameters of autoregressive exogenous models can be estimated using ordinary least square (OLS). This has the advantage that many programming libraries[9] contain this method by default and the parameters can be estimated relatively fast by using for instance the so-called QR-decomposition or cholesky factorization.[10] In combination with the comparatively small number of model parameters, it is feasible to run ARX models on low computational edge devices (e.g., remote terminal units). Furthermore, the model approach has established itself over many years which is why linear time series models are still often used in practical applications [57]. In the comparative study in Ref. [48][11] these models also provided similar probabilistic prediction accuracy to several neural network architectures.

In addition, (deep) MLPs are studied in detail. Their structure is summarized in Figure 2.13. They consist of input and output layer with one or several hidden layers in between, where each layer is fully connected to the next. The layers are in turn composed of a number of perceptrons[12]. Mathematically, the propagation characteristic of a single perceptron can be described as follows [153]:

$$h_j^{(l)} = \phi_j^{(l)} \left(\sum_{i=1}^{N_{l,j}} \omega_{i,j}^{l} \cdot h_i^{(l-1)} + b_j^{(l)} \right), \tag{2.16}$$

with $l \in \mathbb{N}$ denoting the number of the respective layer and $j \in \mathbb{N}$ the number of the perceptron of the layer, $i \in \mathbb{N}$ the number of the perceptron from the previous layer, $h_j^{(l)} \in \mathbb{R}$ the output

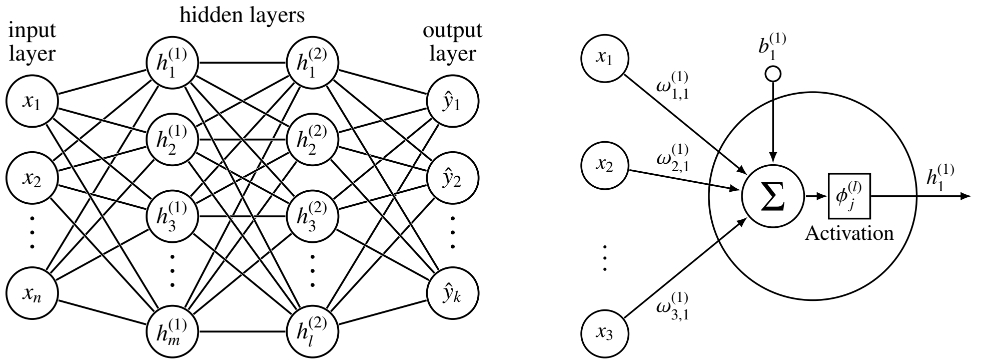
---

[9] For instance, the MATLAB coder supports the transformation of OLS methods into C code, thus providing easy rapid prototyping and programming for remote terminal units [140].

[10] Although the parameters of OLS in the so-called closed form can already be estimated by using matrix multiplication and inversion, in practice the numerically more robust methods mentioned above are commonly used [33].

[11] In this reference solar irradiance forecasts, using only endogenous data with an hourly resolution were studied.

[12] In the hidden layer, the perceptrons are also often called hidden units [153].



(a) MLP structure with two hidden layers, adopted and modified from Ref. [182].

(b) Mathematical characteristic of a perceptron, adopted and modified from Ref. [182].

Figure 2.13: Example MLP structure (a) and detailed depiction of a perceptron (b).

of the perceptron, $\phi_j^{(l)}$ the activation function and $N_{l,j} \in \mathbb{N}$ the overall number of perceptrons in the previous layer. The weights $\omega_{i,j}^{(l)} \in \mathbb{R}$ of the respective perceptron inputs and the bias $b_j^{(l)} \in \mathbb{R}$ are the parameters of the model that are estimated during the training. A nonlinear function is often chosen as activation function, in order to be able to model nonlinear behavior [153]. As evident from Figure 2.13 and 2.16, the MLP initially has a static model structure without memory. Nevertheless, it can be used for modeling time series by temporally shifting the respective associated inputs and outputs, analogous to the ARX model. MLPs have the advantage compared to ARX models that they can depict considerably more complex behavior. For instance, given a sufficient amount of hidden units a shallow MLP with only one hidden layer is a universal function approximator and can therefore theoretically model any function to any desired accuracy level [43, 102]. Nevertheless, deep networks have demonstrated better performance than shallow ones both in theoretical and practical studies in recent years [172]. One reason is their ability to learn better and more detailed abstract relationships between the input data, because each subsequent layer can leverage the generated features of the previous one. However, MLPs generally require significantly higher computing capacity for model training than ARX models, which means that they may not be able to be trained on all remote terminal units. In these cases, cloud solutions or high-performance clients on-site have to be used.

Besides the depictable complexity, MLPs have the advantage that many of the probabilistic extensions listed in the following chapter can be applied to them. Furthermore, preliminary investigations by the author with more complex dynamic neural networks (e.g., LSTM) have not yielded significantly better deterministic predictions. Further information on MLPs can be found, for instance, in Ref. [153].

2.5 From a deterministic to a probabilistic forecast

2.5.1 Sources and types of uncertainty

While deterministic forecasting predicts only a single value $\hat{y}$, probabilistic forecasting estimates the conditional probability $\mathrm{p}(\hat{y} \,|\, \boldsymbol{x}, \mathcal{D})$ at each time point, while considering the used training data set and the respective input features for the current forecast.

To characterize the inherent uncertainty of forecasts, it is beneficial to comprehend their respective causes. According to Hyndman [106] there are generally four major sources affiliated with the uncertainty of time series models:

1. The random noise of the undelaying process which ideally corresponds to the model error term
2. The choice of the model (structure) to replicate the historical process behaviour and subsequently extrapolate it into the future
3. The estimated parameters of the chosen model
4. The assumption that the process being forecasted will behave in a similar way in the future as it did in the past

Applied to the use case of PV power forecasts, the first source can be, for instance, the non-consideration of possible relevant features such as wind speed and wind direction changes in the prediction model or present inaccuracies in the input signals e.g., NWP.

Uncertainties due to the model structure, as mentioned in the second point, are caused by the general choice of the forecasting algorithm and its respective structure (e.g., model order for ARX model and architecture hyperparameters for neural networks). Often, an attempt is made to reduce this uncertainty and the associated forecasting error by data exploration and hyperparameter tuning combined with cross validation during model selection.

Uncertainty in the model parameters described by the third point may result from an insufficient amount of available training data, an (overly) high model complexity, or the inability to determine the global minimum during training due to non-linear models with respect to their parameters (e.g., neural networks). Especially in the case of flexible model structures such as neural networks, the second and third point become sometimes indistinguishable.

The listed fourth cause of uncertainty is almost impossible to quantify and at the same time inherent and unavoidable in forecasting. Nevertheless, by understanding the physics of the process and a subsequent consideration of relevant exogenous variables, this effect is tried to be minimized. For the other three sources of uncertainty, however, there are approaches to model them.

In the literature, the respective sources are generally categorized into two different types of uncertainty. Aleatoric (lat. aleator: dice player; alea: game of chance) describes in this context the general non-modeled randomness of the underlying process and thus both the first and the last cause. Epistemic (greek episteme: knowledge) uncertainty in turn, refers to the existing model uncertainty (second and third point) which occurs due to the lack of knowledge about the perfect predictor. Mathematically, the composition of the conditional probability can be represented as follows [136, 153]:

$$\mathrm{p}(\hat{y}\,|\,\boldsymbol{x},\mathcal{D}) = \int \underbrace{\mathrm{p}(\hat{y}\,|\,\boldsymbol{x},\boldsymbol{\theta})}_{\substack{\text{primarily}\\ \text{aleatoric}}}\,\underbrace{\mathrm{p}(\boldsymbol{\theta}\,|\,\mathcal{D})}_{\substack{\text{primarily}\\ \text{epistemic}}}\,\mathrm{d}\boldsymbol{\theta}, \tag{2.17}$$

with the term $\mathrm{p}(\boldsymbol{\theta}\,|\,\mathcal{D})$ denoting the probability distribution function of the model parameter $\boldsymbol{\theta}$ given the training data set $\mathcal{D}$ and with the term $\mathrm{p}(\hat{y}\,|\,\boldsymbol{x},\boldsymbol{\theta})$ depicting the probability distribution function of the estimated model output $\hat{y}$ given a parameterized model with $\boldsymbol{\theta}$ and the respective input $\boldsymbol{x}$. Consequently, epistemic uncertainty is modeled by placing a probability distribution over the parameter whereas aleatoric uncertainty is modeled by placing a distribution over the model output [114].[13]

The distinction between uncertainty types is important since epistemic uncertainty decreases with more training data, while this has no influence on aleatoric uncertainty [104].[14] Additionally, neglecting one of them will likely lead to an underestimation or misrepresentation of the overall uncertainty. For instance, in practical applications, the models often have a significantly higher epistemic uncertainty, given that instead of individual adjusted hyperparameters, default settings based on intial investigations e.g., from different sites are used. If this higher epistemic uncertainty is not taken into account, it leads to too narrow prediction intervals. Yet, several current studies do not consider a combined assessment of both uncertainties [82]. Accordingly, Hyndman notes that "almost all prediction intervals from time series models are too narrow" [106].

Due to the relevance of the uncertainty types and causes as well as their sometimes difficult distinction at first glance, the concept is demonstrated in Figure 2.14 by a simplified statistical example.

---

[13] This concept can be confusing at first for the interested reader. Of course, a distribution over the model parameters also leads to a distribution at the output. However, this distribution then only depicts the uncertainty caused by the model parameters and therefore the epistemic uncertainty.

A distribution, which is taken only at the output with a single set of model parameters, in turn, considers only the distribution of the model error, which depicts the aleatoric uncertainty. Consequently, slightly different training data could in this case change the model parameters for the forecast and therefore the model error as well as the recorded output distribution. The total uncertainty can thus only be represented by combining both effects.

[14] Although epistemic is often referred to in the literature as reducible and aleatoric as irreducible, this refers only to the situation with the respective information content. For instance, by increasing measurement precision or considering additional information (input signals), some of the seemingly random noise in the process can be reduced. Related to PV Power this could be, for instance, the live consideration of cloud movements in the respective environment with cameras. This also shows that the distinction between the two types of uncertainty cannot always be made precisely in practice and can change due to modifications in the setup or the considered model features. For a more detailed insight into this topic, the interested reader is guided to Ref. [104].

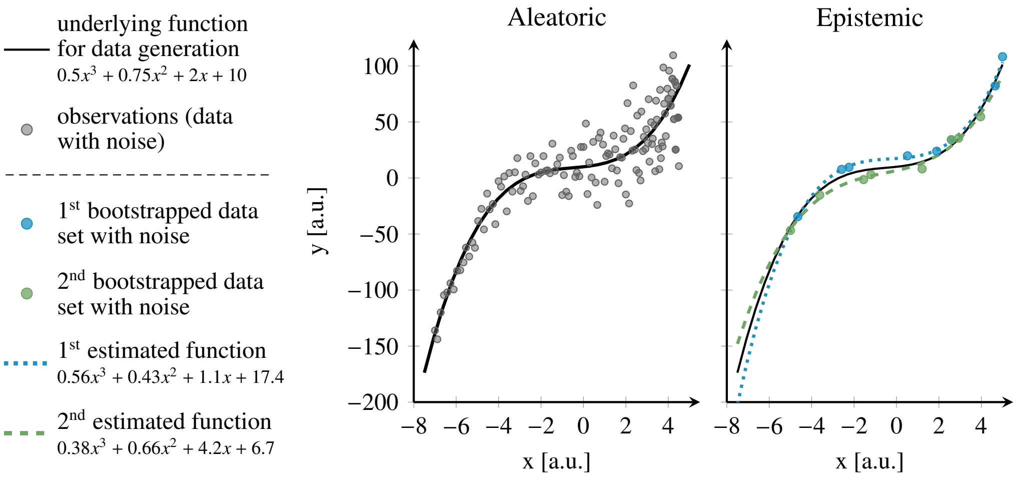


Figure 2.14: Illustration of aleatoric and epistemic uncertainty using a simple example. As indicated in the left diagram, even if the underlying function is precisely estimated, the individual observations cannot be accurately predicted given the present (and not modeled) noise. The noise can thereby also be heteroscedastic and thus not constant over the complete range of values. The diagram on the right shows that, despite having the correct model structure, different functions can be determined depending on the available training data. The deviations between the two estimated functions are mainly caused by epistemic and should diminish with an increasing amount of training data. Moreover, further deviations from the underlying function may occur, if the model structure (e.g., order) of the underlying process is assumed incorrectly.

2.5.2 Difference between a prediction and a confidence interval

Although the confidence interval and the prediction interval describe different concepts, both terms are sometimes mistakenly used as synonyms even in the scientific literature (see e.g, Refs [3, 4, 47, 87, 99, 118, 121, 126, 134, 208, 212]) or in software documentations (see e.g, [94, 199]). However, this inconsistency can lead to confusion and incorrect conclusions. Therefore, their distinction is explained below, in particular with reference to the respective considered types of uncertainty.

Confidence intervals are a frequentist concept, which are often associated with the uncertainty of estimated parameters. Mathematically, an α % confidence interval estimated for a parameter θ based on a given data set $\mathcal{D}$ states that in a study repeated with infinitely sampled data sets, about α % of the different confidence intervals will cover the true value of θ [211].[15] Thus, confidence intervals are based on the expected value or mean of the respective (true) parameter.

---

[15] The commonly used expression that an empirical confidence interval is the range in which the true value lies with x % probability can be used as a rough guide but is strictly speaking inaccurate. From the frequentist point of view, θ is a fixed constant value, so it can either be in an interval or not. [153]

Applied to the model output of, e.g., a linear model

$$
\begin{aligned}
y &= \hat{y} + \varepsilon, \\
&= x \cdot \hat{\theta} + \varepsilon,
\end{aligned}
\tag{2.18}
$$

confidence intervals refer to the so-called mean response. Hence, assuming that the model error is $\varepsilon \sim \mathcal{N}\left(0, \sigma^2\right)$, the width of the confidence interval is proportional to $\sqrt{\mathrm{Var}(\hat{y})} = \sqrt{x^2 \mathrm{Var}(\hat{\theta})}$.

In practice, however, one often does not want to know the mean response, but rather the uncertainty associated with an individual forecast. The so-called prediction interval also takes into account the variability of the individual observation and therefore the model error. Applied to the linear model example, the latter has a width proportional to $\sqrt{\mathrm{Var}(\hat{y} + \varepsilon)} = \sqrt{x^2 \mathrm{Var}(\hat{\theta}) + \sigma^2}$.

To connect the two concepts, one could also simplistically say that if predictions are estimated for infinitely many different samples of the input x, their distribution should lie within the prediction interval while their mean should lie within the respective confidence interval, both according to the associated probability. Confidence intervals therefore account only for the epistemic uncertainty of the model parameters and not for the aleatoric uncertainty of the underlying process. As a consequence, confidence intervals are usually significantly narrower than prediction intervals (see e.g., Figure 2.15).

In this thesis, the prediction interval is examined, as it is more important for the described practical use cases.

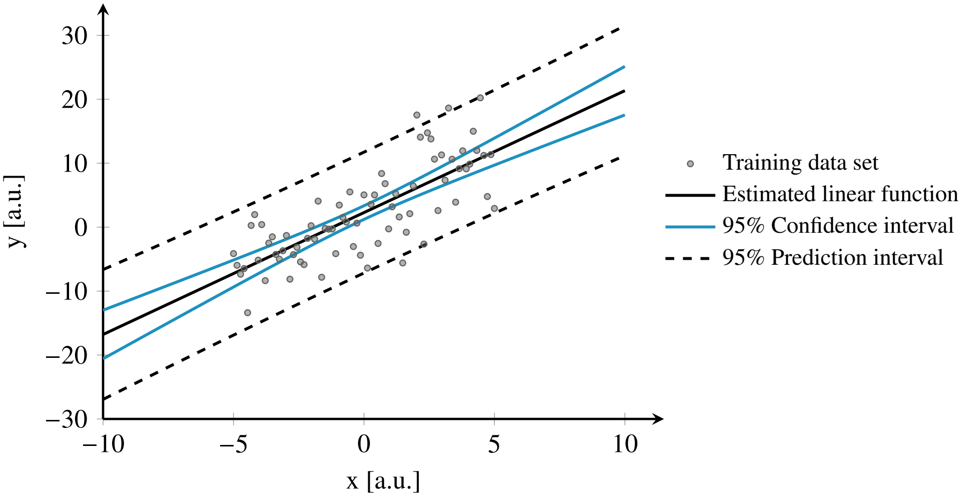


Figure 2.15: Width of a 95 % confidence and 95 % prediction interval exemplarily illustrated. The data set was generated with the underlying linear function $y = 1.75x + 3$.

2.5.3 Types of probabilistic forecasts

Three different basic concepts (see Figure 2.16) are widely used for the representation and generation of probabilistic forecasts: (1) creation of ensemble forecasts, (2) identifying a discrete cumulative distribution function by e.g., quantiles, or (3) determining a continuous probability function via a parametric distribution or non parametric depiction (e.g., kernels). In the following, these different approaches are discussed in detail referencing several state of the art examples.

Ensemble approaches

Ensemble forecasts consist of different ensemble members – typically point forecasts – which are generated by e.g., bootstrapping [90], multi model approaches [22, 228], scenario analysis of model inputs [198], determination of possible input deviations [192] or by selection outputs from comparable situations in the past [3]. They can therefore assume any occurring distribution function. However, ensembles may require a postprocessing for the calibration of the prediction interval and are often in comparison more computational demanding [192]. Furthermore, depending on the used ensemble generation technique, they commonly only model one type of uncertainty. Training data bootstrapping, for instance, only depicts epistemic uncertainty, as one basically estimates a distribution of model parameters [57]. Bootstrapping

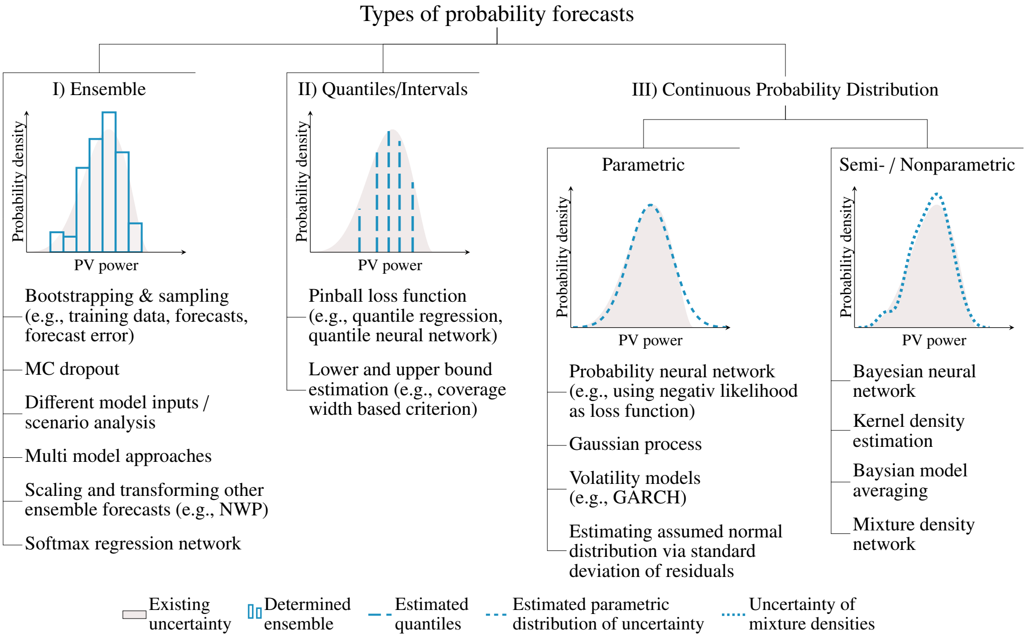

Types of probability forecasts
I) Ensemble
Probability density
PV power
Bootstrapping & sampling (e.g., training data, forecasts, forecast error)
MC dropout
Different model inputs / scenario analysis
Multi model approaches
Scaling and transforming other ensemble forecasts (e.g., NWP)
Softmax regression network
II) Quantiles/Intervals
Probability density
PV power
Pinball loss function (e.g., quantile regression, quantile neural network)
Lower and upper bound estimation (e.g., coverage width based criterion)
III) Continuous Probability Distribution
Parametric
Probability density
PV power
Probability neural network (e.g., using negativ likelihood as loss function)
Gaussian process
Volatility models (e.g., GARCH)
Estimating assumed normal distribution via standard deviation of residuals
Semi- / Nonparametric
Probability density
PV power
Bayesian neural network
Kernel density estimation
Baysian model averaging
Mixture density network
Existing uncertainty
Determined ensemble
Estimated quantiles
Estimated parametric distribution of uncertainty
Uncertainty of mixture densities

Figure 2.16: Overview of different representation types for probabilistic forecasts and exemplary methods to generate them. While continuous probability distributions can be both parametric and non-parametric, ensembles and quantile representations never assume a parametric distribution. A combination of several approaches as well as the conversion of the different representation forms into each other is also possible.

of the model residuals, in turn, only depict the aleatoric uncertainy, as a distribution is only estimated over the model output.

In addition to their standalone use, ensemble concepts can also be combined with other probabilistic approaches to enable the modeling of all types of uncertainties. In these cases, ensemble members are often used to indirectly depict the distribution of model parameters, while other methods represent the aleatoric uncertainty (e.g., an ensemble of quantile forecasts). This is based on the assumption that the influence of epistemic uncertainty can be compensated by averaging the individual ensemble members, resulting in (2.17) being approximated to [136]:

$$\mathrm{p}(\hat{y}\,|\,\boldsymbol{x}, \mathcal{D}) = \int \mathrm{p}(y\,|\,\boldsymbol{x}, \theta)\,\mathrm{p}(\theta\,|\,\mathcal{D})\,\mathrm{d}\theta \approx \frac{1}{M}\sum_{i=1}^{M} \mathrm{p}\big(\hat{y}\,|\,\boldsymbol{x}, \hat{\boldsymbol{\theta}}_i\big), \qquad \hat{\boldsymbol{\theta}}_i \sim \mathrm{p}(\theta\,|\,\mathcal{D}), \tag{2.19}$$

where $M \in \mathbb{N}$ denotes the number of ensemble members and $\hat{\boldsymbol{\theta}}_i \in \mathbb{R}$ the estimated parameters of the respective ensemble members. This concept is also applied to purely deterministic identifications. For instance, bagging is used to improve the prediction quality in random forests by reducing the model uncertainty when compared to decision trees. In addition to the classical methods of ensemble generation, new approaches have been developed in the field of machine learning, such as Monte Carlo (MC) dropout. There, dropout is not only activated during training, but also during the forward pass of the network. By randomly dropping various units during forecasting, different results are generated. These can even be interpreted overall as a deep Gaussian process approximation when dropout is applied to each hidden layer [84].

Quantile and interval approaches

Estimating the cumulative distribution function (CDF) of probabilistic prediction discretely using e.g., quantiles is the most commonly used approach for probabilistic forecasts [123]. In doing so, for each quantile $\upsilon \in [0; 1]$, a forecast $\hat{y}_\upsilon$ is estimated for the signal y where the probability of $y[t]$ to be lower than $\hat{y}_\upsilon[t]$ is exactly υ:

$$\mathrm{Pr}(y[t] < \hat{y}_\upsilon[t]) = \upsilon\,. \tag{2.20}$$

While in classical deterministic forecasts the parameters are estimated by minimizing the MSE of the residuals, in quantile regression an asymmetric weighted error is used for the loss function e.g., the so-called pinball loss. Hence, this approach can be applied to several algorithms and consequently often used to easily transform an existing deterministic forecast model into a probabilistic one [142]. For example, Ref. [123] and Ref. [57] provide a comparison and overview of different linear prediction models based on quantile regression. The authors in Ref. [65], in turn, applied quantile regression to an encoder-decoder architecture that uses LSTM neural networks in combination with an MLP. However, quantile regression only depicts the aleatoric uncertainty and therefore the uncertainty in the data, as the adjusted loss function only characterizes a distribution over the model output. In addition, for each quantile that one wants to determine, a separate model training is usually performed.

In addition to the determination of quantiles, probabilistic intervals (e.g., an 80 % prediction interval) can also be determined directly. This can be done by adapting the cost function, e.g., using the coverage width-based criterion for the lower and upper bound estimation, which considers both the sharpness and the coverage of the intervals (see also 3.3.2) [115]. Another approach is to estimate conformal intervals based on past prediction errors [195].

Nevertheless, of all the representation forms of probabilistic forecast the quantile and interval approaches have the lowest information content, as only discrete values of the CDF are estimated. As a result, the information content may not be sufficient for some subsequent applications.

Parametric approaches with a continuous probability distribution

Alternatively, the complete probability distribution can also be estimated directly. In the parametric approaches, a set distribution (e.g., Gaussian distribution) is assumed ex ante for the uncertainty and its parameters are subsequently estimated. This can be realized with additive volatility models such as generalized autoregressive conditional heteroscedasticity (GARCH) models, which provide a probabilistic extension for any deterministic prediction while requiring minimal computational effort [47]. Another possibility, implemented in many publicly available R (e.g., tsibble [206]) or Python (e.g., pmdarima [199]) forecasting packages thanks to its simplicity, is the estimation of a normal distribution using the standard deviation of the training residuals. Yet, this approach assumes homoscedasticity and only considers the deviation of the random error term and therefore aleatoric uncertainty. This can lead to an common underestimation of prediction intervals of up to 25 % [107]. Analogous to quantile regression, the cost functional of the prediction model can also be adapted in order to apply a parametric approach to different model structures (e.g., distributional neural networks). By using the negative log likelihood of a Gaussian distribution as the cost function, for instance, the mean and the standard deviation can be estimated directly.

All these approaches have in common that the uncertainty of the estimated model should correspond to the predefined distribution. However, previous studies on irradiance forecasts have shown that the assumption of e.g., a fixed Gaussian distribution of the error terms is not always supported by the data due to a lack of symmetry, resulting in inferior forecast quality [48].

Semi-/ and non-parametric approaches with a continuous probability distribution

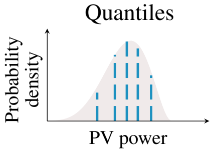
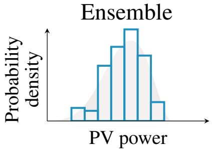
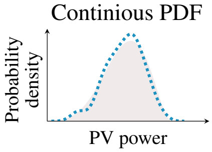
Flexible density estimation, e.g., via kernel functions, can solve this challenge. One of the best-known methods for this is the Gaussian process, which was used, for example, in Ref. [155] to predict PV power probabilistically. Ref. [63] in turn used Bayesian Model Averaging as a postprocessing method to generate a probabilistic mixture model out of NWP ensembles, combining a discrete component for power clipped at the inverter rating and a continuous portion for the lower output. Ref. [132] used a coupled input and forget gate network in combination with quantile regression to initially generate individual quantiles and afterwards converted them into a continuous probability distribution using kernel density estimation.

Another promising approach, which has been successfully applied to other forecasting domains, are mixture density networks (mixture density networks (MDNs)) [229]. They can be seen as extensions of distributional neural networks, as they combine different distributions using the sum of weighted negative log likelihoods of kernel functions as minimizing objective. Consequently, they can estimate flexible uncertainty distributions with almost all neural network structures and thereby benefit from the advances in machine learning, e.g., deep neural networks. Moreover, with a sufficiently high number of Gaussian distributions, it is theoretically possible to represent any other distribution form [17]. Ref. [229] have used it, for instance, for the probabilistic forecast for regional wind power. Furthermore in Ref. [203] an MDN with four distributions was able to achieve a significantly better deterministic forecast than a linear transformation model for the solar irradiance. Analogous to quantile regression, however, MDNs only depict the aleatoric uncertainty with the change of the minimizing objective/cost function.

2.6 Selected probabilistic approaches for this thesis

As the different representation forms of probabilistic forecasts have different advantages, all three concepts will be applied and compared in this thesis for ARX time series models as well as for MLPs (see Table 2.1).

Table 2.1: Overview of the different to be analysed probabilistic appraoches in this thesis.

		Underlying forecasting concept	
		ARX	MLP
Type of representation		• Quantile Regression	• Quantile neural network
		• Training data bootstrapping • Residual bootstrapping • Extended sieve bootstrapping	• MC dropout with model output calibration
		• GARCH model	• Mixture density network (MDN)

For the discrete representation of the CDF, the most commonly used approach of quantile regression will be adopted. The focus is thereby on the question to what extent epistemic uncertainty plays a role for the application case at hand, with limited training data and without manual hyperparameter optimization. Furthermore, it will be investigated to which degree it can be reduced, if necessary. This can be of particular importance, as pure quantile regression depicts only the aleatoric.

As ensemble approaches, training data bootstrapping (epistemic uncertainty), model residual bootstrapping (aleatoric uncertainty) and an extended sieve bootstrapping approach (epistemic and aleatoric uncertainty) will be adapted for the prediction with the ARX model. Since each of the bootstrapping approaches considers different types of uncertainty, this allows to analyze their respective impact better. Ensemble approaches for MLPs usually have the disadvantage that the generation of multiple deterministic predictions requires considerably more computational power caused by the comparatively longer training time. As an alternative, MC dropout is adapted for PV forecasts in this thesis, since it requires only one training session. However, since MC dropout depicts only the epistemic uncertainty, a model output calibration will be also applied.

For the continuous description of the CDF in this thesis, the ARX model will be incorporated with the GARCH model. In comparison to other parametric approaches, which are only based on the standard deviation of the training residuals, the GARCH model has the advantage that it allows the modeling of time-varying volatilities. This is important, because the uncertainty in PV predictions is heteroscedastic [47]. In addition, GARCH approaches require minimal computational effort and are therefore also suited for use on edge devices. For MLPs, MDN will be adopted as an approach for PV power prediction. By combining MLPs with several Gaussian distributions, it is possible to describe non-symmetric uncertainties continuously. Given that both parametric approaches only represent the aleatoric uncertainty, this thesis will also analyze the influence of epistemic extensions and will investigate to what extent these can further improve the prediction quality.

A detailed description of the algorithms and the adaptations made for PV power forecasts can be read in Chapter 4.

2.7 Summary of the specified scope and derived research objectives

The objective of this work is to advance the field of probabilistic PV power prediction by addressing remaining questions for practical application in multi-modal DES. Therefore, first the detailed requirements for the forecasting algorithms were derived in this chapter based on the specified practical applications. Afterwards, they were used to identify the to be resolved gaps in the current state of the art.

In particular, the commissioning process and related challenges have been insufficiently addressed in the scientific literature. This includes both the dealing with comparatively little training data and the commissioning with as little manual effort as possible. A summary of the use case scope, which will be analyzed, and the associated forecast specifications can be found in Table 2.2. The third column contains references to the detailed reasoning for the selection.

To address these gaps and advance the field of probabilistic PV forecasting, the following specific aspects will be explored in this thesis:

- **Simulation and analysis of forecast commissioning and operation under practical conditions** – To the best of the author's knowledge, there are no studies regarding the probabilistic prediction quality of PV power forecasts with limited amount of data. However, as this is indispensable for commissioning in practice, the prediction quality of different methods is investigated in this thesis, both for the initialization operation period with little data (7 days) and also for a regular operation period (182 days of training data). In order to do this, multiple temporal forecast initialization start points are also simulated for each site.

  Furthermore, the optimal combination of training hyperparameters and network structure depends on the underlying data in each case. Accordingly, the optimal choice varies by location, the number of training data, and sometimes the time of year (e.g., weather in spring and fall is more volatile than in summer). Hence, there is arguably no general specification that is truly "optimal" for all circumstances. For this reason, scientific studies often perform extensive manual optimization for their published forecasting algorithms [119]. However, in practical applications this is not possible due to limited capacities.

Table 2.2: Specified focus for the analysis in this work with reference to the segments for the respective explanation.

Parameter	Specifiction/Scope for this thesis		Reference
Forecast type	Direct PV power forecast		2.4.1
Temporal resolution	15 minutes		2.4.2
Spatial resolution	On-site roof top systems		2.4.2
Forecast horizon	6 hours		2.4.2
Amount of available data for training	Two scenarios: • "Start" of commissioning process (7 days) • "End" of commissioning process (182 days)		2.4.2
Commissioning	(semi) automated		2.4.2
Forecasting algorithm	ARX	MLP	2.4.3
Probabilistic extension	• Quantile regression • Residual-, training data and sieve bootstrapping • GARCH	• Quantile regression • MC dropout with output calibration • MDN	2.6

Hence, this thesis also investigates the feasibility of generating and updating the forecasts over the commissioning period without manual intervention.

- **Consideration of both aleatoric and epistemic uncertainty** – Non-optimal model structures and a lack of training data can lead to high epistemic model uncertainty, which in turn can degrade forecast quality if not taken into account. Moreover, considering the previous paragraph, both aspects are infeasible to avoid in practice. Thus, it seems particularly important for practical applications to investigate methodological approaches that also consider and compensate these epistemic uncertainties and therefore generate very good results even with limited data or model structures that are not perfectly application specific.

  However, previous studies on PV power mostly do not differentiate between the different types of uncertainty and do not consider both [82]. They focus instead commonly only on the aleatoric component (see, e.g., [47, 48, 65, 142]). This thesis will focus in particular on the consideration of both types of uncertainties e.g., by using epistemic extensions. Consequently, to the best of the author's knowledge, for a number of used probabilistic PV power approaches (e.g., MDN, GARCH), this thesis will investigate epistemic extensions in detail for the first time. This also enables a specific analysis of the influence of the different uncertainty types.

- **Extension and adaptation of probabilistic forecasting algorithms for the prediction of PV power** – There are several advanced studies on probabilistic machine learning approaches e.g., for computer vision use cases [122], while probabilistic solar forecasting is the least mature area in the field of energy time series forecasting [9]. Accordingly, this thesis leverages a number of approaches that have yielded very good results in other fields (e.g., MDN, MC dropout) and adapts them for PV power for the first time.

- **Comparison of several probabilistic methods for PV power** – As outlined in Section 2.4.3, a cross comparison of different forecasting algorithms between several papers is always very challenging. Hence, large comparative studies between different approaches provide a clear added value for the PV power forecasting community. This thesis intends to fulfill this need by comparing eight different approaches, some of them with and without different epistemic extensions.

“[M]any researchers draw equivalence between accuracy and value [of a forecast]. Whereas accuracy can indeed reflect the significance of a forecasting research, it does not necessarily translate to value.”

Dazhi Yang [219]

(Professor at the Harbin Institute of Technology and currently the world’s leading published solar forecaster)

3

Test and evaluation framework

This Chapter provides an overview of the applied analysis framework. First, the data sets and preprocessing steps along with the creation of the simulated forecast initialization instances are outlined. Afterwards, methods and metrics for evaluating the probabilistic forecasting algorithms are introduced.

3.1 Data sets used and preprocessing applied

The algorithms in this study are compared using data from three different sites in Central Europe. An overview of the site characteristics is summarized in Table 3.1. Thereby the solar variability $\sigma\Delta\kappa t$ describes the standard deviation of the changes of the clear sky index, which is the GHI in relation to the GHI under clear sky conditions. As such, it characterizes the volatility of local weather conditions and serves as a relative reference for how difficult it is to predict PV power at that site. All PV systems are rooftop installations with mixed orientation. Alongside the measured PV power, the GHI and outside temperature predicted by an external provider on the previous day serve as input signals for the forecasts. In order to achieve practical conditions, the forecasts of the provider Meteonorm were continuously recorded at midnight of the respective previous day. Figure 3.1 illustrates exemplarily the temporal behavior of the individual signals for the location North Bavaria (for the other location see Appendix: A.3). The measured values were available at a sample rate of one minute.

Table 3.1: Main information concerning the data used in this work.

	North Bavaria (Germany)	South Bavaria (Germany)	Vienna (Austria)
Elevation [m]	280	725	150
Annual GHI [MWh/m^2]	1.77	1.88	1.79
Time period	08/19 – 02/21	01/19 – 09/21	05/17 – 04/19
Sample rate [min]	15	15	15
Ratio of missing and removed days [%]	1.3	4.9	12.2
Solar variability ($\sigma\Delta\kappa t$)	0.188	0.186	0.195
$\overline{P_{\text{peak, daily}}}$ of PV panel [kW]	1.29	14.584	14.95

First, outlier and erroneous measured values (e.g., variance of signal is too long zero, signal exceeded physical feasible thresholds) were deleted. Subsequently, the signals were resampled on a 15 minute basis, as this is also the time resolution of the forecast[1]. Thereby data gaps smaller than 30 minutes were linearly interpolated and days with gaps of more than 30 minutes were omitted. It was assumed that the panels were covered with snow, if the days had an

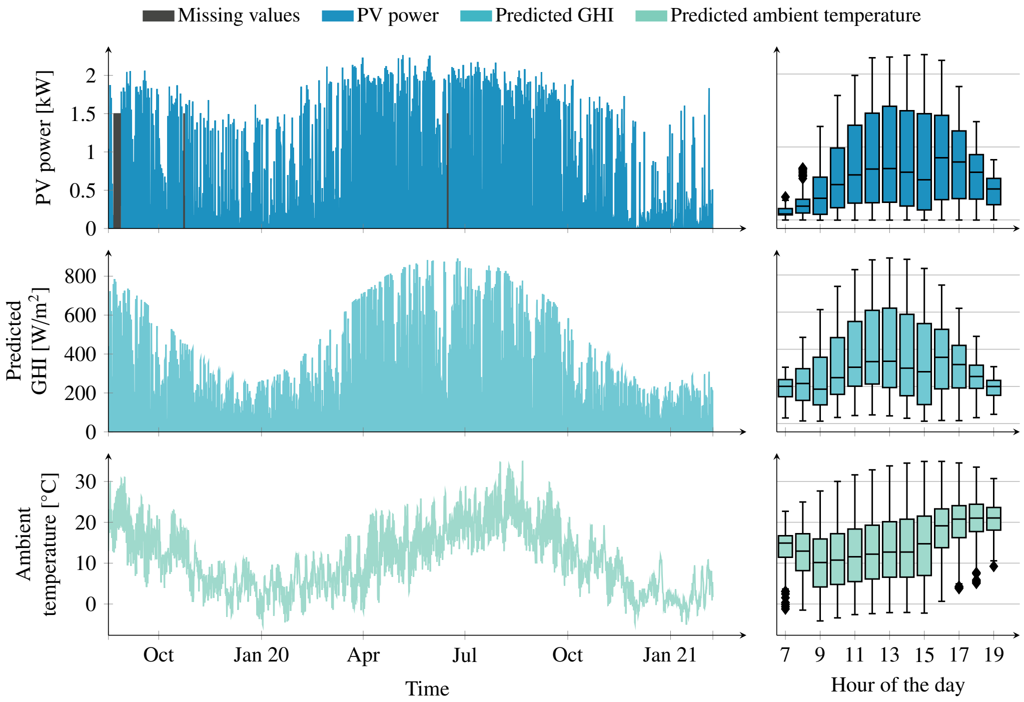


Figure 3.1: Temporal profile of the signals for the PV power system in northern Bavaria.

---

[1] The authors in Ref. [148] observed in their research that the use of smaller sample times during training did not increase the forecast accuracy and merely made the learning phase more time-consuming.

average power of less than five percent of the mean PV power of the previous month. As the methods in this paper are only intended to forecast nominal operation, these days were also ignored.[2]

3.2 Simulation setup

It is common for forecasting studies to perform analyzes by cross-validation or prediction using a single test set for each site. However, this does not provide a very good representation of a commissioning process. As an alternative approach, this work simulates 24 forecast initialization instances for each site (see Figure 3.2). In each case, the next seven days after each forecast initialization instance serve as test data.

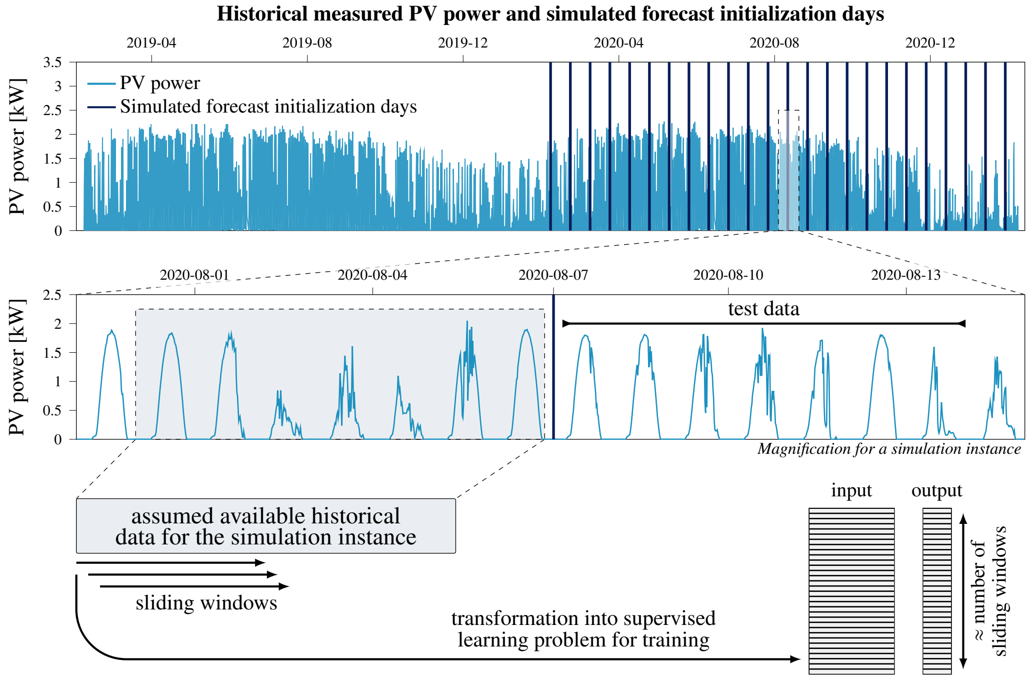


Figure 3.2: 24 simulated forecast initialization instances are defined for each of the three data sets (see top graph). The forecast quality for the next seven days (test data) is determined for these instances. A distinction is made between 2 scenarios. In the first scenario, the respective last 7 days are used as training data (see bottom graph left), while in the second scenario the last 182 days are used (not depicted in the figure).

---

[2] It is preferable to use a monitoring system in practice combined with multiple forecasting models for the different operating states (e.g., snow on panel, nominal operation.). Furthermore, the PV power signal has a very low volatility when the panel is covered, which would distort the accuracy metrics, if the dates with snow on the panel where not ignored.

Given that forecasting models are rarely adjusted over time, the probabilistic forecasts should be able to produce satisfactory results over the complete commissioning process. Accordingly, this thesis analyzes two different scenarios. The first scenario depicts the early stages of the commissioning process, where only seven days of training are available and therefore the data is limited. In the second scenario, 182 days of available measurements are considered. If days are thereby missing due to a data gaps for one of the instances, the previous days are included until the corresponding number of days with training data are available.

The forecasts are generated every 15 minutes over a forecast horizon of six hours, resulting in 24 different forecasts for each time instant. During the conversion to a supervised learning problem for the individual methods, the rows which output contained only sunset times were deleted. This leads to an even stronger focus on the nominal operation during the day for the model training.

3.3 Forecast evaluation

3.3.1 What constitutes a good forecast?

In order to evaluate and compare forecasts properly, it is crucial to initially define what actually constitutes a good forecast. This varies depending on the specific application and is particularly dependent on the individual factors described in Section 2.4 (e.g., interpretability, computation time). According to Murphy [152] three different ways for the characterization of the goodness of a forecast can be distinguished:

- **Consistency** – The extent to which the prediction reflects the forecaster's best estimate of the situation based on his or her level of knowledge (no influence e.g., due to bias)
- **Quality** – The extent to which the prediction matches actual events / observations
- **Value** – The extent to which the forecast helps a decision maker achieve additional economic and/or other benefits.

Forecast quality is the best-known criterion in the scientific field, and also the focus of this thesis, since a comparison of the informative value of a forecast is generally only possible with regard to a concrete use case and the applied methods. In addition, the consideration of consistency is mainly relevant for manually created or influenced forecasts. Nevertheless, as forecast value should not be overlooked when comparing multiple methods, the shortcomings of evaluating forecasting algorithms solely on the basis of their quality will be briefly discussed.

In order to compare the forecast quality of ensemble based, quantile/interval based and continuous CDF based approaches on an equal basis, they have to be transformed into a common form of presentation. Otherwise, they would contain varying degrees of information about the estimated probability density. This transformation is often performed by down sampling the estimated CDF with respect to the information density, which generally corresponds to a representation via quantiles e.g., $[10\,\%, 20\,\%, \ldots, 90\,\%]$. Afterwards, the evaluation is carried out using the metrics outlined in the next section. However, the necessary transformation may lead to the quantile-based methods being superior in terms of forecast quality, as they are designed directly for the determination of quantiles. Depending on the use case, though, the value of the estimated quantiles may not be as high given the lower information content. For example, to use a probabilistic prediction in stochastic optimization, a continuous CDF (e.g., parametric representation) is beneficial [128]. Yet, the authors in Ref. [142] have found that 69 % of published studies are non-parametric. One of the reasons could be that studies and publications primarily focus on the forecast quality and only secondarily on the often abstract and use-case dependent value. As the author stated in Ref. [219]: “many researchers draw equivalence between accuracy and value”. The same effect occurs not only when comparing the representation types but also when comparing the neural network and time series model based methods, as the latter have significantly more value for solutions on edge devices due to the limited computing capacities. Therefore, the difference between forecast value and forecast quality as well as the fact that the quality is not the sole criterion for the goodness of a forecast should be kept in mind when reading the results of this thesis.

3.3.2 Evaluating forecast quality

For deterministic forecasts, it is common to quantify the forecast quality with metrics based on the scalar forecast error, which is the difference between forecast and observation. Probabilistic forecasts, however, are not as straightforward to evaluate, since they have multiple quality objectives and involve comparing a probability distribution with a scalar observation in each case. This complexity, combined with the lack of established assessment methods, has often led to the use of inconsistent practices in the past (i.e., inappropriate metrics and benchmarks) [142]. As a result, several papers [97, 124, 143, 164, 216] focused on the subject to introduce a standard and to promote consistence and sensible methodological comparison. In the following, the recommended procedures are outlined and if necessary adjusted for the use case of probabilistic prediction of PV power.

One of the fundamental characteristics a good probabilistic forecast should exhibit is reliability, sometimes also referred to as calibration [171]. It characterizes whether the predicted distribution corresponds to the observed distribution over a sufficiently long data sets. If one e.g., predicts a 10 % to 90 % coverage interval, 80 % of the values should also fall into the interval. A well calibrated prediction therefore avoids systematic bias and ensures statistical consistency, which could otherwise lead to a systematic bias in the subsequent decision process [171]. Reliability can be quantified with the prediction interval coverage probability (PICP) [142], which is defined as:

$$\mathrm{PICP} = \frac{1}{N}\sum_{t=1}^{N} \zeta[t], \tag{3.1a}$$

$$\text{with} \quad \zeta[t] = \begin{cases} 1, & \text{if } y[t] \in \left[\mathrm{p}(\hat{y})_{\rfloor}[t], \mathrm{p}(\hat{y})^{\rceil}[t]\right] \\ 0, & \text{otherwise} \end{cases}, \tag{3.1b}$$

where $y[t] \in \mathbb{R}$ denotes the observed data pointand $N \in \mathbb{N}$ represents their overall amount. $\mathrm{p}(\hat{y})_{\rfloor}[t] \in \mathbb{R}$ and $\mathrm{p}(\hat{y})^{\rceil}[t] \in \mathbb{R}$ are the lower and upper bound of the predicted interval respectively at time t. Consequently, the PICP value should be equal or near the expected coverage rate to have a reliable forecast.

A possibility to check deviations in reliability graphically is the rank histogram[3]. There, the predicted CDF is divided into $M \in \mathbb{N}$ bins on the x-axis, while the y-axis shows the PICP value for each predicted interval of bins, which in this context is often termed relative frequency (see Figure 3.3). If the division of the bins is equidistant and the forecast is well calibrated, the rank histogram possesses a uniform distribution[4] [170]. In this dissertation, the CDF is divided into ten ranks for each of the rank histograms. Therefore, 10 % of the observations should fall into each decile, which is represented by the dashed line in the illustration. Deviations from this line indicate an over- or underestimation of the respective quantile. Accordingly, a ∪-shape indicates an underdispersed and a ∩-shape indicates an overdispersed forecast. A triangular shape, in turn, indicates a systematic bias. However, the uniform distribution of a rank histogram is a necessary but not a sufficient criterion for reliability. As the authors in [91] and [88] showed, one can obtain a seemingly perfect rank histogram while the forecasted and true underlying probability still differ.[5] Consequently, a combination with other evaluation methods is recommended.

Since the primary objective of probabilistic forecasting is to estimate future uncertainty, predicted probabilistic bands should be kept as wide as necessary to ensure reliability, but also as narrow as possible to provide the maximum amount of information. The latter is

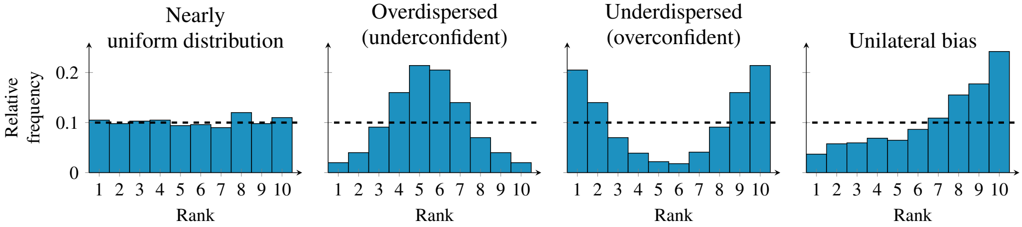


Figure 3.3: Example cases for the calibration of probabilistic forecasts visualized as rank histograms. With a sufficient number of data points a good calibration leads to a nearly uniform distribution. If the histogram is ∩-shaped, the forecast is overdispersed, so that in particular the middle percentiles disproportionately represent the actual measured values. With a ∪-shaped histogram, the forecast is underdispersed. As a result, a high proportion of the measured values are within the 10% percentile and above the 90% percentile. Disproportionately high values on a side of the histogram in turn indicate a systematic bias of the model.

---

[3] The rank histogram was originally developed and only used for ensemble forecasts [91]. For continuous probability depiction the analogous solution is often called probability integral transforms diagram [88]. However, since the term rank histogram has also been adopted in the literature for other forms of representation (e.g., quantiles [48]), it will be used for a better readability for all forms of representations in this thesis.

[4] Slight differences may still occur depending on the sample size investigated due to random noise. In this case, one can also determine confidence intervals for the bins based on a binomial distribution [170].

[5] For example, if the forecasted distribution is multimodal and the true underlying probability Gaussian, a generated rank histogram could possess a uniform distribution [91].

referred to as sharpness of a forecast and is in most practical use cases essential for a good forecast value. It can be considered as a measure of the efficiency of the forecasting model and can be quantified with the prediction interval average width (PIAW) [204]:

$$\mathrm{PIAW} = \frac{1}{N}\sum_{t=1}^{N}(\mathrm{p}(\hat{y})^{\rceil}[t] - \mathrm{p}(\hat{y})_{\rfloor}[t]). \tag{3.2}$$

However, sharpness alone is not a good measure of prediction quality. As can be seen in (3.2), the PIAW depends solely on the probabilistic forecast itself and therefore does not provide any information on how significant a deviation from the observation is. Consequently, a good score must consider both reliability and sharpness simultaneously. For instance, the authors in Ref. [88] state, that the objective of a forecast should be to maximize sharpness while having reliability as a constraint. In addition, a probabilistic metric should be proper to ensure consistency [24, 201]. For a scoring function $S\left(\mathrm{p}(\hat{y}), y\right)$ with an estimated forecast probability density $\mathrm{p}(\hat{y})$ and observation $y \in \mathbb{R}$ the expected outcome can be defined as:

$$S\left(\mathrm{p}(\hat{y}), \mathrm{p}(y)\right) = \int S\left(\mathrm{p}(\hat{y}), y\right) \mathrm{d}\mathrm{p}(y), \tag{3.3}$$

where $\mathrm{p}(y)$ is the true probability distribution [24, 201]. The scoring function is defined proper, if:

$$S\left(\mathrm{p}(y), \mathrm{p}(y)\right) \leq S\left(\mathrm{p}(\hat{y}), \mathrm{p}(y)\right), \tag{3.4}$$

whereby a lesser score denotes a more successful forecast [24, 201]. In other words, with a proper scoring function, the best score can only be generated by estimating the true underlying probability.

A metric that fulfills these three requirements is the continuous ranked probability score (CRPS). It is defined as the squared difference between the predicted CDF $\mathrm{P}(\hat{y}[t])$ and the observed CDF $\mathrm{P}(y[t])$ for signal y at time point t, and can be denoted as [93]:

$$\mathrm{CRPS}(\mathrm{P}(\hat{y}[t]), y[t]) = \int_{-\infty}^{\infty} \left(\mathrm{P}(\hat{y}[t]) - \mathrm{P}(y[t])\right)^2 \mathrm{d}\mathring{y}, \tag{3.5}$$

where $\mathring{y}$ represents the variable of interest[6] for $y[t]$ and the observed CDF $\mathrm{P}(y[t])$ is described by the Heaviside step function:

$$\mathrm{P}(y[t]) = \mathbb{1}(y[t] - \mathring{y}) \tag{3.6a}$$

$$= \begin{cases} 1 & \text{for } \mathring{y} \geq y[t] \\ 0 & \text{for } \mathring{y} < y[t] \end{cases}. \tag{3.6b}$$

---

[6] The distinction between y and $\mathring{y}$ is made to clarify that the integral is not calculated over the temporal axis of the signal y but rather for the respective CDF of the time series y at time t. See also Figure 3.4

For a better understanding of the CRPS, Figure 3.4 illustrates scenarios with varying sharpness and reliability of the probability forecast. CRPS has the same dimension as the predicted variable, with a lower value meaning higher prediction quality. If a deterministic forecast is provided, (3.7) transforms into an absolute error [93]. Consequently, its minimal value of zero is achieved if $\mathrm{P}(\hat{y}[t]) = \mathrm{P}(y[t])$, which means that the forecast is firstly deterministic and therefore perfectly sharp and secondly also perfectly reliable.

As discussed in the previous section, the respective representations are converted for a better comparability of the CRPS values between them into the quantiles $\upsilon \in [10\,\%, 20\,\%, \ldots, 90\,\%]$ beforehand. In the case of the ensemble forecast, the so-called classical method is thereby applied. This method assumes that each of the ensemble members has the same probability mass and that no forecast will fall outside the ensemble (see also Figure 3.5) [124].[7]

For a systematic analysis and better comparability between sites, the CRPS score is averaged over all observations N and normalized with the mean maximum daily produced power

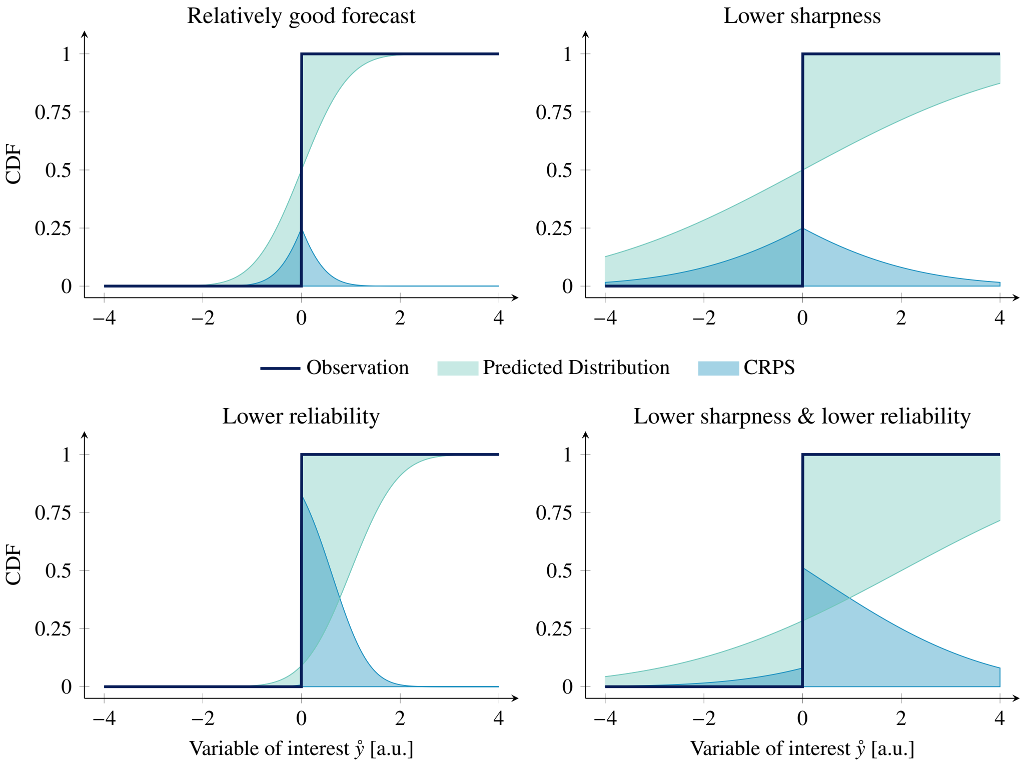


Figure 3.4: In the four different forecasting scenarios, the CRPS value is represented by the blue area. As can be seen, the CRPS depends on both reliability and sharpness. Adopted and modified from Ref. [117].

---

[7] For more information on this topic: Ref. [124] explains the advantages and disadvantages of different conversion options from ensembles to a CDF.

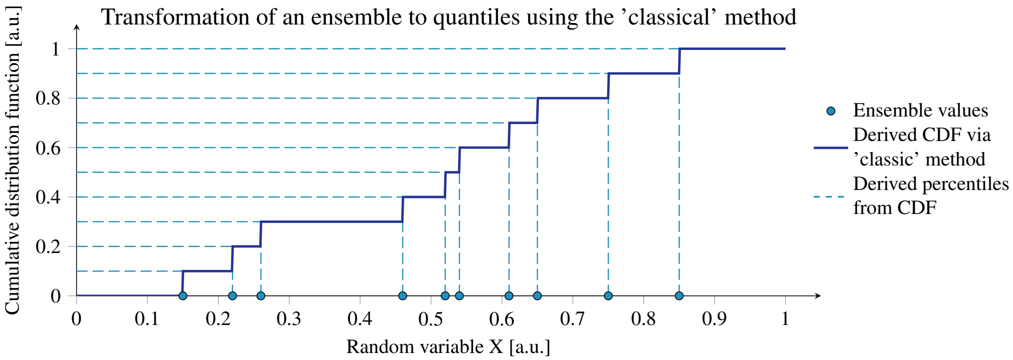


Figure 3.5: Conversion from an ensemble representation to its CDF using the 'classical' method adapted and modified from Ref. [124].

$\overline{P}_{\text{peak, daily}} \in \mathbb{R}_{\geq 0}$ (see Table 3.1) of the PV panels resulting in the normalized continuous ranked probability score (NCRPS):

$$\text{NCRPS} = \frac{1}{\overline{P}_{\text{peak, daily}}} \frac{1}{N} \sum_{t=1}^{N} \text{CRPS}(\text{P}(\hat{y}[t]), y[t]). \tag{3.7}$$

Furthermore, time points with marginal PV power generation (PV power < 3 % of the respective $\overline{P}_{\text{peak, daily}}$) are neglected, as these would affect the NCRPS disproportionately [220]. The remaining examined time instants possess more than 97 % of the produced electrical energy. The Python package properscoring [200] is used for the determination of the NCRPS, which adopts the approach from [93] using discrete[8] CDFs.

As the forecast quality depends particularly on the local weather conditions, it is advisable to make a comparison with a reference forecast in addition to the evaluation via a proper score. For this purpose, the complete-history persistence ensemble (CH-PeEn) is adapted to the use case of PV power in this study (see Algorithm 1 and Figure 3.6). The forecasting algorithm is recommended in [97] based on a comparative study against other benchmark methods for solar irradiance and can be interpreted as a probabilistic extension of the described smart persistence in equation (2.14).

To quantify the improvement related to the benchmark method and to analyze the benefit of different extension and hyperparameter combinations, this study applies the skill score (SS), which is calculated as follows:

$$\text{SS} = 1 - \frac{\text{NCRPS}_{\text{forecast}}}{\text{NCRPS}_{\text{ref}}}. \tag{3.8}$$

---

[8] Originally, the approach was developed for ensemble forecast assuming a so-called classical spacing of the CDF [124]. However, since then it is also often used for evaluating CDFs with quantiles [48, 123, 124, 225].

Algorithm 1: Complete-history persistence ensemble

```
Calculate clear sky index $\kappa$ for the PV power, $P_{\mathrm{PV}}$ at time $t$: $\kappa[t] = P_{\mathrm{PV}}[t]/P_{\mathrm{PV,csp}}[t]$.
  Since there is no clear sky value for PV power as there is for radiation, and a simple
  conversion is not possible due to the mixed orientation of the panels, the clear sky
  profile $P_{\mathrm{PV,csp}}$ is based on a moving horizon of the last seven days' maximum values
  at the same time of day.
Generate a forecast ensemble by using all past values of $\kappa$ in the same hour
Multiply the ensembles of clear sky indices with $P_{\mathrm{PV,csp}}$
```

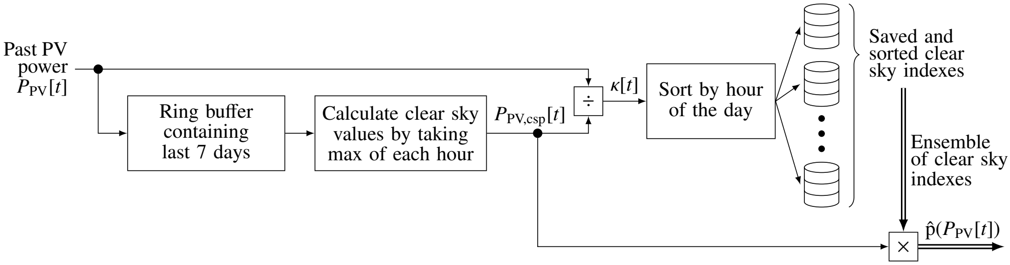

Past PV power $P_{\mathrm{PV}}[t]$
Ring buffer containing last 7 days
Calculate clear sky values by taking max of each hour
$P_{\mathrm{PV,csp}}[t]$
÷
$\kappa[t]$
Sort by hour of the day
Saved and sorted clear sky indexes
Ensemble of clear sky indexes
×
$\hat{\mathrm{p}}(P_{\mathrm{PV}}[t])$

Figure 3.6: Complete-history persistence ensemble.

A higher value for the latter indicates a relative greater improvement of the forecast $\mathrm{NCRPS}_{\mathrm{forecast}} \in \mathbb{R}_{>0}$ compared to the reference case $\mathrm{NCRPS}_{\mathrm{ref}} \in \mathbb{R}_{>0}$, with a value of one characterizing a perfect forecast. Negative values, in turn, signify a deterioration compared to the benchmark and a value of zero denotes no change in the forecast quality quantified by the NCRPS.

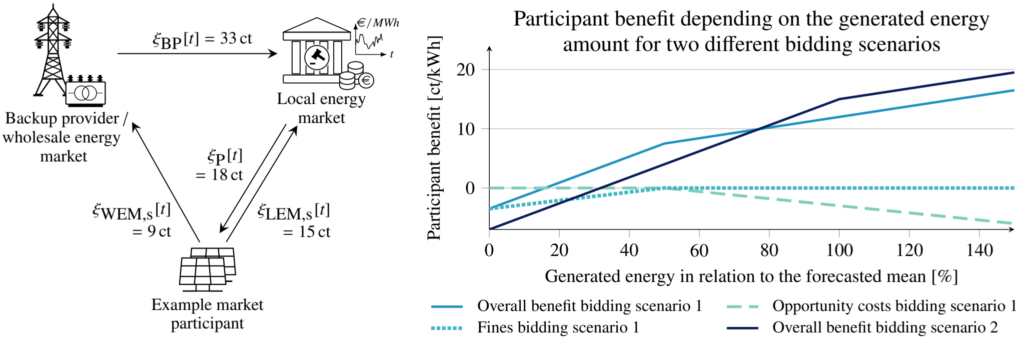
However, the skill score should be treated with caution. Even though the skill score is state of the art and widely used [97, 143], it is not proper according to the definition in (3.4). Unconsistencies may occur, for instance, with a limited number of samples caused by noise and dividing by the reference case [209]. Accordingly, the skill score is only taken into account after averaging the NCRPS values over all measured values, which usually comprise more than 150,000 samples depending on the specific analysis.

3.3.3 How does the forecast quality affect potential use cases?

As already explained in Section 2.1, it is preferable to have not only a benchmark but also a baseline for the required forecast accuracy. It is also important to know whether further increases in forecast quality also lead to more value or if a saturation occurs above a certain threshold value. In the following, this relationship and the difficulty of quantifying a baseline will be addressed with the example use case local energy markets. Therefore, the concept of the LEM and the exemplary use of probabilistic forecasts for a bidding strategy will be briefly explained.

In LEMs, DESs trade with each other in a geographically and socially close community (e.g., within a city district). This results in improved self-consumption and self-sufficiency of the local energy system, which in turn reduces the strain on the higher-level grid and reduces load or generation peaks [95]. However, smaller DESs have on average both a more volatile generation and a more volatile consumption profile due to the lack of aggregation effects resulting in higher forecasting uncertainties (e.g., compared to virtual power plants). Moreover, if a so-called prosumer provides a lower amount of energy than previously offered due to an incorrect forecast, additional costs arise for him. If this happens, the difference in energy must be provided by a backup supplier. By incorporating energy reserves (e.g., consistently offering only 50 % of the forecast PV generation on the market) in their bidding strategies, these penalties can be minimized. Nevertheless, this may result in opportunity costs, as the remaining energy is then no longer sold for comparable higher prices prices achievable on the LEM. Figure 3.7 illustrates this relationship using a simplified application example with different scenarios for the amount of generated energy.

Taking into account the probability $\Pr(E) \in \mathbb{R}_{\geq 0}$ that the energy $E \in \mathbb{R}_{\geq 0}$ is generated, the benefit $\xi_{\mathrm{WEM,s}} \in \mathbb{R}_{\geq 0}$ for selling to the wholesale energy market (WEM), the penalty costs $\xi_{\mathrm{p}} \in \mathbb{R}_{\geq 0}$ for non-compliance with the bid and the benefit $\xi_{\mathrm{LEM,s}} \in \mathbb{R}_{\geq 0}$ for compliance with the bid, overall losses compared to selling to the WEM can be avoided over a longer period of time.



(a) Overview of simplified bidding example. **(b)** Participant benefit for two bidding scenarios.

Figure 3.7: Participant benefit for a simplified bidding example of a prosumer with the LEM. It is assumed that the benefits for selling to the LEM $\xi_{\mathrm{LEM,s}}$ are uniform for the entire quantity of energy traded, the complete undelivered energy segments must be acquired via a backup provider for ξ_{BP} and any surplus energy is sold at a fixed price $\xi_{\mathrm{WEM,s}}$, analogous to the current concept of surplus feed-in. The price differences between ξ_{BP} and $\xi_{\mathrm{LEM,s}}$ result also from assumed savings in taxes and grid fees.
In the first bidding scenario, 50% of the forecast value was offered on the local market and in the second scenario 100%. Due to penalty costs by non-adherence, the particitpant benefit can also be negative (e.g., forecast mean offered but less than ~ 50 % are generated). Accordingly, only as much energy should be offered as can be expected to be delivered. However, a bid that is too low leads to opportunity costs if more energy is generated. Hence, if 100 % of the forecasted power is generated the benefit of bidding the full 100 % at the LEM is greater than when bidding, for instance, only 50 %.

For this, the individual bids have to ensure the following:

$$\underbrace{\Big(\Pr(E)[t]\cdot\xi_{\mathrm{LEM,s}}[t]-(1-\Pr(E)[t])\cdot\xi_{\mathrm{p}}[t]\Big)}_{\text{Expected benefit for each kWh sold via the LEM}}\cdot E\geq\xi_{\mathrm{WEM,s}}[t]\cdot E. \tag{3.9}$$

The penalty costs in turn result from

$$\xi_{\mathrm{p}}[t]=\xi_{\mathrm{BP}}[t]-\xi_{\mathrm{LEM,s}}[t]+\xi_{\mathrm{LEM,ic}}[t], \tag{3.10}$$

where $\xi_{\mathrm{BP}}\in\mathbb{R}_{\geq 0}$ is the cost of the energy from the backup provider (e.g., wholesale energy market) and $\xi_{\mathrm{LEM,ic}}\in\mathbb{R}$ is the internal price premium or discount of the LEM after consolidation of all non-compliant bids and including any penalty. Consequently, the necessary minimum price $\xi_{\mathrm{LEM,s}\rfloor}\in\mathbb{R}_{\geq 0}$ to prevent losses can be calculated for each given probability of the probabilistic prediction using

$$\xi_{\mathrm{LEM,s}\rfloor}[t]=\xi_{\mathrm{WEM,s}}[t]+\xi_{\mathrm{BP}}[t](1-\Pr(E)[t])-\xi_{\mathrm{LEM,ic}}[t]\Pr(E)[t]. \tag{3.11}$$

Since there is also a price ceiling (price for consuming from the WEM $\xi_{\mathrm{WEM,b}[t]}\in\mathbb{R}_{\geq 0}$), a probabilistic range results that should be traded on the local energy market. Assuming that $\xi_{\mathrm{WEM,b}}[t]\sim\xi_{\mathrm{BP}}[t]$, the marginal probability threshold results in

$$\Pr_{\mathrm{mt}}[t]=\frac{\xi_{\mathrm{WEM,s}}[t]}{\xi_{\mathrm{BP}}[t]+\xi_{\mathrm{LEM,ic}}[t]}, \tag{3.12}$$

and is therefore dependent on the time-varying price variables. Accordingly, a continuous CDF is preferable for this use case, as the corresponding amount of energy is predicted for each probability. Figure 3.8 illustrates an example of $\Pr_{\mathrm{mt}}$ and the characteristic of $\xi_{\mathrm{LEM,s}\rfloor}$

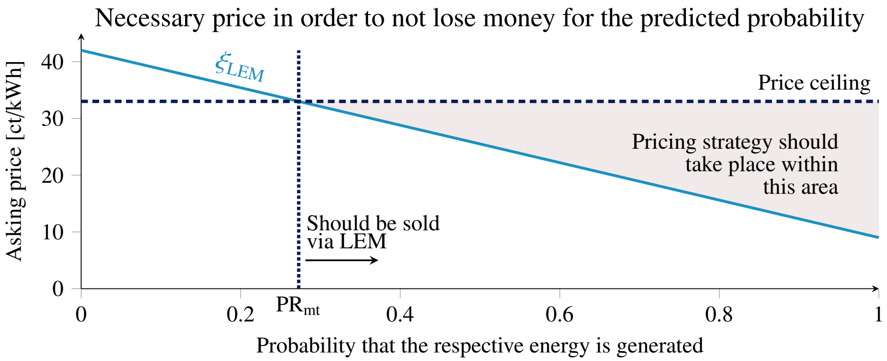


Figure 3.8: Necessary minimum price $\xi_{\mathrm{LEM,s}\rfloor}$ to prevent losses over a long period of price depending on the respective generation probability. The values of the example scenario from Figure 3.7 were used for the diagram. Due to the price ceiling, which corresponds to trading via the WEM, a marginal probability threshold for bidding via the LEM results at the intersection point.

depending on the respective probability of generating energy. The resulting amount of energy $E_{\mathrm{mt}} \in \mathbb{R}_{\geq 0}$ that should be traded results from the probabilistic prediction via $\mathrm{P}^{-1}(\mathrm{Pr}_{\mathrm{mt}}) \cdot \Delta t$ or, according to the notation used so far for the PV prediction: $y_{\mathrm{Pr}_{\mathrm{mt}}} \cdot \Delta t$. For this amount of energy, $\xi_{\mathrm{LEM,s}\rfloor}$ should only serve as a minimum bid. An individual pricing strategy based on the market situation for profit maximization is still recommended (see e.g., Refs. [8, 158]).

The question in relation to the use case is now what consequences a bad probabilistic forecast has. A forecast with a lower sharpness also has a lower energy quantity E_{mt}, which should be traded on the LEM. Accordingly, the opportunity costs increase in this case and the following applies:

$$\mathrm{PIAW}[t] \propto (\xi_{\mathrm{LEM,s}}[t] - \xi_{\mathrm{WEM,s}}[t]). \tag{3.13}$$

To examine the influence of reliability, two different cases must be distinguished. Systematic overpredictions will lead to opportunity costs, while underpredictions will lead to penalty costs. The additional energy costs $\xi_{\mathrm{rel}} \in \mathbb{R}_{\geq 0}$ result accordingly:

$$\xi_{\mathrm{rel}}[t] = \begin{cases} (y_{\mathrm{Pr}_{\mathrm{mt}}}[t] - \hat{y}_{\mathrm{Pr}_{\mathrm{mt}}}[t]) \cdot (\xi_{\mathrm{LEM,s}}[t] - \xi_{\mathrm{WEM,s}}[t]) & \text{for } \hat{y}_{\mathrm{Pr}_{\mathrm{mt}}}[t] < y_{\mathrm{Pr}_{\mathrm{mt}}}[t] \\ (\hat{y}_{\mathrm{Pr}_{\mathrm{mt}}}[t] - y_{\mathrm{Pr}_{\mathrm{mt}}}[t]) \cdot \xi_{\mathrm{p}}[t] & \text{for } \hat{y}_{\mathrm{Pr}_{\mathrm{mt}}}[t] > y_{\mathrm{Pr}_{\mathrm{mt}}}[t] \end{cases}. \tag{3.14}$$

Nevertheless, defining a necessary accuracy limit, e.g., in the form of a maximum NCRPS score, is difficult. On the one hand, the respective weighting of the individual costs is dependent on multiple time-dependent variables, as can be seen in (3.13) and (3.14). In particular, $\xi_{\mathrm{LEM,s}}$ depends not only on one's own bidding behavior but also on the bidding behavior of other market participants. At the same time, the true $y_{\mathrm{Pr}_{\mathrm{mt}}}$ cannot be determined for a single point in time but can only be derived systematically over a longer period. Finally, in multi-modal DESs the effects of inaccurate predictions can often be reduced by optimal operation using e.g., electrical storage or multi-modal coupling. Consequently, simulations of different market participants and bidding behaviors for varying prediction errors are useful for more accurate estimates of the impact of prediction quality on costs. To the best of the author's knowledge, this has so far only been done for local energy markets in Ref. [185] for deterministic forecast errors of load forecasts at the LEM.

However, it can be deduced from (3.13) and (3.14) that there is no threshold for maximum accuracy or a range from which its significance decreases. This statement was also supported by the simulations in Ref [185]. Each improvement of the forecast automatically leads to lower overall costs over time and should therefore be pursued taking into account the respective effort.

In addition to the presented method, there are also other approaches to compensate the inaccuracies and uncertainties of forecasts in LEM. For instance, the author of this thesis has co-authored three pending patents (Refs [59–61]) on the integration of probabilistic forecasts in LEMs.

"A large part of research efforts is actually still focused on point forecasting only, with the main objective of increasing forecast accuracy. [...] [However], there will always be an inherent and irreducible uncertainty in every prediction."

Pierre Pinson [169]
(Prof. at Imperial College London and Editor-in-Chief of the International Journal of Forecasting)

4

Methodology

This chapter explains the forecasting methods used in this work in detail. First, the applied and for the use case adapted deterministic forecasting structures of the ARX model and the MLPs are specified. This includes how the commissioning can be ensured without the need for manual adjustments. Subsequently, the probabilistic methods for the two approaches including their extensions are discussed. Thereby content of previous conference papers and journal publications by the author about the respective probabilistic methods are partly integrated (ARX: [57], MLP with MDN: [58], MLP with MC dropout: [56]).

For a comprehensible structure, the probabilistic approaches are sorted according to the representation forms (ensemble, quantiles, continuous CDF), whereby the probabilistic extension for the ARX model is outlined first followed by the MLP approach.

4.1 Underlying forecasting frameworks

4.1.1 ARX model

Preprocessing and overall concept

As ARX models can only depict stationary behavior (signals without seasonality and with constant mean and homoscedasticity) a stationarization process for the respective input and output signals needs to be carried out beforehand. For solar irradiation predictions, the GHI signal is generally stationarized by dividing it with the GHI signal under clear sky conditions, which is a common signal provided by weather services [47]. The resulting, so-called, clear sky index kt can then be forecasted with the time series model.

However, there is no corresponding signal readily available for PV power. In Ref. [13] the authors calculated the “clear sky PV power“ using a two-dimensional smoothing kernel along the days and the respective time of the day as an alternative for an ARX model. Nevertheless, this approach requires either much more historical data or, as in the reference, future PV power data, which of course are not available ex ante in practice. Another common alternative for the preprocessing of ARX models is the differentiation of the signal values, which would indirectly correspond to the integrative part of the 1st order of the ARIMA model family [130].

Instead, this thesis proposes a time series decomposition approach as an alternative. The seasonal daily component of the PV forecast can be compensated using day ahead forecasts. This allows additional information about the behavior of the physical system and the signals of the weather forecast already to be incorporated during the stationarization. Afterwards, only the remaining, commonly referred to as stochastic, component is modeled with the ARX model.

For a rudimentary daily forecast, the nonlinear relationship between the GHI and the generated PV power (see also Section 2.2) is first adaptively linearized for the respective 15-minute intervals of a day. This is performed by calculating the ratio between the measured PV power $P_{\mathrm{PV}} \in \mathbb{R}_{\geq 0}$ and the predicted GHI $\in \mathbb{R}_{\geq 0}$ for the last seven days at the same time of day t.[1] Multiplication by the weather service provider’s GHI forecast then yields the forecast $\hat{P}_{\mathrm{PV,\,day\;ahead}} \in \mathbb{R}_{\geq 0}$:

$$\hat{P}_{\mathrm{PV,day\;ahead}}[t] = \frac{\frac{1}{7}\sum_{i=1}^{7} P_{\mathrm{PV}}\left[t - i\frac{24\mathrm{h}}{T}\right]}{\frac{1}{7}\sum_{i=1}^{7} \mathrm{GHI}\left[t - i\frac{24\mathrm{h}}{T}\right]} \cdot \mathrm{GHI}[t]. \tag{4.1}$$

Despite its relative simplicity, this forecasting method has shown better results than a physical based modeling approach and only slightly inferior results compared to more advanced

---

[1] Linearization along the temporal component has the advantage of selecting close operating points with respect to the irradiation/power characteristic. In addition, systematic shadowing effects are taken into account, as they hardly differ from one day to the next.

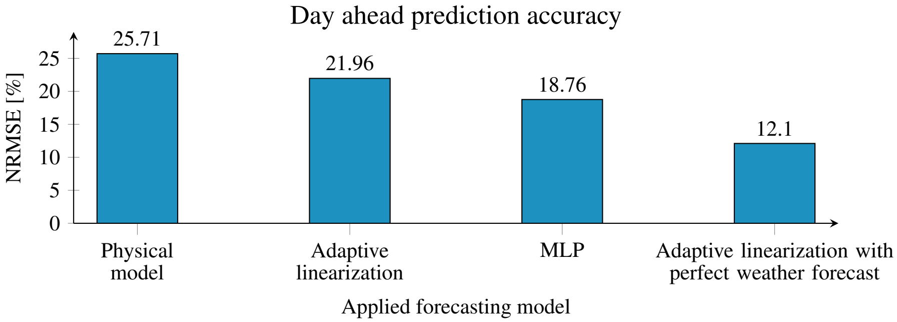


Figure 4.1: Day ahead prediction accuracy of the applied adaptive linearization of operating points compared to a physical approach as well as an MLP with one year of training data. The deterministic investigations were performed by the author in a previous research project based on five different PV power sites. The adaptive linearization model performs better than a physical model and almost as well as an MLP. In addition, ≈ 44.9 % of the error is due to the uncertainties in the input signal, as a perfect weather forecast yields only an error of 12.1 %. Adopted and modified from Ref. [54].

forecasting methods (MLP) in past analyzes of the author regarding deterministic day ahead forecasts (see also Figure 4.1).

Analogous to the clear sky index, the PV power is afterwards stationarized by dividing the measured power with the generated forecast:

$$\tilde{P}_{\mathrm{PV}} = \frac{P_{\mathrm{PV}}(t)}{\hat{P}_{\mathrm{PV,day\ ahead}}(t)}. \tag{4.2}$$

For the stationarization of the exogenous signals of the ARX model, additional day ahead forecasts are generated using smart persistence models (see (2.14)) of the last seven days. Consequently, the GHI $x_{\mathrm{GHI}}[t] \in \mathbb{R}$ and ambient temperature $x_{\mathrm{Tamb}}[t] \in \mathbb{R}$ provided from the weather forecaster are stationarized as follows:

$$\tilde{x}_{\mathrm{GHI}}[t] = \frac{x_{\mathrm{GHI}}[t]}{\frac{1}{7}\sum_{i=1}^{7} x_{\mathrm{GHI}}\left[t - i\frac{24\mathrm{h}}{T}\right]} \tag{4.3a}$$

$$\tilde{x}_{\mathrm{Tamb}}[t] = \frac{x_{\mathrm{Tamb}}[t]}{\frac{1}{7}\sum_{i=1}^{7} x_{\mathrm{Tamb}}\left[t - i\frac{24\mathrm{h}}{T}\right]}. \tag{4.3b}$$

The overall ARX forecasting approach of this thesis is summarized in Figure 4.2.

Applied to the use case, the mathematical notation of the ARX model with an added intercept $\theta_{\mathrm{c}} \in \mathbb{R}$ is as follows:

$$\begin{aligned}\tilde{P}_{\mathrm{PV}}[t] = \theta_{\mathrm{c}} + \sum_{i \in \mathcal{T}_{\mathrm{ar}}} \theta_{\mathrm{ar},i} \cdot \tilde{P}_{\mathrm{PV}}[t - i \cdot T] + \sum_{j \in \mathcal{T}_{\mathrm{GHI}}} \theta_{\mathrm{GHI},j} \cdot \tilde{x}_{\mathrm{GHI}}[t - j \cdot T] \\ + \sum_{k \in \mathcal{T}_{\mathrm{Tamb}}} \theta_{\mathrm{Tamb},k} \cdot \tilde{x}_{\mathrm{Tamb}}[t - k \cdot T],\end{aligned} \tag{4.4}$$

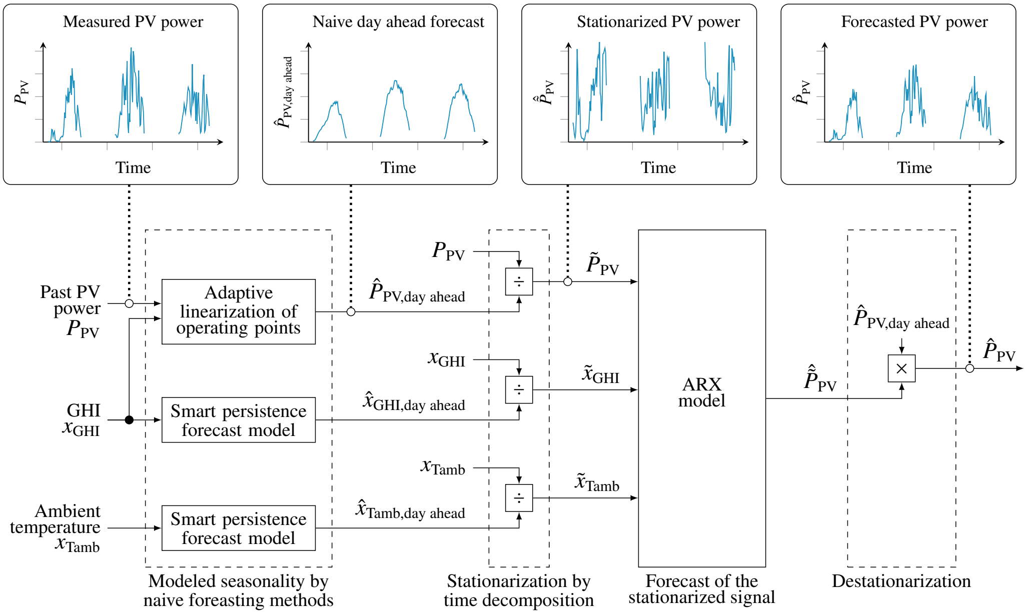

Measured PV power
P_{PV}
Time
Naive day ahead forecast
$\hat{P}_{\mathrm{PV,day\ ahead}}$
Time
Stationarized PV power
$\hat{\tilde{P}}_{\mathrm{PV}}$
Time
Forecasted PV power
$\hat{P}_{\mathrm{PV}}$
Time
Past PV power P_{PV}
Adaptive linearization of operating points
GHI x_{GHI}
Smart persistence forecast model
Ambient temperature x_{Tamb}
Smart persistence forecast model
P_{PV}
$\hat{P}_{\mathrm{PV,day\ ahead}}$
x_{GHI}
$\hat{x}_{\mathrm{GHI,day\ ahead}}$
x_{Tamb}
$\hat{x}_{\mathrm{Tamb,day\ ahead}}$
÷
÷
÷
$\tilde{P}_{\mathrm{PV}}$
$\tilde{x}_{\mathrm{GHI}}$
$\tilde{x}_{\mathrm{Tamb}}$
ARX model
$\hat{\tilde{P}}_{\mathrm{PV}}$
$\hat{P}_{\mathrm{PV,day\ ahead}}$
×
$\hat{P}_{\mathrm{PV}}$
Modeled seasonality by naive foreasting methods
Stationarization by time decomposition
Forecast of the stationarized signal
Destationarization

Figure 4.2: Function block diagram of the used overall forecasting concept with an ARX model.

whereby $\mathcal{T}_{\mathrm{ar}} \in \mathbb{N}^{D_{\mathrm{ar}}}, \mathcal{T}_{\mathrm{GHI}} \in \mathbb{Z}^{D_{\mathrm{GHI}}}, \mathcal{T}_{\mathrm{Tamb}} \in \mathbb{Z}^{D_{\mathrm{Tamb}}}$ are the sets of used time lags of the respective features, $\theta_{\mathrm{ar},i} \in \mathbb{R}, \theta_{\mathrm{GHI},j} \in \mathbb{R}, \theta_{\mathrm{Tamb},k} \in \mathbb{R}$ are the to be estimated parameters for the respective features and $D_{\mathrm{ar}} \in \mathbb{Z}_{\geq 0}, D_{\mathrm{GHI}} \in \mathbb{Z}_{\geq 0}, D_{\mathrm{Tamb}} \in \mathbb{Z}_{\geq 0}$ are the respective number of considered lags.[2] Converted into matrix notation, this results for N training data points in:

$$\underbrace{\begin{pmatrix} \tilde{P}_{\mathrm{PV}}[t] \\ \vdots \\ \tilde{P}_{\mathrm{PV}}[t-\zeta] \end{pmatrix}}_{\boldsymbol{Y}\in\mathbb{R}^{(\zeta-1)}} = \underbrace{\begin{pmatrix} 1 & \left[\tilde{\boldsymbol{P}}_{\mathrm{PV}}[t-iT]\right]^{\mathsf{T}}_{i\in\mathcal{T}_{\mathrm{ar}}} & [\tilde{\boldsymbol{x}}_{\mathrm{GHI}}[t-jT]]^{\mathsf{T}}_{j\in\mathcal{T}_{\mathrm{GHI}}} & [\tilde{\boldsymbol{x}}_{\mathrm{Tamb}}[t-kT]]^{\mathsf{T}}_{k\in\mathcal{T}_{\mathrm{Tamb}}} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \left[\tilde{\boldsymbol{P}}_{\mathrm{PV}}[t-iT-\zeta]\right]^{\mathsf{T}}_{i\in\mathcal{T}_{\mathrm{ar}}} & [\tilde{\boldsymbol{x}}_{\mathrm{GHI}}[t-jT-\zeta]]^{\mathsf{T}}_{j\in\mathcal{T}_{\mathrm{GHI}}} & [\tilde{\boldsymbol{x}}_{\mathrm{Tamb}}[t-kT-\zeta]]^{\mathsf{T}}_{k\in\mathcal{T}_{\mathrm{Tamb}}} \end{pmatrix}}_{\boldsymbol{X}\in\mathbb{R}^{(\zeta-1)\times(1+D_{\mathrm{ar}}+D_{\mathrm{GHI}}+D_{\mathrm{Tamb}})}}$$

$$\cdot \underbrace{\begin{pmatrix} \theta_{\mathrm{c}} \\ \boldsymbol{\theta}_{\mathrm{ar}} \\ \boldsymbol{\theta}_{\mathrm{GHI}} \\ \boldsymbol{\theta}_{\mathrm{Tamb}} \end{pmatrix}}_{\boldsymbol{\theta}\in\mathbb{R}^{(1+D_{\mathrm{ar}}+D_{\mathrm{GHI}}+D_{\mathrm{Tamb}})}} \text{, with}$$

$$\zeta = N - 1 - \max(\mathcal{T}_{\mathrm{ar}}, \mathcal{T}_{\mathrm{GHI}}, \mathcal{T}_{\mathrm{Tamb}}) - \max(0, -\mathcal{T}_{\mathrm{ar}}, -\mathcal{T}_{\mathrm{GHI}}, -\mathcal{T}_{\mathrm{Tamb}}). \tag{4.5}$$

---

[2] As can be seen from the specified value ranges, for $\tilde{x}_{\mathrm{GHI}}$ and $\tilde{x}_{\mathrm{Tamb}}$ future lags are also included, as the forecasts of the weather provider are available for future time steps.

There are already a variety of proven implementations for identifying ARX models for time series (e.g., System Identification Toolbox in Matlab, pmdarima in Python and fable in R). However, for this work, a new algorithm is implemented to determine the model parameters. This involves preprocessing the input data with appropriate time shifts into matrix form according to (4.5) and a subsequent determination of the model parameters using OLS. Compared to the above-mentioned implementations, the more flexible and direct access to the optimization problem for the parameter identification results in the following advantages for the given use case:

- **Ignoring nighttime values for the parameter estimation** – In practice, PV power forecasts provide added value only when the sun is present, i.e., for about 50% of the time of day, depending on the location and season. Accordingly, it is beneficial to optimize the model parameters only for these time periods without taking nighttime values into account during training. However, the mentioned existing implementations require continuous time series as input, which renders the disregard of the nighttime values infeasible.[3]

  In the matrix notation of (4.5), in turn, the rows for the time periods t can be disregarded where $P_{\mathrm{PV}}[t]$ is smaller than 1 % of $\overline{P}_{\mathrm{peak,\,daily}}$. Thus, the respective night values are still considered as input for the parameter identification, but not as model output.

- **Enabling bootstrapping of the training data** – This thesis studies the influence of the epistemic uncertainty for the ARX time series models by bootstrapping the training data. Thereby, repeated random samples are drawn from the used rows of the matrix in (4.5) during the parameter identification process for each ensemble member to approximate the uncertainty of the estimated parameters.

  Analogous to the previous item, bootstrapping is difficult without direct access to the optimization problem, especially when there are few training days, as otherwise only whole days could be sampled in each case.

- **Enabling cross validation** – The validation of time series models during initialization is traditionally done through simulations with a moving horizon. In this case, the last temporal section is mostly used as validation data set. Analogous to the points already described, the matrix notation, on the other hand, allows a fraction of the rows to be used as a test data set at a time, which also enables multiple cross validation.

---

[3] Other publications (e.g., Refs. [13, 47, 130]) handle this issue instead by deleting the nightly signal segments in preprocessing and connecting the signals synthetically afterwards. However, the deleted time segments in this case should be of equal length, so that in the case of the autoregressive component the temporal offset to the day before is constant over the entire year. As a result of the different sunrise and sunset times during a year, however, many night values are then still included in the parameter identification, especially during winter. Moreover, the sunset time periods of the previous day are used as model input for the prediction of PV power during sunrise, which leads to relatively large errors.

Automatic model order selection

In addition to the parameter estimation, the model structure which characterizes the physical behavior in the best way must also be determined during the initialization of the ARX model. This should be as automated as possible for the present application in order to avoid manual intervention and to adapt the model structure throughout the entire commissioning process. The choice of the model structure corresponds to the selection of the considered lags and thus the choice of $\mathcal{T}_{\text{ar}}, \mathcal{T}_{\text{GHI}}$ and $\mathcal{T}_{\text{Tamb}}$. For the classical ARX model, all lags up to a selected model order are considered (e.g., $\mathcal{T}_{\text{ar}} = \{1, 2, \ldots, D_{\text{ar}}\}$). However, at a sample rate of 15 minutes, this would require a total of 97 autoregressive parameters to include the same time point of the previous day, which in turn would unnecessarily increase the parameter uncertainty and thus the epistemic.

As an alternative, isolated important lags are used for the forecast in this work. For this purpose, the partial autocorrelation function (PACF) and the cross correlation function (CCF) are applied to determine the lags with the highest correlation to future time points. These lags are also the most important for the forecast, given that the ARX models are linear and the correlation analyzes describe the linear dependency relationship between the lags in each case.

Subsequently, the number of the respective lags of the features is determined. A greedy search algorithm is therefore developed (see Algorithm 2), which is based on the principle of Occam's razor. Hence, the model order is increased consecutively until the more complex model can no longer describe the underlying process better.

For the determination of the prediction quality of the respective model orders, cross validation is employed, as it enables the identification of the model with the lowest generalization error.[4] Additionally, a patience counter $\varrho_{\text{MAX}} = 3$ is implemented, as a strict greedy algorithm does not necessarily select the best model structure or lag combination. Although a lag can be higher sorted according to the PACF or CCF, the information it can provide may be overlapping with already selected lags. Thus, the additional information may be smaller in comparison to other available lags.

The exact procedure of the developed approach for the automatic selection of the optimal model structure can be seen in Algorithm 2.

---

[4] As an alternative to cross validation, other studies (e.g., Refs. [34, 148, 161, 202]) use information criteria, such as the Akaiken information criterion (AIC) to determine the optimal model structure [218]. AIC introduces a penalty term for the number of parameters used in addition to the goodness of the fit in the cost functional. Nevertheless, the performance of cross validation should be generally higher, because in addition to the number of parameters, it implicitly includes the two other influencing factors of model complexity, sample size and functional form [112].

ALGORITHM 2: Greedy search algorithm to determine best ARX model structure

Preliminary steps: Extract for all F features ($\tilde{P}_{\mathrm{PV}}, \tilde{x}_{\mathrm{GHI}}, \tilde{x}_{\mathrm{Tamb}}$) the D lags ψ with the highest PACF (endogen) or CCF (exogen) values and sort them by size into the respective feature set $\mathcal{T}_f$, s.th. $\mathcal{T}_f = \{\psi_{f,l}\}\ \forall f \in (1, F), d \in (1, D)$ and $F, D \in \mathbb{N}$.

Summary description: Select successively for each possible ARX model order $\Theta \in [1, F \cdot D]$ the lag ψ_{best} out of $\mathcal{T}_f$ that produces the smallest mean squarred error $\mathrm{MSE}_{\mathrm{low}}$ until no further improvement can be detected using cross-validation. A patience counter $\varrho \in \mathbb{N}$ is introduced as additional lags with higher PACF/CCF values do not necessarily provide the most additional information due to cross correlation between the lags.

```
$\mathbf{MSE_{low}} := [\mathrm{MSE}_{\mathrm{low},\Theta}]$, whereby $\mathrm{MSE}_{\mathrm{low},\Theta} = \infty\ \forall\ \Theta \in [1, F \cdot D]$    // initialization
for model order $\Theta$ in $[1, F \cdot D]$ :
    for feature set $\mathcal{T}_f$ in $[\mathcal{T}_1, \dots, \mathcal{T}_F]$ :
        $\varrho := 0$    // initialization
        for lag $\psi_{f,d}$ in $[\psi_{f,1}, \psi_{f,D}]$ :
            Determine $\mathrm{MSE}_{\Theta,f,d}$ of ARX model with additional (or first) lag $\psi_{f,d}$ using 3-fold cross validation
            if $\mathrm{MSE}_{\Theta,f,d} \leq \mathrm{MSE}_{\mathrm{low},\Theta}$ :
                $\mathrm{MSE}_{\mathrm{low},\Theta} := \mathrm{MSE}_{\Theta,f,d}$
                $\psi_{\mathrm{best},\Theta} := \psi_{f,d}$
            elif $\varrho \leq \varrho_{MAX}$ :
                $\varrho := \varrho + 1$    // increase $\varrho$, as this lag does not lead to a lower MSE
            else:
                break    // break for loop, as no other lag of this feature results in a lower MSE
    if $\Theta > 1$ and $\mathrm{MSE}_{\mathrm{low},\Theta} \geq \mathrm{MSE}_{\mathrm{low},\Theta-1}$ :
        break    // end greedy search, as no additional lag resulted in a lower MSE
    Add determined best additional lag $\psi_{\mathrm{best},\Theta}$ to selected ARX model structure
```

Multistep forecast generation

The classical ARX model estimates the output for one step into the future. However, PV forecasts with a forecast horizon of six hours and respectively 24 time steps are investigated in this work, due to its relevance for use in multimodal DES. Such a multistep forecast is achieved with a classical ARX model by iteration. First $\hat{y}[t]$ is estimated and afterwards $\hat{y}[t+1]$ is used to determine the next iteration step as if it were a real measurement point ($y[t] \approx \hat{y}[t]$).[5] However, this approach has some challenges for probabilistic PV power forecasts, as their uncertainty increases with the lead time. In other words, a forecast that extends further into

---

[5] This mathematical notation assumes a fixed temporal reference point with respect to t. In terms of the common notation of ARX models, where $y[t]$ is estimated, one can also think of it as if the relative reference point of t shifts. In this case, all time lags shift by one and $y[t-1]$ is approximated by the previously determined forecast value.

the future is generally subject to more uncertainty. Accordingly, the additional uncertainty of the iteratively used forecasts in comparison to the PV power measurements would need to be taken into account via some form of uncertainty propagation.

A more elegant alternative to account for the increasing uncertainty with increasing lead time is to compute the multistep forecast directly via different ARX models, expanding (4.4) to:

$$\begin{aligned}\tilde{P}_{\mathrm{PV}}[t+\tau] = \theta_{\mathrm{c}} &+ \sum_{i \in \mathcal{T}_{\mathrm{ar},\tau}} \theta_{\mathrm{ar},i,\tau} \cdot \tilde{P}_{\mathrm{PV}}[t - i \cdot T] + \sum_{j \in \mathcal{T}_{\mathrm{GHI},\tau}} \theta_{\mathrm{GHI},j,\tau} \cdot \tilde{x}_{\mathrm{GHI}}[t - j \cdot T] \\ &+ \sum_{k \in \mathcal{T}_{\mathrm{Tamb},\tau}} \theta_{\mathrm{Tamb},k,\tau} \cdot \tilde{x}_{\mathrm{Tamb}}[t - k \cdot T], \quad \forall \tau \in [0, 23]\end{aligned} \tag{4.6}$$

whereby only the time steps up to $t - 1$ are available to the respective model. The Greedy search algorithm described in the previous section is used for each of these models. Hence, for each forecast horizon lag, the ARX models possess presumably varying model structures.

4.1.2 MLP approach

Preprocessing and overall concept

Analogous to the ARX model, a general data preparation was performed (deletion of gaps, outliers, etc., see also Chapter 3.1). However, no time decomposition or adaptive linear stationarization of the data was performed, as this is not necessary for MLPs. Furthermore, due to their nonlinear structure, better day ahead forecasts were achieved in preliminary investigations with an MLP than with the rudimentary forecasts used for the stationarization of the ARX model (see also Figure 4.1, p. 57).

All features were standardized by:

$$\tilde{\boldsymbol{x}} = \frac{\boldsymbol{x} - \overline{\boldsymbol{x}}}{\sigma_x} \tag{4.7}$$

with $\sigma_x \in \mathbb{R}_{\geq 0}$ being the standard deviation of the feature $x \in \mathbb{R}$. This ensures that the input data for the MLP has a common scale and thereby improves the numerical condition of the optimization problem and accelerate convergence during training [103, 125].

In each case, the PV power over the last 24 h, the predicted GHI over the forecast horizon as well as the previous day, and the predicted ambient temperature over the forecast horizon and the last three hours are used as input data for the neural networks (see also Figure 4.3).[6] As with the ARX approach, rows whose prediction output are only during sunset times were deleted in the supervised learning problem.

---

[6] As the outdoor temperature has a much lower variability than the other two features and its impact on both the volatility and the absolute level of the PV power is minor, not the entire past day was taken into account.

In addition, for PV power and the predicted GHI signal, no improvements with respect to the deterministic prediction accuracy could be achieved in preliminary investigations with input data from even further back in time.

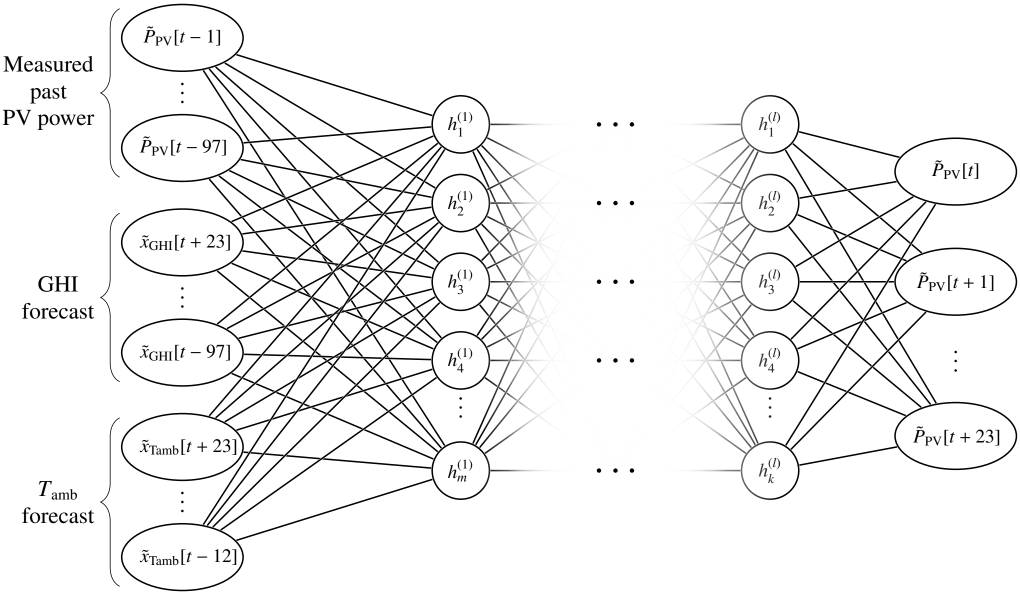

Measured past PV power
$\tilde{P}_{\mathrm{PV}}[t-1]$
$\tilde{P}_{\mathrm{PV}}[t-97]$
GHI forecast
$\tilde{x}_{\mathrm{GHI}}[t+23]$
$\tilde{x}_{\mathrm{GHI}}[t-97]$
T_{amb} forecast
$\tilde{x}_{\mathrm{Tamb}}[t+23]$
$\tilde{x}_{\mathrm{Tamb}}[t-12]$
$h_1^{(1)}$
$h_2^{(1)}$
$h_3^{(1)}$
$h_4^{(1)}$
$h_m^{(1)}$
$h_1^{(l)}$
$h_2^{(l)}$
$h_3^{(l)}$
$h_4^{(l)}$
$h_k^{(l)}$
$\tilde{P}_{\mathrm{PV}}[t]$
$\tilde{P}_{\mathrm{PV}}[t+1]$
$\tilde{P}_{\mathrm{PV}}[t+23]$

Figure 4.3: Principle network architecture of the underlying MLP.

Hyperparameter selection

Analogous to the greedy search algorithm used in the ARX model, there are also approaches for the automated determination of the optimal hyperparameter for MLPs. Under the popular term “automated machine learning“, both training hyperparameters and the network architecture are optimized in a multistage optimization process [105, 113]. For this purpose, e.g., grid search, random search, greedy search or Bayesian optimization can be employed [105, 113].

However, due to the large number of model parameters, the nonlinear model structure and the number of hyperparameters to be optimized, MLPs require significantly longer for hyperparameter optimization than the ARX model. Moreover, the number of available data points varies over the commissioning process, which arguably makes it more advantageous to perform the determination of the hyperparameters even multiple times.

Nevertheless, the objective of this thesis is not to find the optimal model structure in each case, but rather to use a constant network architecture as it is common in practice.[7] Thus, only the model parameters are adjusted by continuous training with the newly available data. Hence, regularization methods are used to mitigate overfitting. Additionally, the aim is to generate sufficiently accurate probabilistic forecasts also with non-perfectly tuned models, by taking the epistemic into account. This approach is supported by the hypotheses in Ref. [41],

---

[7] This approach also in line with that of Andrew Ng, cofounder and former head of Google Brain, who advocates a shift away from the model-centric approach prevalent in the academic world towards a data-centric approach for practitioners [174].

according to which large networks with appropriate regularization methods are preferable to smaller networks in practice. There the authors argue that with increasing number of neurons, the probability of capturing a local minimum with poor generalization error decreases [41].[8] By accounting for epistemic uncertainty through ensembles, this effect is likely to be further amplified, as multiple local minima are thus incorporated into the result.

Consequently, in this thesis, the network architecture for each representation is determined by manual hyperparameter selection based on additional data sets. Initially, the network architecture is chosen via random search and subsequently refined via a specified grid search. Afterwards, the network architecture and hyperparameters are adopted for all three sites.[9]

Nevertheless, all probabilistic methods used in this thesis have common hyperparameters and training specifications, the selection of which is briefly discussed below:

- **General network structure: rather deep than shallow** – Although in theory neural networks can act as universal function approximator already with one layer [43, 102], deep networks have performed better in empirical studies in recent years for a wide variety of uses cases than shallow ones (see Refs. [89, 153, 172] for summaries of reference cases). One reason is their ability to learn better and more detailed abstract relationships between the input data, because each subsequent layer can leverage the generated features of the previous one. Hence, mathematical functions with a compositional[10] structure can be represented with significantly fewer neurons [145].[11] In the case of probabilistic PV power predictions these intermediate steps can be e.g., general preprocessing, time decomposition, generation of volatility time features or elements of the physical chain in Section 2.2.

  Accordingly, sparser deep network structures are prioritized in the respective initial search of the network hyperparameters in this work.

- **Rectified linear unit (ReLU) as activation function in the hidden layers** – The hidden layer activation function used in this thesis is ReLU, a piece wise linear function that directly passes a positive input and returns zero otherwise (see also Figure 4.4) [81, 153]:

  $$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.8a}$$

  $$= \max(0, x) \tag{4.8b}$$

  Its non-saturating nature for positive values has been shown to minimize the occurrence of vanishing gradient. Hence it is more suitable for deep neural networks than e.g., a

---

[8] The authors also suggest that it is not necessarily useful to find the global minima of the training data, as this is often accompanied by overfitting [41].

[9] In contrast to the ARX model, an individual tuning of the hyperparameters/model order per location is not mandatory, since MLPs have a significantly higher capacity to represent varying mathematical functions due to the incorporated nonlinearity as well as the number of parameters even with constant model structure.

[10] A simple function composition is e.g., $f_3(x) = f_2(f_1(x))$.

[11] For instance, Ref. [68] proves that for approximation of certain functions neural networks with only one less layer need an exponentially larger width for the same accuracy.

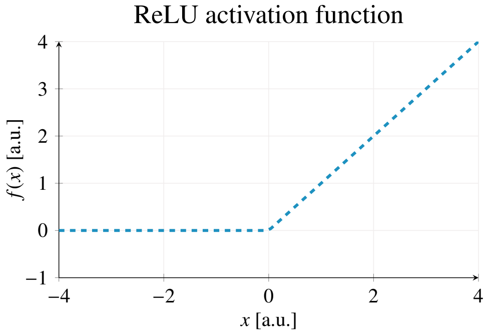


Figure 4.4: The rectified linear activation function. Adapted and modified from Ref. [153].

sigmoid oder hyperbolic tangent activation function [153]. Furthermore it is relatively computationally efficient, as only the mathematical max(·) function is applied and its gradient is either zero or one. For these reasons it is the most common activation function [153] and according to Ian Goodfellow also “the default activation function recommended for use with most feedforward neural networks” [89]. Consequently also several papers focussing on probabilistic or energy forecasting applied ReLU successfully (e.g., Ref. [25] for MDN, Ref. [232] for quantile neural networks, Ref. [122] for deep ensembles, Ref. [139] for energy forecasts with deep neural network architectures).

- **Applied regularization methods: dropout in each hidden layer, max-norm and early stopping** – To prevent the MLP from overfitting and to enable better generalization, several regularizing approaches are used in this work. The first one is dropout, which was introduced in 2014 by Srivastava et al. [193]. Its basic concept is to randomly switch off the output of individual perceptrons according to a specified probability $\mathrm{Pr}_{\mathrm{drop}} \in \mathbb{R}_{\geq 0}$. Accordingly, the propagation characteristics of a single pereceptron can be expanded from Equation 2.16 (p. 28) to:

$$h_j^{(l)} = z_j^{(l)} \cdot \phi_j^{(l)}\left(\sum_{i=1}^{N_{l,j}} \omega_{i,j}^{l} \cdot h_i^{(l-1)} + b_j^{(l)}\right), \tag{4.9}$$

  with the dropout variable $z_j^{(l)} \sim \mathrm{Bernoulli}(1 - \mathrm{Pr}_{\mathrm{drop},j})$. The used dropout mask (whether a perceptron is activated or not) is thereby updated for each new training step (size of the MiniBatch). Consequently, a different randomly selected thinner neural network is trained on each model parameter update. Accordingly, only around $n \cdot \mathrm{Pr}_{\mathrm{drop}}$ perceptrons are available for each training step, with $n \in \mathbb{Z}_{\geq 0}$ being the total number of hidden units (see also Figure 4.5).[12] This reduces the so-called co-adaptation [153, 193], i.e. that single nodes specialize too much on single features and other units become too reliant on them. Instead, the hidden units are motivated to be more independent and generate meaningful features on their own [193]. Consequently, “dropout can be interpreted as a way of regularizing a neural

---

[12] Accordingly, networks with dropout should have generally more neurons per layer than those without dropout.

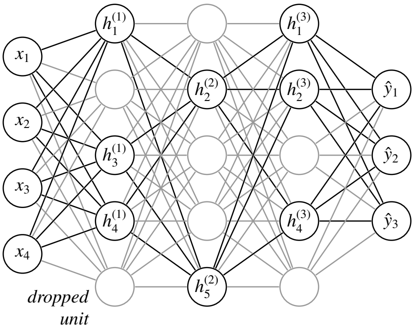


Figure 4.5: An example neural network with three hidden layers and a dropout probability of 50 %. Due to the dropped units (shaded grey) the network is thinner and the amounts of connections between the layers is reduced.

network by adding noise to its hidden units" [193]. This leads to increased robustness by preventing an overly specialized network for the training data.

Besides regularization, dropout also has the advantage of being able to generate ensemble members for neural networks without having to retrain them [84] (see segment 4.2.2 for more information). This is also one of the primary reasons for choosing it as a regularization method in this work.

To prevent individual network parameters from becoming too large, it is recommended to combine dropout with max-norm regularization [193]. This constrains the model parameters of each neuron to be smaller than a scaled ℓ_2 norm of all weights of the neuron [40]:

$$\omega_{i,j}^{(l)} = \begin{cases} \omega_{i,j}^{(l)} & \text{if } \|\boldsymbol{\omega}_{\boldsymbol{j}}^{(\boldsymbol{l})}\|_2 \leq C_{\text{Max-norm}} \\ C_{\text{Max-norm}} \cdot \frac{\omega_{i,j}^{(l)}}{\|\omega_j^{(l)}\|_2} & \text{otherwise} \end{cases}, \tag{4.10a}$$

$$\text{with} \quad \|\boldsymbol{\omega}_{\boldsymbol{j}}^{(\boldsymbol{l})}\|_2 = \sqrt{\omega_{1,j}^2 + \omega_{2,j}^2 + \cdots + \omega_{N_i,j}^2}, \tag{4.10b}$$

whereby $C_{\text{Max-norm}} \in \mathbb{R}_{>0}$ denotes the hyperparameter, which determines the scaling value of the ℓ_2 norm, $N_i \in \mathbb{N}$ the overall number of perceptrons of the previous layer and $\omega_{i,j} \in \mathbb{R}$ the model weight from the perceptron i of the previous layer $(l-1)$ to the perceptron j of the layer l.

Another regularization method used in this thesis is early stopping. One disadvantage of having a very large network for a given amount of data is that, if it is trained for too long (over too many epochs), the training data error may still be reduced, while the generalization error increases due to overfitting [153]. Therefore, early stopping considers the validation data error as estimation of the generalization error for the model selection. If the validation error does not decrease any further, the training process is stopped and the model parameter set with the lowest error is selected. A so-called patience parameter defines thereby how many epochs a model should be continued to be trained even without loss decrease in order to find a more optimal solution eventually (see Figure 4.6). This should prevent a premature

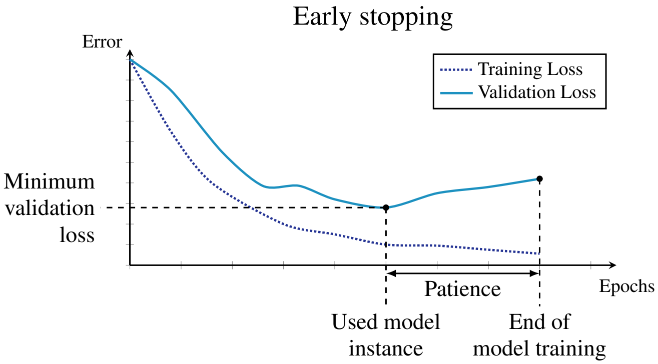


Figure 4.6: An exemplary illustration for early stopping. The generalization error is approximated by the validation loss, which is therefore sought to be minimized during model training. As soon as there is no lower validation error over a certain number of epochs (defined by the parameter patience) the training stops. Afterwards, the parameter set with the lowest validation loss is adopted.

termination of the training process while the best possible parameter combination has not yet been found [173]. Early stopping is particularly suited for a (semi) automated training of forecasts, since the hyper-parameter of the amount of training epochs does not have to be extensively adapted. This is significant for the analyzed use case, as the number of optimal training epochs will vary depending on the location and the amount of available training data. To quote Geoff Hinton: "Early stopping [is] beautiful free lunch" [96].[13]

- **Applying Adam as optimizer** – In this thesis, Adaptive Moment Estimation (Adam) is used as a gradient descent optimization algorithm. It considers both the momentum of the last gradients and an adaptive learning rate based on the second momentum of the gradients to update the model weights individually [116]. Accordingly, the model parameters $\boldsymbol{\theta}$, which include both model weights and biases, are updated over each training step t as follows[14]:

$$\boldsymbol{\theta}[t] = \boldsymbol{\theta}[t-1] - \frac{\beta_\mathrm{s}}{\sqrt{\boldsymbol{\delta}_\mathrm{v}[t]} + 10^{-8}} \boldsymbol{\delta}_\mathrm{m}[t], \tag{4.11a}$$

$$\text{with} \quad \boldsymbol{\delta}_\mathrm{m}[t] = \beta_\mathrm{m} \boldsymbol{\delta}_\mathrm{m}[t-1] + (1-\beta_\mathrm{m}) \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}, \boldsymbol{\theta}[t-1]), \tag{4.11b}$$

$$\text{and} \quad \boldsymbol{\delta}_\mathrm{v}[t] = \beta_\mathrm{v} \boldsymbol{\delta}_\mathrm{v}[t-1] + (1-\beta_\mathrm{v}) \cdot (\nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{D}, \boldsymbol{\theta}[t-1]))^2, \tag{4.11c}$$

whereby $\boldsymbol{\delta}_\mathrm{m} \in \mathbb{R}$ and $\boldsymbol{\delta}_\mathrm{v} \in \mathbb{R}$ are the exponential moving averages of the gradient and squared gradient of the loss function $\mathcal{L}$ with regard to $\boldsymbol{\theta}$ [116]. The parameter $\beta_\mathrm{m} \in \mathbb{R}_{\geq 0}$ and $\beta_\mathrm{v} \in \mathbb{R}_{\geq 0}$, in turn, specify the respective exponential decay rates and $\beta_\mathrm{s} \in \mathbb{R}_{>0}$ the

---

[13] Geoff Hinton is a luminary in the field of deep neural networks and works at Google and the University of Toronto. He alludes in this case to the "no free lunch theorem" [210], which claims that no optimization algorithm is better on average over all possible problems than any other optimization algorithm.

[14] For a better focus on the concept of Adam, the bias correction for the initialization and the smoothing term for numerical stability has not been included in the equation. See Ref. [116] for more detailed information on Adam.

step size. The consideration of the moment $\boldsymbol{\delta}_\mathrm{m}$ is particularly useful for dropout, as past gradients derived with different dropout masks still have an influence on the training. This also stabilizes the training behavior, since otherwise the varying dropout masks could lead to strongly oscillating direction changes of the gradients and also to a strongly fluctuating learning curve. Using an individual adaptive learning rate for each parameter also fits the intended use case, as input data is often sparse given the nighttime values [183]. Moreover, according to empirical analysis from its inventor [116], dropout is fairly robust with respect to the choice of hyperparameters and thus well suited for (semi-) automatic model training.

For these reasons, Adam is commonly used in deep learning architectures in the energy forecasting domain [207] (see, e.g., [25, 82, 147, 229]), and according to Refs. [139, 183], it is also the first choice for a training optimization algorithm.

More details on the MLP hyperparameters used for each probabilistic representation can be found in the corresponding sections that follow.

4.2 Probabilistic representation using ensemble members

4.2.1 Bootstrapping approaches for the ARX model

Bootstrapping was introduced in 1979 by Brad Efron [67].[15] It is a nonparametic method to estimate statistical characteristics of an underlying population using bootstrapped data samples. The basic concept is as follows: Given

- a random data sample $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^{N}$ from a population $\mathcal{P}$, and
- $M \in \mathbb{N}$ bootstrap samples $\mathcal{D}_m^* = \{\boldsymbol{x}_n^*\}_{n=1}^{N}$, $m \in [0, M]$ of the same size $N \in \mathbb{N}$, generated by random sampling with replacement from $\mathcal{D}$,

then $\mathcal{D}$ behaves to the population $\mathcal{P}$ as the bootstrapped samples $\mathcal{D}_{m\in[0,M]}^*$ to $\mathcal{D}$ [77, 120]. Consequently, the distribution of a calculated statistic $\Xi^* = \{\Xi_m^*\}_{m=1}^{M}$ (e.g., mean, median, identified parameter) for all bootstrapped samples is analogous to the unknown distribution of Ξ for $\mathcal{D}$ [77, 120].

Applied to the present use case, the uncertainty of model parameters can be approximated, for instance, by empirically determined distributions of identified parameters from bootstrap samples.[16]

---

[15] The name is derived from the phrase “pull yourself up by your bootstraps”, famously associated with “The Surprising Adventures of Baron Munchausen”. It means to succeed on your own, without any help from outside (in something that seems impossible).

[16] It is important to note that the bootstrap samples should have the same size as $\mathcal{D}$, since many statistical characteristics are dependent on sample size. Accordingly, the replacement during sampling is necessary, as otherwise it would be the case that $\mathcal{D} = \mathcal{D}_{m\in[0,M]}^*$. [120]

This thesis analyzes three different bootstrapping approaches for determining the uncertainty: bootstrapping of training data, bootstrapping of residuals and a two stage bootstrapping approach, which will be explained in the following. The three bootstrapping methods were selected because each incorporates different combinations of uncertainties. This allows to estimate the influence of the respective uncertainty types in a direct comparison.

All bootstrapping approaches are performed for the ARX model and in the domain of the decomposed signals, as it is recommended e.g., in Ref. [167]. In consequence, the residuals exhibit lower heteroscedasticity over time. Furthermore, for each approach 200 ensemble members are generated, as no significant improvement could be observed in exemplary tests with more ensemble members.[17]

Bootstrapping of training data

Training data bootstrapping, sometimes referred to as "case bootstrapping" [49], is the process of generating M random samples $\mathcal{D}^*_{m \in M}$ with the same size from the training data set $\mathcal{D} = \{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^{N}$. Afterwards, for each bootstrap sample $m \in [1, M]$ an ARX model is determined according to the method described in Section 4.1.1 and subsequently a forecast $\hat{y}_m \in \mathbb{R}$ is generated. This results in an ensemble of generated forecasts throughout all bootstrap samples (see also Algorithm 3 and Figure 4.7 for an overview of this appraoch).

The basic principle of bootstrapping of the training data is predominantly used to determine the confidence intervals of the model parameters [49, 77]. Accordingly, if the model structure remains fixed, the different training data will result in varying estimated parameters and, accordingly, a distribution over the parameters. Hence, it is clear that this approach only

ALGORITHM 3: Bootstrapping of training data

```
Preliminary steps: Generate the training data set $\mathcal{D} = \{\boldsymbol{X}_n, \boldsymbol{y}_n\}_{n=1}^{N}$ from the
    stationarized time series in the form of a regression problem (see (4.5))
for ensemble member $m$ in $M$ :
    $\mathcal{D}^*_m := \{\}$                                                    // initialization
    while $\#(\mathcal{D}^*_m) < \#(\mathcal{D})$ :
        Sample a block of six consecutive input-output cases $\{\boldsymbol{X}_n, \boldsymbol{y}_n\}_n^{n+6}$, with
          $n \in [1, N-5]$ from $\mathcal{D}$ and append it to the bootstrapped training data set $\mathcal{D}^*_m$
    Ensure $\#(\mathcal{D}^*_m) = \#(\mathcal{D})$ by deleting surplus elements          // necessary, if $N \bmod 6 \neq 0$
    Determine the best model structure with the the developed greedy search algorithm
      using cross validation (see Algorithm 2)
    Estimate the model parameter $\hat{\boldsymbol{\theta}}_{\boldsymbol{m}}$ for the determined model structure using OLS
    Generate a forecast ensemble member $\hat{y}_m = \boldsymbol{X}_\mathrm{f} \cdot \hat{\boldsymbol{\theta}}_{\boldsymbol{m}}$ with the respective forecast
      input $\boldsymbol{X}_\mathrm{f}$
```

---

[17] For the residual bootstrapping approach the members amount exceeds for instance, already the number of different residuals to sample from, if only two weeks of training data are available.

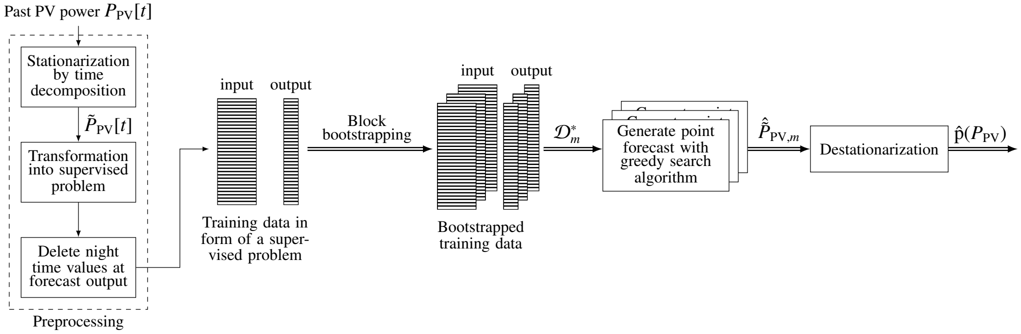

Past PV power $P_{PV}[t]$
Stationarization by time decomposition
$\tilde{P}_{PV}[t]$
Transformation into supervised problem
Delete night time values at forecast output
Preprocessing
input
output
Training data in form of a supervised problem
Block bootstrapping
input
output
Bootstrapped training data
$\mathcal{D}^*_m$
Generate point forecast with greedy search algorithm
$\hat{\tilde{P}}_{PV,m}$
Destationarization
$\hat{p}(P_{PV})$

Figure 4.7: Principle behind the bootstrapping of training data.

considers the epistemic uncertainty. Model residuals and thus the aleatoric is not incorporated. Nevertheless, the approach has been used, e.g., in Ref. [231] for probabilistic load forecasts and in Ref. [205] for solar irradiation forecasts both in combination with several machine learning appraoches. Moreover, the principle is also utilized in the machine learning domain with the so-called bagging method (bootstrap aggregation) [23]. By averaging the ensemble members, a part of the epistemic uncertainty is compensated, which enhances the deterministic forecasting quality [167].

By applying the developed greedy search algorithm, the ARX model structure in this work also varies across the bootstrapping samples. Accordingly, not only the parameter uncertainty but also part of the uncertainty induced by the model structure is taken into account with this approach.[18]

The bootstrapping is applied to the supervised training data set formulated according to Equation (4.5). Otherwise, direct sampling would introduce gaps in the time series.[19] However, when bootstrapping regression problems of time series, it is recommended to sample blocks consisting of several rows rather than individual rows [27]. Thereby some time related information and dependencies are still preserved within each sample [27]. For this purpose, 1.5 hours (six samples) are selected in this work, since the partial autocorrelation values of the stationarized PV power signals were highest over this time span.

Residual bootstrapping

Residual bootstrapping was already applied successfully in several energy forecasting use cases (see e.g., wind power: [86, 169], load: [212]). This approach generates an ensemble

---

[18] It should be noted that probably not all of the uncertainty generated by the assumed model structure is addressed, as specific model constraints are imposed by the assumption of an ARX structure (e.g., linear combination of the lagged inputs)

[19] In comparison: during the bootstrap of the supervised learning data set, certain time steps of the PV power signal might not be sampled as model output, but might appear indirectly in other samples as model input.

member $\hat{y}_m$ of the probabilistic prediction by adding a bootstrapped residual $\varepsilon_m^* \in \mathbb{R}$ to the prediction result of an underlying deterministic forecast:

$$\hat{y}_m[t] = \hat{y}[t] + \varepsilon_m^*. \tag{4.12}$$

Consequently, it depicts only the aleatoric uncertainty, as only the distribution at the model output is modeled. Despite the applied stationarization, a systematic variation of the signal variance often remains in solar forecasts, leading to a heteroscedastic distribution of the residuals (see also Figure 4.8) [90]. Therefore, a simple bootstrapping from all residuals should be avoided and instead a prior selection of possible residuals to sample from each time should be preferred.

In general, a nearest neighbor approach is used to find similar circumstances in the past in order to generate appropriate residual pools. Depending on the use case and available data sources, different influential variables are incorporated in past studies. In Ref. [90], for instance, the selection of possible residuals is based on the respective solar elevation and solar hour angle of the time points. Ref. [3] in turn, also includes additional meterological weather signals as cloud cover and predicted GHI.

Since this thesis focuses on the commissioning period, where data points are limited, considering too many different features for the selection of residual pools is unreasonable. Instead, the underlying concept of the solar positions is adapted by assigning the residuals from the same hour of the day in the past to the respective residual pools. Due to the applied adaptive ARX approach, the solar hour angle and solar elevation are similar over the short observation horizon and additionally, no further information about the location is needed. To prevent systematic bias the residuals are thereby also mean adjusted [50]. A summary of the bootstrapping method is given in Figure 4.9 and Algorithm 4.

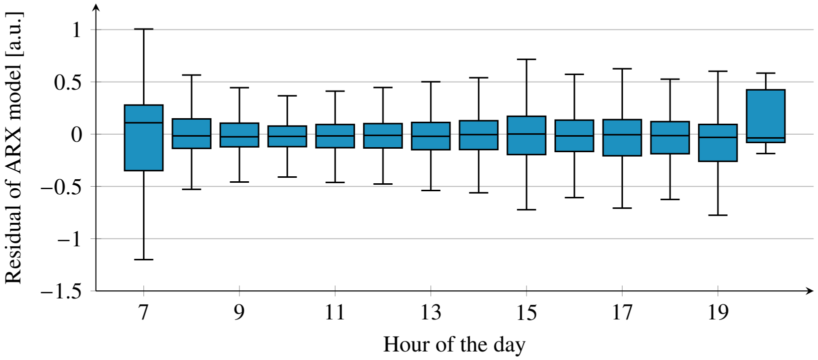
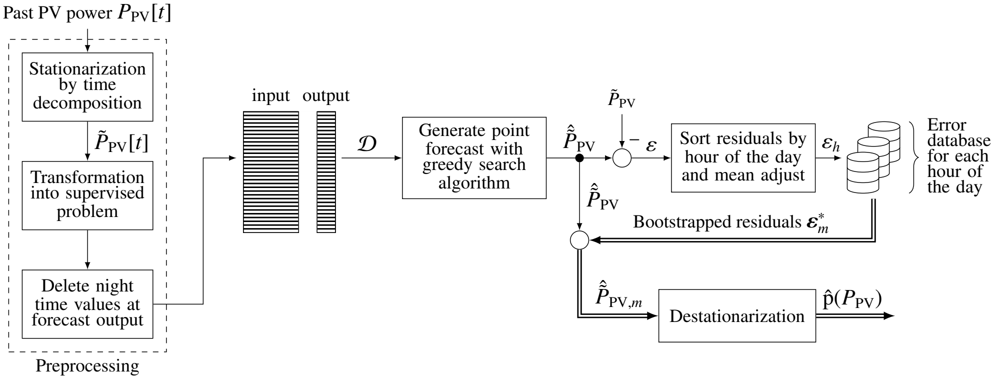


Figure 4.8: Distribution of the residual of the ARX model for the stationarised signal as a function of the hour of the day for the location in north bavaria. Even after stationarization, the variance varies differs throughout the day.

Algorithm 4: Residual bootstrapping

```
Preliminary steps: Determine the best model structure with the the developed greedy
  search algorithm (see Algorithm 2) and generate an according training data set
  $\mathcal{D} = \{\boldsymbol{X}_n, \boldsymbol{y}_n\}_{n=1}^{N} = (\boldsymbol{X}, \boldsymbol{y})$ from the stationarized time series
Estimate $\hat{\boldsymbol{\theta}} = (\boldsymbol{X}^T \boldsymbol{X})^{-1} \boldsymbol{X}^T \boldsymbol{y}$ of the underlying point forecast via OLS
Determine $\hat{\boldsymbol{y}} = \boldsymbol{X} \cdot \hat{\boldsymbol{\theta}}$ and calculate $\boldsymbol{\varepsilon} = \hat{\boldsymbol{y}} - \boldsymbol{y}$
Group residuals $\boldsymbol{\varepsilon}$ on an hourly basis $h \in [0; 23]$ and mean adjust each group
  $\boldsymbol{\varepsilon}_h = \boldsymbol{\varepsilon}_h - \overline{\boldsymbol{\varepsilon}}_h$
Generate the deterministic forecast $\hat{y}$ for the future
for ensemble member $m$ in $M$ :
    Bootstrap sample $\varepsilon_m^*$ from respective hourly pool $\boldsymbol{\varepsilon}_h$
    Generate the ensemble members $\hat{y}_m = \hat{y} + \varepsilon_m^*$
```



Past PV power $P_{\mathrm{PV}}[t]$
Stationarization by time decomposition
$\tilde{P}_{\mathrm{PV}}[t]$
Transformation into supervised problem
Delete night time values at forecast output
Preprocessing
input
output
$\mathcal{D}$
Generate point forecast with greedy search algorithm
$\hat{\tilde{P}}_{\mathrm{PV}}$
$\tilde{P}_{\mathrm{PV}}$
$-$
ε
Sort residuals by hour of the day and mean adjust
ε_h
Error database for each hour of the day
$\hat{\tilde{P}}_{\mathrm{PV}}$
Bootstrapped residuals $\boldsymbol{\varepsilon}_m^*$
$\hat{\tilde{P}}_{\mathrm{PV},m}$
Destationarization
$\hat{\mathrm{p}}(P_{\mathrm{PV}})$

Figure 4.9: Principle behind the residual bootstrapping method.

Extended sieve with residual bootstrapping

The two bootstrapping appraoches presented earlier each consider only one uncertainty type. Thus, a two-stage bootstrapping approach is adapted from Ref. [50] (Algorithm 6.4), which depicts both uncertainty types. To the best of the author's knowledge, this method has not been applied to probabilistic solar forecasts before.

First, in a preliminary for loop, bootstrapped residuals are used to generate synthethic output training data, which in turn is used to identify new model parameters. Thereby, primarily the epistemic is taken into account. Afterwards, the residuals are bootstrapped again in a secondary for loop similar to the residual bootstrapping. This primarily depicts the aleatoric. Analogous to residual bootstrapping, the residuals in both for loops are first sorted into respective pools depending on the hour of the day and afterwards mean adjusted. A summary and detailed description of the algorithm can be found in Figure 4.10 and Algorithm 5. The parameters $M \in \mathbb{N}$ and $R \in \mathbb{N}$ of the for loops thereby define how many bootstrap samples should be used to estimate the respective uncertainties. In this study, M was set to 20 and R to 10, which results overall also in 200 ensemble members.

ALGORITHM 5: Extended sieve with residual bootstrapping

```
Preliminary steps: Determine the best model structure with the the developed greedy
  search algorithm (see Algorithm 2) and generate an according training data set
  $\mathcal{D} = \{\boldsymbol{X}_n, \boldsymbol{y}_n\}_{n=1}^{N} = (\boldsymbol{X}, \boldsymbol{y})$ from the stationarized time series
Estimate $\hat{\boldsymbol{\theta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$ of the underlying point forecast via OLS
Determine $\hat{\boldsymbol{y}} = \boldsymbol{X} \cdot \hat{\boldsymbol{\theta}}$ and calculate $\boldsymbol{\varepsilon} = \hat{\boldsymbol{y}} - \boldsymbol{y}$
Group residuals $\boldsymbol{\varepsilon}$ on an hourly basis $h \in [0; 23]$ and mean adjust each group
  $\boldsymbol{\varepsilon_h} = \boldsymbol{\varepsilon_h} - \overline{\boldsymbol{\varepsilon}}_{\boldsymbol{h}}$
for parameter ensemble $m$ in $M$ :
    Bootstrap samples $\boldsymbol{\varepsilon}^*_{\boldsymbol{m}}$ from the respective hourly pool $\boldsymbol{\varepsilon}_\omega$ to generate a synthesized
      model output with $\boldsymbol{y}_m = \hat{\boldsymbol{y}} + \boldsymbol{\varepsilon}^*_m$
    Reestimate parameter $\hat{\boldsymbol{\theta}}_m = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}_m$ for this bootstrapped replication
    Calculate residuals $\boldsymbol{\varepsilon}_m = \boldsymbol{X}\hat{\boldsymbol{\theta}}_m - \boldsymbol{y}_m$, group them on an hourly basis and mean
      adjust each group $\boldsymbol{\varepsilon}_{h,m} = \boldsymbol{\varepsilon}_{h,m} - \overline{\boldsymbol{\varepsilon}}_{h,m}$
    for residual ensemble $r$ in $R$ :
        Bootstrap sample $\varepsilon^*_r$ from the respective $\boldsymbol{\varepsilon}_{h,m}$ and calculate an overall ensemble
          residual $\varepsilon_{m,r} = \underbrace{\boldsymbol{X}_\mathrm{f}(\hat{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}_m)}_{\text{primary epistemic}} + \underbrace{\varepsilon^*_r}_{\text{primary aleatoric}}$ with the forecast input $\boldsymbol{X}_\mathrm{f}$
        Generate predicted ensemble member $\hat{y}_{m,r} = \hat{y} + \varepsilon_{m,r}$
```

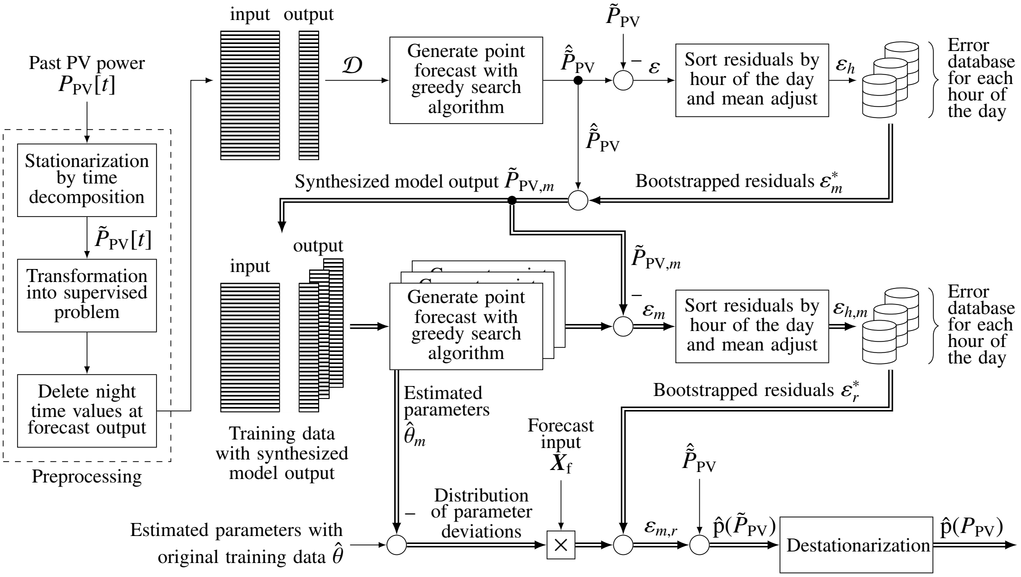

input
output
Past PV power
$P_\mathrm{PV}[t]$
Stationarization by time decomposition
$\tilde{P}_\mathrm{PV}[t]$
Transformation into supervised problem
Delete night time values at forecast output
Preprocessing
$\mathcal{D}$
Generate point forecast with greedy search algorithm
$\hat{\tilde{P}}_\mathrm{PV}$
$\tilde{P}_\mathrm{PV}$
$-$
ε
Sort residuals by hour of the day and mean adjust
ε_h
Error database for each hour of the day
$\hat{\tilde{P}}_\mathrm{PV}$
Synthesized model output $\tilde{P}_{\mathrm{PV},m}$
Bootstrapped residuals ε^*_m
input
output
$\tilde{P}_{\mathrm{PV},m}$
Generate point forecast with greedy search algorithm
$-$
ε_m
Sort residuals by hour of the day and mean adjust
$\varepsilon_{h,m}$
Error database for each hour of the day
Training data with synthesized model output
Estimated parameters $\hat{\theta}_m$
Bootstrapped residuals ε^*_r
Forecast input X_f
$\hat{\tilde{P}}_\mathrm{PV}$
Estimated parameters with original training data $\hat{\theta}$
$-$
Distribution of parameter deviations
$\times$
$\varepsilon_{m,r}$
$\hat{\mathrm{p}}(\tilde{P}_\mathrm{PV})$
Destationarization
$\hat{\mathrm{p}}(P_\mathrm{PV})$

Figure 4.10: Principle behind the extended sieve with residual bootstrapping method.

The first stage of this approach, resembles the so-called Sieve bootstrapping, which is especially recommended for time series [5, 26, 27].[20] The advantage of sieve bootstrapping is that temporal information and dependencies between samples are preserved. It was successfully applied for wind forecasting in Ref. [86] and for solar irradiation forecasting in Refs. [48, 90].[21] In addition, a similar variation of a two-step bootstrapping approach was successfully applied in Ref. [230] for load forecasting. There, first training data and second residuals were bootstrapped in combination with a random forest appraoch.

4.2.2 Monte-Carlo dropout with output calibration

Monte-Carlo dropout

All described bootstrapping approaches of the ARX model could also be adopted for neural networks. However, training MLPs is computationally more demanding than for ARX models due to the nonlinear optimization problem, the number of parameters to be trained, and the training epochs used. Thus, having the network trained individually per ensemble member – e.g., for the consideration of the epistemic – would significantly increase the total training time.

As an alternative approach, Yarin Gal et al. 2016 proposed the use of Monte Carlo (MC) Dropout [83, 84]. The key idea is that the regularization method dropout (see also section 4.1.2) is applied not only during training, but also during inference. This leads to varying network structures and therefore also varying estimated deterministic forecasts $\hat{y}$, despite only a single model training (see also Figure 4.11). This approach leverages thereby the dropout characteristic of decreasing co-adaptation, wich results in sparse subnetworks during inference that tend to accurately reproduce the underlying behavior. Or to say it differently, dropout can also be interpreted as if numerous thinned networks with shared weights are implicitly trained [193].

As already stated in Equation 2.17 (p. 31) and repeated here for better readability, the predictive distribution of the estimated output $\hat{y}$ given the respective model input $\boldsymbol{x}$ and the training data set $\mathcal{D}$ can be calulated by marginalization over the model parameters $\boldsymbol{\theta}$ [83]:

$$\mathrm{p}(\hat{y} \,|\, \boldsymbol{x}, \mathcal{D}) = \int \underbrace{\mathrm{p}(\hat{y} \,|\, \boldsymbol{x}, \boldsymbol{\theta})}_{\substack{\text{primary} \\ \text{aleatoric}}} \underbrace{\mathrm{p}(\boldsymbol{\theta} \,|\, \mathcal{D})}_{\substack{\text{primary} \\ \text{epistemic}}} \, \mathrm{d}\boldsymbol{\theta}, \tag{4.13}$$

---

[20] The difference to the described sieve bootstrapping approaches in the references is that instead of a simple autoregressive model also exogenous variables are considered in this thesis through the ARX model. In addition, the signal is not averaged in advance but stationarized via time series decomposition.

[21] One might ask why sieve bootstrapping was not used for the analysis of epistemic uncertainty in this thesis instead of training data bootstrapping. Since the former uses the output residual to generate new parameters, the aleatoric already has an effect on the ensemble generation. Accordingly, sieve bootstrapping does not model solely the epistemic.

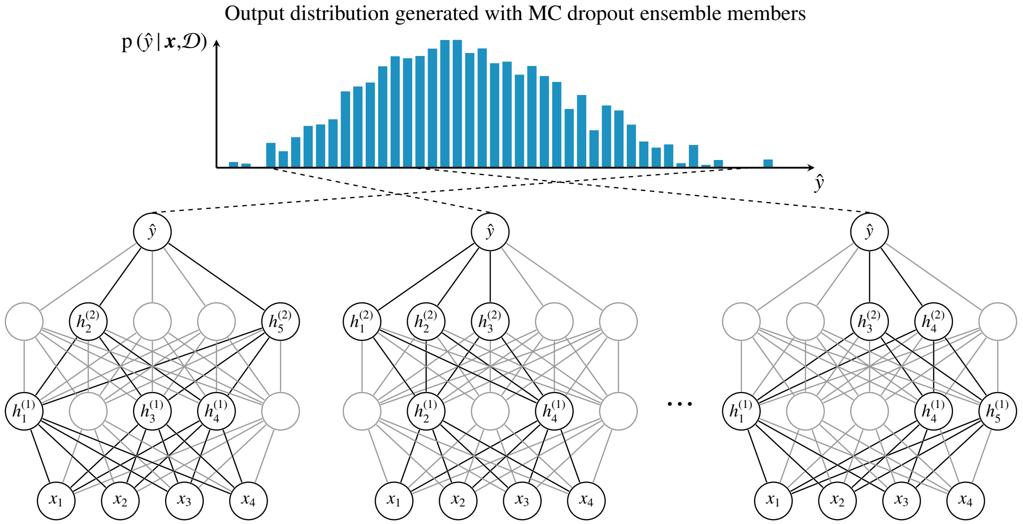


Figure 4.11: The principle behind MC dropout. During inference, nodes are also dropped, leading to a different structure for each forecast of the ensemble.

Gal et al. proved[22] in Ref. [83] that each dropout member corresponds to a sample of the approximated parametric posterior distribution $q(\theta \,|\, \mathcal{D})$, leading to:

$$\hat{\boldsymbol{\theta}}_i \sim q(\theta \,|\, \mathcal{D}) \approx \mathrm{p}(\theta \,|\, \mathcal{D}). \tag{4.14}$$

A sufficient number of MC sample parameters thus approximates $\mathrm{p}(\theta \,|\, \mathcal{D})$, which concludes that the generated ensemble members represent the effects of the epistemic uncertainty on the forecast.

As a result, MC dropout is often combined with other methods such as a mean and variance [131, 146, 157] or quantile [126, 186] estimation at the model output to represent the overall uncertainty. For instance, it has been successfully applied in the past in load forecasting [38, 146, 186], weather forecasting [131], wind power forecasting [157] or in other practical areas including e.g., time series forecasting at Uber [233]. In the field of solar forecasting it has been used only sporadically. Ref. [126] used it in combination with quantile regression for the prediction of an 80 % prediction interval. Ref. [208] in turn estimated with it a 95 % prediction interval for a one step ahead PV power forecast. However, neither reference analyzed the accuracy over the entire PDF, e.g., using multiple quantiles, examined the combination with multiple approaches, or evaluated the impact on multi-step prediction.

---

[22] The authors proved that as long as dropout is applied in each hidden layer, the distribution of the dropout configuration is equivalent to a deep Gaussian process that estimates $\mathrm{p}(\theta \,|\, \mathcal{D})$ by minimizing the Kullback-Leibler divergence.

In this thesis, MC dropout is combined with mean and variance estimation, with mixture density networks (MDNs), as well as with quantiles, and is also compared with other representations methods for the epistemic uncertainty. Nevertheless, the focus in this segment is on using MC dropout while maintaining an ensemble uncertainty representation and simultaneously accounting for the overall uncertainty. For this purpose, post-processing using output calibration is carried out in the following.

Output calibration

The output calibration of a derived predictive uncertainty via postprocessing is predominantly carried out using a held out calibration data set [85]. Under the assumption, that the omitted data set represents the same uncertainty distribution as the underlying process, one can use it to adjust the uncertainty of the Monte Carlo ensemble to depict also the aleatoric uncertainty.

In this thesis the method called "calibrating regression uncertainty distributions empirically" (CRUDE) is being applied. In Ref. [227] it showed better better results in comparison to other output processing methods in combination with MC dropout and is particularly suited for non Gaussian distributions, as no parametric form for the uncertainty is assumed. The general assumption thereby is that the generated MC dropout distribution can be linearly scaled to generate the sought after underlying probabilistic distribution. This principle is also consistent with the calibration approach proposed by Yarin Gal et al. in the appendix to their paper on MC dropout [83].

First, for each ensemble member $m \in [1, M]$, at each forecast time t of the calibration data set $\mathcal{D}_{\mathrm{cal}}$, the respective z-score

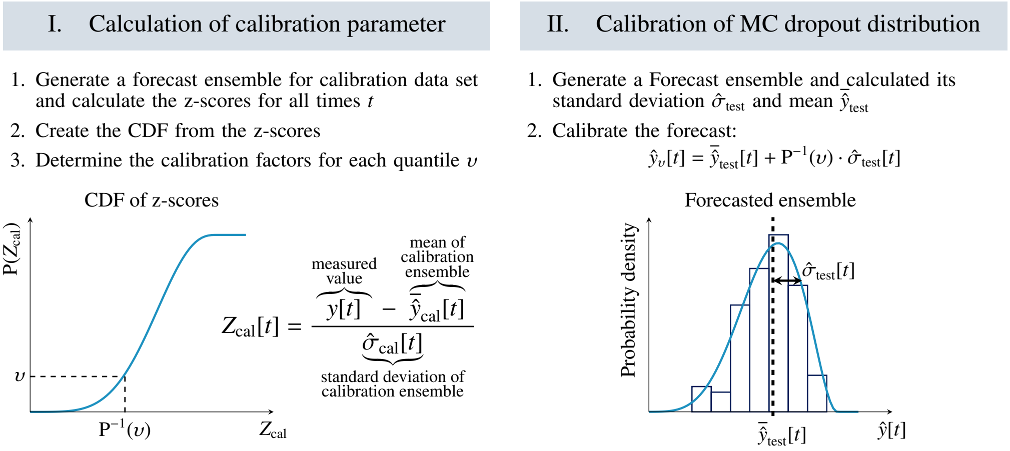
$$Z_{\mathrm{cal}}[t] = \frac{y[t] - \bar{\hat{y}}_{\mathrm{cal}}[t]}{\hat{\sigma}_{\mathrm{cal}}[t]} \tag{4.15a}$$

$$\text{with} \quad \hat{\sigma}_{\mathrm{cal}}[t] = \frac{1}{M}\sum_{m=1}^{M}\left(\hat{y}_{\mathrm{cal},m}[t] - \bar{\hat{y}}_{\mathrm{cal}}[t]\right)^2, \tag{4.15b}$$

$$\text{and} \quad \bar{\hat{y}}_{\mathrm{cal}}[t] = \frac{1}{M}\sum_{m=1}^{M}\hat{y}_{\mathrm{cal},m}[t], \tag{4.15c}$$

is calculated, which basically describes how many standard deviations $\hat{\sigma}_{\mathrm{cal}} \in \mathbb{R}_{\geq 0}$ the observation $y \in \mathbb{R}$ is away from the forecasted mean $\bar{\hat{y}}_{\mathrm{cal}} \in \mathbb{R}$ of the ensemble [226]. Afterwards, the CDF $\mathrm{P}(Z_{\mathrm{cal}})$ is formed over all considered z-scores of the calibration data set (see also Figure 4.12). With its inverse, the percentile function $\mathrm{P}^{-1}(\upsilon)$, the estimated output power for the probability $\upsilon \in [\frac{1}{M}, \frac{2}{M}, \ldots, 1]$ can be calculated as follows:

$$\hat{y}_{\upsilon}[t] = \bar{\hat{y}}_{\mathrm{test}}[t] + \mathrm{P}^{-1}(\upsilon) \cdot \hat{\sigma}_{\mathrm{test}}[t], \tag{4.16}$$



I. Calculation of calibration parameter

1. Generate a forecast ensemble for calibration data set and calculate the z-scores for all times t
2. Create the CDF from the z-scores
3. Determine the calibration factors for each quantile υ

II. Calibration of MC dropout distribution

1. Generate a Forecast ensemble and calculated its standard deviation $\hat{\sigma}_{\text{test}}$ and mean $\bar{\hat{y}}_{\text{test}}$
2. Calibrate the forecast:
$$\hat{y}_{\upsilon}[t] = \bar{\hat{y}}_{\text{test}}[t] + \mathrm{P}^{-1}(\upsilon) \cdot \hat{\sigma}_{\text{test}}[t]$$

$$Z_{\text{cal}}[t] = \frac{\underbrace{y[t]}_{\text{measured value}} - \underbrace{\bar{\hat{y}}_{\text{cal}}[t]}_{\text{mean of calibration ensemble}}}{\underbrace{\hat{\sigma}_{\text{cal}}[t]}_{\text{standard deviation of calibration ensemble}}}$$

Figure 4.12: Illustration of the CRUDE steps involved in the calibration process.

whereby $\hat{\sigma}_{\text{test}}[t] \in \mathbb{R}_{\geq 0}$ is the standard deviation and $\bar{\hat{y}}_{\text{test}}[t] \in \mathbb{R}$ the mean of the forecasted ensemble of the test data set [226, 227]. Essentially, CRUDE utilizes the calibration data to determine the deviation of individual probabilities from the average and subsequently scale the standard deviation of the forecast accordingly.

As the aleatoric uncertainty of the data may increase with higher values, the CRUDE z-scores will be distinguished in this thesis for every hour of the day. This enables a different calibration of the distributions for each hour and also adapts the same approach as the binning of the residual pools of the previously presented bootstrapping method. It is also in line with the suggestions of Yarin Gal et al. who recommend to compensate the influence of the data magnitude when calibrating MC dropout [83]. In addition, an hourly discrimination has shown better results in initial studies.

Simulation setup

Table 4.1 summarizes the specific parameters used for MC dropout in addition to the hyperparameters already described in Section 4.1.2 for the MLP structure. To analyze the influence of the dropout factor as well as the number of ensemble members, these two parameters are varied in the simulation.

Creating the individual training data sets is not straightforward, as a calibration data set is also required. 20 % of the data are used only for calibration and 20 % of the data are used for both validation and calibration.[23]. In this process, complete days of the regression formulation are sampled to make sure that all hours of the day are available and all data

---

[23] In general the calibration data set should be completely independent of the training data set [227]. However, especially with the small number of days studied, and the distinction made between the individual hour slots during the day, it is not feasible to separate the calibration and validation data sets. The influence of the validation set on the training is also comparatively small as it is only used for the determination of the best model during early stopping.

Table 4.1: Used hyperparameter for the MC dropout methd with CRUDE postprocessing. For a specification of the other hyperparameter see also Section 4.1.2.

Hyperparameter	
Number of hidden layers	3
Number of units per hidden layer	50
Batch size	32
Validation split	20 %
Calibration split	40 %
Dropout factor	[0.1, 0.3, 0.5, 0.7, 0.9]
Amount of MC dropout ensemble members	[50, 100, 500]

sets have homogeneous distribution of the times of the day. In addition, the calibration data are predominantly sampled from temporally more recent data. This ensures that for a very large amount of available training data, the sun positions of the individual hour slots of the calibration data are very similar to the forecasting time period. A more detailed description of how the data sets are created is summarized in Figure 4.13.

Strictly speaking, the calibration and the determination of the calibration parameters are also subject to epistemic uncertainty. Given the limited amount of training data, this epistemic could have a significant effect on the overall forecasting quality. In order to evaluate this influence on the calibration, five different sampled versions of the individual data sets are also tested for each training initialization. This corresponds to a bootstrapping of the respective training data for the forecasting model and for the calibration model.

Generation of the data sets for model training

Notation information: The overall available data set for the training is $\mathcal{D}$ while $\mathcal{D}_{\text{poss cal}}$ and $\mathcal{D}_{\text{poss val}}$ respectively represent the subsets of possible input/output pairs out of which the calibration and validation data sets can be sampled (see also illustration on the right).

(I) Randomly select 20% of the total available data from the given range to be used exclusively for calibration to generate $\mathcal{D}_{\text{cal excl}} \in \mathcal{D}_{\text{poss cal}}$

(II) Randomly select 20% of the total available data from the given range to be used for validation to generate $\mathcal{D}_{\text{val}} \in \mathcal{D}_{\text{poss val}}$
Data selected exclusively for calibration in the previous step cannot be included in the sample ($\mathcal{D}_{\text{val}} \cap \mathcal{D}_{\text{cal excl}} = 0$).

(III) Assign the overall calibration data by $\mathcal{D}_{\text{cal}} = \mathcal{D}_{\text{cal excl}} \cup \mathcal{D}_{\text{val}}$

(IV) Generate the training data set by $\mathcal{D}_{\text{train}} = \mathcal{D} - \mathcal{D}_{\text{cal}}$

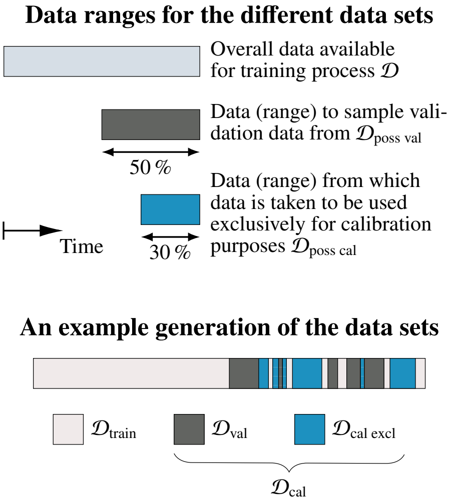

Data ranges for the different data sets
Overall data available for training process $\mathcal{D}$
Data (range) to sample validation data from $\mathcal{D}_{\text{poss val}}$
50 %
Data (range) from which data is taken to be used exclusively for calibration purposes $\mathcal{D}_{\text{poss cal}}$
Time
30 %
An example generation of the data sets
$\mathcal{D}_{\text{train}}$
$\mathcal{D}_{\text{val}}$
$\mathcal{D}_{\text{cal excl}}$
$\mathcal{D}_{\text{cal}}$

Figure 4.13: Specification of the procedure used to generate the different data sets (training, validation and calibration) for the model training.

4.3 Probabilistic representation using quantiles

4.3.1 Pinball loss function

The predominant approach to generate a quantile representation for time series models is to adjust the loss function for the model training. While in classical linear regression the parameters are estimated by minimizing the MSE of the residuals, in quantile regression an asymmetric error function is minimized. A corresponding loss function for achieving this is the so-called pinball loss:

$$\mathrm{PL}_\upsilon(\hat{y}_\upsilon, y) = \begin{cases} \upsilon \cdot |\hat{y}_\upsilon - y|, & \hat{y}_\upsilon \leq y \\ (1-\upsilon) \cdot |\hat{y}_\upsilon - y|, & \hat{y}_\upsilon > y \end{cases}, \tag{4.17}$$

whereby $\upsilon \in [0, 1]$ denotes the quantile and $\hat{y}_\upsilon \in \mathbb{R}$ the respective estimated output for it. As can be seen in Figure 4.14 the lowest error for the pinball loss still occurs, when the forecast matches the measurements and the resulting residual is zero. Nevertheless, depending on the respective quantile the over- and underpredictions are weighted differently. For the illustrated quantile 0.1, for instance, the pinball loss is nine times lower for underprediction than for overprediction, if the absolute residuals are equal.

For each quantile an extra model with a corresponding set of parameters is commonly estimated. This can lead to the effect of quantile crossing, whereby $\hat{y}_{\upsilon_1} > \hat{y}_{\upsilon_2}$ although $\upsilon_1 < \upsilon_2$. Since a CDF should increase monotonicall with υ, quantile crossing is compensated in this thesis by rearranging the respective quantiles as recommended by Ref. [39].

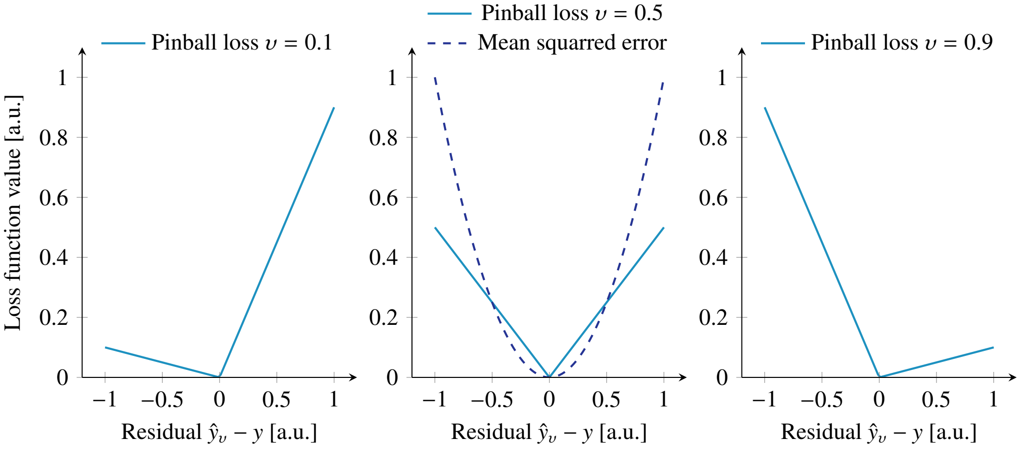


Figure 4.14: Pinball loss for different quantiles. In contrast to the mean square error cost function, the values of the pinball loss increase linearly with the residuals. The origin of the name is easy to see, as the course of the cost function resembles the reflection of a pinball.

4.3.2 Quantile regression with the ARX model

There are different approaches to combine quantile regression with time series models. In Ref. [48] and [161] the deterministic output of an ARMA model was used and afterwards linearly scaled with one parameter

$$\hat{y}_\upsilon[t] = \theta_\upsilon \cdot \hat{y}[t], \tag{4.18}$$

to generate the respective quantile forecasts for solar irradiation and PV power. However, this approach has the disadvantage that additional influences are not taken into account when determining the quantile and, thus, e.g., a remaining heteroscedasticity of the residuals or uncertainty can not be considered. In Ref. [123], the authors were able to achieve better results in a direct comparison for the quantile forecast of solar irradiation by applying the pinball loss directly to the time series model instead. This is also the approach proposed but not evaluated in Ref. [13] for PV power forecasts.

Accordingly, this approach is adopted in this thesis. The entire procedure is summarized in Algorithm 6. In contrast to the referenced sources, the determination of the model order is conducted automatically and without continuous consideration of the lags, using the developed greedy search algorithm.

ALGORITHM 6: Quantile regression with consideration of the epistemic uncertainty

```
Preliminary steps: Generate the training data set $\mathcal{D} = \{\boldsymbol{X}_n, \boldsymbol{y}_n\}_{n=1}^{N}$ from the
  stationarized time series in the form of a regression problem (see (4.5))
for ensemble member $m$ in $M$ :
    $\mathcal{D}_m^* := \{\}$                                         // initialization
    while $\#(\mathcal{D}_m^*) < \#(\mathcal{D})$ :
        Sample a block of six consecutive input-output cases $\{\boldsymbol{X}_n, \boldsymbol{y}_n\}_n^{n+6}$, with
          $n \in [1, N-5]$ from $\mathcal{D}$ and append it to the bootstrapped training data set $\mathcal{D}_m^*$
    Ensure $\#(\mathcal{D}_m^*) = \#(\mathcal{D})$ by deleting surplus elements      // necessary, if $N \bmod 6 \neq 0$
    Determine the best model structure with the the developed greedy search algorithm
      using cross validation (see Algorithm 2)
    for quantile $\upsilon$ in $[0.1, 0.2, \dots, 0.9]$ :
        Estimate the model parameter $\hat{\boldsymbol{\theta}}_{m,\upsilon}$ for the determined model structure by
          solving the nonlinear optimization problem:
          $\hat{\boldsymbol{\theta}}_{m,\upsilon} = \arg\min_{\boldsymbol{\theta}_{m,\upsilon}} \left( \sum_{n=1}^{N} \mathrm{PL}_\upsilon(\hat{\boldsymbol{y}}_\upsilon, \boldsymbol{y}) \right)$.
        Generate a forecast for the ensemble member $\hat{y}_{m,\upsilon} = \boldsymbol{X}_\mathrm{f} \cdot \hat{\boldsymbol{\theta}}_{m,\upsilon}$ with the
          respective forecast input $\boldsymbol{X}_\mathrm{f}$
for quantile $\upsilon$ in $[0.1, 0.2, \dots, 0.9]$ :
    Calculate the overall quantile forecast and compensate epistemic uncertainty by
      $\hat{y}_\upsilon = \frac{1}{M} \sum_{m=1}^{M} \hat{y}_{m,\upsilon}$
```

Quantile regression only considers aleatoric, as only at the model output the distribution is modeled. Consequently, bootstrapping of the training data is applied, which leads to different model orders as well as parameter sets per ensemble (see also Section 4.2.1). The ensemble members are subsequently averaged based on the assumption that:

$$\mathrm{p}(\hat{y} \,|\, \boldsymbol{x}, \mathcal{D}) = \int \mathrm{p}(y \,|\, \boldsymbol{x}, \theta)\, \mathrm{p}(\theta \,|\, \mathcal{D})\, \mathrm{d}\theta \approx \frac{1}{M} \sum_{i=1}^{M} \mathrm{p}\Big(\hat{y} \,|\, \boldsymbol{x}, \hat{\boldsymbol{\theta}}_i\Big), \qquad \hat{\boldsymbol{\theta}}_i \sim \mathrm{p}(\theta \,|\, \mathcal{D}), \tag{4.19}$$

with $\mathrm{p}(\theta \,|\, \mathcal{D})$ being in this case represented by the the respective quantile forecasts.

4.3.3 Quantile neural network

In the quantile neural network, the pinball loss is used for the cost function and averaged over all lags of the forecast horizon. This approach has been successfully used in the context of solar forecasting [65, 126, 132, 139] as well as general energy time series forecasting [223, 224, 232].

In this thesis, a separate network is generated and trained for each quantile. Alternatively, all quantiles could also be determined with a single neural network. However, given the present use case with a forecast horizon of six hours, a sampling rate of 15 minutes, and the quantiles to be determined ranging from 10 % to 90 %, this would result in 216 outputs for the neural network. As the total cost function value would be the average of the pinball loss over all outputs, local minima could be reached during training which yield very good results for a large fraction of the outputs, while neglecting individual lags or quantiles.

In order to take epistemic into account, MC dropout is applied in this thesis. Analogous to the approach in the ARX model, the ensemble members are subsequently averaged to obtain the final forecast for the individual quantiles. To the best of the author's knowledge, the consideration of epistemics with quantile neural networks was only performed in [126]. There, an LSTM model was used to estimate PV power for the next hour at hourly resolution. Even with more than 200 days of training data, it was able to improve sharpness by 33 % and reliability by 4 %. However, the evaluation was done only on a very small test size (≈5 days), for one location and only for the 10 % to 90 % interval. Hence, further investigations are necessary to enable more systematic assessments.

An illustration of the used quantile neural network can be seen in Figure 4.15 and an overview of its specific hyperparameters can be found in Table 4.2.

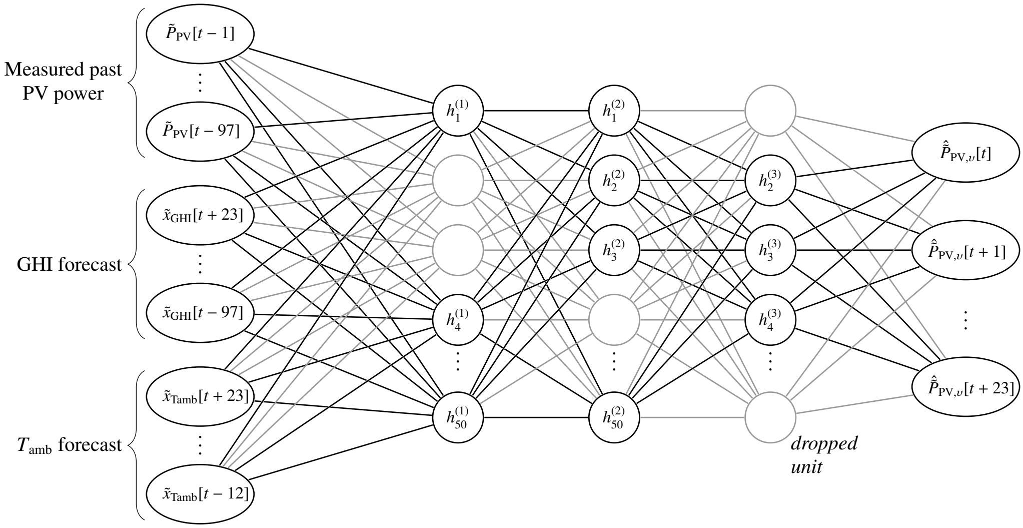

Measured past PV power
$\tilde{P}_{\mathrm{PV}}[t-1]$
$\tilde{P}_{\mathrm{PV}}[t-97]$
GHI forecast
$\tilde{x}_{\mathrm{GHI}}[t+23]$
$\tilde{x}_{\mathrm{GHI}}[t-97]$
T_{amb} forecast
$\tilde{x}_{\mathrm{Tamb}}[t+23]$
$\tilde{x}_{\mathrm{Tamb}}[t-12]$
$h_1^{(1)}$
$h_4^{(1)}$
$h_{50}^{(1)}$
$h_1^{(2)}$
$h_2^{(2)}$
$h_3^{(2)}$
$h_{50}^{(2)}$
$h_2^{(3)}$
$h_3^{(3)}$
$h_4^{(3)}$
dropped unit
$\hat{\tilde{P}}_{\mathrm{PV},v}[t]$
$\hat{\tilde{P}}_{\mathrm{PV},v}[t+1]$
$\hat{\tilde{P}}_{\mathrm{PV},v}[t+23]$

Figure 4.15: Architecture of the used quantile neural network with dropout. For the estimation of each quantile a different model was trained.

Table 4.2: Used hyperparameter for the quantile neural network. For a specification of the other hyperparameter see also Section 4.1.2.

Hyperparameter	
Number of hidden layers	3
Number of units per hidden layer	50
Batch size	32
Validation split	30 %
Dropout factor	0.3
Amount of MC dropout ensemble members	[1, 50, 100, 500]

4.4 Probabilistic representation using a continuous probability distribution

4.4.1 GARCH model in combination with the ARX model

The Generalized AutoRegressive Conditional Heteroscedasticity (GARCH) model is a volatility model developed by Tim Bollerslev in 1986. It has the following basic structure:

$$y[t] = \mu[t] + \varepsilon[t] \tag{4.20a}$$

$$\varepsilon[t] = \sigma[t] \cdot \chi[t] \tag{4.20b}$$

$$\sigma^2[t] = \theta_{\mathrm{res},0} + \sum_{j=1}^{D_{\mathrm{res}}} \theta_{\mathrm{res},j} \cdot \varepsilon^2[t - j \cdot T] + \sum_{i=1}^{D_{\mathrm{v}}} \theta_{\mathrm{v},i} \cdot \sigma^2[t - i \cdot T], \tag{4.20c}$$

where $\mu[t] \in \mathbb{R}$ is the mean model, $\varepsilon[t] \in \mathbb{R}$ is the volatility process also labelled as mean model residuals and $y[t] \in \mathbb{R}$ is the output signal [19]. The underlying principle is that the model used to estimate the mean – in our case the ARX model – is not able to represent the respective volatility around the average given its model structure and cost function. Instead, an additional volatility model is used (see also figure 4.16). For the volatility process, an independent, identically distributed random variable $\chi[t]$ with $\mathbb{E}[\chi] = 0 \wedge \mathrm{Var}(\chi) = 1$ is scaled by an estimated conditional standard deviation $\sigma[t] \in \mathbb{R}_{\geq 0}$. The corresponding conditional variance $\sigma^2[t] \in \mathbb{R}_{\geq 0}$ is in turn defined as a linear combination of past values of the conditional variance and past values of the model residuals, where $\boldsymbol{\theta}_{\mathrm{v}} \in \mathbb{R}^{D_{\mathrm{v}}}$ and $\boldsymbol{\theta}_{\mathrm{res}} \in \mathbb{R}^{D_{\mathrm{res}}}$ are the weighting factors, and $D_{\mathrm{v}} \in \mathbb{Z}_{\geq 0}$ and $D_{\mathrm{res}} \in \mathbb{Z}_{\geq 0}$ are the considered model order.

GARCH models are predominantly used in economics to estimate the volatility of asset prices, stock returns, indices, and currencies [78]. However, in the domain of probabilistic energy time series forecasting, it has also been applied to wind power [37], electricity prices [129], and electrical load [35, 36, 222].

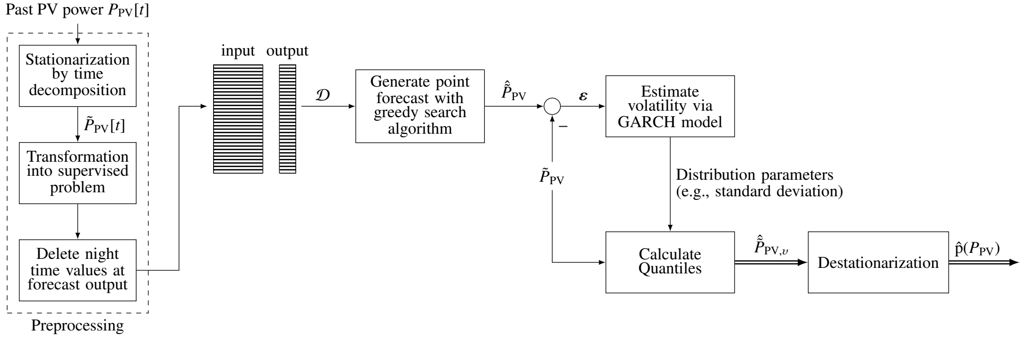

Past PV power $P_{\mathrm{PV}}[t]$
Stationarization by time decomposition
$\tilde{P}_{\mathrm{PV}}[t]$
Transformation into supervised problem
Delete night time values at forecast output
Preprocessing
input
output
$\mathcal{D}$
Generate point forecast with greedy search algorithm
$\hat{\tilde{P}}_{\mathrm{PV}}$
ε
−
$\tilde{P}_{\mathrm{PV}}$
Estimate volatility via GARCH model
Distribution parameters (e.g., standard deviation)
Calculate Quantiles
$\hat{\tilde{P}}_{\mathrm{PV},\upsilon}$
Destationarization
$\hat{\mathrm{p}}(P_{\mathrm{PV}})$

Figure 4.16: Principle of the GARCH model in combination with the ARX model.

In the field of solar forecasting, it has been applied by David et. al. in Ref. [47] and Ref [48] for the probabilistic forecasting of solar irradiance. In these studies, however, the accuracy of the GARCH approach was lower than that of quantile regression. This was attributed to the fact that the model error at hand was not symmetrically distributed and had a larger kurtosis than the assumed Gaussian distribution [48]. However, this may be caused by the fact that only autoregressive information was considered in the generation of the deterministic forecasts, which means that the used ARMA model was insufficient to capture the behavior of the underlying process. In Ref. [87] the GARCH model for the probabilistic prediction of PV power was slightly modified to emphasize the variance residuals. However, as the evaluation was only performed using a 99 % prediction interval, the results do not provide enough insight into the quality for an entire distribution function.

The estimation of the GARCH parameters in this thesis is performed by maximum likelihood estimation. Thereby, the parameters are determined which maximize the likelihood of observing the measured data given a selected statistical model. For time series $y[t]$ with N independent data points, the likelihood function is the product of the probability density functions of all observed values of the time series. Assuming that the density function p_χ of the random noise χ is known (4.20b), and given that the area under a density function must remain equal to one, the density of ε is [78]:

$$\mathrm{p}_\varepsilon(\varepsilon[t]) = \frac{1}{\sigma[t]}\,\mathrm{p}_\chi\!\left(\frac{\varepsilon[t]}{\sigma[t]}\right). \tag{4.21}$$

Hence, the likelihood is given by:

$$\ell(\boldsymbol{\theta}_\mathrm{v}, \boldsymbol{\theta}_\mathrm{res} | \varepsilon[t]) = \prod_{t=1}^{N} \frac{1}{\sigma[t]}\,\mathrm{p}_\chi\!\left(\frac{\varepsilon[t]}{\sigma[t]}\right), \tag{4.22}$$

with σ being recursively defined by (4.20c). For a more stable estimation of the parameters, especially for individual small probabilities, a log transform is commonly applied to the likelihood combined with a change of sign, leading to the minimization of the negative sum of the a density functions called negative log likelihood:

$$\hat{\boldsymbol{\theta}}_\mathrm{v}, \hat{\boldsymbol{\theta}}_\mathrm{res} = \arg\min_{\boldsymbol{\theta}_\mathrm{v},\boldsymbol{\theta}_\mathrm{res}} -\!\left(\sum_{t=1}^{N} \log \frac{1}{\sigma[t]}\,\mathrm{p}_\chi\!\left(\frac{\varepsilon[t]}{\sigma[t]}\right)\right). \tag{4.23}$$

In addition to the commonly used Gaussian distribution, a skewed t-distribution is used in this thesis in order to be able to better represent non-symmetric distributions. The Gaussian probability density function of the signal x is defined as

$$\mathrm{p}(x, \sigma, \mu) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\!\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{4.24}$$

with $\mu \in \mathbb{R}$ being the mean and $\sigma \in \mathbb{R}_+$ the standard deviation [78]. The skewed t distribution was introduced by Handsen in 1994 to the GARCH model family [92]. Its density function is [92]:

$$\mathrm{p}(x, \sigma, \nu, \lambda) = \zeta_2 \zeta_3 \left(1 + \frac{1}{\nu - 2} \left(\frac{\zeta_1 + \zeta_2 x / \sigma}{1 + \mathrm{sgn}\left(\frac{x}{\sigma} + \frac{\zeta_1}{\zeta_2}\right) \lambda} \right)^2 \right)^{-(\nu+1)/2}, \tag{4.25a}$$

$$\text{with} \quad \zeta_1 = 4\lambda c \left(\frac{\nu - 2}{\nu - 1} \right), \tag{4.25b}$$

$$\text{and} \quad \zeta_2 = \sqrt{1 + 3\lambda^2 - \zeta_1^2}, \tag{4.25c}$$

$$\text{and} \quad \zeta_3 = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\pi(\nu - 2)}\Gamma\left(\frac{\nu}{2}\right)}, \tag{4.25d}$$

whereby the two shape parameters $\nu \in \mathbb{R}$ and $\lambda \in \mathbb{R}$ control the kurtosis and skewness, respectively, and Γ denotes the gamma function. An overview of the skewed t-distribution with different parameters for gamma and lambda in comparison to the Gaussian distribution can be seen in Figure 4.17. The different parameters of the distribution functions are also estimated during the maximum likelihood estimation. The percentile function of the distribution and the mean prediction of the ARX model can subsequently be used to calculate the PV power values for the different quantiles.

For the implementation of the GARCH model, the Python package “arch 6.1.0” is used [187]. In order to also represent the epistemic, the training data of the ARX model is bootstrapped, which in turn results in a varying ARX model structure and mean model residuals. These different residuals serve as training data to estimate the different GARCH models. Analogous to the quantile regression, the calculated quantiles of the GARCH ensemble members are subsequently averaged. The model order is the same as in Ref. [47] and Ref. [48] with $D_\mathrm{v} = 1 \wedge D_\mathrm{res} = 1$, since it also yielded the best results in preliminary analysis.

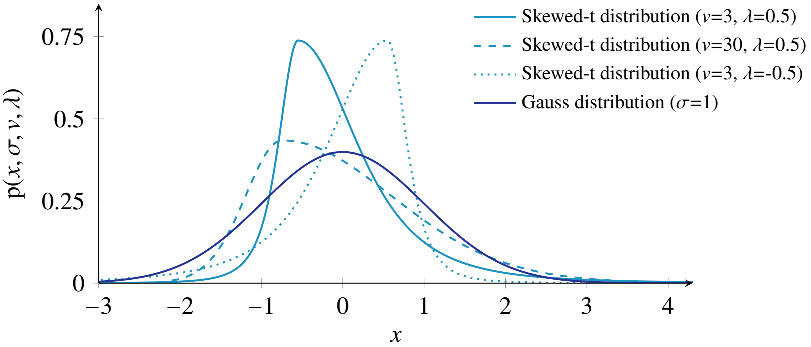


Figure 4.17: Skewed-t distribution with different parameters sets. ν changes the kurtosis and λ the skewness.

4.4.2 Mixture density network

MDNs were initially introduced by Bishop in [17] as a means of estimating general distribution functions. For this purpose, the conditional probability distribution $\mathrm{p}(y|\mathbf{x})$ of the target variable y given the input features $\mathbf{x}$ is represented as a linear combination of kernel functions $\Psi_k(y|\mathbf{x})$:

$$\mathrm{p}(y|\mathbf{x}) = \sum_{k=1}^{K} \varphi_k(\mathbf{x}) \Psi_k(y|\mathbf{x}), \tag{4.26a}$$

$$\text{with:} \quad \sum_{k=1}^{K} \varphi_k(\mathbf{x}) = 1 \tag{4.26b}$$

where $K \in \mathbb{N}$ is the amount and $k \in [1, K]$ is the respective number of the considered components in the mixture model. Furthermore $\varphi_k(\mathbf{x}) \in \mathbb{R}$ constitutes the weighting of the respective mixture component also called mixing coefficient. In this work Gaussian kernels are used, since a neural network with a sufficient number of hidden units and a mixture model with a sufficient number of kernel functions can theoretically approximate any conditional density function [17]. Consequently $\Psi_k(y|\mathbf{x})$ is formulated as follows:

$$\Psi_k(y|\mathbf{x}) = \frac{1}{\sqrt{2\pi\sigma_k^2(\mathbf{x})}} \exp\left(-\frac{(y - \mu_k(\mathbf{x}))^2}{2\sigma_k(\mathbf{x})^2} \right) \tag{4.27}$$

with $\mu_k(\mathbf{x}) \in \mathbb{R}$ as the mean and $\sigma_k(\mathbf{x}) \in \mathbb{R}_{\geq 0}$ as the variance of the kth mixture component. To estimate the neural network parameter the negative log likelihood is used as minimization objective $\mathcal{L}$. This results together with (4.26) and (4.27) in:

$$\begin{aligned} \mathcal{L} &= -\log(\mathrm{p}(y|\mathbf{x})) \\ &= -\log\left(\sum_{k=1}^{K} \frac{\varphi_k(\mathbf{x})}{\sqrt{2\pi\sigma_k^2(\mathbf{x})}} \exp\left(-\frac{(y - \mu_k(\mathbf{x}))^2}{2\sigma_k(\mathbf{x})^2} \right) \right). \end{aligned} \tag{4.28}$$

For a better understanding of the applied MDN, its basic structure as well as exemplary estimated distributions are depicted in Figure 4.18.

In practice, several measures must be implemented to guarantee that, on the one hand, the Gaussian parameters comply with their mathematical constraints and, on the other hand, no numerical instability occurs during training. The formulation in (4.28) is mathematically ill-conditioned, as exponentiating small values can lead to a numerical underflow[24], while logarithmizing small values can lead to a numerical overflow. Given the negative sign, this

---

[24] An underflow occurs when an arithmetic operation attempts to produce a numeric value whose absolute value is smaller than the range that can be represented by a given number of digits.

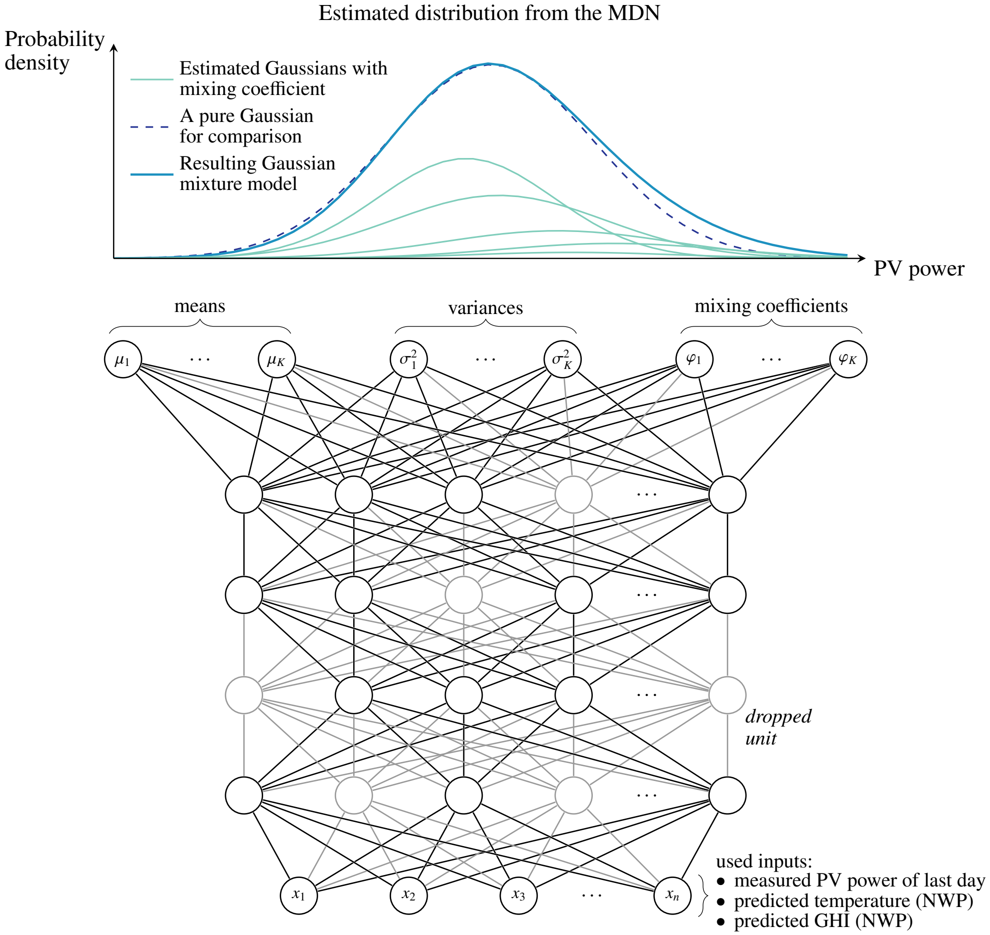

Estimated distribution from the MDN
Probability density
Estimated Gaussians with mixing coefficient
A pure Gaussian for comparison
Resulting Gaussian mixture model
PV power
means
variances
mixing coefficients
μ_1
μ_K
σ_1^2
σ_K^2
φ_1
φ_K
dropped unit
x_1
x_2
x_3
x_n
used inputs:
• measured PV power of last day
• predicted temperature (NWP)
• predicted GHI (NWP)

Figure 4.18: Principle representation of the employed mixture density network (MDN) in combination with dropout in the hidden layers. The weighted combination of the different Gaussian distributions allows to estimate non-Gaussian distribution functions.

situation may occur, for instance, when the estimated variances during the model training are relatively small. To mitigate this effect, the so-called log-sum-exp trick [18]

$$\log \sum_{i=1}^{N} \exp(\zeta_i) = \max_j \zeta_j + \log \sum_{i=1}^{N} \exp\Big(\zeta_i - \max_j \zeta_j\Big), \forall\, N \in \mathbb{N}, j \in [1, N], \tag{4.29}$$

is adopted, whereby $\zeta_i \in \mathbb{R}$ denote arbitrary values. By rearranging the maximum value outside the logarithmic and exponential terms, an approximation for the optimization is obtained, even if an underflow would occur in all summands. Moreover, the sum of the exponents ≥ 1, even if underflow occurs, since at least one of the terms is exp(0). Consequently, there are no numerical issue with the use of the logarithm.

For this purpose the exponential function within (4.28) is reformulated as follows:

$$\mathcal{L} = -\log\left(\sum_{k=1}^{K}\exp\left(\log(\varphi_k(\mathbf{x})) - \overbrace{\frac{1}{2}\log(2\pi)}^{\text{constant}} - \frac{1}{2}\log\Big(\sigma_k^2(\boldsymbol{x})\Big) - \frac{(y-\mu_k(\mathbf{x}))^2}{2\sigma_k(\mathbf{x})^2}.\right)\right), \tag{4.30}$$

leading in combination with (4.29) to:

$$\begin{aligned} \mathcal{L} &= -\log\left(\sum_{k=1}^{K}\exp\Big(\zeta_k - \max_j \zeta_j\Big)\right) - \max_j \zeta_j, \quad \forall j \in [1, K], \\ \text{with} \quad \zeta_k &= \log(\varphi_k(\mathbf{x})) - \frac{1}{2}\log(2\pi) - \frac{1}{2}\log\Big(\sigma_k^2(\boldsymbol{x})\Big) - \frac{(y-\mu_k(\mathbf{x}))^2}{2\sigma_k(\mathbf{x})^2}. \end{aligned} \tag{4.31}$$

For the mixing coefficients $\varphi_k(\mathbf{x})$ a softmax activation function is used, since they must sum to unity (see (4.26)). In addition, clipping $\Big(\varphi_k \in [1 \times 10^{-12}, 1], \forall k \in [1, K]\Big)$ is performed beforehand to guarantee that their values are positive and for numerical stability purposes not too small. To ensure a positive variance Bishop [17] initially suggested an exponential activation function. However, as these can get unstable for large values a softplus function with an additional constant minimum variance term was added instead. This approach was also used in [122] for multiple regression tasks with a deep ensemble and only a single Gaussian distribution.

Consequently, the output layer of the neural network corresponds to a parameter vector $[h_{\mu_k}, h_{\sigma_k^2}, h_{\varphi_k}]^T, \forall k \in [1, K]$, which must be post-processed to get the parameters of the Gaussian mixture model (GMM) model for both the loss function and the forecast, as follows:

$$\mu_k = h_{\mu_k} \tag{4.32a}$$

$$\sigma_k^2 = \log\Big(1 + \exp\Big(h_{\sigma_k^2}\Big)\Big) + 1 \times 10^{-6}, \tag{4.32b}$$

$$\varphi_k = \frac{\exp\Big(\zeta_{\varphi_k}\Big)}{\sum_{j=1}^{K}\exp\Big(\zeta_{\alpha_j}\Big)}, \tag{4.32c}$$

$$\text{with:} \quad \zeta_{\varphi_k} = \begin{cases} 1 \times 10^{-12}, & \text{if } h_{\varphi_k} \leq 1 \times 10^{-12} \\ 1, & \text{if } h_{\varphi_k} \geq 1 \\ h_{\varphi_k}, & \text{otherwise} \end{cases}. \tag{4.32d}$$

MDNs only depict aleatoric uncertainty, as e.g., the uncertainty of the estimated model parameters is not considered. Analogous to the previous methods, members with different estimated parameters are generated for the consideration of the epistemic.

This means for the present approach that the conditional distributions are combined in a higher-level mixture model. Taking (4.26) and (4.27) into account, the overall distribution for the approach can thus be determined as follows:

$$
\begin{aligned}
\mathrm{p}(y|\mathbf{x}) &= M^{-1} \sum_{m=1}^{M} p_{\theta_m}(y|\mathbf{x}, \theta_m) \\
&= \sum_{m=1}^{M} \sum_{k=1}^{K} \frac{\varphi_{k,m}(\mathbf{x})}{M\sqrt{2\pi\sigma_{k,m}^2(\mathbf{x})}} \exp\left(-\frac{(y-\mu_{k,m}(\mathbf{x}))^2}{2\sigma_{k,m}(\mathbf{x})^2}\right),
\end{aligned}
\tag{4.33}
$$

where $M \in \mathbb{N}$ is the overall amount of ensemble members and $m \in [1, M]$ indicates the respective ensemble member number. As can be seen in (4.33), the ensembles also influence the shape of the distribution function. Since the overall amount of distribution is K times M, the density flexibility increases. For instance, even with only one distribution used at the output ($K = 1$) and multiple ensembles ($M > 1$), the result is a GMM. It should be noted, however, that with well-chosen hyperparameters, the distributions between the generated ensemble members are expected to differ less than the generated output distributions within a single forecast.

For the generation of ensemble members, this thesis compares two different approaches, MC dropout and multiple network initialization. The latter captures the deviation from the different local minima of the objective function (4.31) reached during model training. Compared to MC Dropout, better results were obtained with this approach in [122] for different regression tasks. In addition, the multiple network initialization approach has been found to outperform Bayesian neural networks in practice [76]. To take into account the uncertainty caused by the limited training data, the training and evaluation data sets are randomly allocated before each initialization.

An overview of the specific parameter of the applied MDN can be seen in Table 4.3.

Table 4.3: Hyperparameter used for the MDNs. For specification of the other hyperparameters see also Section 4.1.2.

Hyperparameter	
Number of hidden layers	4
Number of units per hidden layer	75
Batch size	32
Validation split	30 %
Dropout factor	0.35
Amount of MC dropout ensemble members	[1, 5, 10, 15]
Amount of network initializations	[1, 5, 10, 15]

> *“Accepting the advantages and limitations of systematic forecasting methods [...] is critical. Such methods do not possess any prophetic powers, they simply extrapolate established patterns and relationships to predict the future and assess its uncertainty.”*
>
> Spyros Makridakis [168]
>
> (Founding chief editor of the Journal of Forecasting and the International Journal of Forecasting)

5

Results

Considering the large number of comparison possibilities, the focus of this chapter is first on analyzing the different variations of the respective methods (e.g., with different hyperparameters) in detail. This is particularly interesting as several of the introduced methods are applied for the first time to PV power forecasts to the best of the author’s knowledge. Furthermore, commissioning simulations with little training data have not yet been carried out.

Afterwards, the overall comparison of the respective best methods and hyperparameter combinations across all representation forms is performed in Section 5.7.

5.1 Bootstrapping approaches for the ARX model

Figure 5.1 illustrates both the probabilistic accuracy and the relative improvement of the bootstrapping approaches compared to the reference forecast, each depending on the amount of training data.[1]

Both Sieve bootstrapping (16.2 %) and Residual bootstrapping (15.9 %) demonstrate a better forecast accuracy than the benchmark with just 7 days of training data. However, training data bootstrapping has a worse forecast quality than the benchmark. Combined with the fact of the small differences (maximal 0.2 %) between residual bootstrapping and sieve bootstrapping, it can be concluded that the influence of epistemic uncertainty for the ARX approach on the present use case is rather small or even negligible. This conclusion is reinforced further by the rank diagrams in Figure 5.2. These show that the rank histogram of the training data bootstrapping approach is far too sharp / underdispersed. This is due to the fact that the modeled parameter uncertainty only accounts for a significantly smaller part of the uncertainty.

The low influence of the epistemic uncertainty also results from the determination of the significant model lags via cross-validation with the greedy search algorithms and by allowing gaps between them. Both prevent an unnecessarily high number of parameters and

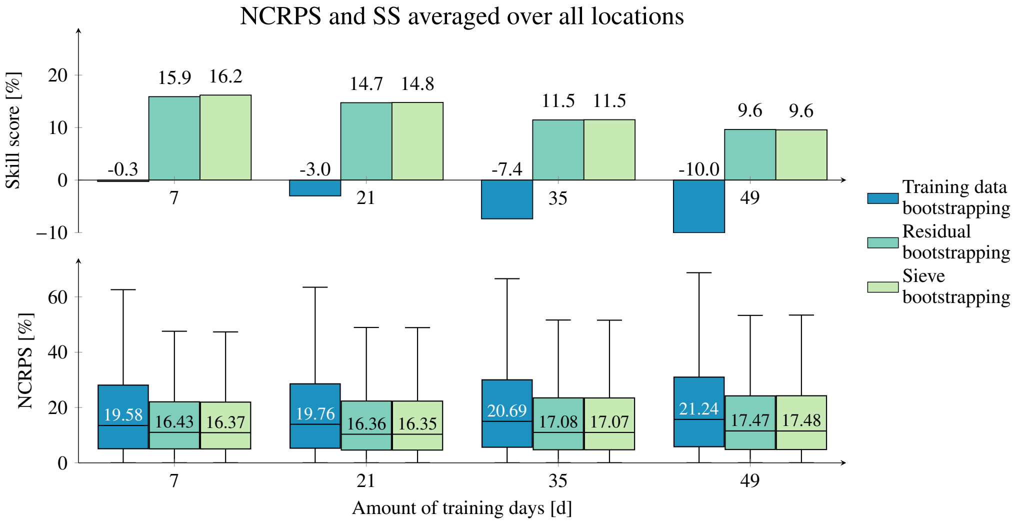


Figure 5.1: The NCRPS depicted as box plots and the SS depicted as bar graph averaged over all locations for the different bootstrapping approaches. As benchmark serves the CH-PeEn. The specific results for the individual locations can be found in the appendix in A.4. The whiskers in the box plot span 1.5 times the interquartile range, which extends from the 25th to the 75th percentile.

---

[1] The skill score is only presented as a bar graph in this thesis. As stated in Section 3.3.2, the skill score is not proper. Hence, the skill score should not be determined for individual data points or small samples to ensure consistent statements. Therefore, the NCRPS values are first averaged over all measured values and subsequently the skill score is calculated.

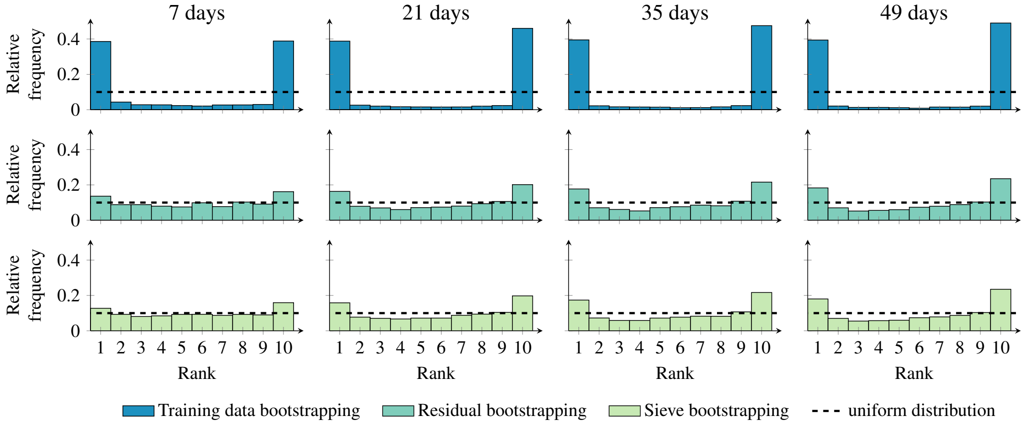


Figure 5.2: The rank histogram for the analyzed bootstrapping algorithms for different amount of training data. The training data bootstrapping rank histogram is underdispersed, leading to the conclusion that this probabilistic forecast approach generates a too sharp distribution.

correspondingly a higher parameter uncertainty. The other two bootstrapping approaches, though, consistently show a good reliability in the rank diagram. Nevertheless, with increasing amounts of training data, all distributions tend to become more and more underdispersed with the bars of the 1st and 10th ranks increasing in particular.

The effect of forecast deterioration with increasing number of training data can also be observed when looking at the NCRPS value. Figure 5.3 shows the relative improvement in NCRPS of each respective forecast as a function of the number of days of training data used. While the forecast quality has increased slightly due to the increase from 7 to 21 days of training data (e.g., 0.44 % for residual bootstrapping)[2] , a deterioration of the forecast quality

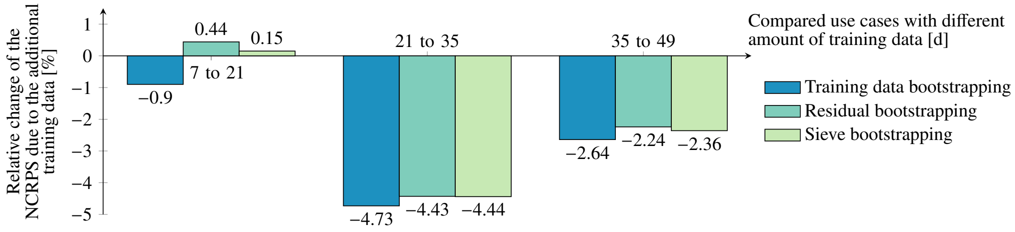


Figure 5.3: The relative change of the NCRPS due to the additional available amount of training data. While the addition of 7 to 21 days of training data still improves the prediction accuracy for two of the three methods, a general deterioration of the prediction accuracy can be seen from 35 days of training data onwards. Due to the comparatively longer time period, the influence of seasonal effects increases, which cannot be captured with the applied ARX model approach.

---

[2] If one only considers the skill score, it seems as if the forecast gets worse with 21 days of training data. However, the reduction is due to a relative improvement in the prediction quality of the benchmark.

is visible starting from the step of 21 to 35 used training data days (e.g., 4.43 % for residual bootstrapping).

Due to the comparatively longer time period, the influence of seasonal effects increases, which cannot be captured with the applied decomposition model approach for the stationarization. For instance, depending on the season and the location the sunrise and sunset times can shift by more than one hour from one month to the next, which in turn reduces the accuracy of the piece wise linearization based on the temporal operating points. Furthermore, the volatility of PV power varies throughout the year. Hence, a comparatively lower volatility in summer could be better predicted e.g., with less highly volatile training data from spring.

State of the art to compensate the occurring seasonal changes with the ARX model would be either to adjust the window width accordingly (e.g., limit it to a climate/site-specific duration of 21 days in our case) or to introduce a forgetting factor for the training data [48]. In this thesis the first approach is chosen. Thus, even if 182 days of training data were available for the selected model approach, only the last 21 days would be considered by using a moving window.

In addition to the prediction accuracy, the computational time required for the individual bootstrapping algorithms was monitored (see Figure 5.4). The evaluation of the required computation time should be interpreted primarily qualitatively, as different results can be expected depending on the available hardware and type of implementation. For this reason, a normalization is carried out with respect to the longest required computing time (15.7 s). In particular, a repeated parameter estimation by least square to model the epistemic uncertainty is more computationally intensive. Consequently, the Sieve bootstrapping and training data bootstrapping take two to three times as long as the residual bootstrapping.

As Sieve bootstrapping was only marginally better for the analyzed systems the residual bootstrapping approach is probably preferable, particularly when using the algorithm on low computational edge devices, e.g., remote terminal units.

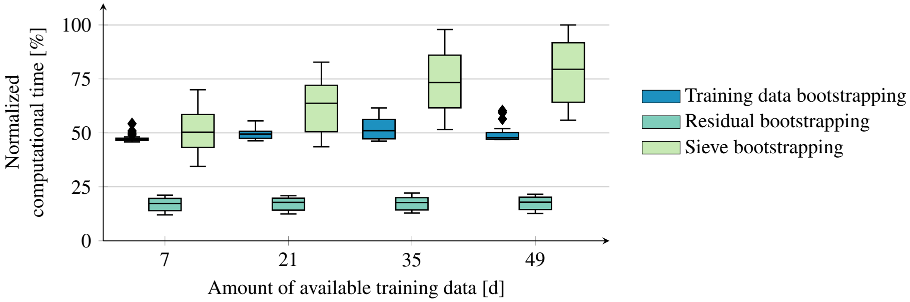


Figure 5.4: Needed computational time to generate a respective forecast for the different bootstrapping forecasting methods. As the absolute time depends highly on the available hardware and type of implementation a normalization was carried out with respect to the longest required computing time (15.7 s).

5.2 Monte-Carlo dropout with output calibration

Figure 5.5 displays the prediction quality of the MC dropout for both the calibrated and uncalibrated cases. It is immediately apparent that the benchmark algorithm performed better with 7 days of training data. This is partly due to the fact that only 4 days are used to train the neural network as the additional calibration data set is required. Moreover, the calibration data set is also too small, which even results in a deterioration compared to the uncalibrated case.

However, from 21 days of training data, the MC dropout approach is already considerably better than the benchmark. This effect also increases with 182 days of training data, where, for example, the calibrated model shows a forecast with a 31.7 % lower NCRPS compared to the CH-PeEn. In addition, the calibrated approach is significantly better in both cases. For instance, the difference between the approaches for 21 days of training data is 3.4 %, which in turn corresponds to a relative improvement of ~ 21 %.

The reason for the significant improvement of the calibrated prediction compared to the uncalibrated one becomes especially clear in the rank histogram (see also Figure 5.6). Only a consideration of the epistemic uncertainty is not sufficient, as this results in a too sharp probabilistic forecast. This can be concluded from the ∪-shape of the rank histogram. Despite this, an improvement compared to the benchmark without calibration, can be seen due to the good underlying point forecast. It keeps many of the sharp predictions close to the true value and thus positively influences the metric. The calibrated forecast, in turn, is significantly more

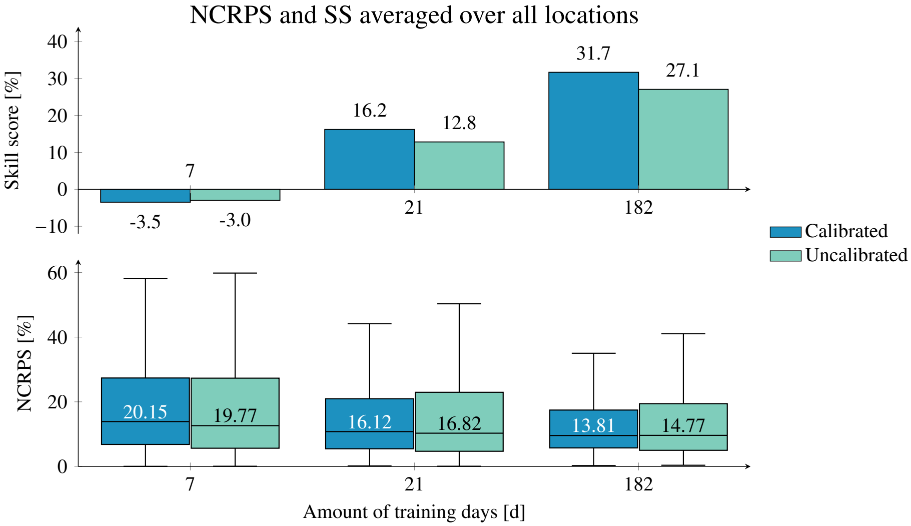


Figure 5.5: The NCRPS depicted as box plots and the SS depicted as bar graph averaged over all locations for Monte-Carlo dropout with output calibration. As benchmark serves the CH-PeEn. The specific results for the individual locations can be found in the appendix in A.4.

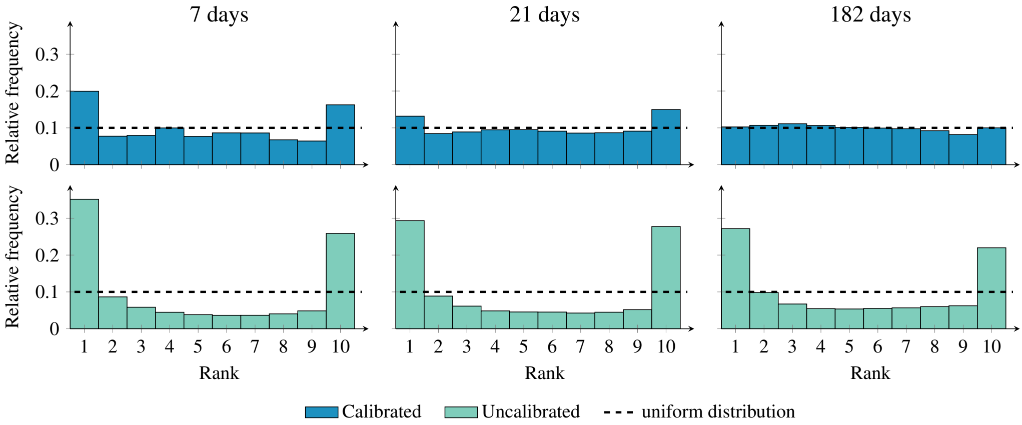


Figure 5.6: The rank histogram for MC dropout for different amount of training data.

reliable and shows almost a uniform distribution in the rank histogram, especially for 182 days of training data. The slight increases in rank 1 and 10 for 21 days of training data are probably due to the fact that the calibration data set is not sufficient to adequately reflect the very high and low extreme values.

The dependence of the forecast quality on the forecast horizon is displayed in Figure 5.7. It clearly shows how much more difficult it is to predict further into the future. For example, the skill score is 37 % lower with a forecast horizon of five to six hours compared to the next hour despite 182 days of training data. This is due to the fact, that the current volatility, e.g., strong winds/calm weather, can change more with a longer forecast horizon. In addition, it can be seen that even with 7 days of training data, the model is better than the benchmark with a low forecast horizon.

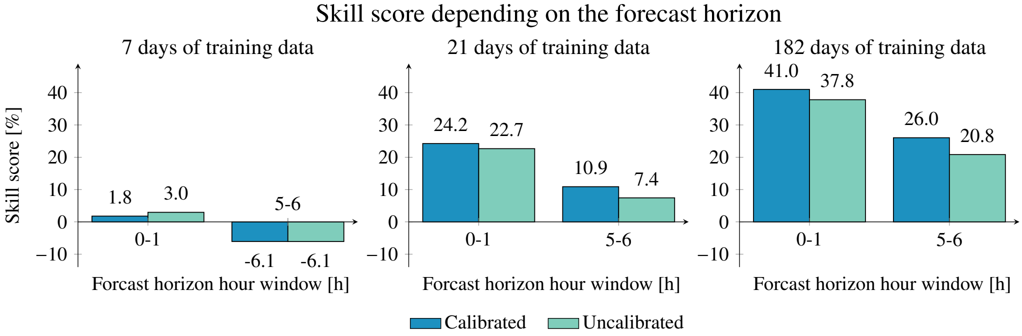


Figure 5.7: The skill score for both the forecast one hour into the future and five to six hours into the future. There is a significant deterioration in the forecast quality for the period further into the future. As benchmark serves the CH-PeEn.

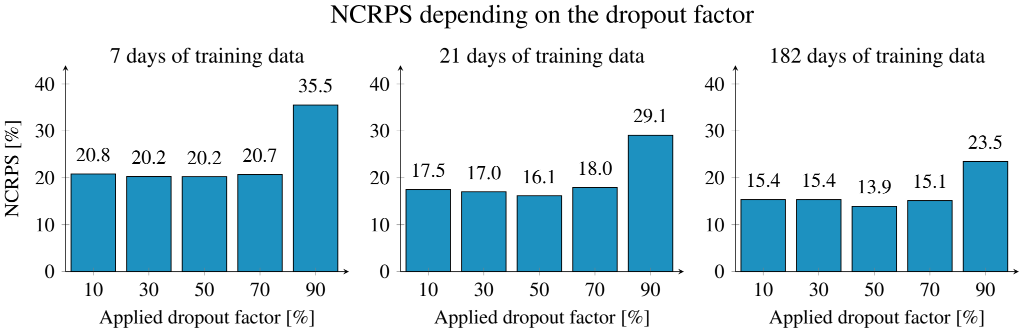


Figure 5.8: NCRPS for the calibrated model as a function of the dropout factor for different numbers of available training days.

The influence of the dropout factor was analyzed in preliminary studies during the hyperparameter selection using the location in North Bavaria. As can be seen from Figure 5.8, the prediction accuracy is almost the same except for the dropout factor of 0.9, due to the subsequent calibration. Since 50 % was slightly better with 182 days of training data, it was selected for the investigations.

Both the number of network initializations and the number of MC dropout ensemble members positively impact the prediction accuracy, as can be seen in Figure 5.9. For instance, with 21 days of training data, multiple network initializations alone can reduce the NCRPS value of the prediction by 8.26 %. The influence of the MC dropout ensemble members is lower, however. This is partly attributable to the subsequent calibration. Furthermore, only in the

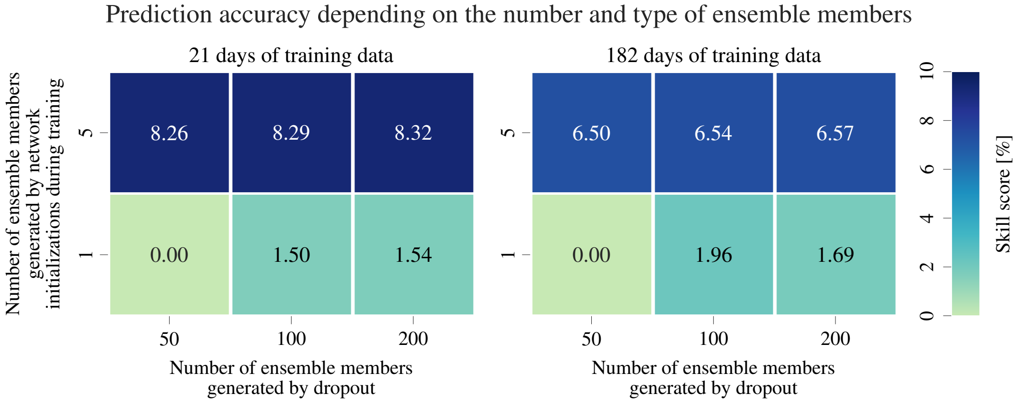


Figure 5.9: Influence of the number of MC dropout members and network initializations during training on the forecast quality depending on the amount of available training data. The data includes all forecasts over the entire horizon (six hours). To enable a clearer analysis of the benefits of the epistemic extensions, the prediction with one dropout member and one initialization member was used as a reference value for the skill score for each respective amount of available training data.

case of multiple initializations, is the epistemic of the calibration model depicted by using different timestamps for the calibration data set of each ensemble member. Consequently, at least a few ensemble members should be considered in practice, since considerable great advances can already be achieved with relatively little effort. It is also noticeable that the improvement due to network initialization decreases as the amount of training data increases. Given the larger training and calibration data sets the influence of the epistemic decreases.

5.3 Quantile regression with the ARX model

As can be seen in Figure 5.10, the ARX model in combination with quantile regression shows a significant improvement compared to the CH-PeEn benchmark. For instance, the skill score is 19 % for 7 days of training data. As with the ARX bootstrapping approaches, the predictive accuracy decreases as the number of training days increases. Thereby, the marginal deterioration from 7 (15.9 %) to 21 (16.0 %) days of training data is caused by individual outliers. For example, the median at 21 days (10.2 %) is even slightly lower than at 7 days (10.6 %). Accordingly, an adaptive approach with a moving training data window is also preferable for this probabilistic method. This is also supported by the histogram (see Figure 5.11), which shows a bias, especially starting from 21 days of training data.

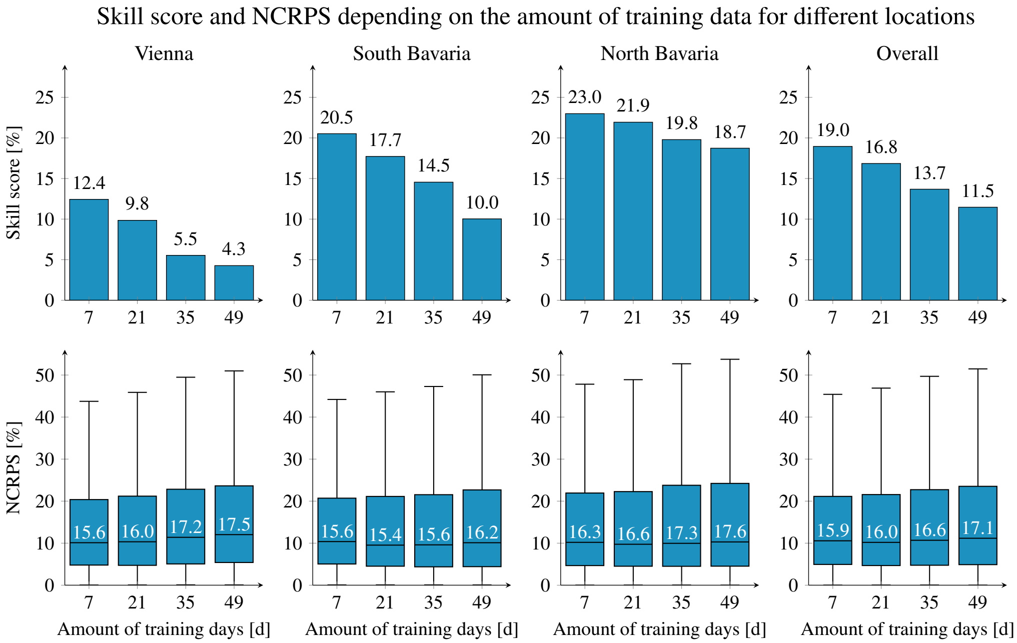


Figure 5.10: The NCRPS depicted as box plots and the SS depicted as bar graph for the quantile regression approach with the ARX model. As benchmark serves the CH-PeEn.

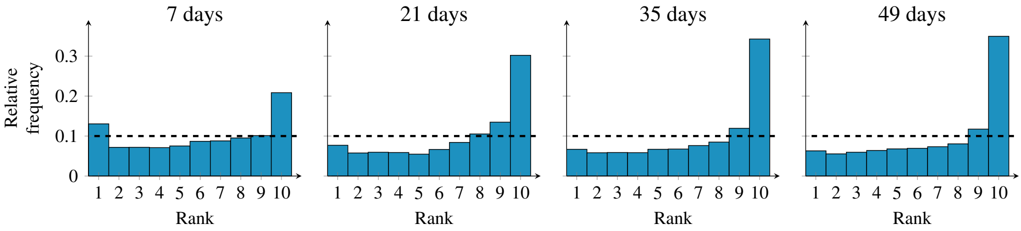


Figure 5.11: The rank histogram for quantile regression in combination with the ARX model approach for different amount of training data.

Although the prediction quality does not improve as the number of training data increases, the influence of the epistemic is still relatively high. Thus, the average of the probabilistic distributions of the ensemble shows a higher prediction quality than even the best ensemble member (see Figure 5.12). Hence, it can be concluded that the model uncertainty is primarily caused by the restrictive linear model structure of the ARX model and not by the available amount of training data. This conclusion is also confirmed in Section 5.7 in the comparison with the GARCH approach.

A potential reduction of the model bias could be achieved by determining the optimal model structure in the greedy search algorithm during cross validation directly on the basis of the probabilistic forecast and thus also with a probabilistic metric.

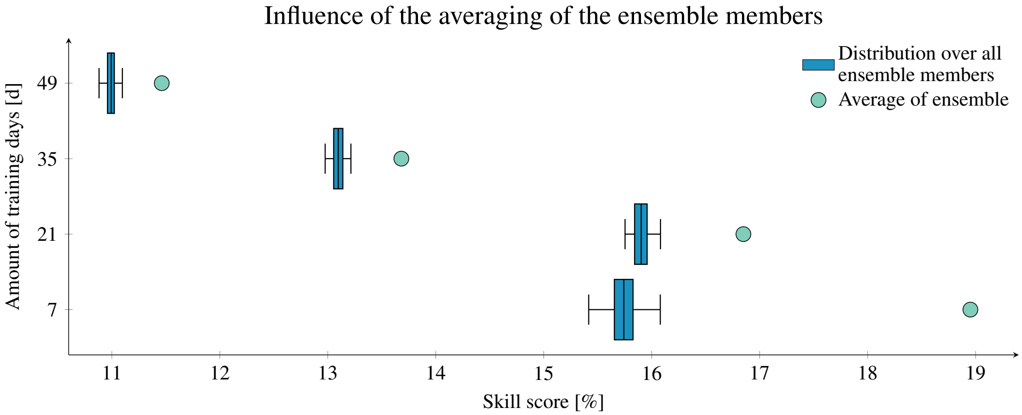


Figure 5.12: Influence of ensemble member averaging on prediction quality. The averaged forecast has better accuracy than any of the individual ensemble members. The difference between the average and the ensemble members declines with an increasing number of training data, as this also decreases the distinction between the determined ARX model structures by the greedy search algorithm.

5.4 Quantile neural network

Averaged over all locations, the quantile neural network already shows a better forecast quality at 7 days than the benchmark of 4.1 % (see Figure 5.13). Nevertheless, this does not apply to all the examined sites, as Vienna has a negative skill score. However, the improvement with increasing number of training days is significant. Both in relation to the benchmark and also in terms of the absolute NCRPS values. For example, the NCRPS value decreases in relative terms by 31 % from 18.4 % to 12.7 % with 182 days of training data. This improvement is also visible in the rank histogram in Figure 5.14. The distribution is almost uniform at 182 days.

These results suggest that the model structure is initially too complex for the available amount of data. However, when 182 days of training data are available, slightly worse results were obtained in the sample tests with 30 instead of 50 neurons per hidden layer.

The influence of the epistemic and the extent to which the ensemble members can compensate it is also shown in Figure 5.15. Here it can be seen that the increase in additional MC dropout ensemble members has no significant influence on the prediction quality.

The influence of the epistemic and the extent to which the ensemble members can compensate for it is also shown in Figure 5.15. The increase of additional MC dropout ensemble members

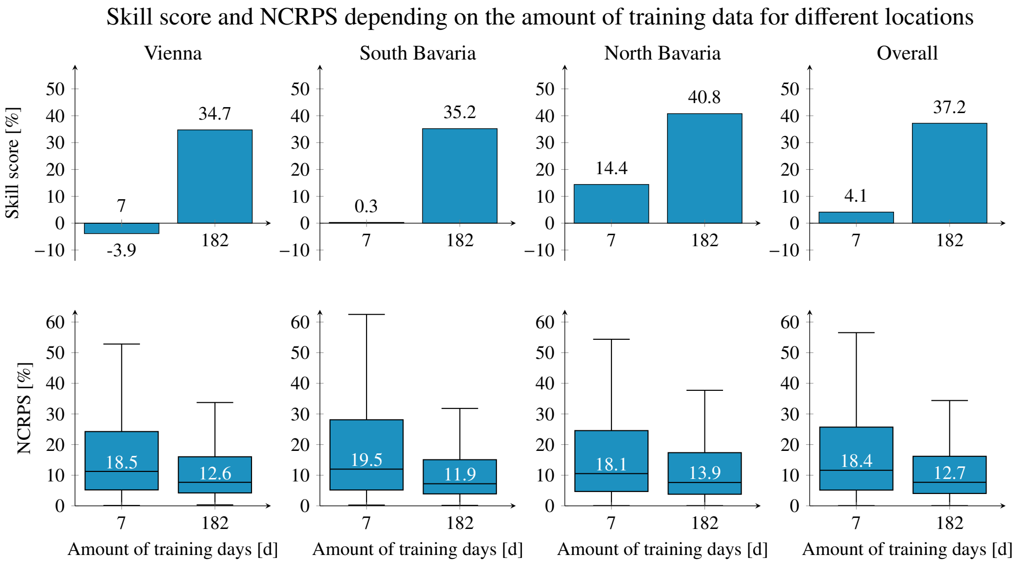


Figure 5.13: The NCRPS depicted as box plots and the SS depicted as bar graph for the quantile neural network. As benchmark serves the CH-PeEn.

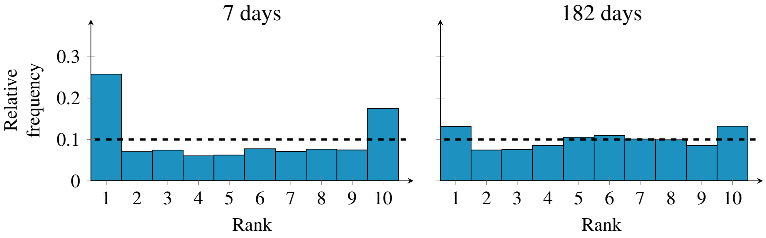


Figure 5.14: The rank histogram for the quantile neural network for different amount of training data.

has no significant influence on the prediction quality.[3] However, the influence of the network initializations is significantly higher, as the forecast quality increases by 4.36 % at 7 days if the network is initialized five times.

There are two reasons for the difference in impact. Firstly, MC dropout tends to focus on a single mode of the loss landscape, whereas different network initializations tend to explore diverse modes in the function space [76]. Secondly, the training and validation data was also resampled for each initialization. Both lead to a greater variance between the ensemble members and also to a better consideration of the epistemic. As was to be expected, the influence of the epistemic also decreases at 182 days due to the greater number of data. Hence, the network initializations only increased the prediction quality by an average of 2.24 %.

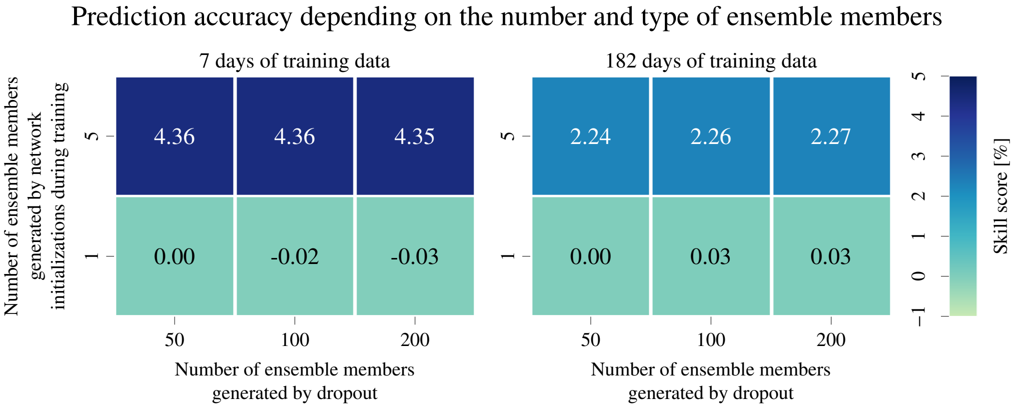


Figure 5.15: Influence of the number of MC dropout members and network initializations during training on the forecast quality of the quantile neural network for different amount of available training data. The illustrated chart contains all forecasts over the entire horizon (six hours). To enable a clearer analysis of the benefits of the epistemic extensions, the prediction with one dropout member and one initialization member was used as a reference value for the skill score for each respective amount of available training data.

---

[3] The reduction of 0.03 % with increasing number of ensemble members via MC dropout is due to the fact that negligibly better members were apparently drawn at random in the 50 members. To reduce the effect of individual ensemble members, five different combinations of ensemble members were used for each of the ensemble scenarios shown. Accordingly, each number in Figure 5.15 also represents the average of 5 different selected ensemble pairs.

5.5 GARCH model in combination with the ARX model

Figure 5.16 depicts both the probabilistic accuracy and the relative improvement of the GARCH model compared to the reference forecast. Already 7 days of training data show a significant improvement of the forecast quality of 30.5 % for the Gaussian distribution. Compared to the other ARX approaches, the forecast quality is consistent for different numbers of training data. Nevertheless, with 21 days of training data, the NCRPS values are marginally better at 13.1 % for the Gaussian distribution. With an additional number of training days, the NCRPS values increase slightly. Accordingly, an adaptive approach or an approach that uses the training data with a moving window is also recommended for the GARCH model.

The accuracy results of the different distributions show a marginal difference. The estimated Gaussian distribution approach has a roughly 0.2 % lower NCRPS value at 21 days than the model with Skewed-t distribution. However, the negligible difference is primarily due to the influence of “more extreme” deviations during averaging. For instance, at 21 days of training days, the median NCRPS for the Gaussian distribution (9.31 %) is even slightly higher than for the Skewed-t distribution (9.26 %).

This aspect is also supported by the rank histograms in Figure 5.17. Given the minimally overdispersed distribution with assumed Gaussian distribution, it can be concluded that the prediction intervals are relatively wide. This leads to slightly higher NCRPS values over several predictions, but at the same time reduces the probability of outliers or very poor predictions. Due to the mostly heavier tails of the t-distribution, the outer ranks one and ten in particular are better calibrated with this distribution. At the same time, a bias is recognizable.

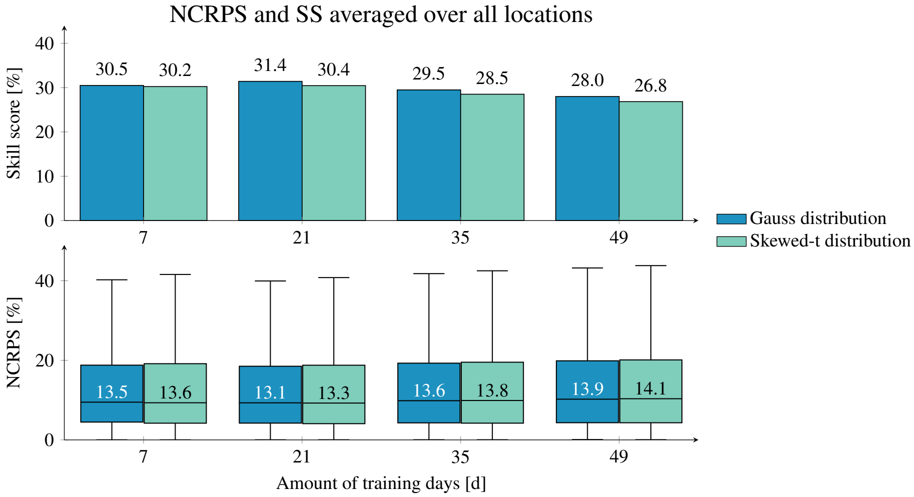


Figure 5.16: The NCRPS depicted as box plots and the SS depicted as bar graph averaged over all locations for the GARCH model. As benchmark serves the CH-PeEn. The specific results for the individual locations can be found in the appendix in A.4.

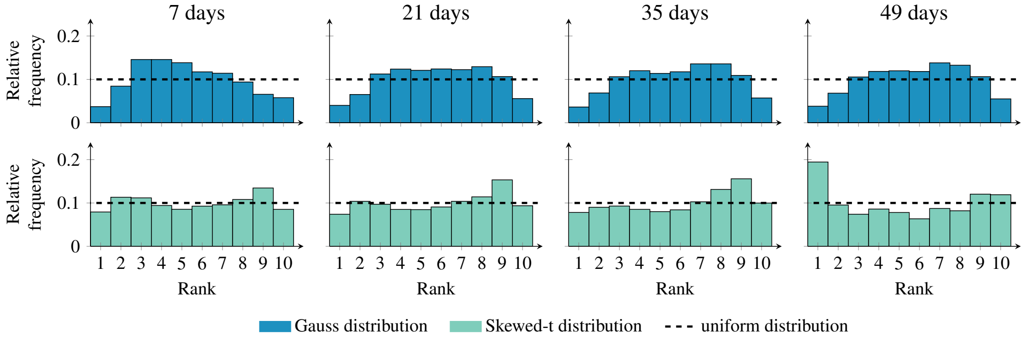


Figure 5.17: The rank histogram for the GARCH model for different amount of training data and different types of assumed residual distributions.

Overall, both the rank histograms and the NCRPS values show better results compared to the previous ARX approaches. The high forecasting accuracy even with a low amount of training data can be attributed to the more rigorous model formulation with assumed uncertainty distribution and therefore lower flexibility. Consequently, the influence of the epistemic is also comparatively low. This is also one of the reasons why averaging the ensemble members only marginally improves the prediction quality (see Figure 5.18). For instance, the average of the ensemble does not have a higher accuracy than the best ensemble member, as it was the case with the quantile ARX approach. Nevertheless, the bagging approach increases the robustness of the prediction, especially with little training data, as individual ensemble members also perform worse. The accuracy of the prediction can potentially be improved in the future if, in addition to the structure of the underlying ARX model, the order of the GARCH model is also varied.

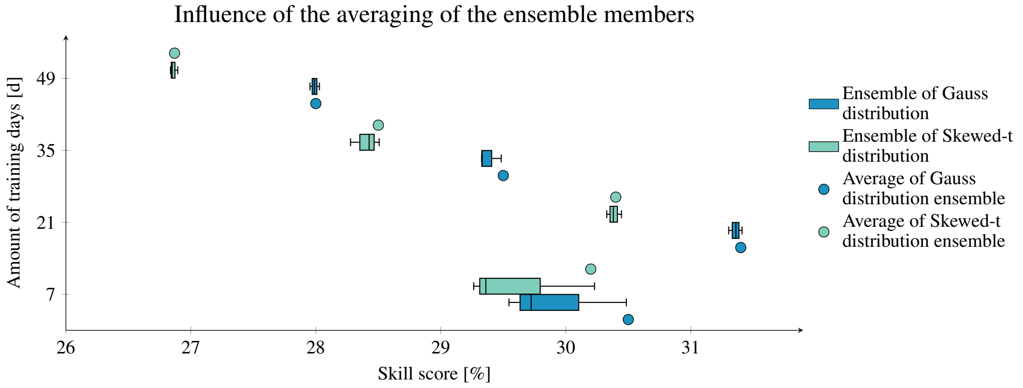


Figure 5.18: Influence of ensemble member averaging on prediction quality. The spread in accuracy between the respective ensemble members decreases with an increasing number of training data, as the influence of the epistemic also lessens.

5.6 Mixture density network

Figure 5.19 illustrates both the probabilistic accuracy and the relative improvement of the MDN compared to the reference forecast, each depending on the amount of training data and the number of mixture distributions. A significant improvement of 14.4 % to 16.0 % compared to the benchmark can already be observed with 7 days of training data and a single Gaussian output distribution. MDNs can therefore be used with relatively limited available training data, e.g., during commissioning. Prediction accuracy increases considerably with half a year of training data, as can be seen by the reduced interquartile ranges in the box plots. The mean NCRPS for ten mixture distributions, for instance, decreases from 16.3 % to 12.0 % for 10 distributions, corresponding to a relative improvement of 26.4 % and an improvement over the CH-PeEn benchmark of 39.8 %.

The impact of the additional output distributions in the mixture model depends on the amount of available training data. The quality initially decreases at 7 days, as the NCRPS increases from 16 % (one distribution) to 16.3 % (ten distributions). This can be attributed to the influence of extreme values on the averaged value, as the median for ten distributions, is relatively 14 % lower than with one. Hence, the significantly more complex model structure of 30 outputs for 10 distributions in comparison to three outputs for one distribution may lead to poorer results if there is too little training data. However, after six months of training data,

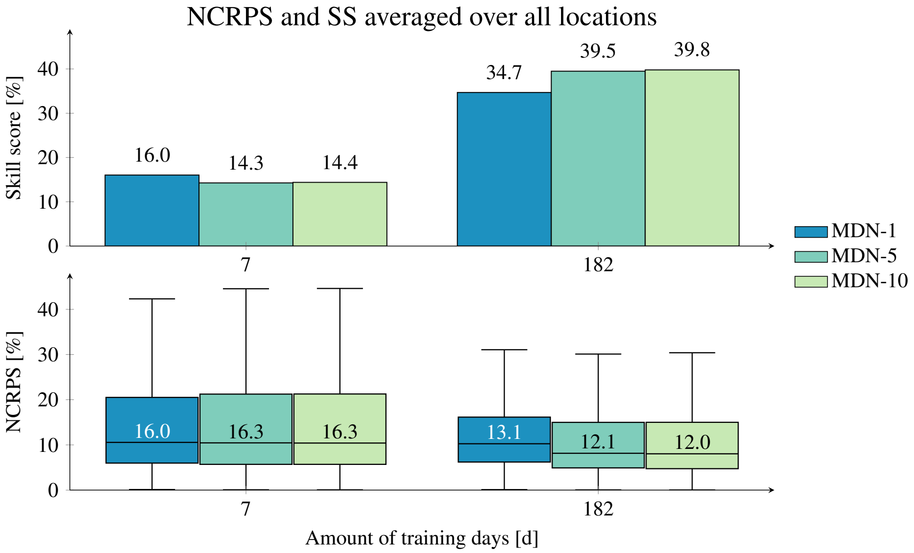


Figure 5.19: The NCRPS depicted as box plots and the SS depicted as bar graph averaged over all locations for the MDN approach. As benchmark serves the CH-PeEn. The results are based on a forecast horizon of 6 hours and MDN ensembles with 15 training initializations, which in turn have 15 dropout ensembles each. The number after the abbreviation MDN in the legend indicates the respective distribution quantity, e.g., MDN-1 means one output distribution.

significant improvements can be observed, resulting in a relative improvement of e.g., 7.6 % from one to 5 mixture distributions. Consequently, firstly, the more complex model structure is better at utilizing the additional information provided by the extra data and, secondly, a certain amount of training data is required to exploit the potential of the more flexible distribution mixture. From five to ten distributions no significant additional improvement occurred, indicating that the underlying uncertainty distribution of the forecast can already be estimated relatively accurately with five output distributions. Moreover, it should not be forgotten that slightly different Gaussian distributions are already included in the mixture model due to the ensemble members.

The rank diagrams in Figure 5.20 show a good reliability in comparison to the benchmark method and a slight improvement with the added mixture distributions. Since the underlying uncertainty at the output does not exactly resemble a normal distribution, a slight bias occurs at the output with only one distribution. Consequently, the sixth percentile is slightly overestimated. This context is demonstrated further by Figure 5.21, which shows the parameters (standard deviation, mean value, weighting factor) of the individual output distributions normalized by the measured PV power for different numbers of mixture distributions. In the case of a single distribution, the mean is slightly overestimated, and the standard deviation is comparatively higher, in order to include and represent also extreme values in the uncertainty estimation. With multiple mixture models, these extreme values can, in turn, be modeled by the additional distributions with higher normalized mean values and lower standard deviations and weighting factors. Thus, instead of one broad distribution, multiple narrower distributions are combined with each other. Thereby, the distributions with the smallest distance to the true value, which is in the normalized representation the value one, have the largest weighting factors. As the number of mixture distributions increases, their weighting factors decrease significantly. This also leads to the conclusion that additional distributions probably do not improve the forecast quality, and at the same time may cause numerical problems considering a possible underflow in (4.31).

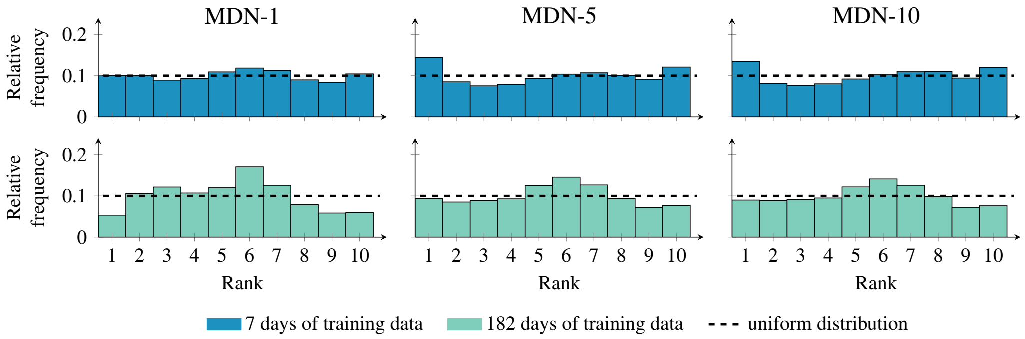


Figure 5.20: The rank histogram for the MDNs for different amount of training data and different numbers of distribution at the output.

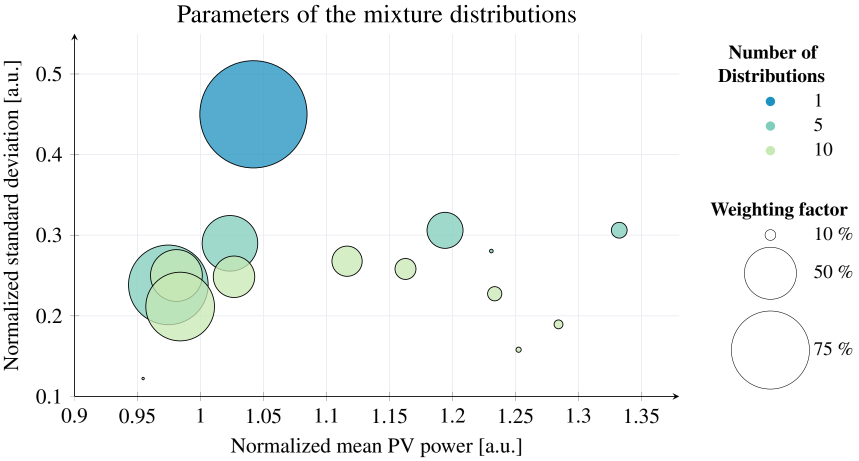


Figure 5.21: Representation of the parameters of the different mixture distributions depending on their number in the model output. The size of the markers reflects their respective weighting factor in the mixture model. In order to enable comparability for varying power levels, both the standard deviation and the mean of the distributions were normalized by dividing the values with the respective measured PV power.

The influence of the extensions to estimate epistemic uncertainty are summarized in Figure 5.22. Both the network initializations and MC dropout have a significant positive effect on the forecast performance. For example, the use of dropout ensemble members alone improves the forecast quality by up to 10.05 % and additional network initializations by up to 18.41 % for 7 days of training data. The impact of MC dropout is therefore slightly lower. As already mentioned in the case of the quantile neural network, the reason for this is that the members of the dropout ensemble typically gravitate around a single mode of the loss landscape, while the different network initializations exhibit a more diverse exploration [76]. Moreover, since the training and validation data are also sampled, the variety and information in the training data is also higher for the multiple network initializations. Nevertheless, MC dropout needs less computational resources and is faster, as the model training does not have to be performed multiple times. For both methods, the added value decreases significantly as the number of ensemble members increases. At least a few ensemble members should therefore be considered in practice, since considerable great advances can already be achieved with relatively little effort.

The improvement of the forecast quality by the two approaches is larger for 7 days of training data than for 182 days, since the epistemic uncertainty decreases with increasing number of training data. Accordingly, the epistemic extensions to the MDN are particularly recommended for applications in practice, when the number of training data may be limited during commissioning and no corresponding individual adjustment of the network structure is made.

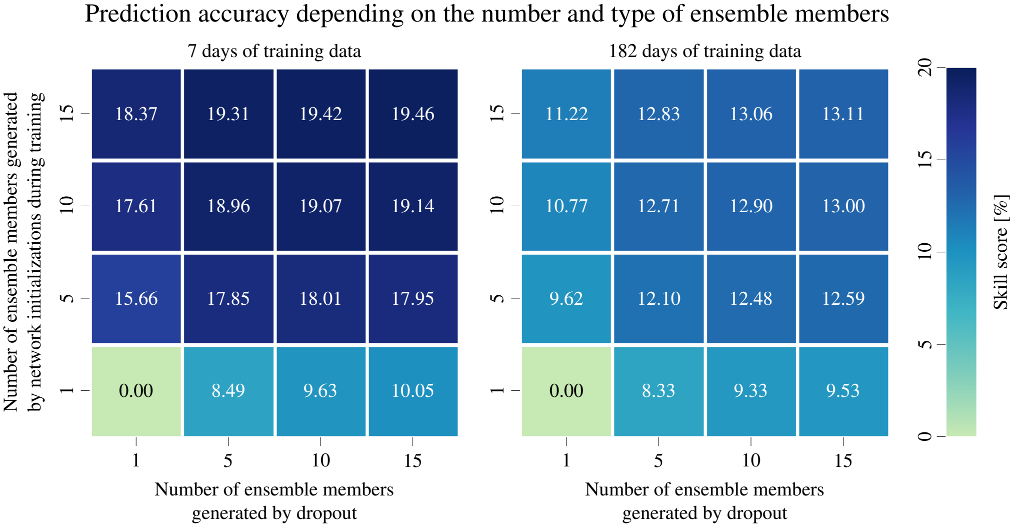


Figure 5.22: Influence of the number of MC dropout members and network initializations during training on the forecast quality depending on the amount of available training data. The data includes all forecasts over the entire horizon (six hours) with ten mixed distributions. To enable a clearer analysis of the benefits of the epistemic extensions, the prediction with one dropout member and one initialization member was used as a reference value for the skill score for each respective amount of available training data.

5.7 Overall Comparison

Figure 5.23 provides the overall comparison of the algorithms in this work each with the best specification for both 7 and 182 days of training data.[4]

Overall, the ARX-based probabilistic prediction methods perform better than the neural networks with fewer available training days for all representation forms. Thus, the increased model flexibility of the MLP models leads to a lower prediction quality despite the regularization methods and methods used to compensate for the epistemic in the studies. This is particularly evident in the MC dropout, which has an average skill score of -3.5 %. Due to the necessary calibration data set, this approach has even less data available for training, meaning that even the benchmark algorithm performs better.

Nevertheless, it can be seen that the developed MDN structure with an average of 14.4 % performs better than the quantile regression approach with 4.1 %. It therefore performs better than the other MLP approaches. The comparatively low model flexibility is the reason for this. By estimating the entire distribution and the model specification of the mixed Gaussian distributions, the influence of the epistemic is reduced. This is further reinforced

---

[4] The individual analyses have shown that the accuracy of the ARX approaches decreases beyond a certain point if more training data is used. As a result, even with 182 training days available, the last 21 days were used adaptively for the GARCH and residual bootstrapping approach and the last 7 days for the quantile regression approach.

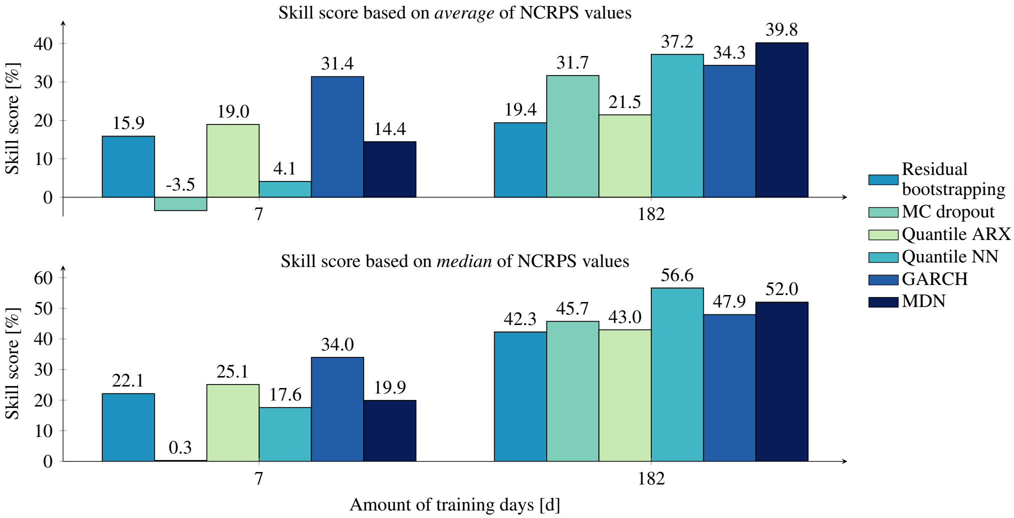


Figure 5.23: Comparison of the skill score of the different algorithms using the persistent ensemble. For the individual algorithms, the best specification, e.g., maximum number of ensemble members, was used. Differences in the values of the skill score compared to the previous subchapters arise since for the ARX based algorithms 182 days instead of 21 days of training data were used for the benchmark. The first row corresponds to the SS based on the average NCRPS values of the respective algorithms and the benchmark and the second row to the respective medians.

by a comparison of the skill score values based on the medians. In particular, the quantile neural network (NN) shows a considerable difference between the median and mean-based skill score. This indicates negative outliers, presumably due to the increased flexibility, which distort the overall mean value. The MDN approach is therefore advantageous compared to the other MLP approaches during commissioning with smaller data sets.

Overall, with 7 days of training data, the GARCH ensemble model performed best with an average improvement of 31.4 % compared to the benchmark. This means that it even has better accuracy than the used multistep quantile regression model (19.0 %), which has more model flexibility due to the form of representation via quantiles and the use of an individual model for each quantile. Accordingly, it can be assumed that the differences in quality lie in the underlying model concept. In the GARCH model, past volatility is taken into account directly in order to estimate future volatility. As can be seen in Figure 5.24 and also in the forecast results, the use of past volatility and thus uncertainty is a good indicator for the future. As the ARX model with quantile regression estimates the quantiles using a linear combination of past performance values, past volatility can only be taken into account indirectly.

This information is particularly significant, as the GARCH approach had performed worse in past studies [48] in the irradiation forecast with a pure autoregressive ARMA approach compared to different probabilistic methods (including quantile regression). Consequently, the developed adaptive ARX approach used with greedy search algorithm and time decomposition as well as including exogenous features is better able to capture the behavior of the underlying

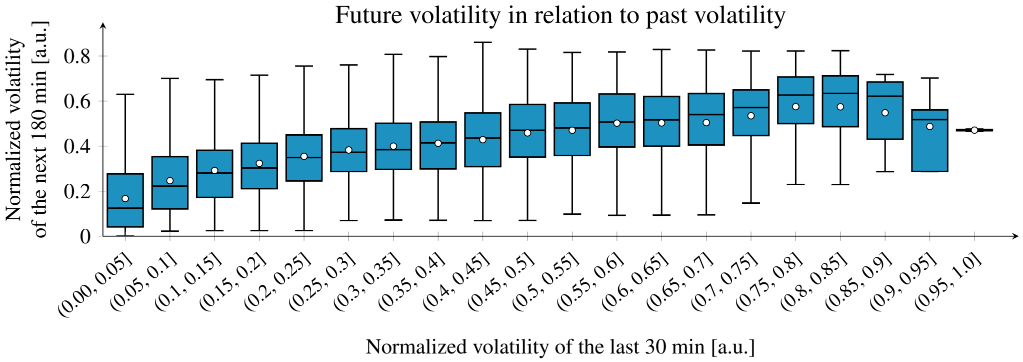
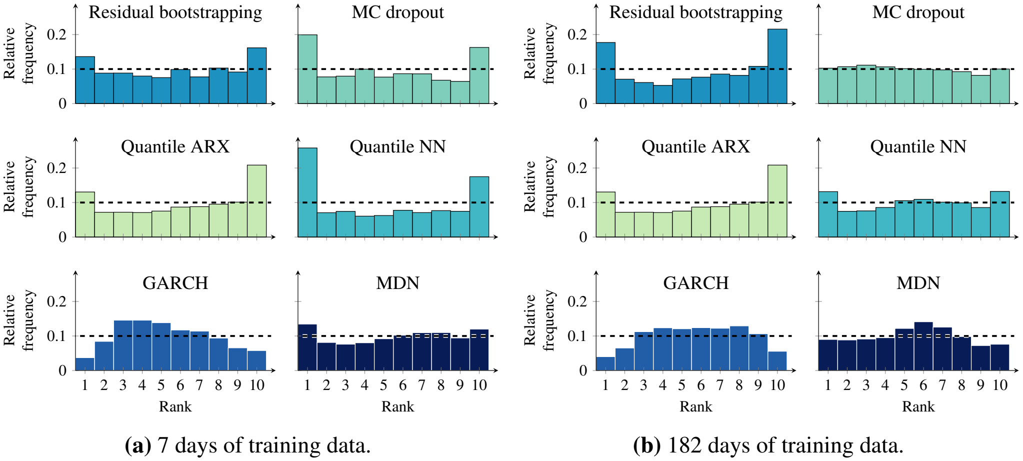


Figure 5.24: Volatility (standard deviation) of the next 180 minutes dependent on the volatility of the last 30 minutes. The mean value is shown by the white marker. A clear direct positive trend can be seen.

process. This leads to more Gaussian-like distributions of the residuals and better depiction of the forecasting uncertainty with the GARCH approach.

Although the ARX-based probabilistic prediction methods perform better than the MLP-based methods during the initial commissioning process, this is reversed after six months of available training data. At this point, the MLP-based approaches perform better on average than their ARX-based counterparts. A comparison of the rank histograms also illustrates this (see figure 5.25), as the MLP-based approaches show a better calibration with 182 days of training data.



(a) 7 days of training data.

(b) 182 days of training data.

Figure 5.25: The rank histogram for different amount of training data. For the individual algorithms, the best specification, e.g., maximum number of ensemble members, was used. The subsequent calibration in the MC dropout algorithm becomes visible, as the distribution almost resembles a uniform distribution.

In particular, the more flexible quantile neural network approach shows the greatest improvement with an average SS of 37.2 %. Nevertheless, the developed MDN approach is still more accurate with an average SS of 39.8 %. Since the median SS is the highest for the quantile neural network, it can be concluded that individual poor results distort the overall mean analogous to the case with few training days.

In general, it can be concluded that the pure ensemble methods have the lowest accuracy in comparison with 19.7 % on average for residual bootstrapping and 31.7 % for MC dropout with output calibration. Accordingly, they should rather be used as a supplement for the compensation of the epistemic. As shown in the previous detailed analyses, they have led to considerable improvements.

Furthermore, it can be seen overall that the approaches used with a continuous distribution have the best prediction accuracy. Based on the available data, the GARCH model is recommended over the entire commissioning period, as it achieved in comparison excellent results both with a small and large amount of available training data. MDN, on the other hand, has the best overall performance given sufficient amounts of training data.

"It is very important for researchers and practitioners to understand that a universally best technique simply does not exist."

Tao Hong [99]

(Founder of the Global Energy Forecasting Competition and the IEEE working group on energy forecasting)

6

Summary and outlook

6.1 Summary and conclusion

This thesis addressed the generation of probabilistic PV power forecasts for multi-modal DESs. The focus was thereby particularly on relevant questions that need to be answered for the transition to an applied industrial use.

Based on several use cases, the practical requirements for on-site energy systems were first identified (Section 2.4.2), followed by the derived definition of the forecast specifications for the studied scenario (Section 2.7). In addition, the example of local energy markets was used to demonstrate that there is no threshold for maximum forecast accuracy or a point after which the importance of further accuracy improvements decreases (Section 3.3.3).

By comparing the practical requirements with the current state of the art in the field of probabilistic PV predictions, research gaps were analyzed and identified (Section 2.7). To close these gaps, the following three primary research objectives were explored in this thesis:

- **Comparison and adaption of several probabilistic methods for PV power** – Probabilistic solar forecasting is the least mature area in the field of energy time series forecasting. Moreover, solar forecasts in the scientific literature are generally made indirectly via forecasting models for solar irradiation, although this approach has several disadvantages in practice compared to PV power forecasts (Section 2.4.1). In addition, the comparison of different forecasting algorithms across several papers is often not feasible (Section 2.4.3).

Accordingly, four approaches (extended sieve with residual bootstrapping, MC dropout with output calibration, GARCH model, MDN), which have proven successful in solar irradiance forecasting or other forecasting domains, were adapted for PV power and examined in detail, for the first time in this paper or its corresponding publications to the best of the author's knowledge. These methods were compared with two other already established algorithms (ARX model with quantile regression and quantile NN), two additional ARX bootstrapping approaches and a benchmark (CH-PeEn). Thus, all three forms of representation for uncertainties (ensemble, quantiles and continuous probability distribution) are reflected by the selected methods. Furthermore, comparatively simpler model structures with less required computing capacity (ARX models) as well as more complex models (e.g., MDN) were compared.

- **Simulation and analysis of forecast commissioning and operation under practical conditions** – There were no studies regarding the probabilistic prediction quality of PV power forecasts with limited amount of data.

  However, as this is indispensable for commissioning in practice, the prediction quality of different methods was investigated in this thesis, both for the initialization operation period with little data (7 days) and also for a regular operation period (182 days of training data). In order to do this, a simulation setup with 24 different initialization start points for each site was set up (Section 3.2). In practice extensive manual optimization for each site and forecast initialization is not possible due to limited capacities. However, the optimal choice of model structure and hyperparameter varies by location, the amount of training data, and sometimes the time of year (e.g., weather in spring and fall is more volatile than in summer). Hence, this thesis also investigates the feasibility of generating and updating the forecasts over the commissioning period without manual intervention. Instead, several regularization methods are applied to the MLP based approaches (Section 4.1.2). Furthermore, a time decomposition approach was developed for the ARX models using rudimentary forecasting methods followed by a higher-level greedy search algorithm for the automatic determination of the model order (Section 4.1).

- **Consideration of both aleatoric and epistemic uncertainty** – Non-optimal model structures and a lack of training data can lead to high epistemic model uncertainty, which in turn can degrade forecast quality if not taken into account. Moreover, considering the previous paragraph, both aspects are infeasible to avoid in practice. However, previous studies on PV power forecasting mostly do not differentiate between the different types of uncertainty and do not consider both.

  Therefore, this thesis focused on the consideration of both types of uncertainties e.g., by using extensions for modeling the epistemic. Consequently, for several probabilistic PV power approaches (e.g., MDN, GARCH), this thesis investigated epistemic extensions and the influence of the different uncertainty types in detail for the first time.

Some work in this thesis was previously presented in corresponding publications (Refs. [56–58]). However, this thesis also contains many new elements. These are in particular, an even more profound analysis of the individual methods (Section 5.1 – Section 5.6), the applied GARCH model approach (Section 4.4), a comprehensive comparison between all the analyzed methods (Section 5.7) and the overarching, broader characterization of the topic including its context.

Data from three different sites in Central Europe (Section 3.1) was used for the analysis. All investigated methods except MC dropout with output calibration showed better forecast accuracy than the CH-PeEn benchmark with seven days of available training data. However, the MC dropout approach had even less data available for the training due to the necessary calibration data set. For all uncertainty representation forms, the ARX-based probabilistic prediction were better than their respective neural network counterparts given only seven days of training data. Furthermore, it was apparent that an adaptive approach with consideration of 21 days of training data is preferable for the ARX models. If longer periods are taken into account, the influence of seasonal effects cannot be sufficiently captured with the applied decomposition model approach for the stationarization, leading even to a reduction in forecast accuracy.

After six months of available training days, the behavior reversed and the MLP-based approaches performed better on average. Due to the higher model flexibility, they were better able to represent the existing underlying uncertainty. In general, the approaches with a continuous distribution had the best forecasting accuracy. Hence, the GARCH model in combination with the ARX model is recommended over the entire commissioning period, as it achieved in comparison excellent results both with a small (SS: 31.4 %) and large (SS: 34.3 %) amount of available training data. Thus, contrary to previous studies on solar irradiation, it achieved significantly better results in direct comparison with other probabilistic methods. Due to the developed adaptive ARX approach with greedy search algorithm and time decomposition as well as the inclusion of exogenous features, the method is significantly better able to capture the behavior of the underlying uncertainty. However, when provided with sufficient amounts of data, the MDN model surpasses the other models in terms of overall forecasting accuracy with an improvement of 39.8 % compared to the benchmark.

In general, it can also be concluded that the ensemble methods should not be used individually for uncertainty characterization, as the predictive accuracy was systematically worse. However, the additional ensembles in MLP based methods led to significant improvements and should therefore be used as an extension. Ensembles generated by repeated network initialization were able to achieve significantly better performance gains due to the better exploration of the loss landscape leading to a better representation of the underlying epistemic. For the ARX approaches, on the other hand, the influence of the epistemic was rather small, due to the comparable easier model structure and the use of the greedy search algorithm for the individual determination of the model order.

6.2 Outlook

For the transition to scaled industrial use, further aspects need to be investigated in the area of probabilistic PV prediction. One aspect is that the methods should be tested with data from other climate zones. Furthermore, a practical assessment of the workload on low computational edge devices is also recommended, as they are used for several services at once.

In addition, the analyzed algorithms reflect the operation under standard conditions. In practice, a holistic forecasting framework including a monitoring system is necessary, which integrates models for different operating scenarios (e.g., snow on panel, standard operation, fall back solution in the case of missing measurements). The recommended probabilistic methods GARCH and MDN as well as CH-PeEn can serve as the basis for such a framework.

It was shown that, depending on the respective application, theoretically any improvement in forecast accuracy also yields added value. As a result, further investigations should be carried out by using more advanced model architectures, particularly for large amounts of existing training data. For instance, temporal convolutional neural networks and models based on transformers have achieved exceptional performance in the fields of computer vision and natural language processing in the recent past. These architectures could be combined with the presented MDN approach.

Finally, one of the next priorities should be to further investigate the optimal use of probabilistic predictions for the listed use cases (e.g., bidding strategies with external markets) in DESs. This can be achieved through comprehensive simulations and field trials to advance the evaluation and quantification of the added value under different scenarios.

Bibliography

[1] M. Aghaei. *Solar Radiation - Measurement, Modeling and Forecasting Techniques for Photovoltaic Solar Energy Applications*. Rijeka: IntechOpen, Oct. 2022.

[2] R. Ahmed, V. Sreeram, Y. Mishra, and M. Arif. “A review and evaluation of the state-of-the-art in PV solar power forecasting: Techniques and optimization.” In: *Renewable and Sustainable Energy Reviews* 124 (May 2020), p. 109792.

[3] S. Alessandrini, L. Delle Monache, S. Sperati, and G. Cervone. “An analog ensemble for short-term probabilistic solar power forecast.” In: *Applied Energy* 157 (Nov. 2015), pp. 95–110.

[4] D. AlHakeem et al. “A new strategy to quantify uncertainties of wavelet-GRNN-PSO based solar PV power forecasts using bootstrap confidence intervals.” In: *2015 IEEE Power & Energy Society General Meeting*. IEEE, July 2015, pp. 1–5.

[5] M. Alonso et al. *Forecasting time series with sieve bootstrap*. DES - Working Papers. Statistics and Econometrics. WS. Universidad Carlos III de Madrid. Departamento de Estadística, 2000, pp. 1–11.

[6] S. P. Aly, S. Ahzi, and N. Barth. “Effect of physical and environmental factors on the performance of a photovoltaic panel.” In: *Solar Energy Materials and Solar Cells* 200 (2019), p. 109948.

[7] R. Amaro e Silva and M. Brito. “Spatio-temporal PV forecasting sensitivity to modules’ tilt and orientation.” In: *Applied Energy* 255 (Dec. 2019), p. 113807.

[8] A. Angulo et al. “Algorithms for Bidding Strategies in Local Energy Markets: Exhaustive Search through Parallel Computing and Metaheuristic Optimization.” In: *Algorithms* 14 (Sept. 2021), p. 269.

[9] J. Antonanzas et al. “Review of photovoltaic power forecasting.” In: *Solar Energy* 136 (2016), pp. 78–111.

[10] F. Antonanzas-Torres, R. Urraca, J. Polo, O. Perpiñán-Lamigueiro, and R. Escobar. “Clear sky solar irradiance models: A review of seventy models.” In: *Renewable and Sustainable Energy Reviews* 107 (2019), pp. 374–387.

[11] Arbeitsgruppe Erneuerbare Energien Statistik. *Erneuerbare Energien 2021* -. Tech. rep. Berlin: Bundesministerium für Wirtschaft und Klimaschutz, 2022.

[12] C. Aswin Raj, E. Aravind, R. Sundaram, and S. Vasudevan. “Smart Meter Based on Real Time Pricing.” In: *Procedia Technology* 21 (2015), pp. 120–124.

[13] P. Bacher, H. Madsen, and H. A. Nielsen. “Online short-term solar power forecasting.” In: *Solar Energy* 83 (Oct. 2009), pp. 1772–1783.

[14] W. El-Baz, M. Seufzger, S. Lutzenberger, P. Tzscheutschler, and U. Wagner. “Impact of probabilistic small-scale photovoltaic generation forecast on energy management systems.” In: *Solar Energy* 165 (May 2018), pp. 136–146.

[15] W. El-Baz, P. Tzscheutschler, and U. Wagner. “Day-ahead probabilistic PV generation forecast for buildings energy management systems.” In: *Solar Energy* 171 (2018), pp. 478–490.

[16] R. J. Bessa, A. Trindade, C. S. P. Silva, and V. Miranda. “Probabilistic solar power forecasting in smart grids using distributed information.” In: *International Journal of Electrical Power and Energy Systems* 72 (2015), pp. 16–23.

[17] C. M. Bishop. *Mixture Density Networks*. Tech. rep. Birmingham: Dept. of Computer Science and Applied Mathematics Aston University, 1994.

[18] P. Blanchard, D. J. Higham, and N. J. Higham. “Accurately computing the log-sum-exp and softmax functions.” In: *IMA Journal of Numerical Analysis* 41 (Oct. 2021), pp. 2311–2330.

[19] T. Bollerslev. “Generalized autoregressive conditional heteroskedasticity.” In: *Journal of Econometrics* 31 (Apr. 1986), pp. 307–327.

[20] M. Bouzerdoum, A. Mellit, and A. Massi Pavan. “A hybrid model (SARIMA–SVM) for short-term power forecasting of a small-scale grid-connected photovoltaic plant.” In: *Solar Energy* 98 (2013), pp. 226–235.

[21] G. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1976.

[22] A. Bracale, G. Carpinelli, and P. De Falco. “A Probabilistic Competitive Ensemble Method for Short-Term Photovoltaic Power Forecasting.” In: *IEEE Transactions on Sustainable Energy* 8 (Apr. 2017), pp. 551–560.

[23] L. Breiman. “Bagging predictors.” In: *Machine Learning* 24 (Aug. 1996), pp. 123–140.

[24] J. Bröcker and L. A. Smith. “Scoring Probabilistic Forecasts: The Importance of Being Proper.” In: *Weather and Forecasting* 22 (2007), pp. 382–388.

[25] A. Brusaferri, M. Matteucci, S. Spinelli, and A. Vitali. “Probabilistic electric load forecasting through Bayesian Mixture Density Networks.” In: *Applied Energy* 309 (Mar. 2022), p. 118341.

[26] P. Buehlmann. “Sieve bootstrap for time series.” In: *Bernoulli* 3 (1997), pp. 123–148.

[27] P. Buehlmann. “Bootstraps for time series.” In: *Statistical Science* 17 (2002), pp. 52–72.

[28] Bundesministerium der Justiz. *Verordnung über den Zugang zu Elektrizitätsversorgungsnetzen (Stromnetzzugangsverordnung - StromNZV)*. 2021.

[29] Bundesministerium der Justiz. *Gesetz zu Sofortmaßnahmen für einen beschleunigten Ausbau der erneuerbaren Energien und weiteren Maßnahmen im Stromsektor*. Bonn, 2022.

[30] Bundesnetzagentur. *Genehmigung des Szenariorahmens 2023-2037/2045*. Tech. rep. Bonn: Bundesnetzagentur, 2022.

[31] Bundesnetzagentur. *SMARD Strommarktdaten*. 2023. URL: https://www.smard.de/home/downloadcenter/download-marktdaten (visited on 11/28/2023).

[32] Bundesnetzagentur. *This is how the electricity market works*. URL: https://www.smard.de/page/en/wiki-article/5884/5840 (visited on 11/28/2023).

[33] F. B. Carlson. “Machine Learning and System Identification for Estimation in Physical Systems.” PhD thesis. 2019.

[34] T. Carriere, C. Vernay, S. Pitaval, and G. Kariniotakis. “A Novel Approach for Seamless Probabilistic Photovoltaic Power Forecasting Covering Multiple Time Frames.” In: *IEEE Transactions on Smart Grid* 11 (May 2020), pp. 2281–2292.

[35] H. Chen, F. Li, Q. Wan, and Y. Wang. “Short term load forecasting using regime-switching GARCH models.” In: *IEEE Power and Energy Society General Meeting* (2011), pp. 1–6.

[36] H. Chen, Q. Wan, F. Li, and Y. Wang. “Short term load forecasting based on improved ESTAR GARCH model.” In: *IEEE Power and Energy Society General Meeting* (2012), pp. 1–6.

[37] H. Chen, J. Zhang, Y. Tao, and F. Tan. “Asymmetric GARCH type models for asymmetric volatility characteristics analysis and wind power forecasting.” In: *Protection and Control of Modern Power Systems* 4 (2019).

[38] K. Chen et al. “Short-Term Load Forecasting with Deep Residual Networks.” In: *IEEE Transactions on Smart Grid* 10 (2019), pp. 3943–3952.

[39] V. Chernozhukov, I. Fernández-Val, and A. Galichon. “Improving point and interval estimators of monotone functions by rearrangement.” In: *Biometrika* 96 (2009), pp. 559–575.

[40] F. Chollet and E. Al. *Keras*. 2015. URL: https://github.com/keras-team/keras/blob/v2.11.0/keras/constraints.py#L104 (visited on 01/27/2023).

[41] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. “The loss surfaces of multilayer networks.” In: *Journal of Machine Learning Research* 38 (2015), pp. 192–204.

[42] Y. Chu et al. “Short-term reforecasting of power output from a 48 MWe solar PV plant.” In: *Solar Energy* 112 (2015), pp. 68–77.

[43] G. Cybenko. “Approximation by superpositions of a sigmoidal function.” In: *Mathematics of Control, Signals and Systems* 2 (1989), pp. 303–314.

[44] J. G. da Silva Fonseca Jr. et al. “Use of support vector regression and numerically predicted cloudiness to forecast power output of a photovoltaic power plant in Kitakyushu, Japan.” In: *Progress in Photovoltaics: Research and Applications* 20 (2012), pp. 874–882.

[45] M. L. Darby and M. Nikolaou. “Identification test design for multivariable model-based control: An industrial perspective.” In: *Control Engineering Practice* 22 (2014), pp. 165–180.

[46] U. K. Das et al. “Forecasting of photovoltaic power generation and model optimization: A review.” In: *Renewable and Sustainable Energy Reviews* 81 (2018), pp. 912–928.

[47] M. David, F. Ramahatana, P. J. Trombe, and P. Lauret. “Probabilistic forecasting of the solar irradiance with recursive ARMA and GARCH models.” In: *Solar Energy* 133 (2016), pp. 55–72.

[48] M. David, M. A. Luis, and P. Lauret. “Comparison of intraday probabilistic forecasting of solar irradiance using only endogenous data.” In: *International Journal of Forecasting* 34 (2018), pp. 529–547.

[49] A. C. Davidson and D. Kuonen. “An introduction to the bootstrap with application in R.” In: *Statistical Computing & Statistical Graphics Newsletter* 13 (2003), pp. 6–11.

[50] A. C. Davison and D. V. Hinkley. *Bootstrap Methods and their Application*. Vol. 42. Cambridge University Press, Oct. 1997, p. 216.

[51] C. J. Dent, J. W. Bialek, and B. F. Hobbs. “Opportunity Cost Bidding by Wind Generators in Forward Markets: Analytical Results.” In: *IEEE Transactions on Power Systems* 26 (Aug. 2011), pp. 1600–1608.

[52] Die deutschen Übertragungsnetzbetreiber. *Modalitäten für Regelreserveanbieter gemäß Artikel 18 Abs. 5 der Verordnung (EU) 2017/2195 der Kommission vom 23. November 2017 zur Festlegung einer Leitlinie über den Systemausgleich im Elektrizitätsversorgungssystem*. Tech. rep. 2019.

[53] R. P. B.-M. Dittmar and B. M. Pfeiffer. *Modellbasierte prädiktive Regelung - Eine Einführung für Ingenieure*. Oldenbourg, 2004.

[54] O. Doelle. *Day ahead PV prediction - Comparison of different model structures*. Tech. rep. Erlangen: Internal report - CT REE ENS DEH-DE, 2020.

[55] O. Doelle, A. Amthor, and C. Ament. “Automated parameter identification: robust model validation of a compression chiller based on an uncertainty and consistency analysis.” In: *Proceedings of 11th International Conference on Applied Energy*. Vol. Volume 4. Sweden, 2019, pp. 1–4.

[56] O. Doelle, A. Amthor, and C. Ament. “Probabilistic intraday forecasting of solar power using Monte Carlo dropout and deep neural networks.” In: *10th IEEE PES Innovative Smart Grid Technologies Conference – Asia*. 2021.

[57] O. Doelle, I. Kalysh, A. Amthor, and C. Ament. “Comparison of intraday probabilistic forecasting of solar power using time series models.” In: *2021 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, Sept. 2021, pp. 1–6.

[58] O. Doelle, N. Klinkenberg, A. Amthor, and C. Ament. “Probabilistic Intraday PV Power Forecast Using Ensembles of Deep Gaussian Mixture Density Networks.” In: *Energies* 16 (2023), pp. 1–17.

[59] O. Dölle, S. Niessen, S. Schreck, S. Jochen, and S. Thiem. *DEVICE AND METHOD FOR CONTROLLING ENERGY FLOWS BETWEEN PARTICIPANTS OF AN ENERGY SYSTEM*. 2022.

[60] O. Dölle, S. Niessen, S. Schreck, and S. Thiem. *CONTROL PLATFORM FOR EXCHANGES OF ENERGY BETWEEN A PLURALITY OF ENERGY SYSTEMS, AND ENERGY EXCHANGE SYSTEM*. 2019.

[61] O. Dölle, S. Niessen, S. Schreck, and S. Thiem. *VERFAHREN ZUR ERMITTLUNG EINES AUSFALLRISIKOS*. 2021.

[62] Z. Dong, D. Yang, T. Reindl, and W. M. Walsh. “Short-term solar irradiance forecasting using exponential smoothing state space model.” In: *Energy* 55 (2013), pp. 1104–1113.

[63] K. Doubleday, S. Jascourt, W. Kleiber, and B.-m. Hodge. “Probabilistic Solar Power Forecasting Using Bayesian Model Averaging.” In: *IEEE Transactions on Sustainable Energy* 12 (Jan. 2021), pp. 325–337.

[64] J. A. Duffie and W. A. Beckman. “Solar energy thermal processes.” In: (1974). Ed. by I. John Wiley and Sons.

[65] J. Dumas, C. Cointe, X. Fettweis, and B. Cornelusse. “Deep learning-based multi-output quantile forecasting of PV generation.” In: *2021 IEEE Madrid PowerTech*. IEEE, June 2021, pp. 1–6.

[66] E. Durán, J. M. Andújar, J. M. Enrique, and J. M. Pérez-Oria. “Determination of PV Generator I-V/P-V Characteristic Curves Using a DC-DC Converter Controlled by a Virtual Instrument.” In: *International Journal of Photoenergy* 2012 (2012). Ed. by S. Dai, pp. 1–13.

[67] B. Efron. “Bootstrap Methods: Another Look at the Jackknife.” In: *The Annals of Statistics* 7 (1979), pp. 1–26.

[68] R. Eldan and O. Shamir. “The Power of Depth for Feedforward Neural Networks.” In: *Annual Conference Computational Learning Theory* (Dec. 2015), pp. 1–33.

[69] H. Escrig et al. “Cloud detection, classification and motion estimation using geostationary satellite imagery for cloud cover forecast.” In: *Energy* 55 (2013), pp. 853–859.

[70] Europäisches Parlament and Rat der europäischen Union. *Richtlinie (EU) 2019/944 des europäischen Parlaments und des Rates vom 5. Juni 2019 mit gemeinsamen Vorschriften für den Elektrizitätsbinnenmarkt und zur Änderung der Richtlinie 2012/27/EU*. Tech. rep. Europäischen Parlament, 2019.

[71] European Commission. “EB GL: Comission Regulation (EU) 2017/2195 establishing a guideline on electricity balancing.” In: *Official Journal of the European Union* 2017 (2017), pp. 312/6–312/53.

[72] European Commission. *EU Solar Energy Strategy*. Brussels, 2022.

[73] A. Fassung, H. Wirth, and S. J. Bächle. *Aktuelle Fakten zur Photovoltaik in Deutschland*. Tech. rep. Freiburg: Fraunhofer ISE, 2022, pp. 1–97.

[74] Federal Ministry of Economy affairs and climate action. *Development-of-Renewable-Energy-Sources-in-Germany-2022*. Tech. rep. 2023.

[75] L. A. Fernandez-Jimenez et al. “Short-term power forecasting system for photovoltaic plants.” In: *Renewable Energy* 44 (2012), pp. 311–317.

[76] S. Fort, H. Hu, and B. Lakshminarayanan. “Deep Ensembles: A Loss Landscape Perspective.” In: (Dec. 2019), pp. 1–15.

[77] J. Fox. *Applied Regression Analysis and Generalized Linear Models*. 3rd. Los Angeles: SAGE Publications, 2016.

[78] C. Francq and J.-M. Zakoian. *GARCH Models: Structure, Statistical Inference and Financial Applications*. Wiley, May 2019.

[79] Fraunhofer ISE. *Photovoltaics Report*. Tech. rep. Fraunhofer Institute for Solar Energy Systems, 2022.

[80] G. Friesen et al. *Intercomparison of Different Energy Prediction Methods Within the European Project "Performance" - Results of the 1st Round Robin*. Milano, 2007.

[81] K. Fukushima. “Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements.” In: *IEEE Transactions on Systems Science and Cybernetics* 5 (1969), pp. 322–333.

[82] M. Al-gabalawy, N. S. Hosny, and A. R. Adly. “Probabilistic forecasting for energy time series considering uncertainties based on deep learning algorithms.” In: *Electric Power Systems Research* 196 (2021), p. 107216.

[83] Y. Gal and Z. Ghahramani. “Dropout as a Bayesian Approximation: Appendix.” In: *33rd International Conference on Machine Learning, ICML 2016* 3 (2016), pp. 1661–1680.

[84] Y. Gal and Z. Ghahramani. “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning.” In: *33rd International Conference on Machine Learning, ICML 2016* 3 (2016), pp. 1651–1660.

[85] J. Gawlikowski et al. “A Survey of Uncertainty in Deep Neural Networks.” In: (July 2021), pp. 1–41.

[86] Y. R. Gel and S. E. Ahmed. “Catching Uncertainty of Wind : A Blend of Sieve Bootstrap and Regime Switching Models for Probabilistic Short-term Forecasting of Wind Speed.” In: (2015), pp. 1–14.

[87] S. Ghosh and P. K. Gupta. “Forecasting of Solar Power Volatility using GJR-GARCH method.” In: *2021 IEEE Electrical Power and Energy Conference (EPEC)*. IEEE, Oct. 2021, pp. 261–266.

[88] T. Gneiting, F. Balabdaoui, and A. E. Raftery. “Probabilistic forecasts, calibration and sharpness.” In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69 (2007), pp. 243–268.

[89] I. J. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[90] A. Grantham, Y. R. Gel, and J. Boland. “Nonparametric short-term probabilistic forecasting for solar radiation.” In: *Solar Energy* 133 (2016), pp. 465–475.

[91] T. M. Hamill. “Interpretation of Rank Histograms for Verifying Ensemble Forecasts.” In: *Monthly Weather Review* 129 (2001), pp. 550–560.

[92] B. E. Hansen. “Autoregressive Conditional Density Estimation.” In: *International Economic Review* 35 (Aug. 1994), p. 705.

[93] H. Hersbach. “Decomposition of the continuous ranked probability score for ensemble prediction systems.” In: *Weather and Forecasting* 15 (2000), pp. 559–570.

[94] J. Herzen et al. *Darts: User-Friendly Modern Machine Learning for Time Series*. 2022. URL: https://unit8co.github.io/darts/README.html (visited on 01/04/2023).

[95] S. Hieronymus Schreck. “Local Energy Markets - Simulative Evaluation and Field Test Application of Energy Markets on Distribution Grid Level.” PhD thesis. Technische Universität Darmstadt, 2023.

[96] G. Hinton, Y. Bengio, and Y. Lecun. “Deep Learning - NIPS’2015 Tutorial.” In: *Twenty-ninth Conference on Neural Information Processing Systems*. Montréal, 2015.

[97] K. D. Hodge, V. V. S. Hernandez, and Bri-Mathias. “Benchmark probabilistic solar forecasts: Characteristics and recommendations.” In: *Solar Energy* 206 (Aug. 2020), pp. 52–67.

[98] T. Hong. “Energy Forecasting : Past , Present , and Future.” In: *Foresight: The International Journal of Forecasting* (2014), pp. 43–49.

[99] T. Hong and S. Fan. "Probabilistic electric load forecasting: A tutorial review." In: *International Journal of Forecasting* 32 (2016), pp. 914–938.

[100] T. Hong et al. *Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond*. July 2016.

[101] T. Hong et al. "Energy Forecasting: A Review and Outlook." In: *IEEE Open Access Journal of Power and Energy* (2020), pp. 1–1.

[102] K. Hornik, M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators." In: *Neural Networks* 2 (1989), pp. 359–366.

[103] L. Huang et al. "Normalization Techniques in Training DNNs: Methodology, Analysis and Application." In: (Sept. 2020), pp. 1–20.

[104] E. Hüllermeier and W. Waegeman. "Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods." In: *CoRR* (Oct. 2019), pp. 1–59.

[105] F. Hutter, L. Kotthoff, and J. Vanschoren. *Automated machine learning : methods, systems, challenges*. 2019, p. 219.

[106] R. Hyndman. *Prediction intervals too narrow*. 2014. URL: https://robjhyndman.com/hyndsight/narrow-pi/ (visited on 03/11/2020).

[107] R. Hyndman and G. Athanasopoulos. *Forecasting: Principles and Practice*. 3rd. Melbourne, Australia: OTexts, 2021.

[108] R. H. Inman, H. T. Pedro, and C. F. Coimbra. "Solar forecasting methods for renewable energy integration." In: *Progress in Energy and Combustion Science* 39 (2013), pp. 535–576.

[109] International Energy Agency. *Renewables 2022 Analysis forecast to 2027*. Tech. rep. 2022, p. 158.

[110] IRENA. *RENEWABLE POWER GENERATION COSTS IN 2021*. Tech. rep. Abu Dhabi: International Renewable Energy Agency, 2022.

[111] G. James and T. Hastie. "Generalizations of the Bias/Variance Decomposition for Prediction Error." In: (1997), pp. 1–13.

[112] Jay I. Myung, Daniel R. Cavagnaro and M. A. Pitt. *Model Evaluation and Selection*. Tech. rep. Cambridge, 2015.

[113] H. Jin, Q. Song, and X. Hu. "Auto-keras: An efficient neural architecture search system." In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2019), pp. 1946–1956.

[114] A. Kendall and Y. Gal. "What uncertainties do we need in Bayesian deep learning for computer vision?" In: *Advances in Neural Information Processing Systems* 2017-Decem (2017), pp. 5575–5585.

[115] A. Khosravi et al. “Lower Upper Bound Estimation Method for Construction of Neural Network-Based Prediction Intervals.” In: *IEEE Transactions on Neural Networks* 22 (2011), pp. 337–346.

[116] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization.” In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. arXiv, 2015, pp. 1–15.

[117] N. Klinkenberg. *Probabilistic forecasting of photovoltaic power generation*. Tech. rep. Westfälische Hochschule, 2020.

[118] P. Koponen, J. Ikäheimo, J. Koskela, C. Brester, and H. Niska. “Assessing and comparing short term load forecasting performance.” In: *Energies* 13 (2020).

[119] D. S. Kumar, G. M. Yagli, M. Kashyap, and D. Srinivasan. “Solar irradiance resource and forecasting: a comprehensive review.” In: *IET Renewable Power Generation* 14 (2020), pp. 1641–1656.

[120] S. Lahiri. *Resampling Methods for Dependent Data*. 1st ed. New York: Springer, 2003, p. 374.

[121] A. Lahouar, A. Mejri, and J. B. H. Slama. “Probabilistic day-ahead load forecast using quantile regression forests.” In: *Proceedings - 2017 International Conference on Engineering and MIS, ICEMIS 2017* 2018-Janua (2018), pp. 1–6.

[122] B. Lakshminarayanan, A. Pritzel, and C. Blundell. “Simple and scalable predictive uncertainty estimation using deep ensembles.” In: *Advances in Neural Information Processing Systems* 2017-Decem (2017), pp. 6403–6414.

[123] P. Lauret, M. David, and H. T. Pedro. “Probabilistic solar forecasting using quantile regression models.” In: *Energies* 10 (2017), pp. 1–17.

[124] P. Lauret, M. David, and P. Pinson. “Verification of solar irradiance probabilistic forecasts.” In: *Solar Energy* 194 (Dec. 2019), pp. 254–271.

[125] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller. “Efficient BackProp.” In: *Neural Networks: Tricks of the Trade: Second Edition*. Ed. by G. Montavon, G. B. Orr, and K.-R. Müller. Vol. 4. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–48.

[126] H. Y. Lee, N. W. Kim, J. G. Lee, and B. T. Lee. “Uncertainty-aware forecast interval for hourly PV power output.” In: *IET Renewable Power Generation* 13 (2019), pp. 2656–2664.

[127] S. Leva, A. Dolara, F. Grimaccia, M. Mussetta, and E. Ogliari. “Analysis and validation of 24 hours ahead neural network forecasting of photovoltaic output power.” In: *Mathematics and Computers in Simulation* 131 (2017), pp. 88–100.

[128] B. Li and J. Zhang. “A review on the integration of probabilistic solar forecasting in power systems.” In: *Solar Energy* 210 (2020), pp. 68–86.

[129] C. Li and M. Zhang. “Application of GARCH Model in the Forecasting of Day-Ahead Electricity Prices.” In: *Third International Conference on Natural Computation (ICNC 2007)*. Vol. 1. IEEE, 2007, pp. 99–103.

[130] Y. Li, F. Zheng, Z. Li, L. Zheng, and Q. Ding. “Active Vibration Control of Gear Transmission System.” In: *The 21st International Congress on Sound and Vibration*. Beijing, 2014, pp. 13–17.

[131] R. J. Licata and P. M. Mehta. “Uncertainty quantification techniques for data-driven space weather modeling: thermospheric density application.” In: *Scientific Reports* 12 (May 2022), p. 7256.

[132] R. Liu et al. “A short-term probabilistic photovoltaic power prediction method based on feature selection and improved LSTM neural network.” In: *Electric Power Systems Research* 210 (Sept. 2022), p. 108069.

[133] L. Ljung. “System Identification.” In: *Wiley Encyclopedia of Electrical and Electronics Engineering*. Hoboken, NJ, USA: Wiley, Dec. 1999.

[134] E. Lorenz, D. Heinemann, H. Wickramarathne, H. G. Beyer, and S. Bofinger. “Forecast of ensemble power production by grid-connected PV systems.” In: *Proceedings 20th European Photovoltaic Solar Energy Conference* (2007).

[135] O. Maimon and L. Rokach. *Decomposition Methodology for Knowledge Discovery and Data Mining*. WORLD SCIENTIFIC, 2005.

[136] A. Malinin and M. Gales. “Predictive Uncertainty Estimation via Prior Networks.” In: (2018).

[137] F. Marchesoni-Acland and R. Alonso-Suárez. “Intra-day solar irradiation forecast using RLS filters and satellite images.” In: *Renewable Energy* 161 (2020), pp. 1140–1154.

[138] F. Marchesoni-Acland, P. Lauret, A. Gomez, and R. Alonso-Suarez. “Analysis of ARMA Solar Forecasting Models Using Ground Measurements and Satellite Images.” In: *Conference Record of the IEEE Photovoltaic Specialists Conference* (2019), pp. 2445–2451.

[139] A. Mashlakov, T. Kuronen, L. Lensu, A. Kaarna, and S. Honkapuro. “Assessing the performance of deep learning models for multivariate probabilistic energy forecasting.” In: *Applied Energy* 285 (Mar. 2021), p. 116405.

[140] MathWorks. *Code Generation in Linear Least Squares: Background*. 2022. URL: https://de.mathworks.com/help/optim/ug/code-generation-linear-least-squares.html (visited on 11/12/2022).

[141] M. J. Mayer and G. Gróf. “Extensive comparison of physical models for photovoltaic power forecasting.” In: *Applied Energy* 283 (Feb. 2021), p. 116239.

[142] D. W. van der Meer, M. Shepero, A. Svensson, J. Widén, and J. Munkhammar. "Probabilistic forecasting of electricity consumption, photovoltaic power generation and net demand of an individual building using Gaussian Processes." In: *Applied Energy* 213 (2018), pp. 195–207.

[143] D. van der Meer. "Comment on "Verification of deterministic solar forecasts": Verification of probabilistic solar forecasts." In: *Solar Energy* 210 (Nov. 2020), pp. 41–43.

[144] Meteoblue. *Weather APIs*. 2023. URL: https://content.meteoblue.com/en/business-solutions/weather-apis (visited on 01/28/2021).

[145] H. Mhaskar, Q. Liao, and T. Poggio. "When and why are deep networks better than shallow ones?" In: *31st AAAI Conference on Artificial Intelligence, AAAI 2017* (2017), pp. 2343–2349.

[146] V. Milián, N. Eds, I. Congress, and G. Goos. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Ed. by I. Nyström, Y. Hernández Heredia, and V. Milián Núñez. Vol. 11896. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019.

[147] G. Mitrentsis and H. Lens. "An interpretable probabilistic model for short-term solar power forecasting using natural gradient boosting." In: *Applied Energy* 309 (2022), p. 118473.

[148] S. Monjoly, M. André, R. Calif, and T. Soubdhan. "Forecast Horizon and Solar Variability Influences on the Performances of Multiscale Hybrid Forecast Model." In: *Energies* 12 (June 2019), p. 2264.

[149] C. Monteiro, T. Santos, L. A. Fernandez-Jimenez, I. J. Ramirez-Rosado, and M. S. Terreros-Olarte. "Short-Term Power Forecasting Model for Photovoltaic Plants Based on Historical Similarity." In: *Energies* 6 (2013), pp. 2624–2643.

[150] D. C. Montgomery, C. L. Jennings, and M. Kulahci. *Introduction to Time Series Analysis and Forecasting*. 2nd ed. Wiley Series in Probability and Statistics. Wiley, 2015.

[151] L. Mora-López and M. Sidrach-de-Cardona. "Multiplicative ARMA models to generate hourly series of global irradiation." In: *Solar Energy* 63 (1998), pp. 283–291.

[152] A. H. Murphy. "What Is a Good Forecast? An Essay on the Nature of Goodness in Weather Forecasting." In: *Weather and Forecasting* 8 (1993), pp. 281–293.

[153] K. P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[154] J. I. Myung, M. A. Pitt, and W. Kim. *Handbook of Cognition*. London, 2005. URL: https://sk.sagepub.com/reference/hdbk_cognition.

[155] F. Najibi, D. Apostolopoulou, and E. Alonso. “Enhanced performance Gaussian process regression for probabilistic short-term solar output forecast.” In: *International Journal of Electrical Power and Energy Systems* 130 (2021), p. 106916.

[156] T. N. Nguyen and F. Müsgens. “What drives the accuracy of PV output forecasts ?” In: *Applied Energy* 323 (2022), p. 119603.

[157] Novin Shahroudi. “Probabilistic Forecasting with Monte-Carlo Dropout in Neural Networks.” Master’s Thesis. University of Tartu, Institute of Computer Science, 2019, p. 73.

[158] G. C. Okwuibe et al. “Intelligent Bidding Strategies for Prosumers in Local Energy Markets Based on Reinforcement Learning.” In: *IEEE Access* 10 (2022), pp. 113275–113293.

[159] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio. “N-BEATS: Neural basis expansion analysis for interpretable time series forecasting.” In: *CoRR* abs/1905.1 (2019).

[160] L. Özkan et al. “Advanced autonomous model-based operation of industrial process systems (Autoprofit): Technological developments and future perspectives.” In: *Annual Reviews in Control* 42 (2016), pp. 126–142.

[161] H. Panamtash and Q. Zhou. “Coherent Probabilistic Solar Power Forecasting.” In: *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*. IEEE, June 2018, pp. 1–6.

[162] I. de la Parra et al. “PV performance modelling: A review in the light of quality assurance for large PV plants.” In: *Renewable and Sustainable Energy Reviews* 78 (2017), pp. 780–797.

[163] H. T. C. Pedro and C. F. M. Coimbra. “Assessment of forecasting techniques for solar power production with no exogenous inputs.” In: *Solar Energy* 86 (2012), pp. 2017–2028.

[164] H. T. Pedro, C. F. Coimbra, M. David, and P. Lauret. “Assessment of machine learning techniques for deterministic and probabilistic intra-hour solar forecasts.” In: *Renewable Energy* 123 (2018), pp. 191–203.

[165] Z. Peng et al. “3D cloud detection and tracking system for solar forecast using multiple sky imagers.” In: *Solar Energy* 118 (2015), pp. 496–519.

[166] R. Perez, R. Seals, P. Ineichen, R. Stewart, and D. Menicucci. “A new simplified version of the perez diffuse irradiance model for tilted surfaces.” In: *Solar Energy* 39 (1987), pp. 221–231.

[167] F. Petropoulos, R. J. Hyndman, and C. Bergmeir. “Exploring the sources of uncertainty: Why does bagging for time series forecasting work?” In: *European Journal of Operational Research* 268 (July 2018), pp. 545–554.

[168] F. Petropoulos et al. “Forecasting: theory and practice.” In: *International Journal of Forecasting* (2022).

[169] P. Pinson and G. Kariniotakis. “Conditional Prediction Intervals of Wind Power Generation.” In: *IEEE Transactions on Power Systems* 25 (Nov. 2010), pp. 1845–1856.

[170] P. Pinson, P. McSharry, and H. Madsen. “Reliability diagrams for non-parametric density forecasts of continuous variables: Accounting for serial correlation.” In: *Quarterly Journal of the Royal Meteorological Society* 136 (2010), pp. 77–90.

[171] P. Pinson, H. A. Nielsen, J. K. Møller, H. Madsen, and G. N. Kariniotakis. “Non-parametric probabilistic forecasts of wind power: required properties and evaluation.” In: *Wind Energy* 10 (2007), pp. 497–516.

[172] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao. “Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review.” In: *International Journal of Automation and Computing* 14 (2017), pp. 503–519.

[173] L. Prechelt. “Early Stopping - But When?” In: *Neural Networks: Tricks of the Trade*. Ed. by G. B. Orr and K.-R. Müller. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 55–69.

[174] G. Press. *Andrew Ng Launches A Campaign For Data-Centric AI*. 2021.

[175] H. Quan and D. Yang. “Probabilistic solar irradiance transposition models.” In: *Renewable and Sustainable Energy Reviews* 125 (2020), p. 109814.

[176] V. Quaschning. *Simulation der Abschattungsverluste bei solarelektrischen Systemen*. Berlin: Dr. Köster, 1996, pp. 1–210.

[177] V. Quaschning. *Understanding Renewable Energy Systems*. Routledge, Mar. 2016, pp. 9–25.

[178] M. Rana, I. Koprinska, and V. G. Agelidis. “2D-interval forecasts for solar power production.” In: *Solar Energy* 122 (2015), pp. 191–203.

[179] G. Reikard. “Predicting solar radiation at high resolutions: A comparison of time series forecasts.” In: *Solar Energy* 83 (2009), pp. 342–349.

[180] G. Reikard. “Comment on Verification of deterministic solar forecasts: Choice of models, and estimation procedure.” In: *Solar Energy* 210 (2020), pp. 47–48.

[181] Renewable International Agency Energy. *Global Renewables Outlook: Energy Transformation 2050*. Tech. rep. Abu Dhabi: International Renewable Energy Agency (IRENA), 2020.

[182] J. F. Rodríguez, J. Macías, M. J. Castro, M. de la Asunción, and C. Sánchez-Linares. “Use of Neural Networks for Tsunami Maximum Height and Arrival Time Predictions.” In: *GeoHazards* 3 (June 2022), pp. 323–344.

[183] S. Ruder. “An overview of gradient descent optimization algorithms.” In: (Sept. 2016), pp. 1–14.

[184] S. Schreck. “Implications of Sub-Hourly Solar Radiation Variability on Decentralized Energy Systems.” PhD thesis. Universität Stuttgart, 2018.

[185] S. Schreck et al. “A Methodological Framework to support Load Forecast Error Assessment in Local Energy Markets.” In: *IEEE Transactions on Smart Grid* 11 (2020), pp. 3212–3220.

[186] M. Selim, R. Zhou, W. Feng, and P. Quinsey. “Estimating Energy Forecasting Uncertainty for Reliable AI Autonomous Smart Grid Design.” In: *Energies* 14 (Jan. 2021), p. 247.

[187] K. Sheppard. *Arch.* 2023. URL: https://github.com/bashtage/arch (visited on 06/09/2023).

[188] Siemens AG. *SICAM A8000 Series CP-8031, CP-8050*. 2022.

[189] Siemens AG. *Automation and remote terminal units - SICAM A8000*. URL: https://new.siemens.com/global/en/products/energy/energy-automation-and-smart-grid/substation-automation/automation-and-remote-terminal-units-sicam-a8000-series.html (visited on 09/25/2022).

[190] J. Sjöberg and L. Ljung. “Overtraining, Regularization, and Searching for Minimum in Neural Networks.” In: *IFAC Proceedings Volumes* 25 (1992), pp. 73–78.

[191] Solcast. *Live and Forecast Data - Overview*. 2023. URL: https://solcast.com/data-specifications (visited on 01/10/2023).

[192] S. Sperati, S. Alessandrini, and L. Delle Monache. “An application of the ECMWF Ensemble Prediction System for short-term solar power forecasting.” In: *Solar Energy* 133 (2016), pp. 437–450.

[193] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting.” In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958.

[194] Stadtwerke Augsburg Energie GmbH. *swa Strom Basis - Preisblatt gültig ab 1. Januar 2024*. Augsburg, 2024.

[195] K. Stankeviciute, A. M. Alaa, and M. van der Schaar. “Conformal Time-series Forecasting.” In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. S. Liang, and J. W. Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 6216–6228.

[196] P. Stluka, G. Parthasarathy, S. Gabel, and T. Samad. “Architectures and Algorithms for Building Automation - An Industry View.” In: *Intelligent Building Control Systems: A Survey of Modern Building Control and Sensing Strategies*. Ed. by J. T. Wen and S. Mishra. Cham: Springer International Publishing, 2018, pp. 11–43.

[197] R. Stuhlmann, M. Rieland, and E. Paschke. “An Improvement of the IGMK Model to Derive Total and Diffuse Solar Radiation at the Surface from Satellite Data.” In: *Journal of Applied Meteorology and Climatology* 29 (1990), pp. 586–603.

[198] M. Sun, C. Feng, and J. Zhang. “Probabilistic solar power forecasting based on weather scenario generation.” In: *Applied Energy* 266 (2020), p. 114823.

[199] Taylor G Smith. *pmdarima: ARIMA estimators for Python*. 2022. URL: https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.ARIMA.html (visited on 01/04/2023).

[200] The Climate Corporation. *properscoring*. 2015. URL: https://github.com/TheClimateCorporation/properscoring (visited on 07/28/2022).

[201] A. Tsyplakov. “Evaluation of Probabilistic Forecasts : Proper Scoring Rules and Moments.” In: *SSRN Electronic Journal* (2013), pp. 1–21.

[202] S. I. Vagropoulos, G. I. Chouliaras, E. G. Kardakos, C. K. Simoglou, and A. G. Bakirtzis. “Comparison of SARIMAX , SARIMA , Modified SARIMA and ANN-based Models for Short-Term PV Generation Forecasting.” In: *2016 IEEE International Energy Conference (ENERGYCON)* (2016), pp. 1–6.

[203] D. Vallejo and R. Chaer. “Mixture Density Networks applied to wind and photovoltaic power generation forecast.” In: *2020 IEEE PES Transmission & Distribution Conference and Exhibition - Latin America (T&D LA)*. IEEE, Sept. 2020, pp. 1–5.

[204] D. Van Der Meer. “Spatio-temporal forecasting and optimization for integration of solar energy in urban energy systems.” PhD thesis. Uppsala University, 2020.

[205] C. Voyant et al. “Prediction intervals for global solar irradiation forecasting using regression trees methods.” In: *Renewable Energy* 126 (Oct. 2018), pp. 332–340.

[206] E. Wang, D. Cook, and R. J. Hyndman. “A new tidy data structure to support exploration and modeling of temporal data.” In: *Journal of Computational and Graphical Statistics* 29 (2020), pp. 466–478.

[207] H. Wang, Z. Lei, X. Zhang, B. Zhou, and J. Peng. “A review of deep learning for renewable energy forecasting.” In: *Energy Conversion and Management* 198 (2019), p. 111799.

[208] K. Wang, H. Du, R. Jia, and H. Jia. “Performance Comparison of Bayesian Deep Learning Model and Traditional Bayesian Neural Network in Short-Term PV Interval Prediction.” In: *Sustainability* 14 (Oct. 2022), p. 12683.

[209] E. Wheatcroft. “Interpreting the skill score form of forecast performance metrics.” In: *International Journal of Forecasting* 35 (2019), pp. 573–579.

[210] D. H. Wolpert. “The Lack of A Priori Distinctions Between Learning Algorithms.” In: *Neural Computation* 8 (1996), pp. 1341–1390.

[211] J. M. Wooldridge. *Introductory Econometrics*. 2013, p. 910.

[212] L. Xiao, M. Li, and S. Zhang. “Short-term power load interval forecasting based on nonparametric Bootstrap errors sampling.” In: *Energy Reports* 8 (Nov. 2022), pp. 6672–6686.

[213] G. M. Yagli, D. Yang, and D. Srinivasan. “Automatic hourly solar forecasting using machine learning models.” In: *Renewable and Sustainable Energy Reviews* (2019), pp. 487–498.

[214] D. Yang. “Solar radiation on inclined surfaces: Corrections and benchmarks.” In: *Solar Energy* 136 (Oct. 2016), pp. 288–302.

[215] D. Yang. “A guideline to solar forecasting research practice: Reproducible, operational, probabilistic or physically-based, ensemble, and skill (ROPES).” In: *Journal of Renewable and Sustainable Energy* 11 (2019).

[216] D. Yang. “A universal benchmarking method for probabilistic solar irradiance forecasting.” In: *Solar Energy* 184 (2019), pp. 410–416.

[217] D. Yang and J. Boland. “Satellite-augmented diffuse solar radiation separation models.” In: *Journal of Renewable and Sustainable Energy* 11 (Mar. 2019), p. 023705.

[218] D. Yang, J. Kleissl, C. A. Gueymard, H. T. Pedro, and C. F. Coimbra. “History and trends in solar irradiance and PV power forecasting: A preliminary assessment and review using text mining.” In: *Solar Energy* 168 (July 2018), pp. 60–101.

[219] D. Yang, W. Li, G. Mert, and D. Srinivasan. “Operational solar forecasting for grid integration : Standards , challenges , and outlook.” In: *Solar Energy* 224 (2021), pp. 930–937.

[220] D. Yang et al. “Verification of deterministic solar forecasts.” In: *Solar Energy* (2020), pp. 1–18.

[221] D. Yang et al. “A review of solar forecasting , its dependence on atmospheric sciences and implications for grid integration : Towards carbon neutrality.” In: *Renewable and Sustainable Energy Reviews* 161 (2022), p. 112348.

[222] H. C. S. Yotto et al. “Estimation and Forecasting Electricity Load in Benin: Using Econometric Model ARIMA/GARCH.” In: *3rd International Conference on Electrical, Communication and Computer Engineering, ICECCE 2021* (2021), pp. 1–6.

[223] Y. Yu, X. Han, M. Yang, and J. Yang. “Probabilistic Prediction of Regional Wind Power Based on Spatiotemporal Quantile Regression.” In: *IEEE Transactions on Industry Applications* 56 (2020), pp. 6117–6127.

[224] Y. Yu, X. Han, M. Yang, and Y. Zhang. “A Regional Wind Power Probabilistic Forecast Method Based on Deep Quantile Regression.” In: *2020 IEEE/IAS 56th Industrial and Commercial Power Systems Technical Conference (I&CPS)*. 2020, pp. 1–8.

[225] M. Zamo, O. Mestre, P. Arbogast, and O. Pannekoucke. "A benchmark of statistical regression methods for short-term forecasting of photovoltaic electricity production. Part II: Probabilistic forecast of daily production." In: *Solar Energy* 105 (2014), pp. 804–816.

[226] E. Zelikman and C. Healy. "Improving Regression Uncertainty Estimates with an Empirical Prior." In: (2020).

[227] E. Zelikman, C. Healy, S. Zhou, and A. Avati. "CRUDE: Calibrating Regression Uncertainty Distributions Empirically." In: *ICML 2020 Workshop on Uncertainty & Robustness in Deep Learning*. Vienna: arXiv, May 2020.

[228] N. Zemouri, H. Bouzgou, and C. A. Gueymard. "Multimodel ensemble approach for hourly global solar irradiation forecasting." In: *The European Physical Journal Plus* 134 (Dec. 2019), p. 594.

[229] H. Zhang et al. "Improved Deep Mixture Density Network for Regional Wind Power Probabilistic Forecasting." In: *IEEE Transactions on Power Systems* 35 (July 2020), pp. 2549–2560.

[230] J. Zhang, Y. Wang, M. Sun, and N. Zhang. "Two-Stage Bootstrap Sampling for Probabilistic Load Forecasting." In: *IEEE Transactions on Engineering Management* 69 (June 2022), pp. 720–728.

[231] J. Zhang, Y. Wang, M. Sun, N. Zhang, and C. Kang. "Constructing Probabilistic Load Forecast from Multiple Point Forecasts: A Bootstrap Based Approach." In: *International Conference on Innovative Smart Grid Technologies, ISGT Asia 2018* (2018), pp. 184–189.

[232] W. Zhang et al. "Improving Probabilistic Load Forecasting Using Quantile Regression NN with Skip Connections." In: *IEEE Transactions on Smart Grid* 11 (2020), pp. 5442–5450.

[233] L. Zhu and N. Laptev. "Deep and Confident Prediction for Time Series at Uber." In: *IEEE International Conference on Data Mining Workshops, ICDMW*. New Orleans, LA, 2017, pp. 103–110.

A

Appendix

A.1 List of contributions

First-author publications and conference contributions

1. O. Doelle, N. Klinkenberg, A. Amthor, and C. Ament. “Probabilistic Intraday PV Power Forecast Using Ensembles of Deep Gaussian Mixture Density Networks.” In: *Energies* 16 (2023), pp. 1–17
2. O. Doelle, I. Kalysh, A. Amthor, and C. Ament. “Comparison of intraday probabilistic forecasting of solar power using time series models.” In: *2021 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, Sept. 2021, pp. 1–6 - **Honored with the best presentation award**
3. O. Doelle, A. Amthor, and C. Ament. “Probabilistic intraday forecasting of solar power using Monte Carlo dropout and deep neural networks.” In: *10th IEEE PES Innovative Smart Grid Technologies Conference – Asia*. 2021
4. O. Doelle, A. Amthor, and C. Ament. “Automated parameter identification: robust model validation of a compression chiller based on an uncertainty and consistency analysis.” In: *Proceedings of 11th International Conference on Applied Energy*. Vol. Volume 4. Sweden, 2019, pp. 1–4

Excerpt of published and pending patents

5. Dölle, O., Houssame, H., Schreck, S., & Thiem, S. (2021). METHOD FOR POWER PREDICTION OF AN ENERGY SYSTEM (Patent No. WO 2021/259540 Al).
6. Amthor, A., Dölle, O., Schreck, S., & Schütz, T. (2023). METHOD FOR CHECKING THE PLAUSIBILITY OF A FORECAST (Patent No. WO 2023/179992 Al).
7. Dölle, O., Niessen, S., Schreck, S., & Thiem, S. (2019). CONTROL PLATFORM FOR EXCHANGES OF ENERGY BETWEEN A PLURALITY OF ENERGY SYSTEMS, AND ENERGY EXCHANGE SYSTEM (Patent No. WO 2021/175463 Al).
8. Dölle, O., Schreck, S., & Thiem, S. (2021). CONTROLLING AN ENERGY EXCHANGE (Patent No. WO 2021/032326 Al).
9. Dölle, O., Niessen, S., Schreck, S., Jochen, S., & Thiem, S. (2022). DEVICE AND METHOD FOR CONTROLLING ENERGY FLOWS BETWEEN PARTICIPANTS OF AN ENERGY SYSTEM (Patent No. WO 2022/012933 Al).
10. Dölle, O., Niessen, S., Schreck, S., & Thiem, S. (2021). VERFAHREN ZUR ERMITTLUNG EINES AUSFALLRISIKOS (Patent No. EP 3 872 719 A1).
11. Amthor, A., Dölle, O., LANGEMEYER, S., & Schütz, T. (2022). IDENTIFICATION OF THE TOPOLOGY OF METER STRUCTURES OF AN ENERGY SYSTEM (Patent No. WO 2022/223633 Al).
12. Amthor, A., Dölle, O., & Schütz, T. (2022). TESTING A PARAMETERIZATION OF ONE OR MORE MEASURING DEVICES (Patent No. WO 2022/152438 Al).
13. Amthor, A., & Dölle, O. (2021). METHOD FOR IDENTIFYING PARAMETERS OF A BLACK BOX MODEL FOR ONE OR MORE ENERGY INSTALLATIONS IN AN ENERGY SYSTEM (Patent No. WO 2021/063568 Al).
14. Amthor, A., & Dölle, O. (2020). METHOD FOR VALIDATING SYSTEM PARAMETERS OF AN ENERGY SYSTEM (Patent No. WO 2020/229051 Al).
15. Dölle, O., Schreck, S., & Thiem, S. (2021). CERTIFICATION OF AT LEAST ONE FACILITY-SPECIFIC AMOUNT OF ENERGY OF AN ENERGY SYSTEM (Patent No. WO 2021/148288 Al).

A.2 List of supervised students' theses

1. Klinkenberg, N. (2020). Probabilistic forecasting of photovoltaic power generation. Westfälische Hochschule.
2. Schlabritz, A. (2022). Integration of Probabilistic Forecasts into an Internal Prediction Tool. Technische Hochschule Nürnberg.
3. Arbes F. (2019). Automatisierte Parameteridentifikation eines Eisspeichers für den Einsatz in einer modellprädiktiven Regelung. Universität Stuttgart.

A.3 Data analysis of remaining sites

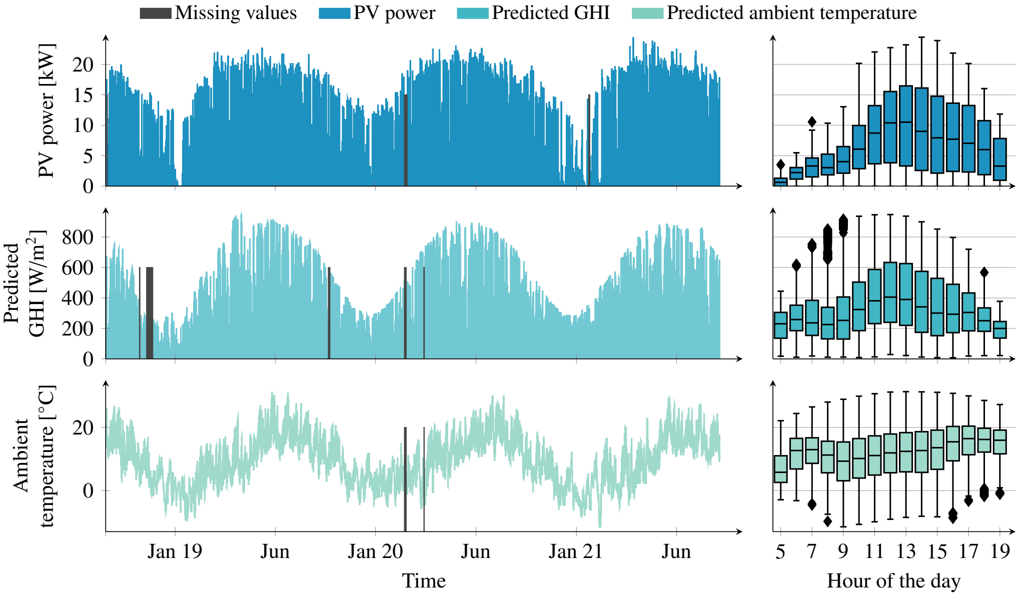


Figure A.1: Temporal profile of the signals for the PV power system in southern Bavaria.

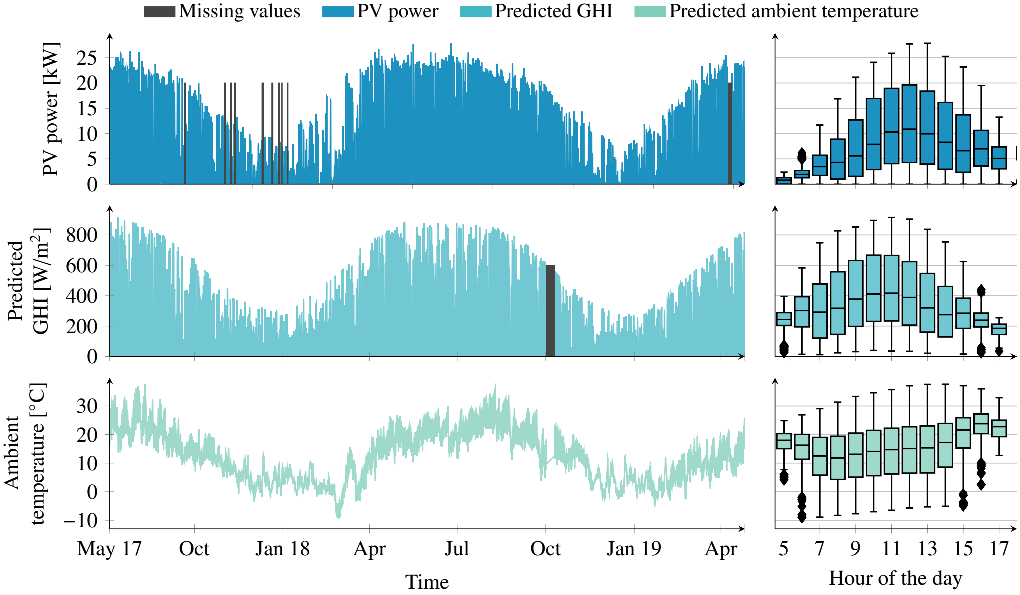


Figure A.2: Temporal profile of the signals for the PV power system in Vienna.

A.4 Additional result diagramms

A.4.1 Bootstrapping approaches for the ARX model

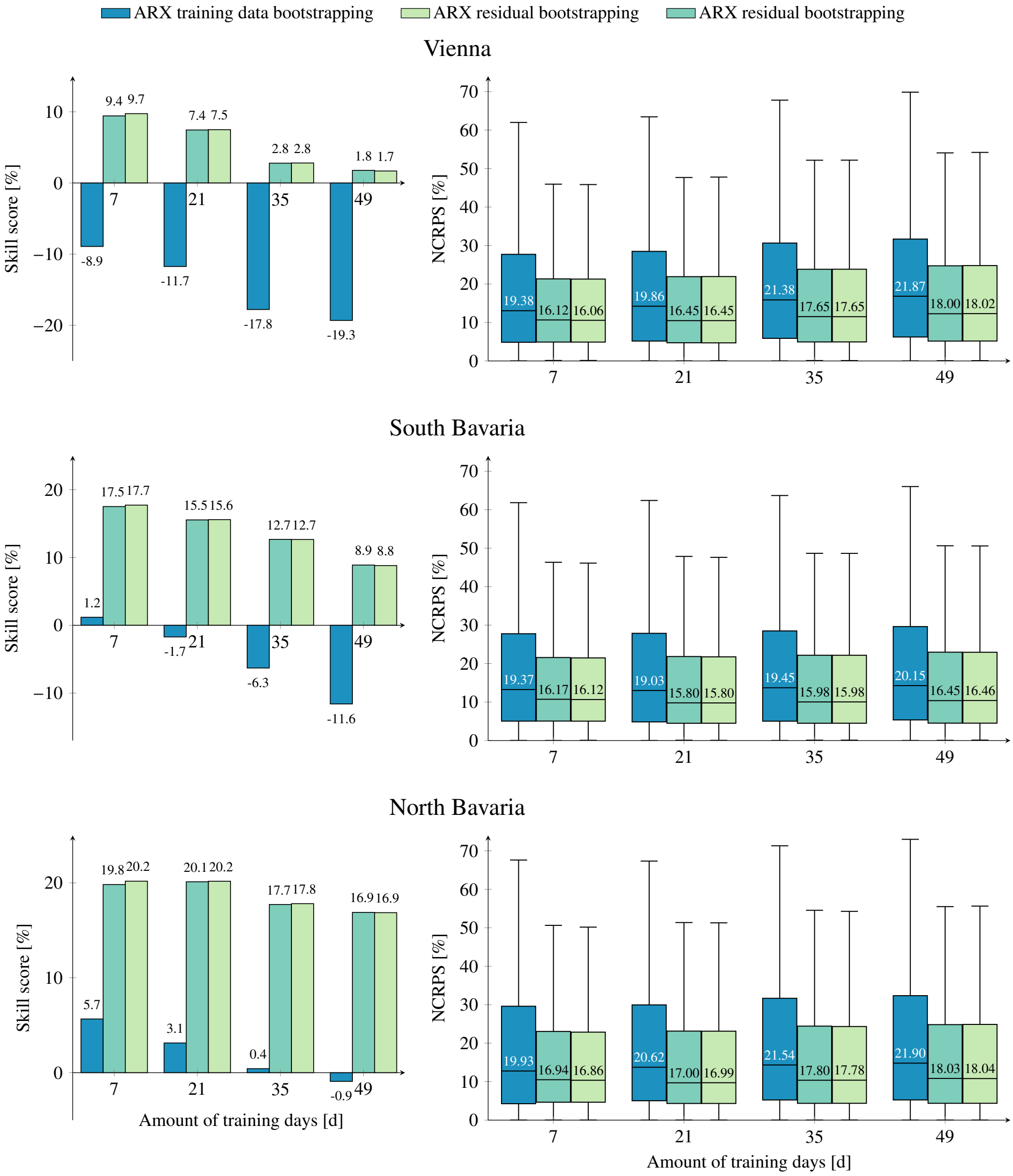


Figure A.3: Specific results for the individual locations for the boostrapping approaches for the ARX models.

A.4.2 Monte-Carlo dropout with output calibration

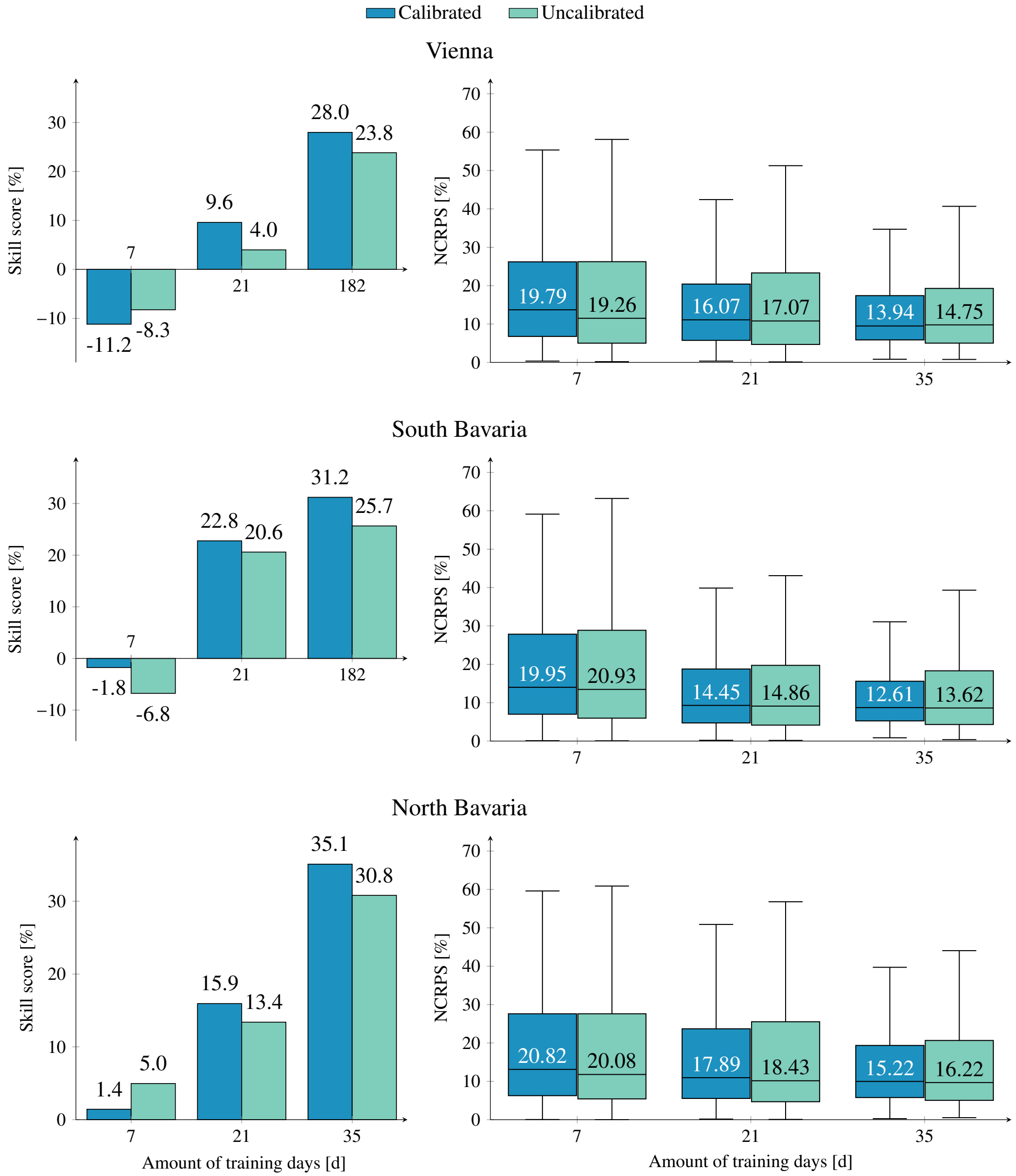


Figure A.4: Specific results for the individual locations for the boostrapping approaches for MC dropout with output calibration.

A.4.3 GARCH model in combination with the ARX model

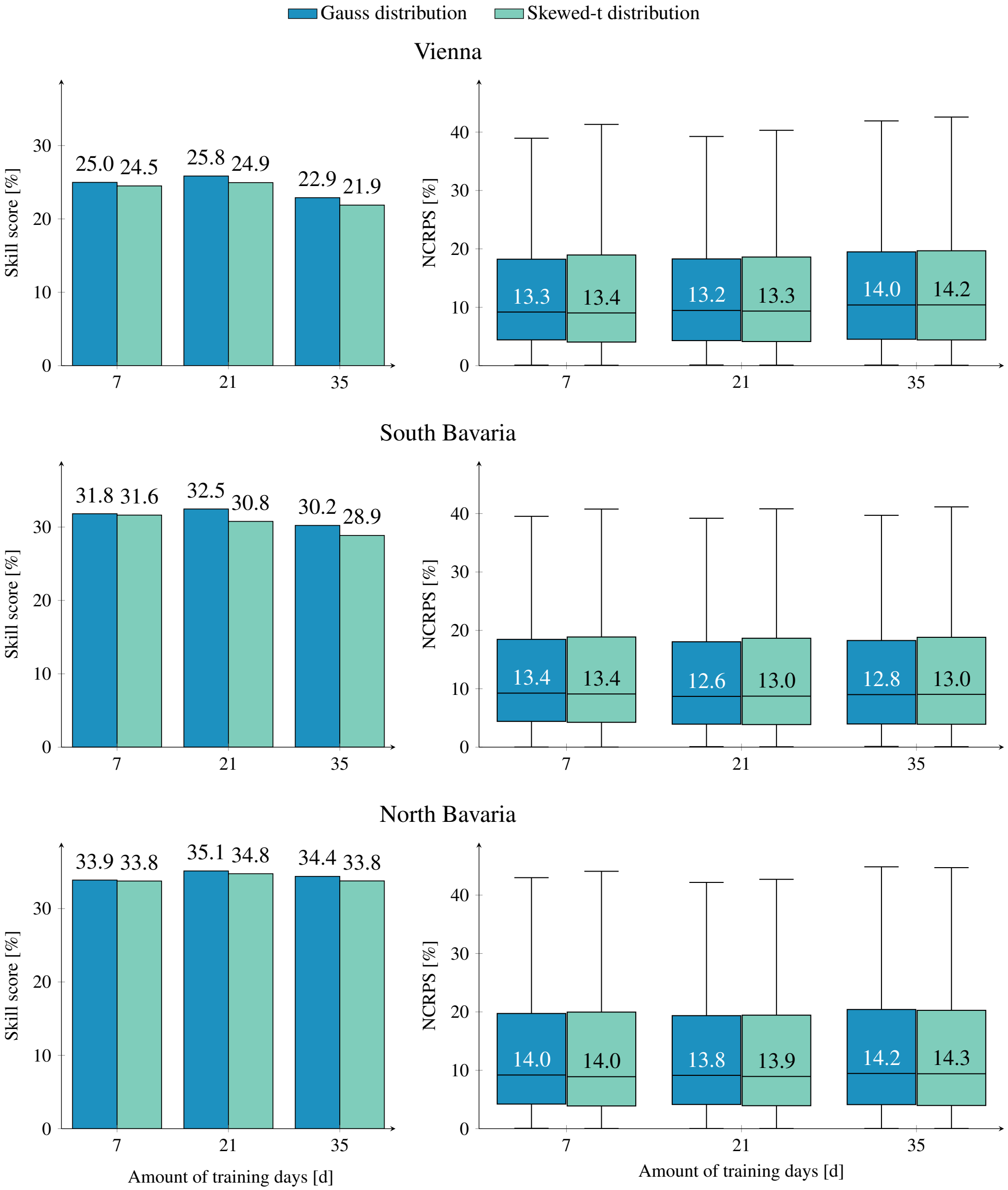


Figure A.5: Specific results for the individual locations for the GARCH model in combination with the ARX model.

A.4.4 Mixture density network

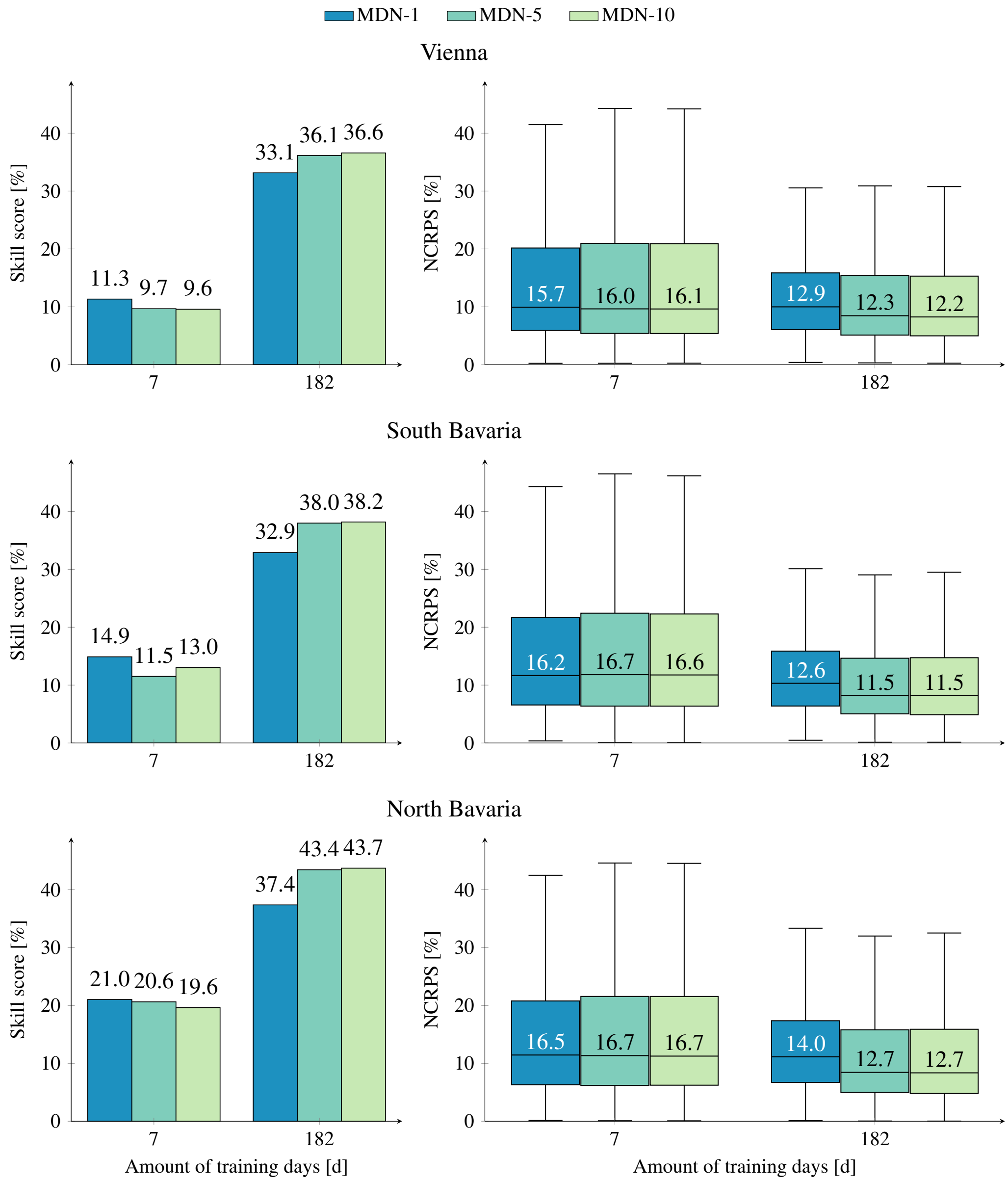


Figure A.6: Specific results for the individual locations for the MDN.

Index

Adam optimizer, 67
Aleatoric, 31
ARX, 28, 56
Automatic model order selection, 60

Bias-variance trade-off, 11
Bootstrapping, 68

Commissioning process, 23
Complete-history persistence ensemble, 49
Confidence interval, 32
Continuous probability distribution, 36
CRPS, 48
CRUDE, 76

Dropout, 65

Early stopping, 66
Ensemble, 34
Epistemic, 31

Forcast horizon, 21
Forecast, 9
Forecast quality, 45
Forecast type, 20

GARCH, 83
Gaussian distribution, 84
Gaussian mixture model, 88
Global horizontal irradiance, 13
Greedy search algorithm, 60

Irradiance, 13

Local energy markets, 17, 50

Maximum likelihood estimation, 84
Mixture density network, 86
Model predictive control, 16
Monte-Carlo dropout, 74
Multi-layer perceptron, 28, 62
Multi-modal decentralized energy system, 2

NCRPS, 48

Pinball loss function, 79
Prediction interval, 32
Probabilistic forecast, 4, 34
PV system, 13

Quantile, 35
Quantile neural network, 81
Quantile regression, 80

Rank histogramm, 46
Residual bootstrapping, 70

Sieve bootstrapping, 72
Skewed-t distribution, 85
Skill score, 49

Time series decomposition, 56

Uncertainty, 30